

УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА
ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници
Година: XXVI
Број: 7/2011

Нови Сад

Едиција: „Техничке науке – Зборници“
Година: XXVI Свеска: 7

Издавач: Факултет техничких наука Нови Сад

Главни и одговорни уредник: проф. др Илија Ћосић, декан Факултета техничких Наука у Новом Саду

Уређивачки одбор:

др Илија Ћосић	*др Бранко Шкорић*
др Владимир Катић	*др Јован Владић*
др Илија Ковачевић	*др Иван Пешењански*
др Јанко Ходолич	*др Бранислав Боровац*
др Срђан Колаковић	*др Зоран Јеличић*
др Вељко Малбаша	*др Властимир Радоњанин*
др Вук Богдановић	*др Горан Вујић*
др Мила Стојаковић	*др Драган Спасић*
др Ливија Цветићанин	*др Дарко Реба*

Редакција :

др Владимир Катић	*др Драгољуб Новаковић*
др Жељен Трповски	*мр Мирослав Зарић*
др Зора Коњовић	*Мирјана Марић*

Штампа: ФТН – Графички центар ГРИД, Трг Доситеја Обрадовића 6

Техничка обрада: Графички центар ГРИД

Штампање одобрио: Савет за издавачко-уређивачку делатност ФТН у Н. Саду

Председник Савета: проф. др Радомир Фолић

CIP-Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

378.9(497.113)(082)
62

ЗБОРНИК радова Факултета техничких наука / главни и одговорни уредник Илија Ћосић. – Год. 7, бр. 9 (1974)-1990/1991, бр.21/22 ; Год. 23, бр 1 (2008)-. – Нови Сад : Факултет техничких наука, 1974-1991; 2008-. – илустр. ; 30 цм. –(Едиција: Техничке науке – зборници)

Двомесечно

ISSN 0350-428X

COBISS.SR-ID 58627591

ПРЕДГОВОР

Поштовани читаоци,

Пред вама је седма овогодишња свеска часописа „Зборник радова Факултета техничких наука“.

Часопис је покренут давне 1960. године, одмах по оснивању Машинског факултета у Новом Саду, као „Зборник радова Машинског факултета“, а први број је одштампан 1965. године. Након осам публикованих бројева у шест година, пратећи прерастање Машинског факултета у Факултет техничких наука, часопис мења назив у „Зборник радова Факултета техничких наука“ и 1974. године излази као број 9 (VII година). У том периоду у часопису се објављују научни и стручни радови, резултати истраживања професора, сарадника и студената ФТН-а, али и аутора ван ФТН-а, тако да часопис постаје значајно место презентације најновијих научних резултата и достигнућа. Од броја 17 (1986. год.), часопис почиње да излази искључиво на енглеском језику и добија поднаслов «Publications of the School of Engineering». Једна од последица нарастања материјалних проблема и несрећних догађаја на нашим просторима јесте и привремени прекид континуитета објављивања часописа двобројем/двогодишњаком 21/22, 1990/1991. год.

Друштво у коме живимо базирано је на знању. Оно претпоставља реорганизацију наставног процеса и увођење читавог низа нових струка, као и квалитетну организацију научног рада. Значајне промене у структури високог образовања, везане за имплементацију Болоњске декларације, усвајање нове и активне улоге студената у процесу образовања и њихово све шире укључивање у стручне и истраживачке пројекте, као и покретање нових дипломских-мастер докторских студија, доносе потребу да ови, веома значајни и вредни резултати, постану доступни академској и широј јавности. Оживљавање „Зборника радова Факултета техничких наука“, као јединственог форума за презентацију научних и стручних достигнућа, пре свега студената, обезбеђује услове за доступност ових резултата.

Због тога је Наставно-научно веће ФТН-а одлучило да, од новембра 2008. год. у облику пилот пројекта, а од фебруара 2009. год. као сталну активност, уведе презентацију најважнијих резултата свих дипломских-мастер радова студената ФТН-а у облику кратког рада у „Зборнику радова Факултета техничких наука“. Поред студената дипломских-мастер студија, часопис је отворен и за студенте докторских студија, као и за прилоге аутора са ФТН или ван ФТН-а.

Зборник излази у два облика – електронском на web сајту ФТН-а (www.ftn.uns.ac.rs) и штампаном, који је пред вама. Обе верзије публикују се више пута годишње у оквиру промоције дипломираних инжењера-мастера.

У овом броју штампани су радови студената мастер студија, сад већ дипломираних инжењера – мастера, који су дипломирали у периоду 01.06.2011. до 31.07.2011. год., а који се промовишу 21.09.2011. год. То су оригинални прилози студената са главним резултатима њихових дипломских радова. Неколико радова раније је објављено на конференцији ЕТРАН у Бањи Врућици и на конференцији SISY у Суботици. У Зборнику су ови радови дати као репринт уз мање визуелне корекције.

Велик број дипломираних инжењера–мастера у овом периоду био је разлог што су радови поводом ове промоције подељени у четири свеске.
У овој свесци, са редним бројем 7, објављени су радови из области електротехнике и рачунарства и мехатронике.

Уредништво се нада да ће и професори и сарадници ФТН-а и других институција наћи интерес да публикују своје резултате истраживања у облику регуларних радова у овом часопису. Ти радови ће бити објављивани на енглеском језику због пуне међународне видљивости и проходности презентованих резултата.

У плану је да часопис, својим редовним изласком и високим квалитетом, привуче пажњу и постане довољно препознатљив и цитиран да може да стане раме-уз-раме са водећим часописима и заслужи своје место на СЦИ листи, чиме ће значајно допринети да се оствари мото Факултета техничких наука:

„Високо место у друштву најбољих“

Уредништво

SADRŽAJ

Strana

Radovi iz oblasti: Elektrotehnika i računarstvo

1. Laslo Norbert, Miloš Živanov,
   OLED – ORGANSKA ELEKTROLUMINISCENTNA DIODA 1543
2. Богдан Тадић, Владимир Катић,
   ПРЕГЛЕД И КАРАКТЕРИСТИКЕ МИКРОМРЕЖА 1547
3. Филип Маличевић,
   РЕГУЛАЦИЈА НАПОНА У ДИСТРИБУТИВНИМ МРЕЖАМА СА ГЕНЕРАТОРИМА 1551
4. Jasenka Števanov,
   IMPLEMENTACIJA IPSSP PROTOKOLA ZA ELEKTRONSKE IGRE 1555
5. Miroslav Popov,
   AUTOMATIZOVANA KONFIGURACIJA ALFRESCO REPOZITORIJUMA 1558
6. Damir Vujaklija,
   OPTIMIZACIJA ŽELJENE VREDNOSTI TEMPERATURE RASHLADNOG FLUIDA CILERA PRIMENOM FAZI LOGIKE 1562
7. Jelena Kartal, Vladimir Katić,
   FARME VETRENJAČA NA MORU 1566
8. Bojan Cvetinov,
   UTICAJ PROMENE NAPONA NA SNAGU POTROŠAČA 1570
9. Risto Obradović,
   JEDNO REŠENJE GRAFIČKE APLIKACIJE ZA PRIKAZ I ANALIZU ISTORIJATA SCADA SISTEMA 1574
10. Saša Stojanović,
    ANALIZA RADA DISTANTNE ZAŠTITE U REALNOJ DISTRIBUTIVNOJ MREŽI 1578
11. Nemanja Paunović, Jelena Kovačević, Ivan Rešetar,
    JEDNO REŠENJE METODOLOGIJE ISPITIVANJA SLOŽENIH SISTEMA PROFESIONALNE ELEKTRONIKE, Konferencija ETRAN, Banja Vrućica, juni 2011. 1582
12. Siniša Božić,
    CYME APLIKACIJA ZA PRORAČUN KRATKIH SPOJEVA – OSNOVNE KARAKTERISTIKE ... 1586
13. Branislav Rodić, Vladimir Katić,
    SISTEM ZA ODREĐIVANJE KARAKTERISTIKA FOTONAPONSKIH PANELA 1590
14. Jovan Brkljača, Laslo Nađ,
    ELEKTRONIKA ZA UPRAVLJANJE HIDRAULIČNOM PRESOM ZA BRIKETIRANJE 1594
15. Nebojša Dragić,
    REALIZACIJE DVB-T2/SFN MREŽE PREDAJNIKA/REPETITORA 1598
16. Borislav Polić,
    DLNA PLEJER ZA OPERATIVNI SISTEM WINDOWS 1602
17. Karolina Valentović,
    NEKI PRAKTIČNI ASPEKTI PRENOSA IPTV SIGNALA 1606
18. Luka Komatina,
    IMPLEMENTACIJA WMS-WFS KLIJENTA ZASNOVANA NA PRISM PATERNU I WPF TEHNOLOGIJI 1609
19. Andraš Čizik,
    KOMPARACIJA ŠEMA HEŠIRANJA 1613
20. Siniša Knežević,
    AUTOMATIZACIJA RADA PEĆI NA BIOMASU PRIMENOM FUZZY LOGIKE 1617

21. Martina Joševska,
SISTEM ZA DIJAGNOZU KVAROVA NA KUGLIČNIM LEŽAJEVIMA 1620
22. Ivan Babić,
DIGITALNO MERILO BROJA UDISAJA 1624
23. Marko Dojčinović,
PROŠIRENJE OPERATERSKE KONZOLE SCADA OASyS DNA SISTEMA C#.NET PROGRAMSKIM SLOJEM ZA KONTROLU SLOŽENIH KORISNIČKIH PRIKAZA 1628
24. Perica Dejanović,
INTEGRACIJA LON SERVERA SA SISTEMOM KUĆNE AUTOMATIKE 1632
25. Dragan Miljković, Ljubiša Gajić, Aleksandar Kovačević, Zora Konjović,
THE USE OF DATA MINING FOR BASKETBALL MATCHES OUTCOMES PREDICTION; konferencija SISY, Subotica, 2010 1636
26. Dragan Krajnović,
RAZVOJ OPC-DA MODULA ZA INTEGRACIJU HETEROGENIH SISTEMA SA RADOM U REALNOM VREMENU 1640
27. Nataša Majstorović,
OBUČAVANJE NEURONSKIH MREŽA PRIMENOM LEVENBERG-MARQUARDT ALGORITMA NA PRIMERU FUNKCIJE PRENOSA 1644
28. Zorica Nikolić,
SISTEM ZA ESTIMACIJU MEHANIČKIH PARAMETARA HODA 1648
29. Ivana Todosijević,
OBUČAVANJE NEURONSKIH MREŽA PRIMENOM BAYESIAN REGULATION ALGORITMA NA PRIMERU FUNKCIJE PRENOSA 1652
30. Aleksandar Veselinović,
PODELA GRAFA EVOLUTIVNIM PSO ALGORITMOM 1656
31. Dejan Čotra,
DINAMIČKA PRERASPODELA MODELA DISTRIBUTIVNE ELEKTROENERGETSKE MREŽE PRIMENOM WF I LMSR ALGORITAMA 1660
32. Suzana Kuburić,
ANALIZA I PRIMENA VIŠEFAZNOG ALGORITMA ZA PODELU MODELA PODATAKA ZASNOVANOG NA GRAFOVIMA 1664
33. Jovana Lugonja,
SSH PROTOKOL 1668
34. Петар Зобеница,
PET/CT У НУКЛЕАРНОЈ МЕДИЦИНИ 1672
35. Ivan Perić,
RAZVOJ SISTEMA ZA AUTOMATIZOVANO TESTIRANJE SOFTVERA ZA MULTIMEDIJALNU VIZUALIZACIJU 1676
36. Petar Vasilić,
KOMPRESIJA VREMENSKE SERIJE LINEARNIM INTERPOLACIONIM METODOM 1680
37. Vladimir Ber,
PRISTUP VREMENSKIM SERIJAMA UPOTREBOM OPC UA INTERFEJSA 1684
38. Igor Javni,
EKSTERNA AUTENTIFIKACIJA PO MODELU IDENTITETA ZASNOVANOM NA TVRDNJAMA 1688
39. Goran Knežević,
OPTIMIZACIJA POTROŠNJE ELEKTRIČNE ENERGIJE U STAMBENO - POSLOVNIM OBJEKTIMA 1692
40. Daniel Tertei,
KLASTEROVANJE KOMANDI X10 PROTOKOLA KUĆNE AUTOMATIKE PRIMENOM SUBTRACTIVE CLUSTERING ALGORITMA I NEUROFAZI STRUKTURE 1696
41. Aleksandra Vuković – Rukavina, Josif Tomić,
REALIZACIJA PID ALGORITMA U LabVIEW PROGRAMSKOM PAKETU 1700

42. Nikola Potrebić,
SOFTVER ZA EVIDENCIJU STIPENDISTA 1704
43. Горан Шмит,
СОФТВЕРСКО ОКРУЖЕЊЕ ЗА ВИЗУЕЛНО ПРАЋЕЊЕ SCADA СИГНАЛА У SINAUT SPECTRUM СИСТЕМУ 1708
44. Vanja Radovanović, Vladimir Katić,
FARME VJETRENJAČA U VOJVODINI – PREGLED POGODNIH LOKACIJA 1712
45. Miloš Vasić, Dragan Samardžija, Miodrag Temerinac,
EKSPERIMENTALNA EVALUACIJA RADIOFREKVENCIJSKE METODE ZA ODREĐIVANJE ORIJENTACIJE DALJINSKOG UPRAVLJAČA; ETRAN, Banja Vrućci, jun 2011. 1716
46. Mirjana Vulin,
PROJEKTOVANJE EMBEDDED DIGITALNOG ELEKTROKARDIOGRAFA 1720
47. Mladen Karaklić,
DVOKANALNI DIGITALNI ELEKTROENCEFALOGRAF 1724
48. Slobodan Lolić, Vladimir Katić,
SOLARNI IZVORI ELEKTRIČNE ENERGIJE - PREGLED REŠENJA ZA MALE POTROŠAČE ... 1728
49. Dragoljub Grubović,
JEDNOSTAVNI SCENARIJI ZA TESTIRANJE PERFORMANSI DMS SOFTVERA 1732
50. Anđelko Pikula,
NAGIB SPEKTRA SIGNALA SRČANOG RITMA 1736
51. Milan Milanko, Vladimir Katić, Zoltan Čorba,
IDEJNO REŠENJE HIBRIDNOG VETRO-FOTONAPONSKOG SISTEMA NA ZGRADI NTP FTN..... 1740
52. Igor Bende,
IZRADA SCADA APLIKACIJE, MATEMATIČKO MODELIRANJE I RAZVOJ FUZZY REGULATORA VODOVODNOG SISTEMA..... 1744
53. Borivoj Jovišić,
PRIMER REALIZACIJE FUZZY REGULATORA ZA UPRAVLJANJE SISTEMOM GREJANJA/HLAĐENJA U STAMBENOM OBJEKTU 1748
54. Nebojša Apostolovski,
LON IP GATEWAY KAO KOMUNIKACIONI INTERFEJS KA SISTEMIMA KUĆNE AUTOMATIKE 1752
55. Srđan Farkaš,
UPOTREBLJIVOST VIRTUELNIH TASTATURA NA MOBILNIM UREĐAJIMA SA EKRANIMA OSETLJIVIM NA DODIR 1756
56. Darko Lauš,
DIGITALNA ZEMALJSKA TELEVIZIJA 1760
57. Aleksandra Krneta,
APLIKACIJA ZA ADMINISTRIRANJE SECURITY PODSISTEMA DMS SOFTVERA 1764

Radovi iz oblasti: Mehatronika

1. Дејан Бирманчевић,
АКВИЗИЦИЈА ПОДАТАКА БЕЖИЧНИМ ПУТЕМ КОРИШЋЕЊЕМ ПДА УРЕЂАЈА 1768
2. Ђорђе Васић, Растислав Шостаков,
АНАЛИЗА ЗАЛЕТА ПОГОНА ТРАКАСТИХ ТРАНСПОРТЕРА 1772
3. Kosta Žarkov, Dragan Šešlija,
OPTIMIZACIJA POTROŠNJE VAZDUHA POD PRITISKOM NA LINIJI ZA PROIZVODNJU LIMENKI 1776
4. Milan Vasiljev, Dragan Šešlija,
MODERNIZACIJA KOLONE I SISTEMA VAZDUHA POD PRITISKOM U RAFINERIJI 1780
5. Predrag Dimković,
SISTEM ZA DALJINSKI NADZOR AUTOMOBILA 1784

6. Ferid Sejdović,
SOFTVER ZA ZAKAZIVANJE KORIŠĆENJA PARKING PROSTORA NA UNIVERZITETU U NOVOM SADU 1788

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.38:535

OLED – ORGANSKA ELEKTROLUMINISCENTNA DIODA

OLED – ORGANIC LIGHT EMITTING DIODE

Laslo Norbert, Miloš Živanov, Fakultet tehničkih nauka, Novi Sad

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu opisane su organske elektroluminiscentne diode, mali molekulni OLED i polimerni OLED. Prikazana je njihova struktura, materijali od kojih su izrađeni, kao i njihove elektrode. Takođe prikazana je njihova primena u SSL (tehnologija rasvete čvrstog stanja) i u ravnim panel monitorima.*

Abstract – *This paper describes the organic electroluminescent diodes, small molecular OLED and polymer OLED. It's shown their structure, the materials of which they were made, and their electrodes. It is also shown their application for SSL (solid state lighting technology) and flat panel monitors.*

Ključne reči: *organska elektroluminiscentna dioda (OLED), SSL, ravni panel monitori*

1. UVOD

Svake godine nas impresionira sve veći broj tehnoloških inovacija. Ipak, sa pravom se može reći da su ta tehnološka "čuda" veoma skromna u odnosu na ono što tek dolazi. Pogled na samo deo budućnosti tehnologije izrade elektronskih komponenata i elektronskih naprava fascinira, pa je praktično nezahvalno prognozirati šta će se sve i kako u budućnosti proizvoditi. Kada je reč o poluprovodničkim komponentama, osnovni cilj će i dalje biti smanjivanje dimenzija istih, sve do nivoa molekula, pa čak i atoma. Elektronika bazirana na takvim komponentama već sada se zove molekularna elektronika. Pored znatno većeg stepena integracije, odnosno izuzetno većeg broja aktivnih komponenata na čipu, smanjivanje dimenzija komponenata dovešće do daljnjeg povećanja brzine njihovog rada. Naravno, postojeći poluprovodnički materijali, prvenstveno silicijum, biće zamenjeni drugim, tako da će i principi rada sadašnjih komponenata biti drugačiji. Izazov koji se postavlja pred tehnologije koje se kandiduju da zamene silicijumski mikroprocesor je prevazilaženje granice u veličini i spuštanju elektronike na nivo molekula i atoma. Takve komponente su po svojoj prirodi fundamentalno različiti od onih sa čime se poluprovodnička tehnologija do sada suočavala. Atomi su, uslovno, elementarne čestice od kojih je sačinjen svaki materijal u svetu oko nas. Njihovo ponašanje opisano je zakonima kvantne mehanike, što predstavlja suštinu problema pri svakom pokušaju da se manipuliše tako malim objektima [1].

NAPOMENA:
Ovaj rad proistekao je iz diplomskog - master rada čiji je mentor dr Miloš Živanov, red. prof.

2. KOMPONENTE BAZIRANE NA ORGANSKIM POLUPROVODNICIMA

Optoelektronske komponente bazirane na organskim poluprovodnicima otvaraju nove mogućnosti za veliku raznovrsnost sistema i primene. To je širok opseg raznih organskih poluprovodničkih komponenata, kao što su izvori svetlosti, fotodiode, solarne ćelije, tranzistori, kao i organski laseri. Prilagodljivost organskih komponenti i njihova primena u integrisanim sistemima će privući pažnju na ovu klasu materijala.

2.1. OLED (organska elektroluminiscentna dioda)

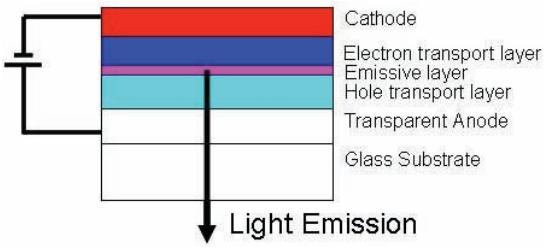
OLED je specijalni tip LED diode. OLED se sastoji od najmanje jednog nedopiranog organskog sloja koji je smešten između dve elektrode, providne anode i metalne katode. Elektroni i šupljine se ubacuju iz obe elektrode, respektivno. Posle primene napona (tipično 2-10 V) oba nosioca kreću prema drugoj strani u primenjenom električnom polju. Kada se elektroni i šupljine sudare, oni se rekombinuju i formiraju eksitone (eksitoni su vezana stanja elektrona i šupljina, koja se međusobno privlače elektrostatičnom Kulonovom silom), čijim se raspadanjem emituje svetlost [3]. OLED su komponente za pretvaranje energije (električne u svetlosnu) bazirane na elektroluminiscenciji. OLED su prvi put privukli pažnju istraživača 1960. godine zbog njihovog potencijala visoke kvantne efikasnosti luminiscencije i mogućnosti da generiše velike varijacije boje. Na nesreću, visok napon na kom su radili (preko 1000 V), je doprineo da se zabrane da se koriste u praksi [2].

2.1.1. Mali molekulni OLED

a) struktura komponente

Mali molekulni OLED se sastoji od jednog ili više organskih slojeva između dve elektrode, anode i katode. Organski materijal mora da služi za tri glavne funkcije: prenos (transport) elektrona, prenos šupljina i emisiju svetlosti. Emisija svetlosti se dobija iz radijativne rekombinacije elektrona i šupljina. U ovom slučaju, stopa ubacivanja oba nosioca treba da bude jednaka da bi se postigla visoka efikasnost. U suprotnom, višak elektrona ili šupljina neće se rekombinovati, što rezultuje niskom radnom efikasnošću.

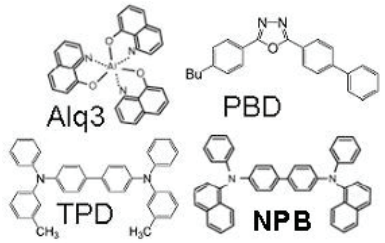
Ove mane se mogu prevazići korišćenjem višestrukih organskih slojeva u OLED strukturi. Osnovna struktura heteroslojne OLED komponente je prikazana na slici 1. Ova struktura ima sloj za transport elektrona (ETL), sloj za transport šupljina (HTL) i sloj za emisiju. Materijal za emisiju može da bude ili sloj između ETL i HTL ili dopant u jednom od ovih slojeva, blizu zone za rekombinaciju.



Slika 1. *Struktura OLED-a*

b) materijali

Tipični materijali n-tipa korišćeni za ETL su: Alq3, PBD, itd. Tipični materijali p-tipa korišćeni za HTL su: NPB, TPD, itd. Njihove hemijske strukture su prikazane na slici 2. Obično korišćeni emiteri su fluoroscentne boje ili fosforne boje.



Slika 2. *Hemijske strukture malih molekula*

c) elektrode

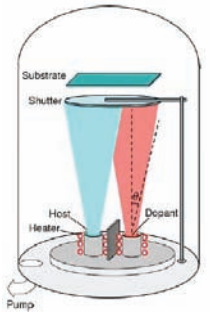
Učinak diode takođe veoma zavisi od materijala od kog je elektroda napravljena. Uopšte radna funkcija katode treba da je dovoljno niska, u cilju da imamo dobro ubacivanje oba nosioca, respektivno. Kod OLED-a anoda treba da je providna da propusti emisiju svetlosti iz komponente.

Zasad, najčešće korišćena providna anoda je ekstremno tanki sloj indijum-kalaj-oksid (ITO), koji ima relativno visoku radnu funkciju (4.5-5.1 eV).

Radna funkcija katode treba da je dovoljno niska, najčešće korišćeni materijali za katodu su: Al, Mg, Ca, itd. Ipak, ovi metali korodiraju i oksidiraju u vazduhu. Zbog toga, legirane elektrode daju veću stabilnost i relativno nižu radnu funkciju. Na primer, legure Mg i Ag su stabilniji od samog Mg.

d) izrada

Toplotno naparavanje u visokom vakuumu je korišćeno za naparavanje ovih malih organskih molekula. Proces je sublimacija materijala, uključujući organske poluprovodnike i metale, u vakuumu iz zagrejane posude na dnu, na podlogu na vrhu, šema je prikazana na slici 3. Vakuum je na oko 10^{-5} do 10^{-7} torr-a (1 torr = 133.32 Pascal).[3]



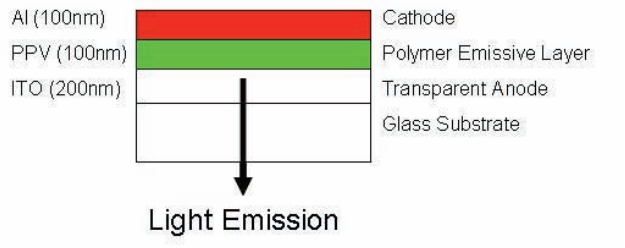
Slika 3. *Šematski prikaz procesa termalnog naparavanja u vakuumu*

2.1.2. Polimerni LED (PLED)

a) struktura komponente

Polimeri korišćeni u PLED su generalno konjugovani polimeri, zato što ova vrsta strukture obezbeđuju visoku konduktivnost (provodnost). Polimeri su sastavljeni od velike količine ponavljajućih sličnih molekula sa ogromnom molekularnom težinom. Prvi polimer koji je korišćen u PLED je poli(para-fenilen vinilen) (PPV). Hemijska struktura je prikazana na slici 5. Struktura ovog polimernog LED je prikazana na slici 4., koji je skoro isti kao kod OLED-a od malih molekula. PPV ima energetski procep od otprilike 2.7 eV, koji emituju zelenu svetlost u oblasti spektra.

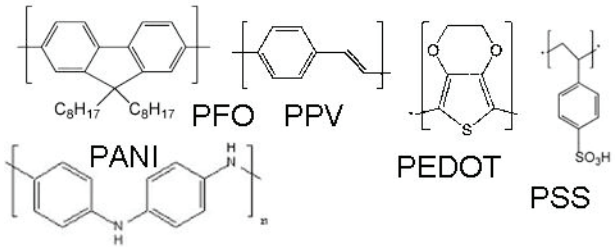
Pri primenjenom naponu ubačeni elektroni i šupljine se rekombinuju u sloju polimera i emituju svetlost. Sloj polimera je oko 100 nm, treba da obezbedi efikasno ubacivanje nosioca i transport nosioca kroz sloj. Struktura komponente takođe može biti i dvoslojna sa dodatnim slojem polimera za provođenje, između ITO i emisionog sloja. Sloj polimera za provođenje će poboljšati ubacivanje šupljina i blokiranje elektrona.



Slika 4. *Konfiguracija OLED-a*

b) materijali

Od konjugovanih polimera najčešće korišćeni polimeri za provođenje su: PANI:PSS i PDOT:PSS, a emisioni polimeri su: R-PPV i PFO. Hemijske strukture ovih polimera su prikazane na slici 5.



Slika 5. *Hemijske strukture polimera*

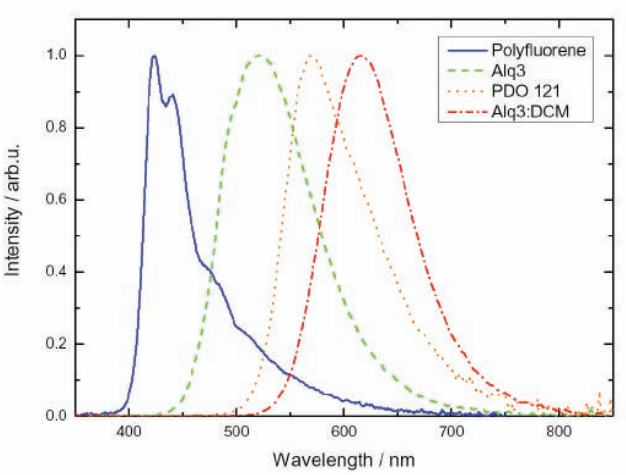
c) izrada

Polimeri ne mogu koristiti isti metod kao mali molekuli, jer polimeri imaju veoma veliku molekularnu težinu. Dakle, polimeri se obično deponuju rastvaranjem u organskom rastvaraču, zatim obrtnim nanošenjem (spin-coating), livenjem kapanjem (drop-casting), ink-džet štampanjem, ili roll to roll web premaz.[3]

2.2. Karakterizacija OLED

Komponente su testirane u računarski kontrolisanom sistemu za karakterizaciju. Ovaj sistem se sastoji od postolja koji se automatski rotira sa držačem uzorka, spektrometra za optička vlakna (StellarNet EPP2000) i jedinice izvora merenja (Keitlhey SMU 238). Instalacija je kalibrisana koristeći standard svetlosnog zračenja (1000 W FEL type) i pruža precizne spektroradiometrijske i fotometrijske veličine. Najbitnije veličine za sen-

zor su ukupan fluks zračenja (W), zračenje eksitanse (W/m^2) i spektar emisije OLED.
Spektralne karakteristike različitih OLED su prikazani na slici 6. Svi OLED pokazuju širok spektar organskog emiterskog materijala. Slično kao u slučaju lasera u boji, ceo vidljivi spektar se može pokriti različitim materijalima. Kao plavi emiter biramo polifluoren sa maksimumom emisije na otprilike 420 nm. Zeleni emiter je nedopiran Alq3, pa je maksimum emisije na talasnoj dužini od otprilike 520 nm. Za narandžasti deo spektra, se mogu koristiti OLED bazirani na polimeru PDO 121, dok se dublja crvena boja u komponentama generiše koristeći mali molekulni Alq3 dopiran sa DCM laserom u boji kao dodatni emisioni materijal.[4]



Slika 6. *Spektar elektroluminiscencije različitih OLED*

3. PRIMENA ORGANSKIH POLUPROVODNIČKIH KOMPONENATA

3.1. Osvetljenje uz pomoć OLED

Možda je jedan od najznačajnijih primena OLED što se tiče čovečanstva, jer ako istraživači dođu do najpogodnijih rešenja u ovoj oblasti, može doprineti u znatnom smanjenju potrošnje električne energije, samim tim smanjenje zagađenja vazduha i očuvanje životne okoline.
Od svih zemalja, SAD je najveći potrošač energije. 2001. godine otprilike 36 Quad (1 Quad = 293,071,000,000 Kilowatt-sati (kWh)) električne energije je proizvedeno u SAD po ceni od 252 biliona dolara. 20 % te energije, što predstavlja 7.2 Quad se koristilo za osvetljenje i koštalo je 50 biliona dolara. Postojeći izvori svetlosti su neefikasni. Trenutno, samo oko 30 % energije utrošene za opštu rasvetu se koristi za generisanje svetla, ostalo se gubi u toploti. Sijalice sa užarenim vlaknom, troše oko 45 % od ukupne energije za osvetljenje, i proizvede samo 14 % svetlosti, izmereno u lumenima. Oko 90 % energije odlazi na proizvodnju toplote. Fluoroscentne lampe su oko četiri puta energetski efikasnije, ali i dalje, značajan deo korišćene energije se rasipa. Ostali izvori svetlosti: halogene lampe, lampe visokog intenziteta pražnjenja (HID), imaju ograničenu upotrebu i nisu mnogo efikasniji od fluoroscentnih lampi. U poslednjih 30-50 godina nije napravljen značajni napredak u energetskoj efikasnosti konvencionalnih izvora svetlosti. Ovi izvori su dostigli tehnološku zrelost, malo se može uraditi da se učine efikasnijim. Zato, nove tehnologije sijalica su poželjne i potrebne [2].

3.1.1. Potencijal SSL (tehnologija rasvete u čvrstom stanju)

Jedna od novih tehnologija osvetljenja koja se pojavila u poslednje dve dekade i ima potencijal da postane energetski efikasnija od postojećih izvora osvetljenja, jeste SSL (Solid State Lighting), tehnologija rasvete u čvrstom stanju koje koriste OLED komponente.
Dostupni podaci o OLED-u i tehničke projekcije pokazuje da iznos energije potrebne da generiše isti iznos svetlosti može se eventualno smanjiti i do 50 %. Ako se ostvari ušteda energije potrebne za osvetljenje od 50 %, to će dovesti do velike uštede svakog društva. Pored uštede, manja *potrošnja* energije će dovesti do manje *proizvodnje* energije, zbog toga i manje zagađenje vode i vazduha. Prema poslednjim izračunavanjima ako se upotreba električne energije smanji za 50 % do 2020. godine, emisiju ugljenika u atmosferu ćemo smanjiti za 45 miliona tona godišnje. Potencionalna ušteda takođe zavisi kako brzo i kako ćemo uspeti do proširiti ove napredke. Iako tehnološki napredak dovede do značajnog smanjenja energije, tržište neće prihvatiti SSL osim i ako se smanji i cena. Ako SSL ostvari niže cene, mnogo više energije će se sačuvati.
Danas, sijalice sa užarenim vlaknom dominiraju na tržištu sijalica za domaćinstva i industriju gde su glavni razlozi inicijalna cena i estetika. Fluoroscentne lampe se koriste u komercijalnom sektoru, gde su principski razlozi kombinacija cene izvora svetlosti (tj. lampe) i potrošena energija.
OLED-i su nekonvencionalni tanki filmovi velike površine, skoro dvodimenzionalne komponente. Oni su distribuirani (difuzioni) izvori svetlosti, jasno drugačiji od tačkastih izvora kao što su sijalice. Takođe, OLED rade na veoma niskim naponima, reda 3-5 V. Dok su značajna istraživanja i dalje potrebna, OLED po postignutoj efikasnosti će uskoro moći da se takmiči sa užarenim izvorima (sijalicama).
Eksperimentalni OLED su već energetski efikasniji od lampi sa užarenim izvorima. Svetlosna efikasnost sijalice je oko 13-20 lm/W, a poslednji eksperimentalni OLED koji emituju zelenu svetlost imaju svetlosnu efikasnost od 76 lm/W. Razvoj je na pravom putu za OLED da se efektno takmiče čak i sa fluoroscentnom lampom, koja ima svetlosnu efikasnost 50-100 lm/W.
Jedna velika prednost OLED-a je mogućnost da se podesi bilo koja boja za emisiju svetlosti, i bilo koja nijansa i intenzitet, uključujući i belu. Postizanje visokog indeksa prikaza boje (CRI) blizu 100 (mogućnost da simuliraju najprijatniju belu boju, dnevnu svetlost) je već u okviru domašaja OLED. Ostale prednosti OLED je da su komponente sa trenutnim pogonom, gde osvetljaj može da varira preko veoma širokog dinamičkog opega i oni ravnomerno rade bez treperenja. Ovo sve kreira veliki optimizam da će OLED biti prihvaćeni i dobrodošli od strane javnosti – sve dok su jeftini.
Još jedna prednost OLED je da se mogu deponovati na bilo koju podlogu: staklo, keramiku, metal, tanke plastične listove, tkanine, fleksibilne i komforne podloge, itd. Zbog toga se mogu praviti u bilo kom obliku i dizajnu. Ovo otvara nove arhitektonske i dizajnerske mogućnosti. Sloboda da se proizvede izvor bilo kog oblika ili boje kreiraće radikalnu novu kulturu osvetljenja.

U suštini, OLE-i imaju potencijal da budu izvori osvetljenja, velike površine koji su: svetli, energetski efikasni, dugotrajni, emituju prijatnu belu svetlost, takođe su ultra tanki, male težine, neravni, konforni i jeftini [2].

3.2. Monitori bazirani na OLED

Ova tehnologija je zasad jedina poznata tehnologija koja može da konkuriše LCD monitorima. Postoje mnogi faktori koji čine da ova tehnologija u budućnosti zauzme mesto LCD.

OLED monitori nude veći kontrast, življe boje, veći odsjaj, širi ugao gledanja, bolju temperaturnu toleranciju i brže vreme odziva, nego što nude LCD monitori. Upoređivanje ovih osobina se vidi u tabeli 1. [5].

Tabela 1. *Upoređivanje osobina LCD i OLED monitora*

Monitori sa tečnim kristalima (LCD)	OLED monitori
- zahtevaju poseban izvor svetlosti	- emituju svetlost
- slika zavisi od ugla gledanja	- slika ne zavisi od ugla gledanja
- sporo vreme odziva(ms)	- brzo vreme odziva(μs)
-niski radni napon (5-15 V)	- niski radni napon (5-15 V)
-fotometrijska efikasnost 2.5 lm/W	- fotometrijska efikasnost 40 lm/W

Pošto monitori bazirani na OLED tehnologiji ne moraju da imaju pozadinsko osvetljenje, imaju potencijal da budu tanji, lakši i fleksibilniji nego LCD. Trenutno se najviše koriste kao ekrani mobilnih telefona, mp3 plejera, u digitalnim kamerama i PDA. Mogu se koristiti i za velike ekrane i fleksibilne ekrane slične papiru. Na slici 7. prikazan je primer korišćenja savitljivog displeja.[1]



Slika 7. *Primer primene savitljivog displeja*

Međutim, OLED-i još nisu spremni da u bliskoj budućnosti sa tržišta potisnu LCD-ove, pošto problemi njihove komercijalizacije i uspešne inkorporacije u proizvode masovne upotrebe još nisu rešeni. Mada proizvodnja OLED-a ima velike potencijalne prednosti, dostizanje tih prednosti zahteva prevazilaženje nekih suštinskih prepreka. Za OLED ne postoji standardizovana proizvodna oprema, niti oprema za testiranje. Mašine, senzori i ostala oprema još su u fazi projektovanja i razvoja. Proizvodne linije tek treba da se izgrade, a proizvodne tehnike da se optimizuju. Velik deo proizvodnje OLED-a još uvek je u nezreloj fazi.

4. ZAKLJUČAK

U cilju dalje minijaturizacije poluprovodničkih komponenti baziranih na poluprovodnicima kao što je silicijum, jednog dana će doći do granice kada se više na može smanjivati. Do tada naučnici moraju pronaći nove tehnologije koje će uspešno zameniti postojeće poluprovodnike. Trenutno postoje mnoge ideje i izvode se razni eksperimenti, međutim nekako se najviše ističu komponente bazirane na organskim materijalima.

U ovoj oblasti se već sada ulažu ogromna sredstva u razvijenim zemljama za istraživanje i razvoj različitih komponenti. One u različitim oblastima primene pokazuju odlične rezultate. Naročito raduje što se polako odstupa od toga da vojne potrebe pokreću industriju, odnosno da je u budućim tehnologijama čovek najbitniji i način da se omogući komforniji, lakši život ljudi.

OLED tehnologija ima potencijal da bude sledeća generacija monitora i sijalica za osvetljenje. Neke kompanije su već demonstrirali određene uređaje kao što su mobilni telefoni, mp3 plejeri i PDA čiji ekrani su u potpunosti od OLED komponenti. Postoje već i televizori, monitori za računare i laptopovi koji koriste ovu tehnologiju. Što se tiče osvetljenja tu postoje različiti problemi, bez čijeg rešavanja OLED ne može biti konkurentan trenutnim izvorima svetla. Njihova velika prednost bi mogla biti mogućnost da emituju belu svetlost kao što je sunčeva svetlost. To nam omogućava organska hemija. Ako bi naučnici rešili sve probleme osvetljenja baziranog na organskim poluprovodnicima i uspešno se sprovela komercijalizacija, to bi čovečanstvu donelo značajne uštede energije.

5. LITERATURA

[1] Stojan Ristić, "Poluprovodničke komponente", Elektronski fakultet Niš, 2010

[2] Milan Stolka, "Organic Light Emitting Diodes(OLEDs) for General Illumination Update 2002", OIDA Optoelectronics Industry Development Association, www.OIDA.org

[3] Minlu Zhang, "Organic Optoelectronic Devices – Organic Light-Emmiting Diodes (OLED)", 2006

[4] Martin Punke, Steffen Mozer, Marc Stroisch, Martina Gerken, Georg Bastian and Uli Lemmer, "Organic semiconductor devices for micro-optical applications", Universität Karlsruhe (TH), Light Technology Institute (LTI), Germany, 2006

[5] Profesor Zoran Popović, "Organske poluprovodničke naprave", Xerox Research Centre of Canada, XRCC, 2007

Kratka biografija



Norbert Laslo rođen je 1985. godine u Zrenjaninu. Završio je Srednju elektrotehničku školu "Nikola Tesla" u Zrenjaninu. 2004. godine je upisao studije na FTN, smer mikroračunarska elektronika.



Miloš Živanov rođen je 1948. godine u Novom Sadu. Doktorirao je na Elektrotehničkom fakultetu u Beogradu 1992. god., a od 2004. god je u zvanju redovnog profesora na FTN. Oblast interesovanja mu je elektronika i optoelektronika.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

УДК: 621.31

ПРЕГЛЕД И КАРАКТЕРИСТИКЕ МИКРОМРЕЖА

OVERVIEW AND CHARACTERISTICS OF MICROGRIDS

Богдан Тадић, Владимир Катић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *Објашњена је улога микромрежа у модерном електроенергетском систему, орјентација ка обновљивим изворима. Представљен је концепт микромрежа – основне идеје, технологије, техничко-економске предности, као и могући проблеми. Дати су неки примјери коришћења или истраживања микромрежа у свијету.*

Abstract – *The paper presents the role of microgrids in modern power systems and the orientation towards renewable sources. The concept of microgrids are presented – basic ideas, technologies, technical and economic advantages and possible problems. Some examples of microgrids usage and research in the world are given.*

Кључне речи: *Микромреже,микроизвори.*

1. УВОД

Пред енергетски сектор у двадесет и првом вијеку постављају се три међусобно повезана изазова. Један се односи на повећање еколошке ефикасности, при чему се мисли на смањење емисије гасова стаклене баште. Други се односи на повећање техничке ефикасности у производњи, преносу и потрошњи електричне енергије. Трећи обухвата развијање обновљивих извора енергије, односно припрему за пост-нафтни период. Наведени проблеми су захтијевали глобално ријешење. Промјене у природи, посебно оне које су последица емисије гасова, услед чега настаје ефекат стаклене баште, захтевале су акцију земаља на међународном нивоу. Рјешење се види у употреби микромрежа чији су основни елементи обновљиви извори енергије као што су PV низови,вјетроелектране,мале хидроелектрана итд.

Уз либерализацију и дерегулацију појединх тржишта долази постепено до глобализације њихова дјеловања, које не мимоилази ни сектор електроенергетике. Један од циљева је и налажење мјере између увођења конкуренције и обавезе јавног снабдијевања електричном енергијом. Пренос и дистрибуција електричне енергије су природни монополи, док се за производњу и снабдијевање успоставља тржиште. У складу с тим, дошло је и до потребе за формирањем нових институција као што су регулаторне агенције у области енергетике. Основне активности регулаторних агенција обухватају регулисање преноса и дистрибуције, а у циљу обезбеђивања сигурне, поуздане и квалитетне испоруке електричне енергије корисницима преносног, односно дистрибутивног система. Са друге стране регулаторне агенције имају задатак да подстичу развој тржишта у дјелатностима које имају потенцијал за развој конкуренције.

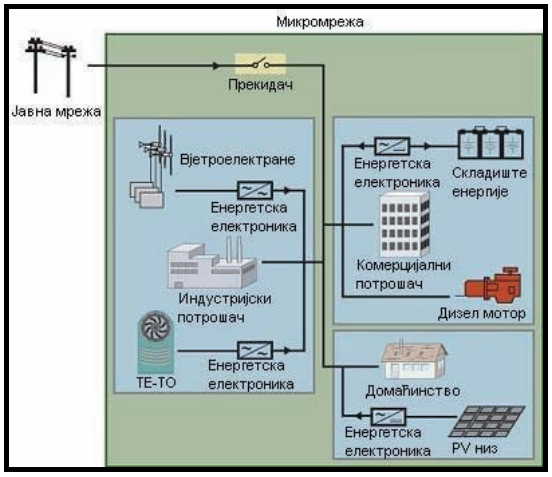
Са аспекта купаца, последица раздвајања регулисаних и тржишних дјелатности је да квалификовани купац бира снабдијевача на слободном тржишту. Са појавом берзе и трговине струјом такође се отвара могућност за већим коришћем мокромрежа. Тиме се ствара конкуренција конвенционалним произвођачима струје и у блиској будћности можемо очекивати струју по јефтинијој цијени.

НАПОМЕНА:
Овај рад је проистекао из мастер рада чији је ментор био проф. др Владимир Катић.

2. КОНЦЕПТ МИКРОМРЕЖА

2.1.Основне идеје микромрежа

Микромреже су мале нисконапонске, комбиноване топлотно-струјне мреже за снабдијевање електричном и топлотном енергијом малих средина, попут стамбених блокова и градских насеља, или академских и јавних области, као што су универзитети и школе, комерцијални квартови, индустријске зоне, трговачки центри или општинске регије. Микромреже могу обухватити и мања насеља, предграђа, села и сеоске атаре. На слици 2.1 представљен је основни концепт микромреже. Види се јавна мрежа и микромрежа које спаја прекидач. Микромрежа може да ради заједно са јавном мрежом а може и острвским начином рада. Микромрежу чине вјетроелектране, PV низ, комбиновани топлотно-енергетски системи (ТЕ-ТО), складиште енергије и пратећа енергетска електроника. Потрошачи електрицне енергије су домаћинства, индустријски и комерцијални потрошачи.



Слика 2.1 *Концепт микромреже*

Генератори или микроизвори који се у некој микромрежи примењују обично су обновљиви/ неконвенционални дистрибуирани енергетски ресурси, који се прикључују на средњи напон и који су интегрисани у једну цјелину ради генерисања струје дистрибутивног напона. Са оперативног становишта, микроизвори треба да буду опремљени енергетско-електронским интерфејсима и регулаторима да би се остварила неопходна флексибилност за функционисање у облику једног интегрисаног система, као и за одржавање специфікованог квалитета и количине струје.

Кључне разлике између микромреже и конвенционалне електране су следеће:

1. Капацитет микромрежа је много мањи у односу на капцитет великих генератора у конвенционалним електранама,
2. Струја која је генерисана на дистрибутивном напону може директно да се унесе у јавну дистрибутивну мрежу,
3. Микроизвори су обично инсталирани близу мјеста гдје се потрошачи налазе, тако да је струју/топлоту могуће ефикасно испоручивати уз задовољавајући напонски и фреквентни профил и занемарљиве губитке у далеководу.

Техничка својства микромреже омогућују испоруку енергије удаљеним областима земље, гдје је снабдијевање из јавне дистрибутивне мреже или отежано због топологије или поремећено због лоших климатских услова или поремећаја које изазива човјек. Може да функционише као једно агрегатно напајање. Тиме је гарантована могућност лаког контролисања, као и сагласност са законима и прописима о мрежама, без угрожавања поузданости и безбиједности енергетског система. Са становишта потрошача, микромреже су корисне у погледу задовољавања локалних захтева за струјом/топлотом. Могу непрекидно да испоручују енергију, побољшавају поузданост на локалном нивоу, редукују губитке на напајању и олакшавају одржавање локалног напона. Са еколошког становишта, путем примене нискоугљеничне технологије, микромреже смањују загађење животне средине и глобално загревање.

2.2. Техничке и економске предности микромрежа

Развој микромрежа може да буде од велике користи за електроенергетску индустрију, и то због следећих предности:

Еколошка питања – Због коришћења обновљивих извора, утицај микромрежа на животну средину много је мањи од утицаја великих конвенционалних термоелектрана. Неке од предности микромрежа у том погледу су следеће:

- Смањење емисије гасова и честица због непосредне контроле над процесима сагоревања, може да помогне у борби против глобалног загријевања.
- Физичка близина потрошача микроизворима може да допринесе повећању њихове свијести о рационалној потрошњи енергије.

Оперативна и инвестициона питања – Смањење физичке и електричне удаљености између микроизвора и оптерећења може да допринесе: побољшању реактивне подршке цјелокупном систему, мањем нагомилавању напајања за пренос и дистрибуцију, смањењу губитака код преноса и дистрибуције на око 3%, смањењу/одлагању потребе за улагањем у проширење система преноса.

Квалитет струје – Побољшање у квалитету и поузданости струје остварује се захваљујући: децентрализацији испоруке, бољом усклађеношћу понуде и потражње, смањењу утицаја великих испада у преносу и генерисању, минимизирању времена испада и побољшању процеса поновног успостављања путем блек-старт операција микромрежа.

Уштеде у трошковима – Коришћењем микромрежа остварују се следеће уштеде у трошковима:

- До значајних уштеда долази путем коришћења отпадне топлоте у комбинованом (ТЕ-ТО) начину рада.
- Уштеде у трошковима се остварују и кроз интеграцију неколико микроизвора.

Тржишна питања – У случају тржишног учешћа, остварују се следеће предности: микромреже се могу искористити за пружање помоћних услуга, распрострањена примена модуларних микроизвора на брзо укључивање и покретање може да допринесе смањењу цијене енергије на енергетском тржишту, одговарајућа економска равнотежа између инвестирања у мрежу и примјене дистрибуираног генерисања (*Distributed generation - DG*) вјероватно ће на дужи рок смањити цијену струје за потрошаче.

2.3. Изазови и недостаци у развијању микромрежа

Упркос потенцијалним предностима, код развоја микромрежа постоје и бројни изазови и потенцијалне мане.

Висока цијена дистрибуираних енергетских ресурса – Висока цијена инсталирања микромрежа представља великуману. То се може ублажити неким обликом субвенционисања од стране државних тијела како би се подстакло инвестирање.

Техничке потешкоће – Односе се на недостатак техничког искуства у контролисању великог броја микроресурса на брзо укључивање и покретање. Овај аспект захтева свеобухватно истраживање у реалном времену у погледу управљања, заштите и контролисања микромрежа, као и у погледу избора, одерђивања величине и размештаја микромрежа. У овој области потребно је развити специфичне телекомуникационе инфраструктуре и комуникационе протоколе.

Непостојање стандарда – Пошто је област микромрежа сразмјерно нова, још увијек не постоје стандарди за рјешавање проблема око операције и заштите. Потребно је установити податке о квалитету струје за различите типове извора, стандарде и протоколе за интегрисање микроизвора и њихово учешће на конвенционалним и дерегулисаним енергетским тржиштима, као и безбиједносне заштитне смијернице.

Административне и законске препреке – У већини земаља не постоје стандардни прописи и закони који би регулисали рад микромрежа.

Тржишни монопол – Ако се микромрежама омогући да самостално снабдијевају енергијом приоритетна оптерећења током неког великог поремећаја у главној мрежи, најважније питање које се поставља јесте ко ће управљати цијеном струје током периода када главна мрежа буде недоступна. Пошто ће главна мрежа да буде искључена, а тренутно тржиште струје изгубиће контролу над цијеном енергије, микромреже ће можда продавати енергију по веома високој цијени, користећи тржишни монопол.

2.4. Питања управљања и рада микромрежа

Најважнија питања која се односе управљање и рад микромреже су следећа:

1. За одржавање квалитета струје, у микромрежама је краткорочно потребно одржавати равнотежу активне и реактивне снаге.
2. У регионима гдје јавна мрежа снабдијевања не постоји, микромрежа треба да ради самостално, а у оквиру веће јавне дистрибутивне мреже у прикљученом начину рада. Оператер микромреже треба да одабере начин рада унутар одговарајућег законодавног оквира.
3. Генерисање, испорука и складиштење потребно је планирати на одговарајући начин у односу на потражњу оптерећења у микромрежи и дугорочну енергетску равнотежу.
4. У микромрежне централне и микроконтролере потребно је уградити мјерне, контролне и заштитне функције засноване на SCADA системима.
5. Економски исплатив начин рада мора се осигурати кроз планирање генерисања, економски исплатив начин испоручивања и рад по оптималном протоку енергије.
6. Системску безбиједност потребно је одржавати кроз анализу непредвиђених ситуација и операције у кризним условима (као што је управљање потражњом, скидање оптерећења, изолација и гашење неке од јединица).
7. Привремени несклад између генерисања и оптерећења потребно је избалансирати путем правилног предвиђања оптерећења и управљања потражњом.
8. За свеукупно управљање, заштиту и регулисање енергије потребно је примјенити одговарајуће телекомуникационе инфраструктуре и комуникационе протоколе

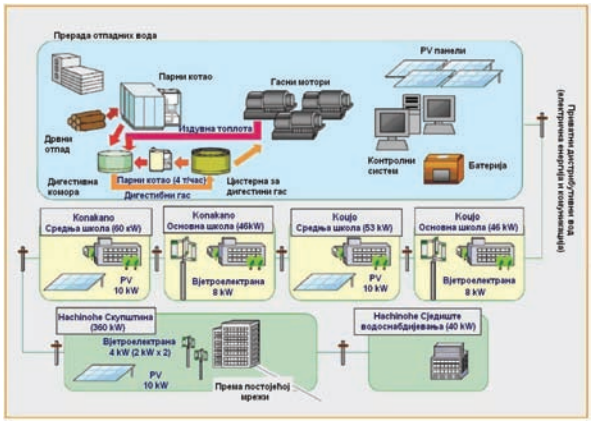
3. ПРИМЈЕРИ МИКРОМРЕЖА

Широм свијета постоје бројни активни експерименти у области микромрежа који покривају један широк спектар технологија. Као део овог истраживања дефинисане су топологије микромреже и оперативне конфигурације и установљени су критеријуми пројектовања за све могућности примјене микромрежа.

3.1. Пројекат из Јапана

Пројекат Hachinohe (слика 3.1) састоји се од једног система микромреже који је конструисан коришћењем једног приватно дистрибутивног вода дужине преко 5 километара. Тај вод је конструисан за пренос струје која је првенствено генерисана од стране система на гасни мотор. На микромрежу је прикључено и неколико PV система, као и мале вјетроелектране.

На електрану на отпад инсталирана су три гасна мотора од 170 kW и један PV систем од 50 kW. Да би се подстакло стварање дигестивног гаса, због недостатка термичке топлоте за одржавање бактерија, на електрану је инсталиран један парни котао на отпадна дрва. Између погона за прераду отпадних вода и градске канцеларије, четири школе и једно сједиште водоснабдијевања прикључене су на приватну дистрибутивну линију. На мјестима гдје се налазе школе коришћени су обновљиви енергетски ресурси ради добијања струје која осцилира у зависности од временских прилика како би се провјериле могућности контролног система микромреже да усклади понуду и потражњу. Контролни систем који је коришћен за балансирање понуде и потражње састоји се од три аспеката: планирање недељне понуде и потражње, контрола економског отпремања једном у три минута, и контрола протока струје у тачкама повезивања секлунду по секунду. Циљ контроле јесте да се оствари маргина грешке између понуде и потражње мања од 3% за сваки интервал од шест минута. Остварена величина маргине грешке током тестирања износила је мање од 3% током 99.99% времена рада система. У новембру 2007. пројекат Hachinohe радио је недељу дана аутономно од главне мреже. У том начину рада, неравнотежа између три фазе надокнађена је помоћу PV конвертора [3].

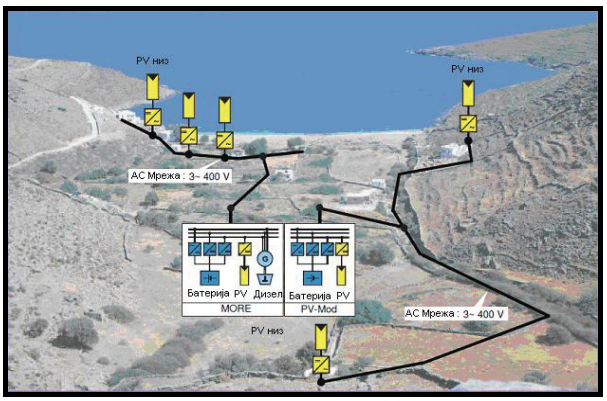


Слика 3.1. *Преглед Hachinohe пројекта.*

3.2. Пројекат из Грчке

Овај систем као што је приказано на слици 4.6 напаја 12 кућа у малој долини Кинтоса, острва које се налази на Архипелагу Егејског мора. Производни систем обухвата 10 kW PV-а, мали акумулатор од 53 kWh и 5 kW дизел мотор. Други PV систем имао је 2 kW, смјештених на крову контролне зграде повезан са SMA инвертором и акумулатором јачине 32 kWh да омогући снагу и контролу. Уређаји кориштени за становништво су били покретани са три SMA акумулатора повезани са паралелним помоћним конфигурацијама које обезбјеђују проток. Више од једног од 3.6 kW акумулаторских инвертора користи

се само да би се задовољиле потребе потражиоца. Акумулаторни инвертори могу да функционишу системом кап по кап и тиме омогуће доток до главног прекидача ако је напуњеност акумулатора слаба и ограничава излаз из PV инвертора ако је акумулатор пун [2].



Слика 3.2 *Пилот микромрежа Китнос*

4. ДАЉА ИСТРАЖИВАЊА МИКРОМРЕЖА

Добри резултати истраживања микромрежа широм свијета охрабрили су исртаживаче да наставе са изналажењем рјешења уочених проблема преко нових пројеката. Тако је у Европској Унији настављен пројекат *Microgrids*, али сада под новим називом *More Microgrids* [3].

Један од пројеката *More Microgrids* налази се у парку забаве Бронсберген, у близини Зутпена у Холандији. Састоји се од 210 објеката од којих је 108 опремљено PV системима прикљученим на главну мрежу. Парк је електрификован традиционалном трофазном мрежом од 400 V, која је преко дистрибутивног трансформатора прикључена на средњенапонску мрежу од 10 kV. Дистрибутивни трансформатор не напаја ништа изван парка. У парку се из дистрибутивног трансформатора интерно дистрибуира струја од 400 V преко четири кабла од којих је сваки заштићен осигурачима од 200 A на три фазе. Вршно оптерећење износи отприлике 90 kW. Укупна инсталирана снага свих PV система износи 315 kW. Циљ овог пројекта је да се путем интелигентног складиштења (праћеног флексибилним дистрибутивним системом) експериментално провере изоловане микромреже, укључујући проверу изолованог начина рада, аутоматску изолацију и поновно прикључење, ниво квара микромреже, дисторзију напона хармоника, управљање енергијом и оптимизацију радног вијека система складиштења, као и паралелен начин рада конвертора [3].

5. ЗАКЉУЧАК

Микромрежа представља алтернативни приступ снабдијевању дистрибутивном енергијом путем интегрисања малих дистрибуираних енергетских ресурса у нисконапонске и средњенапонске дистрибутивне мреже. Тиме се олакшава симултано генерисање струје и топлоте да би се на локалном нивоу задовољиле потребе потрошача.

Микромреже могу да раде самостално, као аутономна острва, синхроно са главном мрежом.

Технички гледано, микромреже представљају аутономну групу контролисанисаних микроизвора и уређаја за складиштење енергије који раде по принципу брзог укључивања и покретања и који су размештени на оптималан начин и на добробит потрошача. Микроизвори у основи представљају обновљиве неконвенционалне дистрибуиране енергетске ресурсе.

Микромреже, као активне нисконапонске и средњенапонске мреже могу потенцијално да имају велики број предности по главну мрежу, повећањем њене радне ефикасности и побољшањем поузданости и квалитета услуга. Међутим, потребно је пажљиво ријешити један велики број индустријских и техничких питања, што широм свијета захтева прилично обимно истраживање.

6. ЛИТЕРАТУРА

[1] Predrag Djapic, Charlotte Ramsay, Danny Pudjianto, Goran Strbac, Joseph Mutale, Nick Jenkins, and Ron Allan, “Taking an active approach”, IEEE Power and Energy Magazine, July/Aug. 2007, pp. 68-77.

[2] Nikos Hatziargyriou, Hiroshi Asano, Reza Iravani, and Chris Marnay, “Microgrids“, IEEE Power and Energy Magazine, July/Aug. 2007, pp.78-94.

[3] Benjamin Kroposki, Robert Lesseter, Toshifumi Ise, Satoshi Morozumi, Stavros Papathanassiou, and Nikos Hatziargyriou, “Making Microgrids Work“, IEEE Power and Energy Magazine, May/June 2008, pp. 41-53.

[4] S.Chowdhury, S.P.Chowdhury, and P.Crossley, “Microgrids and Active Distribution Networks“, The Institution of Engineering and Technology, London, United Kingdom, 2009.

[5] V.Katić и saradnici, „Atlas vetrova za teritoriju AP Vojvodine“, Studija za Pokrajinski sekretarijat za energetiku i mineralne sirovine, Novi Sad, 2008, http://www.psemr.vojvodina.gov.rs/studije_pdf_2009/Studija_Atlas_Vetrova_APV_2008.pdf

[6] http://en.wikipedia.org/wiki/Main_Page

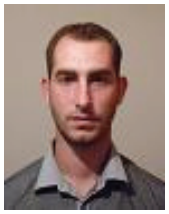
[7] http://www.mem.gov.rs/

[8] http://www.panonske.rs/

[9] http://www.izvorienergije.com

[10] *http://www.obnovljiviizvorienergije.rs*

Кратка биографија:



Богдан Тадић - рођен у Бијељини 1984. год. Дипломски - Мастер рад на Факултету техничких наука у Новом Саду из области Електротехнике и рачунарства, на смеру Електроенергетски системи одбранио је јуна 2011. год.



Владимир Катић је докторирао на Универзитету у Београду 1991. Редовни професор Факултета техничких наука у Новом Саду је од 2002. за област Енергетска електроника, машине и погони. Продекан је ФТН-а и шеф Катедре за енергетску електронику и претвараче. Области интересовања су енергетска електроника, квалитет електричне енергије и обновљиви извори електричне енергије.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.311.16

РЕГУЛАЦИЈА НАПОНА У ДИСТРИБУТИВНИМ МРЕЖАМА СА ГЕНЕРАТОРИМА

VOLT CONTROL IN DISTRIBUTION NETWORK WITH GENERATORS

Филип Маличевић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *У овом раду је дефинисанпроблем регулације напона и токова реактивних снага у дистрибутивним мрежама (ДМ) са генераторима. У циљу решавања постављеног проблема проширен је класичан модел за регулацију напонско реактивних прилика у ДМ. Проширење се састоји пре свега у додавању ограничења на могућности рада дистрибутивних генератора.*

Abstract – *Voltage regulation and reactive power flow problem for distribution networks (DN) with generators is defined in this paper. In order to solve defined problem, classical model for volt-var regulation in DN is expanded. Expansion is done by adding constraints on possibility of injectionof distributed generation.*

Кључне речи: *регулација напона, дистрибутивне мреже, обновљиви извори енергије*

1. УВОД

Постојање тока струје у мрежи доводи до промене напона, како у времену тако и у простору (од елемента до елемента). Последица је да ће на потрошачи који су прикључени на различитим местима у ДМ бити наметнут напон који мање или више одступа од номиналног напона потрошача. Електрична енергија са напоном који одступа од номиналног (на горе или на доле) на потрошачима изазива штету (неодговарајуће, неефикасно функционисање, или испад опреме) [1,2,3]. Зато дистрибутивна предузећа имају обавезу да одржавају напон на свим елементима ДМ унутар дозвољених (толерантних) граница које су дефинисане техничким прописима.

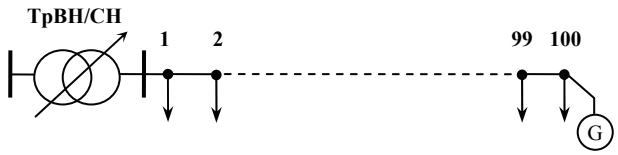
Најмасовнији обновљиви извори електричне енергије, ветрогенератори и соларне електране, имају интермитентну природу рада, тј. производња им је изразито променљива у времену (зависи од временских прилика: јачине, правца, смера ветра за ветрогенераторе, и осунчаности плоча за соларне електране). Када су такви извори прикључени у мрежу, они могу да проузрокују веома брзе и велике промене режима мреже. Такве промене стварају низ проблема које треба решити: проблем регулације напона, оптималних токова снага, подешења заштите, итд [1].

Овај рад се бави решавањем проблема регулације напона ДМ са прикљученим генераторима. У другом делу је дефинисан проблем који у ДМ уносе интермитентни извори електричне енергије – нарушење техничких ограничења напона. У трећем делу је предложено решење описаног проблема унапред прорачуном сигурних граница напона којима управљају аутоматски регулатори напона (АРН). У четвртом делу проширена градијентна метода (ГМ) напонско-реактивне оптимизације (НРО) рада ДМ. У петом делу је верификована ГМ НРО. У шестом делу је описани проблем двосмерних токова снага кроз АРН. Након закључка је наведена коришћена литература.

НАПОМЕНА:
Овај рад проистекао је из дипломског-мастер рада чији је ментор био др Горан Швенда, в.проф.

2. НАРУШАВАЊЕ ТЕХНИЧКИХ ОГРАНИЧЕЊА НАПОНА

За илустрацију проблема регулације напона у ДМ, који могу да настану као последица занемарења ефеката могућности рада генератора (као што су нарушавање техничких напонских граница), разматра се идеализована дистрибутивна тест мрежа (ДТМ) са слике 1. Њу чине: регулациони трансформатор (Тр) ВН/СН; вод са 100 кабловских деоница једнаке дужине; 100 трансформаторских станица (ТС) СН/НН са потрошачима чија је потрошња моделована као нелинеарно зависна од вредности напона [1] и ветрогенератор (ВГ) који је прикључен на СН сабирнице последње ТС СН/НН. Ефекти ВГ су моделовани инјектирањем активне снаге у СН чвор. ВГ у систем инјектира 67% номиналне снаге (S_{NG}=3 MW). Регулациона склопка Тр ВН/СН је у неутралном положају (t=0).



Слика 1. *Дистрибутивна тест мрежа*

Потребно је одредити оптималну позицију регулационе склопке Тр ВН/СН, за коју су губици активне снаге минимални и нису нарушена напонска ограничења. У наставку су разматрани примери у којима је при избору подешења регулационог трансформатора узет у обзир (унапређен модел), односно занемарен утицај могућности рада генератора (класичан модел). Притом истакнути су примери у којима су занемарењем могућности рада генератора, нарушене минималне и максималне вредности напона. У свим примерима, избор оптималне позиције извршен је делимичним претраживањем свих решења (применом градијентне методе) [1].

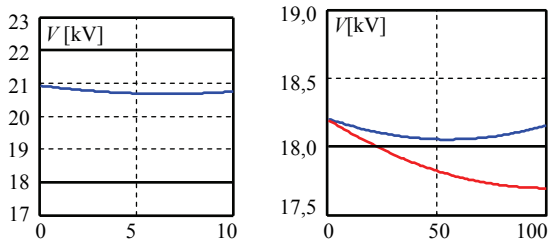
2.1. Нарушена минимална ограничења напона

За основни режим ДТМ оптерећење потрошача при номиналном напону од 400 V износи (50+j16,5) kVA, агенератор инјектира Pgen=2 MW. Профил напона је приказан на слици 2а. Губици активне снаге су ΔP=30,1 kW. Применом класичног модела оптимална

позиција је t=+12, а минимални губици активне снаге ΔP=19,4 kW. Остварен је режим чији је профил напона приказан на слици 2б, плава линија.

Уколико ветар престане да дува, ВГ ће престати да генерише активну снагу, P_{gen}=0 kW. Тада ће се: потрошачи у потпуности снабдевати кроз Тр ВН/СН; повећати струја кроз деонице СН вода; повећати пад напона и значајно смањити напони на потрошачима. Новонастали профил напона приказан је црвеном линијом.

Искључењем ВГ разматрана ДМ долази у недопустив режим – напони великог дела СН вода су испод технички дозвољених граница (18 kV, назначено пуном црном линијом).



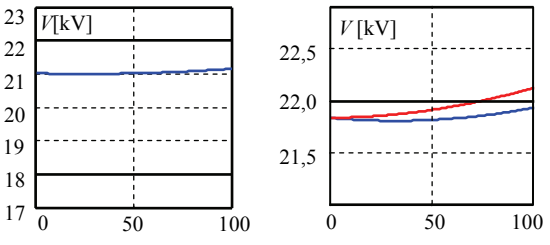
а) *б)*

Слика 2. *Нарушене минималне границе напона*

2.2. Нарушена максимална ограничења напона

Посматра се иста ДТМ у режиму када је потрошња на (25+j8,25) kVA, на номиналном напону од 400V. Регулациона склопка Тр ВН/СН је у неутралном положају (t=0). Профил напона је приказан на слици 3а. Губици активне снаге су тада ΔP=9,6 kW. Применом класичног модела оптимална позиција је t=–3, а губици активне снаге су 8,8 kW. Остварен је режим у ком нису нарушена напонска ограничења, чији је профил напона приказан на слици 3б, плава линија. Ако тада ветар нагло почне да дува и достигне јачину за коју ВГ генерише номиналну активну снагу, $P_{gen}=P_{nom}$, мењају се токови снага у СН воду, део потрошача ће се напајати из ВГ. Смањиће се струја кроз деонице СН вода. Промена генерација активне снаге доводи до повишења напона на сабирницама које су ближе ВГ. Новонастали профил напона приказан је црвеном линијом.

Укључењем ВГ расмтрана ДМ долази у недопустив режим – напони дела СН вода су преко технички дозвољених граница (22 kV, назначено пуном црном линијом).



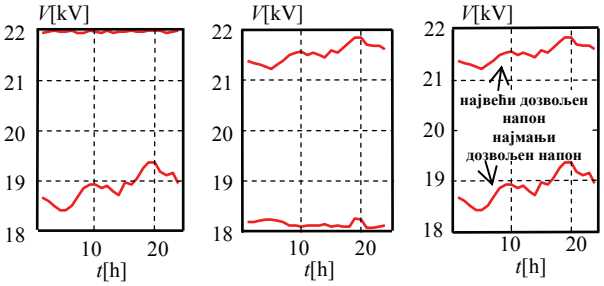
а) *б)*

Слика 3. *Нарушене максималне границе напона*

3. ПРОРАЧУН СИГУРНИХ ГРАНИЦА

За сигуран рад ДМ у присуству генератора, могу се унапред, ван реалног времена, одредити границе у којима мора да се одржава напон на секундару регулационог трансформатора (надгледани напони). Наведене границе се одређују тако да, када се надгледани напони држе у њиховим оквирима напонска ограничења не буду нарушена ни у једној тачки ДМ независно од производње прикључених генератора.

Граничне вредности надгледаног напона V_{SN} једноставне ДТМ са једним генератором, слика 1, када генератор не ради (P_{gen}=0) и када ради са номиналним инјектирањем активне снаге ($P_{gen}=P_{nom}$), приказане су на сликама 4а и 4б, респективно. Са ова два режима (P_{gen}=0 и $P_{gen}=P_{nom}$) дефинисане су граничне вредности напона на секундару регулационог трансформатора за сваки сат дана (слика 4в) – пресек области са слика 4а и б. Овим једноставним поступком дефинисана су напонска ограничења на могућности рада генератора. Ако се та ограничења додају у класичне поступке оптимизације напонско реактивних прилика, као ограничења унутар чијег опсега може да се врши контрола напона, тада ће без обзира на рад прикљученог генератора напон у ДМ бити унутар граница техничких ограничења напона.



а)P_{gen}=0 *б)$P_{gen}=P_{nom}$* *в) дозвољена област*

Слика 4. *Дозвољене границе напона*

Проблем који отежава примену методе прорачуна сигурних граница је време потребно за прорачун пресека дозвољених области свих могућих конфигурација за 24, односно 96 тренутака дана (ако се ради са сатном, односно 15-о минутном дискретизацијом).

Повећањем броја регулационих ресурса, нпр. додавањем кондензаторских батерија и регулатора напона, нагло се повећава број могућих решења које приликом одређивања оптималног решења потребно обрадити. Тиме се значајно повећава време потребно за прорачун сигурних граница.

Други проблем је смањење регулационих могућности додатних регулатора напона, тако да ова метода није ефикасна за примену на дугачким дистрибутивним водовима са више регулатора напона [1].

4. ОПТИМИЗАЦИЈА НАПОНСКО-РЕАКТИВНИХ ПРИЛИКА (ГРАДИЈЕНТНА МЕТОДА)

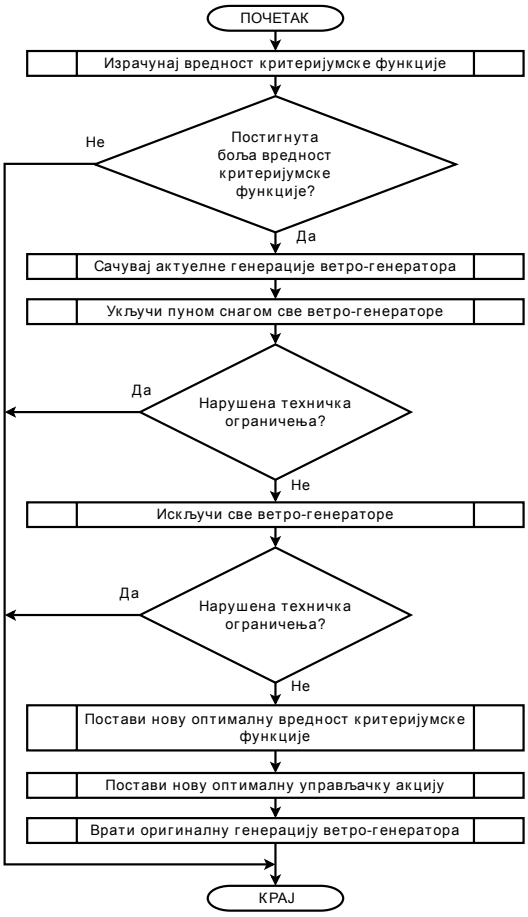
Рад ДМ се може оптимизовати по више критеријумских функција:

– минимални губици активне снаге,
– минимална инјектирана активна снага,
– минимални пад напона,
– трошкови, итд.

У системима у којима се софтверски управља регулационим уређајима до оптималног решење се уобичајено долази делимичним претраживањем свих могућих решења – применом градијентне методе пре-

траге. Применом градијентне методе се из почетног стања мреже одређују акције које својом реализацијом "најбржим" путем преводе ДМ из почетног, затеченог стања у најбоље могуће стање. Поступак проналаска најбољих акција је итеративан. У свакој итерацији се разматрају режими ДМ након промена:

– позиције регулационе склопке за једну позицију навише од почетне, за сваки трансформатор (или регулатор напона) којим се управља;
– позиције регулационе склопке за једну позицију наниже од почетне, за сваки трансформатор (или регулатор напона) којим се управља;
– промена статуса укључености сваке од КБ.



Слика 5. *Алгоритам избора оптималне управљачке акције*

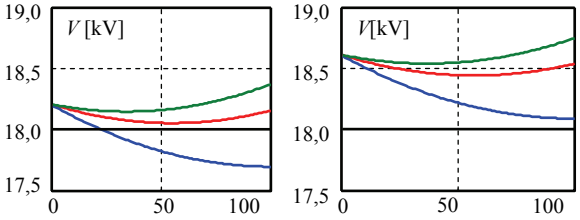
У текућој итерацији се од свих испробаних акција бира једна, ако постоји, која највише поправља вредност критеријумске функције. Да би се та акција прихватила проверава се да ли ће након њене примене, режим са номиналним ангажовањем и режим са искљученим генератором имати проблема са техничким ограничењима напона. Најбоља акција, чија примена не нарушава напонске границе, чак и у ситуацијама са нултим и номиналним инјектирањем генератора представља решење текуће итерације. Нова итерација представља понављање претходно описаног поступка сада на мрежи на којој је задања оптимална акција примењена. Итеративни поступак се завршава итерацијом у којој није пронађена управљачка акција која поправља вредност критеријумске функције, која је таква да при испаду и номиналном ангажовању прикључених генератора одржава систем унутар толерантних граница техничког ограничења напона.

Овај поступак гарантује да свака од изабраних акција неће изазвати режим са нарушеним активним ограничењима. Наравно, режим у који се пређе најчешће представља локални минимум изабране критеријумске функције. Алгоритам за избор оптималних управљачких акција приказан је на слици 5.

5. ВЕРИФИКАЦИЈА УНАПРЕЂЕНОГ МОДЕЛА

Унашређен модел, ГМ са ограничењима на могућности рада генератора примењен за напонско-реактивну оптимизацију верификован је на примеру ДТМ, поглавље 2.1. Вредности напона дуж извода приказане су на слици 6. Профил напона за разматрану ДМ када генератор у мрежу инјектира 67% номиналне снаге наглашен је црвеном линијом, када је генератор искључен плавом, а када инјектира 100% номиналне снаге наглашен је зеленом линијом. На слици 6а позиција регулационе склопке изабрана без ограничења на могућности рада генератора је t=+12. На слици 6б позиција регулационе склопке изабрана са ограничењима на могућности рада генератора је t=+10. Треба уочити да ће се за први пример, када је Тр ВН/СН подешен на основу режима у којем је генератор укључен, слика 6а, непланским искључењем (нпр. престанком ветра код ветрогенератора, наиласком облака код соларних електрана, итд.) или испадом генератора услед квара, вредности напона на великом делу извода бити испод минимално дозвољених вредности. Минимална вредност напона од 18 kV наглашена је црном линијом.

Уколико је Тр ВН/СН подешен поступком којим су уважене могућности рада генератора (испад генератора), напон извода ни у једном тренутку неће бити испод минимално дозвољених граница, слика 6б.



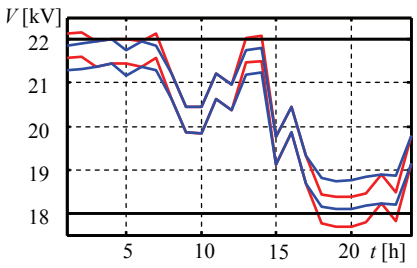
а) *б)*

Слика 6. *Подешење Тр ВН/СН*

У складу са сликом 6, за тренутак 19h, дефинисане су минимална и максимална вредност напона на СН сабирницама 17,7 kV и 18,4 kV, за прорачун позиције регулационе склопке без уважавања граничних могућности, слика 6а, и 18,1 kV и 18,7 kV, са уважавањем граничних могућности рада генератора, слика 6б.

На исти начин (као за један тренутак, 19 часова) извршена је анализа могућности рада регулационог трансформатора за цео дан. На слици 7 су приказане минималне и максималне вредности напонана СН сабирницама ДТМ када су у току решавања оптимизационог проблема узета у обзир – Подешење 1, односно занемарена ограничења на могућности рада генератора – Подешења 2. Црвеним линијама су приказане минималне (кад генератор не ради) и максималне (генератор ради) вредност напона које се могу имати у ДТМ за Подешење 1. Плавим линијама су приказане

минималне (кад генератор не ради) и максималне (генератор ради) вредности напона напона које се могу имати у ДТМ за Подешење 2.
Техничке границе напона за СН сабирнице назначене су црним линијама.
На основу приказаних резултата може се уочити да ако се регулациони ресурси подесе без уважавања ограничења на могућности рада генератора, променом рада генератора могућа су нарушења техничких граница напона. Ако за разматрани пример дође до пораста инјектирања активне снаге генератора (са 2 на 3MW), за Подешење 1, до пробоја напона долази у више тренутака (1, 2, 5, 7, 13 и 14 час). Исто тако ако се прекине инјектирање активне снаге генератора (са 2 на 0MW) напон ће бити испод минималне дозвољене вредности (18, 19, 20, 21 и 23. час).
Уколико се примене Подешења 2, без обзира на промене рада генератора (са 2 на 0 или 3 MW) ни једном није дошло до проблема са напонима.



Слика 7. *Могуће границе напона у ДТМ*

6. ДВОСМЕРНИ РЕГУЛАТОРИ НАПОНА

Регулација напона ДМ се стандардно заснива на примени регулационих трансформатора и регулатора напона са аутоматским регулаторима напона. У случају регулације напона вода у петљи, смер управљања напоном се одређује мерењем тока снаге (активне или реактивне) кроз регулатор. Рад дистрибуираних генератора може значајно да утиче на смер снаге кроз регулатор. Како би се уважиле чињенице да њихова генерација може да буде и већа од потрошње мреже испод регулатора, постојећи модели регулације напона морају да се надограде у складу са следећим чињеницама:

- није довољно да је двосмерни режим рада АРН осетљив само на токове активних снага (велико инјектирање активне снаге може да промени смер напајања вода);
- за регулацију напона вода који није у петљи може је да се користи когенерациони режим рада АРН (када се промени смер снаге кроз регулатор, промене се подешења еквивалентних параметара вода у АРН),
- за водове у петљи потребно је да се користи двосмерни режим рада АРН који је осетљив на токове реактивних снага (постојање КБ које би враћале реактивну снагу кроз АРН се уважавају постављањем прага осетљивости за одређивања инверзних токова снага кроз АРН) [1].

7. ЗАКЉУЧАК

Природа најмасовнијих обновљивих извора електричне енергије је интермитентна, тј. електрична енергија се производи када постоје природни услови за то, и изразито се мења са временским приликама (са јачином, правцем и смером ветра и јачином сунца). Такви генератори, са изразито брзим променама у инјектирању снаге, у опсегу од нулте до номиналне, радикално утичу на режим ДМ. У мрежама са таквим променама режима онемогућена је примена класичних метода регулације напона које су засноване пре свега на компаундованој регулацији напона.
За уважавање изразито интермитентне природе обновљивих извора енергије у овом раду су предложене две методе:

1. метода прорачуна сигурних граница унутар којих је могуће кретање напона којима управљају АРН, (погодна за делове ДМ којима се не управља софтверским алатима);
2. градијентна метода напонско-реактивне оптимизације рада ДМ проширена ограничењима на могућности рада генератора (ова метода се заснива на савременим софтверским ДМС алатима [1]).

ГМ НРО са ограничењима на могућности рада генератора је детаљно верификована у [1]. Верификација је извршена у мрежама за које се применом ГМ НРО без ограничења на могућности рада генератора добијају режими са напонима до 3,2% (700 V) преко горње (22 kV) и 2,2% (400 V) испод доње (18 kV) границе техничког ограничења напона. Уважавањем граничних могућности рада генератора у систему не долази до нарушавања техничких ограничења напона.
Увођење ограничења на могућности рада генератора смањује се број управљачких акција. Тиме се, ради сигурности режима ДМ, смањује ефекат оптимизационе процедуре (остају веће вредности губитака активне снаге).
Додата ограничења на могућности рада генератора, у току оптимизационе процедуре повећавају број прорачуна токова снага. Иако спорији од класичног поступка, извршење ГМ НРО са ограничењима на могућности рада генератора је довољно брзо да се може применити за управљање напонско-реактивним приликама у реалном времену.

8. ЛИТЕРАТУРА

[1] Ф.Маличевић: *"Регулација напона у дистрибутивним мрежама са генераторима"*, дипломски-мастер рад, Универзитет у Новом Саду, ФТН, јун 2011.

[2] В.С.Стрезоски: *"Систем регулације напона дистрибутивних мрежа"*, Универзитет у Новом Саду, ФТН, 1997.

[3] T.A.Short: *"Electric power distribution handbook"*, CRC Press, New York, 2004.

Кратка биографија:

Филип Маличевић рођен је у Новом Саду 1984. год. Средњу електротехничку школу „Михајло Пупин" је завршио у Новом Саду. Дипломски-мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – специјализованих софтвера у електроенергетици је одбранио 2011. године.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007:004.7

IMPLEMENTACIJA IPSSP PROTOKOLA ZA ELEKTRONSKE IGRE

AN IMPLEMENTATION OF IPSSP PROTOCOL FOR ELECTRONIC GAMES

Jasenka Števanov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu izvršena je analiza aktuelnih standarda (GLI) i protokola (SAS, IPSSP) na polju elektronskih igara. GLI i SAS su analizirani teoretski, dok je IPSSP predstavljen na konkretnom aparatu (Poker 100).*

Abstract – *The main goal of this work is the analysis of actual standards (GLI) and protocols (SAS, IPSSP) in world of electronic games. GLI and SAS are analized theoretically and protocol IPSSP is represented on real machine (Poker 100).*

Ključne reči: *electronic games, IPS, GLI, IPSSP*

1. UVOD

Kako bi se osigurala bezbednost i unificiranost aparata za elektronsko igranje, potrebno je uvesti standarde i odgovarajuće protokole koji bi unificirali ponašanje aparata. Protokoli su tu da obezbede sigurnost u upravljanju i zaštitu igračevog novca kako ne bi došlo do pronevere sredstava. Veliki deo protokola SAS je podređen upravo sigurnosnim putevima novca, dok se GLI više fokusira na fizičku bezbednost aparata (zaštita od krađe, povrede) i ravnopravnost za svakog igrača koji dodje za aparat (npr. RNG (random number generator da ne favorizuje određene brojeve). IPS standardni protokol je podređen specifičnostima zakona o elektronskim igrama na teritoriji Španije i kao takav, veoma se razlikuje od SAS-a.

2. GLI

GLI (Gaming Laboratories International) je oformljen 1989. godine sa ciljem da testira aparate za elektronsko igranje igara. Kao takvom, svrha mu je da testira aparat na:

1. subjektivnost pri analizi i sertifikaciji aparata za igranje.
2. kredibilitet i integritet mašine na kojoj se igra sa tačke gledišta igrača i prihoda koje obezbedjuje.
3. poštenost , proverivost i upravljivost.

Testiranje aparata ima 2 faze:

1. testiranje hardvera i
2. testiranje softvera.

2.1. Hardverski zahtevi

Za testiranje hardvera potrebno je da aparat može da podnese forsiran ilegalan upad u kućište mašine. Bilo koji deo aparata ne sme ni na koji način da ugrozi korisnika aparata. Hardverski zahtevi za aparat su definisani na sledeći način:

1. aparat mora biti upravljan mikroporocesorom na takav način da rezultat igre zavisi isključivo od mikroprocesora.
2. prekidač za napajanje mora biti dostupan iznutra kako se spolja ne bi mogao isključiti dovod struje.
3. od aparata se očekuje otpornost na toplotu i vlažnost. Ukoliko aparat nije u stanju da nastavi pravilan rad u nekontrolisanim uslovima, mora se regularno ugasiti, čuvajući status igre i fondove.
4. igrački aparat mora biti tako dizajniran da kablovi za podatke i struju ne budu dostupni igračima kako bi se sprečila manipulacija.
5. svaki aparat mora imati identifikaciju koja se sastoji od naziva proizvođača, jedinstvenog serijskog broja, broja modela i datuma sastavljanja.
6. ukoliko aparat prima kovanice ili žetone aparat mora osigurati razdvajanje različitih kovanica u hopper (ukoliko ga aparat poseduje) ili u drop-box. Drop-box mora biti zaključan i nedostupan igraču.
7. sva vrata aparata moraju biti zaključana i praćena senzorima na ilegalan pristup.
8. igrački aparati moraju imati logičku oblast u kojoj se nalazi elektronika odgovorna za logiku igre. Ova elektronika mora biti fizički odvojena od mesta na kome se odvajaju tokeni i ni u kom slučaju se ne sme dozvoliti pristup logičkoj jedinici.
9. Za aparat koji prima tikete i izdaje tikete, mora se pratiti pravilo za njegovo izdavanje kako bi isti tiket bio upotrebljiv i na drugom aparatu koji prima tikete, ili za mesta za unovčavanje istih.
10. kompletan hardver mora biti sasvim ispravan i svi njegovi delovi dokumentovani. U slučaju otkaza, aparat mora preći u stanje greške i prestati sa radom.

2.2 Softverski zahtevi

Softverski zahtevi se bave tehničkom podobnošću programa i pravilima igara. Video ekran na kome je igra mora biti jasno označen i mora tačno prikazivati pravila igre i sumu koja treba da bude isplaćena igraču (suma i jedinica u kojoj se kredit isplaćuje).

Igrački aparati uvek moraju pokazivati sledeće podatke: trenutni balans u kreditima, trenutni iznos opklade, sve moguće pobedničke kombinacije, pobednički iznos za sve moguće pobedničke kombinacije, osvojenu sumu u poslednjoj igri, selektovane opcije, denominaciju, i odricanje odgovornosti u slučaju greške aparata (disclaimer). Softver takođe treba da poseduje 27 brojača koji broje sve bitne događaje aparata (ulaz i izlaz novca, odbačen novac, džek-pot, fizički ulaz i izlaz kovanica, tiketa i novčanica, prenosi sa računa i na račun, broj otvaranja vrata aparata...).

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Branko Milosavljević, vanred.prof.

3. SAS

SAS (Slot Accounting System) specificira fizički i logički interfejs zasnovan na principu sistema za računanje na slot aparatima. Komunikacija između aparata se odigrava preko serijskog porta sa 19,2K impulsa/sekundi, koristeći wakeup format. Aparati su povezani sa hostom i imaju adrese 0-127(gde je 0 adresa aparata koji je u stanju otkaza).

Komunikacija sa aparatom se odvija po principu upita – odgovora. Host zahteva podatke šaljući upit aparatu koji odgovara jednobajtnim kodom izuzetka javljajući da se izuzetak desio.

Ukoliko host traži računovodstvene podatke, šalje komandu kojom traži specifične podatke. Kada odgovara na komandu hosta, aparat u odgovor uključuje svoju adresu, komandu koju je dobio, podatke i dvobajtni CRC.

Upiti koje host šalje aparatu mogu su u jednom od 2 formata (opšti upiti i dugi upiti). Ukoliko stigne više komandi, aparat ih mora držati u FIFO redu od barem 20 elemenata.

Ukoliko aparat ne odgovori na komandu u roku od 20 milisekundi, host ga isključuje iz lanca komandi.

	Bajtova	Vrednost	Opis
Adresa	1 binarno	01 – 7F	Adresa aparata
Komanda	1 binarno	01-7F, 0A, 0B	Enable-disable
CRC	2 binarno	0000 - FFFF	CRC

Tabela 1. *Format opšte komande hosta*

	Bajtova	Vrednost	Opis
Adresa	1 binarno	01 – 7F	ACK
		81 - fF	NACK

Tabela 2. *Format odgovora igračkog aparata*

SAS se naslanja na GLI i koristi tehnologije koje su definisane GLI-om. Na taj način, SAS ima tačno definisano ponašanje za unos, kovanica, novčanica i tiketa, kao i proceduru za isplatu ili za direktno (i bezbedno) prebacivanje na račun (AFT – Advanced Funds Transfer). Implementacijom AFT, u stanju smo da pratimo svaku transakciju koja se desila na svakom aparatu koji je povezan sa hostom.

AFT transferi su odvojeni u 3 kategorije: *in-house*, bonus i dug. Bonusi se osvajaju preko eksternog sistema za davanje premija, dok su dugovi oni transferi koji podrazumevaju prebacivanje sa računa. Sve druge vrste transfera se smatraju *in-house*.

Pored toga sto SAS prati tokove novca, on takodje i može da proverava stanje programa na aparatu i stanje uređaja svih aparata sa kojima je povezan, tako da u svakom trenutku može tražiti autentifikaciju komponenti.

4. IPSSP

Cilj IPSSP (*Informatica Programes Serveis Standard Protocol*) je da obezbedi uniformnost za različite aparate: igračke aparate A i B tipa, aparate za usitnjavanje novca i druge, kako bi se obezbedila pouzdanost svih očitavanja, posmatranja i automatizacija naplaćivanja.

Aparat je sa hostom povezan preko serijskog porta, pri čemu su podešavanja sledeća: bitrate 9600bps, nema parnosti, jedan zaustavni bit i DTR i RTS signali nisu u upotrebi.

Aparati međusobno komuniciraju tako što šalju komande i podatke u frejmovima različite dužine.

Terminal/online sistem	Pravac	Mašina
STX zaglavlje (1 bajt)	→	
Komandni kod (1 bajt)		
Broj bajtova podataka (2 bajta)		
Podaci (n bajtova)		
Checksum (2 bajta)		

Tabela 3. Frejmovi koje prima igrački aparat

Terminal/ online sistem	Pravac	Mašina
	←	STX zaglavlje (1 bajt)
		Komandni kod (1 bajt)
		Broj bajtova podataka (2 bajta)
		Podaci (n bajtova)
		Checksum (2 bajta)

Tabela 4. Frejm odgovora igračkog aparata

Igrački aparat može slati frejmove samo u dva slučaja: ako je od njega tražen odgovor na komandu, ili ako se javio izuzetak. Kao odgovor na komandu se prvo šalje ACK ili NACK (sa dodatnim kodom greške) i tek nakon toga odgovor, osim za događaje.

Vreme odgovora na komandu je 1 sekunda. Za svaki odgovor koji dodje nakon isteka vremena, šalje se NACK.

Kao i SAS, IPS takođe barata brojačima. U IPS-u su brojači klasifikovani u grupe, gde svaka grupa sadrži brojače iste vrste (kovanice (10c, 20c, 50c, 1€...), novčanice (5€, 10€, ...)). Brojači u IPS-u su u evrima ili kreditima, za razliku od SAS-a koji je u dolarima.

Grupa 1 brojača se sastoji od 21 brojača predstavljenih kreditima. Brojači grupa 2-9 su uvek u novčanim jedinicama i predstavljaju redom: kovanice koje su ušle u aparat (8 brojača, za svaku vrstu kovanica koje podržava), validirane kovanice koje su prešle u kredite (8 brojača), primljene novčanice (8 brojača), izlaz kovanica na hopper (8 brojača), broj kovanica u hopperu (8 brojača), validirane novčanice spremne za isplatu (8 brojača), novčanice isplaćene na hopper (8 brojača), broj novčanica u hopperu (8 brojača).

Kada aparat šalje podatke o brojačima koristi bit-maske da bi pokazao koji brojače podržava. Za svaku grupu brojača, aparat ima bajt kojim obaveštava koji su brojači dostupni. Ovi bajtovi predstavljaju konfiguraciju aparata. Za grupu brojača 1, vrednost brojača je suma svih brojača ove grupe koju aparat podržava (ako je vrednost bajta 2 to znači da aparat podržava barem brojač 2). Ako aparat ne podržava sve brojače redom, vrednost bajta je vrednost poslednjeg brojača podržanog aparatom. Za brojače grupa 2 – 9, bajtovi po grupi su kodirani. Svaki bit bajta će imati vrednost 1 ili 0 u zavisnosti od toga da li aparat podržava brojač ili ne.

Pravilno postavljanje ovih maski je veoma bitno kako bi host znao konfiguraciju aparata.

Takođe, potrebno je naglasiti da ukoliko aparat ima više igračkih mesta, postoje zasebni brojači za svako igračko mesto i sumarni za ceo aparat.

Aparat može biti konfigurisan da automatski šalje različite tipove informacija koje se tiču aparata i njegovih uređaja, brojača, isplata i sl.

Komande koje aparat prima mogu biti one koje dozvoljavaju da aparat ne bude u sesiji (prijava i odjava sa sistema, provera da li je aparast u sesiji, dobavljanje internih informacija aparata, čitanje identifikacije aparata i čitanje verzije IPSSP-a) i one koje traže da je aparat u sesiji (sve komande koje traže stanje brojača, vrata aparata, čitanje logova, čitanje i pisanje vremena aparata, blokiranje i odblokiranje aparata)
Komande sa komandnim kodom iznad 0x10 su one koje zahtevaju da je aparat u sesiji. Takođe, on može generisati događaje, iako nije u sesiji, ali komande koje zahtevaju sesiju, dobijene od strane hosta može obrađivati samo ako je u tom stanju (npr. može generisati događaj isplate na hopperu, ali komandu odblokiraj aparat može izvršiti samo ako je u sesiji).
IPS standardni protokol je aktuelan u 3 verzije: 1.0, 1.1 i 2.0. IPSSP je implementiran kao Windows servis na aparatu *Texas Hold'em*. Zbog specifičnosti samog aparata i vremena isporuke, nisu podržane sve komande definisane verzijom protokola 2.0. Takođe, događaji u potpunosti nisu podržani na aparatu od strane aplikacije.
Zbog specifičnosti tržišta kome je namenjen, aparat prima samo određene vrste kovanica (20c, 1€, 2€ i novčanice od 5€, 10€, 20€).
To takođe znači da se bitmaske konfiguracije brojača sastoje uglavnom od 0, jer aparat ne podržava sve apoene podržane protokolom).

4. ZAKLJUČAK

Ovim radom je predviđena prezentacija osnovnih principa po kojima funkcionišu elektronski aparati za igre na sreću. Predviđeni su za sve vrste elektronskih aparata (slotova, ruleta, igara sa kartama (black jack, poker), kao i aparata za isplatu ili usitnjavanje novca i isplatu tiketa. Mehanizmi su provereni, korektni i u skladu sa aktuelnim tehnologijama.
IPS standardni protokol je specifičan samo za tržište Španije gde se ne traži da aparat bude u skladu sa GLI i da podržava SAS. Aparati mogu imati implementiran SAS, ali u rudimentiranoj formi jer zakoni ne dozvoljavaju vezu aparata sa bilo kojim uređajem, osim u slučaju testiranja i provere aparata.

GLI i SAS (sertifikovan od strane GLI) se međusobno podržavaju i prilagođavaju promenama u tehnologiji i igrama. SAS je, iako prvobitno predviđen samo kao sistem kontrole slot aparata, vrlo brzo prerastao u sistem kojim se prate sve vrste aparata na polju elektronskih igara na sreću.

5. LITERATURA

[1] Gaming Laboratories International, Inc., GLI-11 v2.0 Standard, 2007.
[2] http://www.gaminglabs.com/
[3] *Slot Accounting System Protocol*, Gaming Standards Association, Reno, NV, USA, 2007.
[4] http://www.gamingstandards.com/
[5] http://www.casino-datasolutions.com/html/slot_ accounting.html
[6] *IPS Standard Protocol*, Informatica Programes Serveis, Barselona, Španija, 2009.

Kratka biografija:

Jasenka Števanov je rođena 1984. godine u Zrenjaninu. Diplomski-master rad odbranila je 2011. godine iz oblasti računarske nauke i informatika.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004

AUTOMATIZOVANA KONFIGURACIJA ALFRESCO REPOZITORIJUMA

AUTOMATED CONFIGURATION OF ALFRESCO REPOSITORIES

Miroslav Popov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Rad opisuje rešenje za automatizaciju post-instalacionih zadataka za Alfresco sistem za upravljanje dokumentima. Ovi zadaci implementiraju se pomoću JavaScript programa. Prikazano rešenje je namenjeno za Alfresco, ali se lako može prilagoditi drugim repozitorijumima dokumenata.*

Abstract – *This paper presents a solution for automated post-install tasks for Alfresco document management system. Those tasks are implemented using JavaScript programs. The presented solution is developed for Alfresco DMS, but can be easily customised for other DMS systems.*

Ključne reči: *post-install configuration, document management, Alfresco*

1. UVOD

Instalacija velikih sistema uvek podrazumeva i unos inicijalnih podataka kojima sistem rukuje. Ovi podaci se mogu unositi u bazu podataka kada je model poznat ili se može koristi postojeći korisnički interfejs za unos. I jedan i drugi pristup imaju ozbiljnih mana. Model baze podataka je često nepoznat, a i podložan izmenama. Logika dodeljivanja sistemskih podataka je uglavnom skrivena u samoj aplikaciji. S druge strane unos podataka preko korisničkig interfejsa je spor i vezan za konkretnu instancu programa (isti unos se mora ponoviti i za druge instalacije).

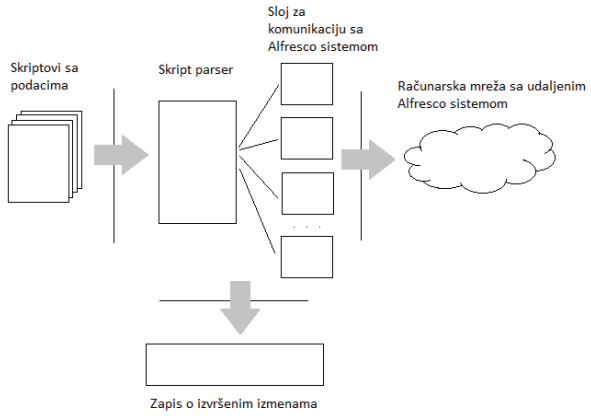
Ovde je izloženo rešenje koje omogućuje unos ovakvih podataka korišćenjem JavaScript jezika i web servisa.

Post-instalaciono postavljanje početnih podataka je standardna procedura kod postavljanja velikih sistema. Ovakvi sistemi u većini slučajeva pružaju više mogućnosti za pristup. Tako se osim korisničkog intefejsa, namenjenog krajnim korisnicima, nudi i mogućnost korišćenja *Application Programing Interface* (API), *Web Service* (WS) i slično. Iako ovakvi načini pristupa omogućavaju automatizaciju unosa podataka, oni zahtevaju poznavanje programiranja i specifičnih softverskih tehnologija što analitičari sistema često ne poseduju.
Da bi analitičarima omogućili brži unos potrebnih podataka, mora im se obezbediti mogućnost jednostavnog definisanja ulaznih podataka. Rešenje problema jeste pisanje programa koji će tumačiti skript fajlove i preko dodatnih interfejsa unositi inicijalne podatke u sistem.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Branko Milosavljević, vanr. prof.

2. ARHITEKTURA SISTEMA

Sama postavka problema nameće razdvajanje bar dva sloja u rešenju. Sa jedne strane potrebno je imati sloj koji će tumačiti skriptove koje unose analitičari, a sa druge strane treba da se nalazi sloj koji vrši komunikaciju sa programom i preko koga se podaci unose u sistem.



Slika 1. *Arhikektura rešenja za post-install konfigurisanje Alfresco sistema*

Slika 1. predstavlja skicu arhitekture sistema. Skript fajlove učitava skript parser i poziva odgovarajuće klase za komunikaciju sa sistemom. Sve izvršene operacije evidentiraju se u log fajlu.

2.1. Skript fajlovi

Analitičare je potrebno osloboditi potrebe poznavanja programskog jezika, odnosno potrebno im je obezbediti pisanje ulaznih podataka u što slobodnijoj formi. Kako je pisanje parsera dodatni posao, najbolje je iskoristiti neko već postojeće rešenje.
JavaScript engine omogućuje lako povezivanje unetih podataka sa pozivanjem JavaScript funkcija. Analitičari bi trebalo da se upoznaju sa određenim brojem funkcija koje pozivaju (a čiji naziv se definiše) i da unose stvarne podatke. Budući da se JavaScript fajlovi evaluiraju pri izvršavanju, njih nije potrebno prevoditi, odnosno posao analitičara se na tom mestu završava.
Ukoliko prilikom inicijalizacije JavaScript interpretera, a pre parsiranja konfiguracionih fajlova, umetnemo definicije funkcija, skript fajlovi postaju oslobođeni definicija i dobijaju formu nosilaca podataka.

Ako pre parsiranja fajla izvršimo sledeći JavaScript kod:

```
engine.eval("function
  cgroup(groupname) {"return
  org.ftn.alfresco.setup.UserAdmin.c
  reateGroup(groupname);} ");
```

sam skript za kreiranje inicijalnih podataka može biti jednostavan:

```
cgroup('group-1');
cgroup('group-2');
```

što je sasvim prihvatljivo i za osobu koja ne poznaje programiranje.

2.2. Parser i web servisi

Korišćenjem JavaScript jezika, od analitičara je sakriveno pozivanje metoda koje obavljaju komunikaciju sa Alfresco sistemom putem web servisa. Pisanjem JavaScript funkcije, putem parsera se poziva odgovarajuća Java funkcija. U ovim funkcijama implementira se validacija, poziv konkretnog servisa, logovanje i slično. Za pristup Alfresco sistemu koriste se klase web servis klijenta koje su sastavni deo Alfresca.

Iako sloj komunicira preko web servisa, to nije ni neophodnost niti pravilo. Na ovom mestu se dobijaju podaci koje unosi analitičar u vidu parametara funkcije, a na programeru je da odabere najbolji način da podatke prenese u sistem.

2.3. Primer implementirane funkcije

Implementacija funkcija za pristup Alfesco sistemu je pojednostavljena korišćenjem već generisanih Alfesco klasa. Na početku je potrebno instancirati odgovarajuću klasu za komunikaciju pomoću WebServiceFactory klase. U zavisnosti od konkretne metode, vrše se provere parametara i generisanje Alfresco parametara nakon čega sledi sam poziv metode.

```
public static String
       createGroup(String groupName)
              throws Exception {

   try {
       AccessControlServiceSoapBindingStub
         acctr =
              WebServiceFactory.
                 getAccessControlService();

       AuthorityFilter af = new
              AuthorityFilter();
       af.setAuthorityType("GROUP");

       String[] auth =
              acctr.getAllAuthorities(af);
       for(int i = 0;i < auth.length;i++) {
          if (auth[i].equals(
              Constants.GROUP_PREFIX +
                    groupName)) {
              logger.warn(
                "Group with name '" +
                     groupName
                     + "' already exists!");
              return auth[i]
          }
       }

       NewAuthority cpGrpAuth = new
              NewAuthority(
              UserAdmin.GROUP_AUTHORITY_TYPE
                    , groupName);
      NewAuthority[] newAuthorities =
              {cpGrpAuth};
      String result[] = acctr.
        createAuthorities(null,
              newAuthorities);

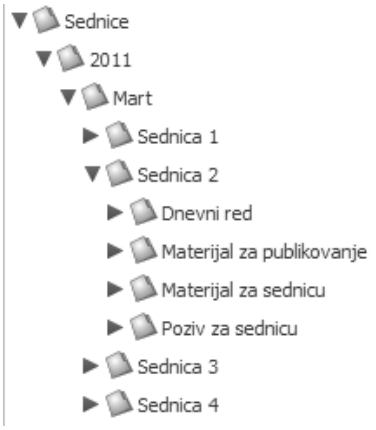
       logger.info(
              "Created group '" + groupName +
                 "'");

          return result[0];
       } catch (Exception e) {
          logger.error(
                 "error creating group", e);
          throw e;
       }
}
```

Metoda na početku pravi referencu ka WS klasi koja joj je potrebna, zatim se proverava postojanje, odnosno validnost podataka. U ovom slučaju poziv kreiranja grupe koja već postoji rezultuje samo upozorenjem, ne i greškom. Ukoliko grupa ne postoji, kreira se i pravi se zapis o uspešnom kreiranju grupe.

3. STUDIJA SLUČAJA

Sistem je primenjen na realizaciji programa za potrebe vođenja sednica republičkog ministarstva za ekologiju i komunikacije ministarstva sa medijima. Pomoću skript fajlova je opisana struktura inicijalnih podataka pa se nakon instalacije samog Alfresco sistema vrlo brzo dolazilo do sistema prilagođenog za konkretan slučaj. Pomoću skript fajlova su kreirani korisnici, grupe, folderska struktura (slika 2) i prava pristupa korisnika.



Slika 2. *Struktura direktorijuma u studiji slučaja*

Da bi se sistem testirao, potrebno je popuniti sistem inicijalnim podacima.

Skriptovi treba da naprave organizacionu strukturu za čuvanje dokumenata o sednicama čija struktura bi bila sledeća:

U folderu Sednice treba da se nalaze folderi koji označavaju godinu održavanja sednice (2010, 2011, ...), u svakom ovom folderu treba da se nalazi folder koji označava mesec održavanja sednice (mart, april...) i u svakom od ovih foldera treba da se nalazi folder sa konkretnom sednicom u datom mesecu (Sednica 1, Sednica 2, ...). Pomoću ovih skriptova kreiraju se četiri sednice koje se održavaju u mesecu martu.

Takođe, svaka sednica ima sledeću strukturu (prava pristupa konkretnom folderu su takođe definisana – naziv grupe iza čega sledi pravo pristupa – **č** (pravo čitanja) i **p** (pravo pisanja).

- Materijal za sednicu – Šef kabineta, čp; Ministar, čp; Državni sekretar, č
- Dnevni red – Šef kabineta, čp; Ministar čp
- Materijal za publikovanje – Šef kabineta, čp; Ministar, čp; PR služba, čp

- Pozivi za sednicu – Šef kabineta, čp; Ministar, čp

Kao što se vidi, svaka sednica ima istu strukturu kao i prethodna tako da je najbolje napraviti šablon za kreiranje sednica (template). Tako da skriptovi treba da kreiraju i template sednica.

Za potrebe PR Službe treba obezebediti model sistema za postavljanje pitanja novinara kabinetu, pa je potrebno kreirati foldere:

- PR Služba
  - Pitanja (PR Služba, čp)
  - Odgovori na pitanja (Kabinet ministra, čp; PR Služba čp)
- Kabinet Ministra
  - Pitanja za ministra (PR Služba, čp; Ministar, čp)
  - Pitanja za državnog sekretara za finansije (PR Služba, čp, Državni sekretar za finansije čp)
  - Pitanja za državnog sekretara za razvoj (PR Služba, čp, Državni sekretar za razvoj čp)
  - Pitanja za državnog sekretara za planiranje (PR Služba, čp, Državni sekretar za planiranje čp)
  - Pitanja za državnog sekretara za energetsku efikasnost (PR Služba, čp, Državni sekretar za energetsku efikasnost čp)
  - Pitanja za Šefa kabineta (PR Služba, čp; Šef kabineta, čp)

Sistem treba da podrži sledeće grupe:

- Ministarstvo
- Kabinet Ministra
- Ministar
- Šef kabineta
- Državni sekretar za finansije
- Državni sekretar za razvoj
- Državni sekretar za planiranje
- Državni sekretar za energetsku efikasnost
- PR
- PR Služba

Za svaku grupu je potrebno registrovati po jednog korisnika.

Pošto sistem treba da obezbedi čitanje i čitanje i modifikaciju u sistemu će se morati definisati bar dve role – čitač i menjač. Prva uloga ima samo pravo čitanja nad objektom dok druga ima prava čitanja i modifikacije objekta.

Da bi se obezbedio još brže kreiranje podataka, pisanje skritpova se može podeliti na dve celine. Pisanje skritpova za unos grupa i korisnika i na pisanje skripta za unos foldera.

Skript za kreiranje grupa i korisnika ima sledeću sadržinu:

```
cgroup('Ministar');
cgroup('Šef kabineta');
cgroup('Državni sekretar za finansije');
cgroup('Državni sekretar za razvoj');
cgroup('Državni sekretar za planiranje');
cgroup('Državni sekretar za energetsku
efikasnost');
cgroup('PR Služba');

cuser('milan', 'x', '/ministar', 'Milan',
'Milanović', 'ministar@ekologija.rs');
...
```

Izvršavanjem ovakvog skripta dobijamo sve grupe i korisnike za testiranje funkcionisanja sistema.

Da bi kreirali template sednica izvršićemo sledeći skript:

```
var templates = cref(
 '/app:company_home/app:dictionary/
 app:space_templates');

var sednica = cfolder(templates,
'Sednica');
var ms = cfolder(sednica, 'Materijal za
sednicu');
mngpermgroup(ms, 'Državni sekretar',
'čitač');
cfolder(sednica, 'Dnevni red');
var mp = cfolder(sednica, 'Materijal za
publikovanje');
mngpermgroup(mp, 'PR Služba', 'čitač');
cfolder(sednica, 'Poziv za sednicu');
```

Za potrebe folderske strukture potrebno je napisati skript sledeće sadržine:

```
var root = cref('/');
var sednice = cfolder(root, 'Sednice');
mngperminherit(sednice, false);

var     2011mart     =     cfolder(sednice,
'2011/Mart');
mngpermgroup(2011mart,    'Šef    kabineta',
'menjač');
mngpermgroup(2011mart,          'Ministar',
'menjač');
mngperminherit(2011mart, true);

var sednica1 = cfolder(2011mart, 'Sednica
1');
var ms1 = cfolder(sednica1, 'Materijal za
sednicu');
mngpermgroup(ms1, 'Državni sekretar',
'čitač');
cfolder(sednica1, 'Dnevni red');
var mp1 = cfolder(sednica1, 'Materijal za
publikovanje');
mngpermgroup(mp1, 'PR Služba', 'čitač');
cfolder(sednica1, 'Poziv za sednicu');

var sednica2 = cfolder(2011mart, 'Sednica
2');
var ms2 = cfolder(sednica2, 'Materijal za
sednicu');
mngpermgroup(ms2, 'Državni sekretar',
'čitač');
cfolder(sednica2, 'Dnevni red');
var mp2 = cfolder(sednica2, 'Materijal za
publikovanje');
mngpermgroup(mp2, 'PR Služba', 'čitač');
cfolder(sednica2, 'Poziv za sednicu');

var sednica3 = cfolder(2011mart, 'Sednica
3');
var ms3 = cfolder(sednica3, 'Materijal za
sednicu');
mngpermgroup(ms3, 'Državni sekretar',
'čitač');
cfolder(sednica3, 'Dnevni red');
var mp3 = cfolder(sednica3, 'Materijal za
publikovanje');
mngpermgroup(mp3, 'PR Služba', 'čitač');
cfolder(sednica3, 'Poziv za sednicu');

var sednica4 = cfolder(2011mart, 'Sednica
4');
```

```
var ms4 = cfolder(sednica4, 'Materijal za
sednicu');
mngpermgroup(ms4, 'Državni sekretar',
'čitač');
cfolder(sednica4, 'Dnevni red');
var mp4 = cfolder(sednica4, 'Materijal za
publikovanje');
mngpermgroup(mp4, 'PR Služba', 'čitač');
cfolder(sednica4, 'Poziv za sednicu');
```

Nakon čijeg izvršenja dobijamo strukturu u Alfresco sistemu kao na slici 5.1

4. ZAKLJUČAK

Odvajanjem slojeva i omogućavanjem pisanja podataka u gotovo slobodnom tekstu postiže se daleko brži unos inicijalnih podataka i omogućuje se njihovo višestruko korišćenje.

Sistem je zasnovan na gotovim rešenjima, predstavlja njihovu integraciju i pokazuje kako se korišćenjem jednostavnih delova i njihovom kombinacijom lako dobijaju korisni alati.

5. LITERATURA

[1] U. Cei, P. Lucidi, "*Alfresco 3 Web Services*", Packt Publishing, 2010.

[2] J. Potts, *"Afresco Developer Guide"*, Packt Publising, 2008.

Kratka biografija:

Miroslav Popov rođen je u Kikindi 1981. god. Usmerava se ka informacionim tehnologijama još od srednje škole. Profesionalno se bavi razvojem java aplikacija od 2005. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

OPTIMIZACIJA ŽELJENE VREDNOSTI TEMPERATURE RASHLADNOG FLUIDA ČILERA PRIMENOM FAZI LOGIKE

OPTIMIZATION OF THE DESIRED CHILLER COOLANT OUTLET TEMPERATURE VALUE USING FUZZY LOGIC

Damir Vujaklija, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj - *Cilj ovog projekta je da se izvrši optimizacija vrednosti izlazne temperature rashladnog fluida čilera, i samim tim, optimizuje potrošnja električne energije čilera. Za rešavanje ovog problema kreiran je simulink model sistema klimatizacije i čiler sistema na osnovu dostupnih podataka realnog sistema, a zatim izvršena optimizacija rada čilera primenom fazi logike.*

Abstract – *The goal of this work is to optimize chiller coolant outlet temperature, and therefore, to optimize consumpton of electric energy in chiller. To solve this problem was created simulink HVAC and chiller system model based on available data of the real system, and then carried out optimization of chiller using fuzzy logic.*

Ključne reči: *HVAC, klimatizacija, čiler, fazi logika, optimizacija, energija, simulink.*

1. UVOD

Potrošnja električne energije u cilju grejanja i hlađenja komercijalnih, industrijskih i stambenih objekata dostiže do 50% ukupne potrošnje energije u svetu. Taj trend rasta potrošnje energije se neizbežno povećava , a samim tim i emisija ugljendioksida .To je navelo Evropsku komisiju da direktivom 2002/91/CE nametne nekoliko akcija kojim bi se racionalizovalo korišćenje energije i poboljšala energetska efikasnost zgrada [1]. Jedan od načina da se izbegne nepotrebno trošenje energije, a ujedno obezbede prijatni uslovi za život i rad u objektima u zavisnosti od spoljašne temperature, je pronalaženje novih metoda upravljanja sistema klimatizacije i čiler sistema u novim i postojećim objektima .

Za razvoj optimalnog upravljanja napravljen je odgovarajući model kompletnog procesa na koji je potom primenjena optimizacija rada čilera. Za realizaciju modela sistema, u ovom radu, korišćeni su primeri modelovanja sistema klimatizacije u radovima [2,3,4,5], kao i primeri modelovanja čilera u radovima [6,7,8].

Optimizacija rada čilera, se može postići kontrolom velikog broja parametara. Mnoga istraživanja koja su se bavila ovim problemom su se bazirala na optimizaciji rada pojedinih komponenti čiler sistema, čija se kontrola bazira na lokalnoj povratnoj sprezi, pa samim tim ne postoji interakcija sa ostalim komponentama sistema i potrebama zgrade. U ovom radu, prilikom optimizacije rada čilera, postoji interakcija sa ostalim komponentama sistema i potrebama zgrade, pa je smanjenje ukupnih troškova lako dostignuto.

Upotreba fazi logike za optimizaciju upravljanja sistema klimatizacije i čiler sistema je znacajno povećana poslednjih godina [9,10,11,12,13]. Zbog dinamike sistema koja u sebi sadrži kompleksnost i neizvesnost u svojim parametrima, primena fazi logike se nameće kao prirodno rešenje, upravo zbog njene sposobnosti da se nosi takvim problemima.

Za kreiranje simulacionog modela sistema korišćen je alat Symulink programskog paketa Matlab, a optimizacija čilera je izvršena korišćenjem fazi logickih kontrolera.

2. MODEL SISTEMA KLIMATIZACIJE I ČILER SISTEMA

Čiler koji je objekat upravljanja u ovom radu se nalazi u komercijalnoj upotrebi, u poslovnoj zgradi kompanije Imel Group na Novom Beogradu.

Podaci koji su korišćeni za izradu simulacionog modela obuhvataju informacije o :

- spoljašnjoj temperaturi
- ulazna temperatura vode u čiler (povratna temperatura)
- temperaturi vode koja izlazi iz čilera (izlazna temperatura)
- stanje svakog pojedinacnog kompresora.

Za dobijanje ovih podataka izvršena su merenja u periodu od nedelju dana, gde se za spoljašnju, povratnu i izlaznu temperaturu vršena merenja na svakih 10 sekundi, dok su za stanje svakog pojedinačnog kompresora vršena merenja na svakih 10 minuta. Ovaj čiler ima četiri centrifugalna kompresora, pri čemu svaki od njih ima snagu 24kW i projektovani su tako da rade u paru tj. postoje samo tri moguća slučaja: ne radi nijedan kompresor, rade dva kompresora, rade sva četiri kompresora.

2.1. Matematički model

Rashladni kapacitet čilera Q_{Ch} je dat jednacinom (1). Na osnovu ove jednačine, izlazna temperatura vode iz čilera $T_{o,Ch}$ racuna se po diferencijalnoj jednačini(2).

$$Q_{Ch} = \dot{m}_{w,Ch} c_{p,w} (T_{i,Ch} - T_{o,Ch}) \quad (1)$$

$$\dot{T}_{o,Ch} = (\dot{m}_{w,Ch} c_{p,w} (T_{i,Ch} - T_{o,Ch}) - Q_{Ch}) / C_{ch} \quad (2)$$

U cooling coil-u količina izgubljene toplotne energija vazduha je jednaka apsorbovanoj energiji vode (3,4). Izlazna temperatura vode iz cooling coil-a se računa diferencijalnom jednačinom (5),

---

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio doc. dr Velimir Čongradac.

$$Q_C = \dot{m}_{w,C} c_{p,w} (T_{o,w,C} - T_{i,w,C}) \quad (3)$$

$$Q_C = \dot{m}_{a,C} c_{p,a} (T_{i,a,C} - T_{o,a,C}) \quad (4)$$

$$\dot{T}_{o,C} = (\dot{m}_{w,C} c_{p,w} (T_{i,w,C} - T_{o,w,C}) + Q_C) / C_c \quad (5)$$

gde su maseni protok vode $\dot{m}_{w,C}$ i maseni protok vazduha $\dot{m}_{a,C}$ konstantne vrednosti.

Izlazna temperatura vazduha iz mixing box-a nastalog u procesu mešanja spoljašnog (atmosferskog) vazduha i povratnog vazduha iz prostorija zgrade se računa jednačinom

$$T_{m,a} = (T_{o,a} \dot{m}_{o,a} + T_{r,a} \dot{m}_{r,a}) / \dot{m}_{m,a} \quad (6)$$

gde je $\dot{m}_{m,a} = \dot{m}_{o,a} + \dot{m}_{r,a}$ za konstantne vrednosti masenih protoka spoljašnjeg $\dot{m}_{o,a}$ i povratnog $\dot{m}_{r,a}$ vazduha, čije vrednosti su 20% i 80% od $\dot{m}_{m,a}$, respektivno.

Diferencijalnim jednačinama je pokušano da se oponaša dinamika sistema, i one su u softverkom paketu Simulink rešavane pomoću s-funkcije.

Nomenklatura:

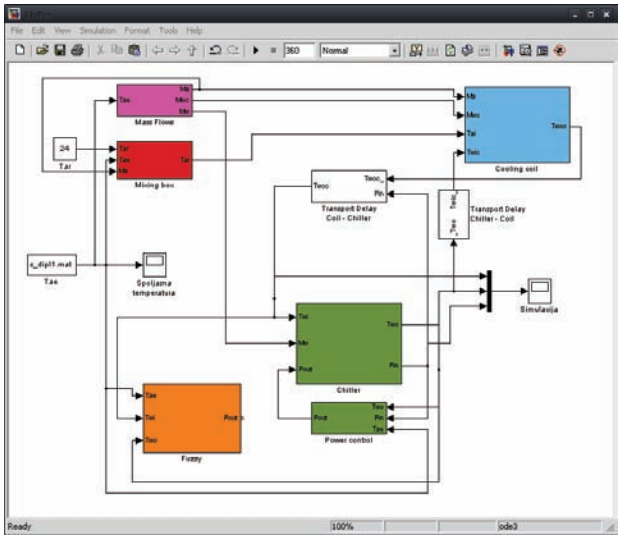
Q	rashladni kapacitet	[kW]
m	maseni protok	[kg/s]
c_p	specifična toplotna kapacitivnost	[kJ/kg^{o}C]
T	temperatura	[^{o}C]
C	termalni kapacitet	[kJ/^{o}C]

indeksi:

o - izlaz	w - voda
i - ulaz	r - povratni
m - mešavina, mix	C - cooling coil
a - vazduh	Ch - čiler

2.2. Simulink model

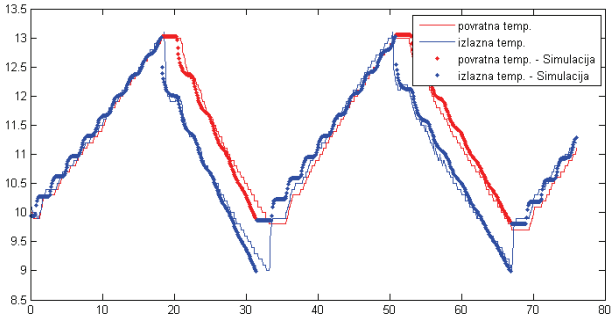
Similacioni model (Slika 2.1.) je organizovan u blokove. Blokovi Chiller i Power control simuliraju rad čilera. U bloku Chiller se računa vrednost izlazne temperature čilera i ona zavisi od vrednosti ulazne promenljive *Pout* za stanja kada rade dva ili sva četiri kompresora. U slučaju kada ne radi ni jedan kompresor, tada je vrednost izlazne temperature jednaka vrednosti ulazne temperature.



Slika 2.1. Simulink model sistema klimatizacije i čiler sistema

Blokom Power control se kontroliše rad bloka Chiller, odnosno njegova izlazna vrednost *Pout* govori koliko para kompresora treba da radi (0, 1, 2). Broj para kompresora koji treba da radi se određuje na osnovu vrednosti izlazne temperature vode iz čilera *Tws* i spoljašnje temperature *Tae*. Granična vrednost izlazne temperature vode iz čilera pri radu jednog para kompresora je od 9 do 13°C , dok za rad oba para kompesora od 5 do 9°C.

U bloku Cooling coil se simulira rad ovog toplotnog izmenjivača i izračunava vrednost temperature vode *Twoc,* koja je ujedno i povratna temperatura čilera. Vrednost temperature vazduha nakon prolaska kroz toplotni izmenjevač je konstantnih 15°C, i prestavlja temperaturu kondenzacije.



Slika 2.2. *Verifikacija povratne i izlazne temperature vode čilera*

U bloku *Mixing box* se izračunava vrednost temperature vazduha *Tai* kojim snabdevamo toplotni izmenjivač, gde je pretpostavljeno da vrednost povratne temperature vazduha *Tar,* iz prostorija zgrade, konstantnih 24°C. Simulacionim modelom sistema pokušano je da se dobro reprezentuje rad realnog sistema. Generisane vrednosti simulacionog modela za izlaznu i povratnu temperaturu vode čilera , kao i rad svakog pojedinačnog kompresora, su približne vrednostima dostupnih podataka sa realnog sistema, za iste vrednosti spoljašne temperature. Na slici 2.2 je prestavljeno poređenje podataka koji su rezultat simulacije i podataka sa rada realnog sistema za određeni vremenski period.

3. FAZI KONTROLER

Pošto je rad kompresora direktno srazmeran potrošnji, jasno je da nalaženjem optimizovanog broja kompresora potrebnih da rade u određenom trenutku , biti sprečeno rasipanje električne energije .

Za optimizaciju rada čilera kreirana su dva fazi kontrolera: *fuzzyChiller_Tk* i *fuzzyChiller_Compressors* .

3.1. fuzzyChiller_Tk

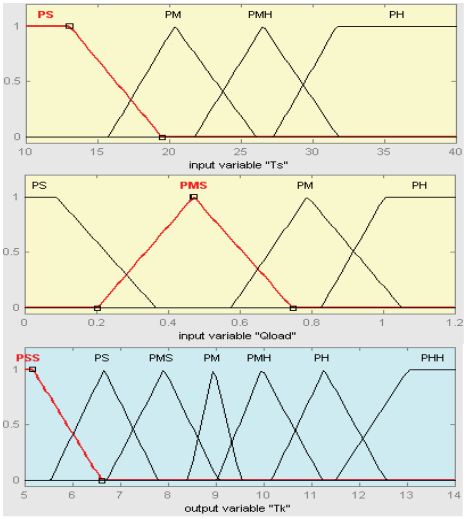
U *fuzzyChiller_Tk* kontroleru se na osnovu ulaznih fazi promenljivih, vrednosti spoljašne temperature *Ts* i rashladnog opterećenja zgrade *Qload,* donosi zaključak o optimalnoj izlaznoj temperaturi vode čilera *Tk* u datom trenutku.

Tabela 1. *Baza pravila za fuzzyChiller_Tk kontroler*

Qload \ Ts	**PS**	**PM**	**PMH**	**PH**
PS	PHH	PH	PMH	PMS
PMS	PHH	PH	PM	PS
PM	PH	PMH	PM	PSS
PH	PH	PM	PS	PSS

Izlazna temperatura *Tk* nije temperatura koju pokušavamo da precizno dostignemo radom kompresora, već tempe-

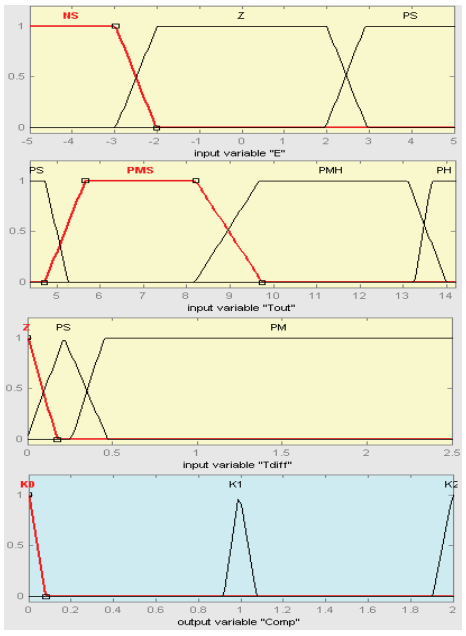
ratura koja ce nam služiti kao kriva vodilja, na osnovu koje ćemo definisati rad kompresora uz određenu grešku odstupanja trenutne od optimalne izlazne temperature. Funkcije pripadnosti lingvističkih vrednosti kao i njihov raspon za svaku definisanu promenljivu su prikazane na slici 3.1. Za potrebe ovog fazi kontrolera, definisano je 16 fazi pravila (Tabela 1.).



Slika 3.1. *Funkcije pripadnosti za Ts, Qload i Tk*

3.2. fuzzyChiller_Compressors

U ovom kontroleru definisane su sledeće ulazne fazi promenljive: a) temperaturna razlika između trenutne izlazne temperature i optimalne izlazne temperature generisane *fuzzyChiller_Tk* kontrolerom, i njena vrednost je $E(t) = Tk(t) - Two(t)$, b) temperaturna razlika između trenutne ulazne i izlazne temperature čilera i njena vrednost je $Tdiff(t) = Twi(t) - Two(t)$, c) trenutna izlazna temperatura čilera *Two*.



Slika 3.2. *Funkcije pripadnosti za E, Tout ,Tdiff i Comp*

Promenljivom *E* pratimo odstupanje vrednosti izlazne temperature od generisane optimalne temperature, i kontrolišemo da ona bude u okviru granica koje smo definisali. Time sprečavamo često paljenje i gasenje kompresora koje uzrokuje daleko vecu potrošnju elekrične energije. Promenjliva *Tdiff* nam govori kolika je iskorišćenost hladne vode, tj. da li je ona dovoljna da zadovoji potrebe zgrade. Koristi se izlazna temperatura *Two* jer njena vrednost određuje koliko kompresora bi trebalo da radi da bi se dostigla ta vrednost.
Funkcije pripadnosti lingvističkih vrednosti kao i njihov raspon za svaku definisanu promenljivu su prikazane na slici 3.2 . Za potrebe optimizacije rada čilera, definisano je 16 fazi pravila (Tabela 2.).

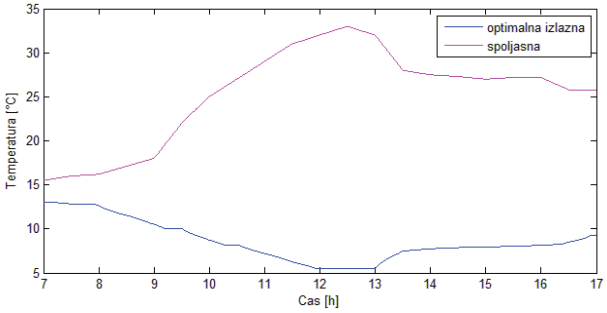
Tabela 2. *Baza pravila za fuzzyChiller_Compressors*

	E	***Tout***	***Tdiff***	***Comp***
1	NS	PMS	Z	K2
2	NS	PMS	PS	K2
3	NS	PMS	PM	K2
4	NS	PMH	Z	K1
5	NS	PS	PS	K2
6	NS	PM	PM	K1
7	Z	PMS	Z	K0
8	Z	PMS	PS	K2
9	Z	PMS	PM	K2
10	Z	PMH	Z	K0
11	Z	PMH	PS	K2
12	Z	PMH	PM	K1
13	PS	PMS	-	K0
14	PS	PMH	-	K0
15	-	PS	-	K0
16	-	PH	-	K1

Vrednost koja izlazi iz *fuzzyChiller_Compressors* kontrolera je broj para kompresora potrebnih da rade u datom trenutku i upravo je to način kojim smo želeli da optimizujemo rad čilera kako bi minimizovali potrošnju električne energije.

4. REZULTAT OPTIMIZACIJE

U standardnom režimu rada ne koristi se nikakva optimizacija rada čilera koja za cilj ima smanjenje potrošnje električne energije u zavisnosti od atmosferskih prilika i potrebe zgrade za hlađenjem, već je izlazna temperatura čilera fiksirana na minimalnu vrednost kako bi potrebe zgrade bile zadovoljene u svakoj situaciji.



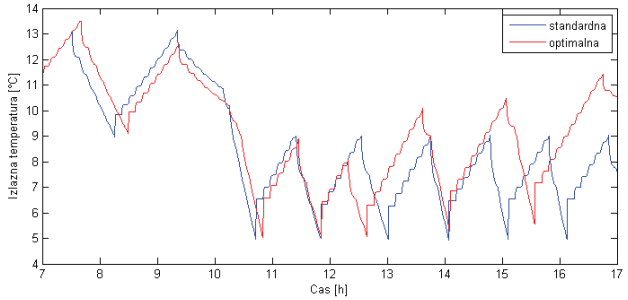
Slika 4.1. *Promena optimalne izlazne temperature Tk i spoljašne temperature u toku radnog dana*

Optimalne vrednosti za izlaznu temperaturu čilera se izračunava u *fuzzyChiller_Tk* kontroleru na osnovu spoljašne temperature i rashladnog opterećenja zgrade.
Promena optimalne vrednosti izlazne temperature u toku radnog dana je prikazana na slici 4.1. Za period između

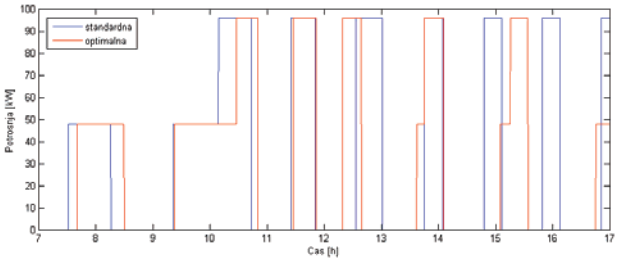
12h i 13h, spoljašna temperatura dostiže najviše vrednosti i rashladno opterećenje zgrade ima visoke vrednosti, pa su vrednosti optimalne izlazne temperature u ovom periodu najniže, sto je očekivano, jer su tada najveće potrebe za hlađenjem zgrade.

Nakon određene vrednosti optimalne temperature, u *fuzzyChiller_Compressors* kontroleru se izračunava koliko kompresora treba da radi u datom trenutku, vodeći računa da se isporuči dovoljna količina hladne vode i time zadovoljile potrebe zgrade.

Na slici 4.2 su prikazane promene izlazne temperature pri standardnom režimu rada i izlazne temperature nakom optimizacije, dok na slici 4.3 su prikazane potrošnje elektricne energije čilera za standardni režim rada i u slučaju optimizacije u toku radnog dana.



Slika 4.2. *Uporedni prikaz izlazne temperature čilera za standardni i optimizovani režim rada u toku radnog dana*



Slika 4.3. *Uporedni prikaz potrosnje čilera za standardni i optimizovani režim rada u toku radnog dana*

Potrošnja električne energije čilera moze imati samo tri moguće vrednosti , i to 0, 48, 96 kW, i odgovara stanjima kada ne radi nijedan, rade samo dva ili sva četiri kompresora, respektivno. Pa na osnovu toga , na slici 4.3 vidimo koliko kompresora je radilo u kom periodu dana. Uočavamo, da je ušteda u potrošnji električne energije postignuta u kasnim jutarnjim i popodnevnim satima.

Analizom potrošnje sistema u slučaju optimizacije je utvrđeno da u odnosu na posmatrani vremenski period, ukupna potrošnja elekrične energije u slučaju optimizovanog rada čilera iznosi 1073 kW, a u slučaju standardnog režima rada 1198 kW, pa vidimo da je postignuta ušteda od 10.4%.

5. ZAKLJUČAK

Osnovna ideja u ovom radu, kada je u pitanju nastojanje da se potrošnja električne energije čilera svede na minimum, je da se odrede optimalne vrednosti za izlaznu temperaturu vode čilera, i na osnovi iste optimizuje broj kompresora koji će raditi. Dobijeni rezultati pokazuju da je optimizacija moguca i da ušteda od 10,4% nije zanemarljiva, kada se uzme u obzir potrošnja i cena elektricne energije. Eksperimentalnim putem smo pokazali da je zgradu moguće klimatizovati i sa višom izlaznom temperaturom u odnosu na onu koja se primenjuje u sistemu pri njegovom standardnom rezimu rada, ne smanjući pritom komfornost u zgradi .

6. LITERATURA

[1] *"Direktiva 2002/91/EC Evropskog parlamenta i saveta o energetskoj efikasnosti zgrada"*, 2002.

[2] M.Anderson, M.Buehner, P.Young, D.Hittle, C.Anderson, J.Tu, D.Hodgson, *"An experimental system for advanced heating, ventilating and air conditioning (HVAC) control",*Energy and Buildings 39, pp. 136–147, 2007.

[3] K. N. Nagabhushan, K.N and D. C. Hittle, *"Modeling Of Non-Linear HVAC system using SIMBAD"*, Clima 2007 WellBeing Indoors, 2007.

[4] P.R. Novak, N. Mendes,G. Oliveira, *"Simulation of HVAC plants in 2 brazilian cities using Matlab/Simulink"*, Building Simulation, pp. 859-866, 2005.

[5] T.T. Chow, J.A. Clarke,A. Dunn, **"***Primitive parts: an approach to air-conditioning component modelling"*, Energy and Buildings 26, pp. 165-173, 1997.

[6] M. Albieri, A. Beghi, C. Bodo, L. Cecchinato, *"Advanced control systems for single compressor chiller units",* ,International Journal of Refrigeration I-9, 2008.

[7] S. Wang, J. Wang, J. Burnett *"Mechanistic model of centrifugal chillers for HVAC system dynamics simulation"*, Building Serv. Eng. Res. Technol., vol 21, pp. 73-83, 2000.

[8] C.P. Underwood, *"Analysing Multivariable Control Of Refrigeration Plant Using Matlab/Simulink",* Building Simulation 2001, pp. 288-294, 2001.

[9] M. Mongkolwongrojn, V. Sarawit, *"Implementation of Fuzzy Logic Control for Air Conditioning Systems"*, ICCAS, Jun 2005.

[10] F.V. Silva, L.C. Filho, V. Silveira Jr., *"Experimental evaluation of fuzzy controllers for the temperature control of the secondary refrigerant in a liquid chiller"*, Journal of Food Engineering Vol.75, pp. 349-354, August 2006.

[11] J.Singh, N. Singh, J.K. Sharma, *"Fuzzy modeling and identification of inteligent control for refrigeration compressor"*, Journal of Scientific & Industrial Research, Vol. 65, pp. 22-30, 2006.

[12] J. Baek, E. Kim, M. Park, *"Adaptive fuzzy output feedback control for nonlinear hating, ventilating, and air conditioning system"*, ITC-CSCC-2008, pp. 1447-1480, 2008.

[13] R. Talebi, "Application Of Fuzzy Logic For Control Of Heating, Chilling, And Air Conditioning Systems", CRC Press, 2001.

Kratka biografija:



Damir Vujaklija rođen u Velesu, R. Makedonija, 1981. godine. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Optimizacija željene vrednosti temperature rashladnog fluida čilera primenom fazi logike odbranio je 2011.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 620.93

FARME VETRENJAČA NA MORU

OFF-SHORE WIND FARMS

Jelena Kartal, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast- ELEKTROTEHNIKA I RAČUNARSTVO

Sadržaj – *Prikazan je koncept farmi vetrenjača na moru kao i njeni osnovni delovi. Prezentovani su načini instalacije svih delova farmi vetrenjača. Navedene su trenutno operativne farme vetrenjača u Severnom moru, a zatim su i opisane najzanimljivije. Na kraju je diskutovana neophodnost izgradnje Super grida.*

Abstract – *The paper presents a concept of off-shore wind farms, as well as its basic parts. Methods of installation of all components of a wind farm are presented. Currently operational off-shore wind farm in the North Sea are noted and the most interesting are described. Finally, the necessity of building the Super grid is discussed.*

Ključne reči: *Vetrenjače na moru, instalacija, Super grid.*

1. UVOD

Farme vetrenjača na moru (*off-shore wind farms*) predstavljaju vetroelektrane postavljene na moru, koje se nalaze minimalno 10 km od kopna. Prednosti ovog pristupa se ogledaju u korišćenju većih brzina vetra nego na kopnu, pa samim tim i značajnijem doprinosu u pogledu isporučene električne energije. Sastoje se od turbine, stuba, temelja i električnih veza sa elektroenergetskim sistemom. U radu će biti opisan i proces njihovog postavljanja i održavanja, koji je složeniji nego kod vetrenjača na kopnu (on-shore), kao i povezivanja na EES.

2. OSNOVNI SKLOPOVI

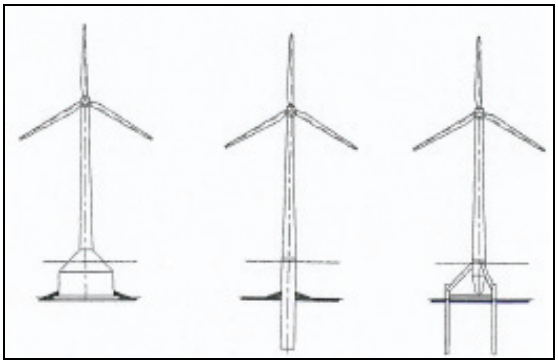
Osnovna jedinica svih off-shore farmi je vetrenjača. Skoro sve vetrenjače se sastoje od turbine u obliku trokrakih elisa, koje rotiraju oko horizontalnog središta. Moguće su i drugačije konstrukcije. Turbina je povezana sa multiplikatorom (menjačem) i generatorom, koji se nalaze unutar gondole. Elisa ima tri, a prečnik im dotiže do 124 metra. Najčešće su napravljene od ojačanog poliestera. Okreću se polako, obično 8 do 18 obrtaja u minuti pri vetru brzine izmedju 4 i 27 m/s za snage preko 1,5 MW, odnosno izmedju 14 i 32 m/s za manje snage.

NAPOMENA:
Ovaj rad je proistekao iz master rada čiji je mentor bio dr Vladimir Katić, red prof.

Stubovi su uglavnom cilindrični i sačinjeni od čelika, a njihove visine se kreću u rasponu od 25m do preko 100m.

Ključni parametar jeste prečnik elise. Što je veći, veće područje se zahvata i dobija se veća energija na izlazu [11]. Dva osnovna tipa vetrenjača su vetrenjače sa konstantnom i promenljivom brzinom.

Vetrenjače na moru zahtevaju različite vrste potpornih struktura. Izbor potporne strukture na prvom mestu zavisi od dubine vode na mestu gde je predviđeno postavljanje vetrenjače. Dva su osnovna tipa: strukture koje se montiraju na dnu mora i plutajuće strukture. Osnovni tipovi potpornih struktura koje se montiraju na dno mora su prikazane na slici 1.



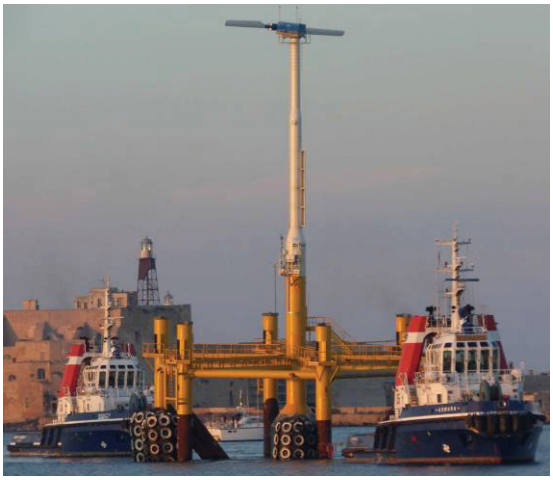
Slika 1. *Tri najčešće vrste potpornih struktura koje se montiraju na dno mora [10].*

2. INSTALACIJA FARMI VETRENJAČA NA MORU

Proces instalacije farmi vetrenjača na moru zavisi od mnogo faktora. Na prvom mestu koja vrsta potpornih struktura se koristi. Zatim zavisi i od samih vetrenjača, da li se sastavljaju na kopnu ili na moru. Prvo se postavljaju osnove, kablovi, a zatim i same vetrenjače. Sledeći, veoma bitan faktor, pri postavljanju vetrenjača je vreme. Naime, za postavljanje osnove i tornja nije potrebno savrseno vreme (bez daška vetra), ali za postavljanje elisa rotora neophodno je krajnje odsustvo vetra, što dodatno komplikuje stvari. Zahtevajući akumulaciju većeg broja vetrenjača na dokovima, kako bi u slučaju lepog vremena bilo moguće postaviti što više vetrenjača.

Način instalacije potpornih struktura i vetrenjača zavisi na prvom mestu od tipa vetrenjača. U slučaju da je vetrenjači potrebna potporna struktura koja se montira na dno mora koriste se standardne Jack-barže i neka namenski napravljena plovila, dok u slučaju vetrenjača koje koriste plutajuće potporne strukture, kao sto su Wind float platforme, za ovim specijalnim brodovima nema potrebe

pošto se cela vetrenjača zajedno sa potpornom strukturom sastavi na kopnu, a zatim običnim remorkerom odvuče na pučinu. Na slici 2. možemo videti kako kompletnu strukturu Blue H vetrenjače vuče jedan obični remorker [1]. Ovo predstavlja veliku prednost, pošto je postavljanje standardnih struktura koje se postavljaju na dno mora, kao i naknadno postavljanje vetrenjača na njih veoma skup proces (što više odvojenih delova, koje treba sastaviti na moru, ima vetrenjača, to će sama instalacija trajati duže i samim tim koštati više), koji pored toga što zahteva skupu opremu, zahteva i veoma obučen ljudski kadar.



Slika 2. *Remorker koji vuče vetrenjaču Blue H tehnologije zajedno sa potpornom strukturom [1].*

Za postavljanje vetrenjača sa potpornim strukturama koje moraju da se postave na dno mora, koriste se brodovi sa ugradjenim kranovima ili Jack barže. Jack barže obično imaju 4-6 nogu koje postavljanju na morsko dno i oslanjanjem na njih izdižu plovilo van vode. Jack barže trenutno mogu da postavljaju vetrenjače do dubine od 35 metara.

Za veće dubine koriste se specijalna plovila koja poseduju plutajuće hidraulične dizalice. Instalacione tehnike offshore vetrenjača još uvek nisu optimizovane za velike količine i brzine. Takođe, postoje mogućnosti da vetrenjače, postolja i priključenja budu tako dizajnirana da olakšaju proces instalacije.

U slučaju gravitacione potporne strukture neophodno je prvo izravnati tlo predviđeno za postavljanje i spremiti lomljeni kamen kojim se fiksira struktura nakon njenog postavljanja. Šematski prikaz postavljanja gravitacione strukture je prikazana na slici 3.

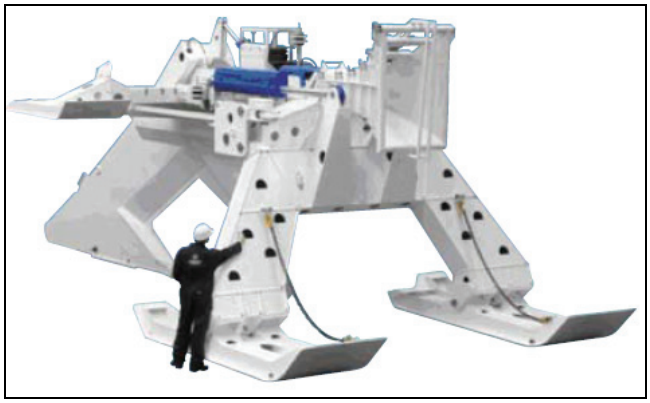


Slika 3. *Grafički prikaz postavljanja gravitacione osnove [7].*

Ako se postavlja monopile osnova neophodno je korišćenje vibracionog čekića uz čiju pomoć se ukopava na željenu dubinu [9]. Vreme potrebno za postavljanje ovakve osnove je relativno kratko (manje od 24 sata).

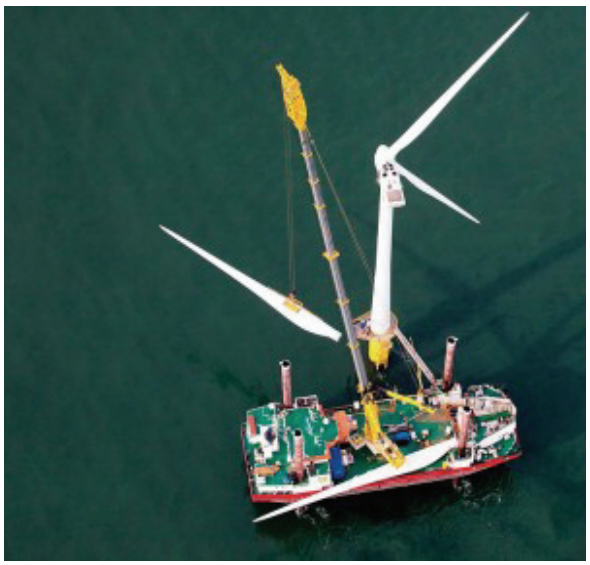
Nakon postavljanja osnova, na red dolaze kablovi. Instalacija kablova se može izvršiti na više načina. Ovde ćemo prezentovati samo jedan.

Da bi se postavio kabl prvo se instalira cev koja se ukopava u dno mora i dalje u obalu. Nakon njenog uspešnog postavljanja, kroz nju se postavlja visoko naponski kabl u smeru sa kopna ka moru. Postavljanje kabla i dodatno kopanje se radi u jednom potezu uz pomoć specijalnog robota prikazanog na slici 4. Kabal se provlači i automatski zakopava na dubinu od 2 metra. Na taj način kabl je zaštićen od oštećenja koja mu mogu naneti sidra brodova, ribarske mreže i druge eventualne pretnje [7].



Slika 4. *Specijalizovani robot za postavljanje i ukopavanje kablova u morsko dno [7].*

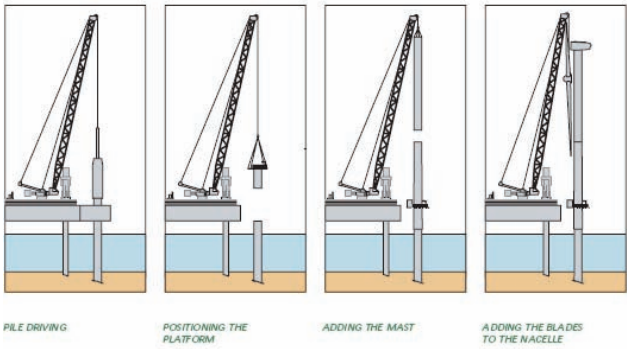
Nakon postavljanja kablova montira se kula ili ceo sklop vetrenjače. Ukoliko je konstrukcija vetrenjače takva da se svaki deo mora posebno sklapati tada se prvo postavlja stub, gondola i tek onda elise, jedna po jedna ili sve zajedno, opet u zavisnosti od same konstrukcije (slika 5) [9]. Međutim, ako je konstrukcija vetrenjače takva da se njen ceo sklop montira na obali, onda se ona jednostavno odnese do željene lokacije i postavi na osnovu (slika 6) Na slici 7 je prikazana hronohogija montiranja osnove i delova vetrenjača.



Slika 5. *Postavljanje elise uz pomoć krana postavljenog na Jack barži [6].*



Slika 6. *Postavljanje celokupne vetrenjače na osnovu uz pomoć namenskog plovila sa hidrauličnim dizalicama [7].*



Slika 7. *Hranologija montiranja osnove i delova vetrenjače [9].*

3. POSTOJEĆE FARME VETRENJAČA U SEVERNOM MORU

Danska je bila pionir u razvoju komercijalnih vetrenjača tokom 1970.-ih. Trenutno ima dve offshore farme vetrenjača u Severnom moru Hormn Rev 1 i Horns Rev 2. Nemačka trenutno ima samo jednu operativnu offshore farmu u Severnom moru, Alpha Ventus snage 60 MW. Holandija ima dve farme vetrenjača u Severnom moru, Princess Amalia i OWEZ. Velika Britanija je lider u broju offshore farmi vetrenjača u Severnom moru, čak šest ih je operativno (Thanet, Lynn and Inner Downsing, Gunfleet Sands, Kentish Flats, Scroby Sands i Beatrice). Belgija ima samo jednu farmu u Severnom moru, Thorntonbank, kapaciteta 30 MW. Norveška, kao i Belgija, ima samo jednu farmu u Severnom moru, Hywind, kapaciteta 2.3 MW.

Thanet je najveća operativna farma vetrenjača na moru u svetu koja je puštena u rad 2010.-e godine, kapaciteta 300 MW. Čine je 100 Vestas vetrenjača tipa V90 snage 3MW, postavljene na monopile. Vetrenjače su postavljene na dubini od 20 do 50 metara i pokrivaju površinu od 35 kilometara kvadratnih, 11 kilometara od kopna. Svaka vetrenjača je visoka 115 metara na svojoj najvišoj tački, razmak između njih je oko 500 metara, a izmedju redova 800 metara. Farma ima offshore transformator koji podiže napon sa 33kV na 132kV odakle dva podvodna kabla prosleđuju energiju mreži na kopnu [3]. Na slici 7 je prikazan izgled farme vetrenjača Thanet.

Beatrice je eksperimentalna farma vetrenjača na moru sa najmanjim kapacitetom od svih Britanskih farmi u Severnom moru, od samo 10 MW. Puštena je u rad 2007.-e. Čine je dve REpower vetrenjače snage 5MW, postavljene na dubini od 45 metara, 23 kilometra od obale. Vetrenjače su postavljene na jacket osnove dužine 70 metara, težine 760 tona. Ovo je prva farma gde su svi delovi vetrenjače sklopljeni na kopnu, a zatim odveženi na more i montirani na osnovu [2]. Na slici 9 je prikazana farma vetrenjača na moru Beatrice.



Slika 8. *Izgled farme Thanet [3].*



Slika 9. *Izgled farme Beatrice [2].*

Hywind je eksperimentalna, prva plutajuča vetrenjača na svetu tipa Siemens SPW 2.3-82 (prikazana na slici 10). Puštena je u rad 2009.-e godine. Postavljena je na dubini od 220 metara, 10 kilometara od obale. Primarna namena je da se testira koliko vetar i talasi utiču na strukturu kako bi se koncept mogao komercijalizovati. Vetrenjača proizvodi energiju napona 22kV koji se dalje prosleđuje do transformatorske stanice na kopnu. Kabal koji povezuje vetrenjaču sa kopnom je dug 10 km [4].



Slika 10. *Izgled farme Hywind [4].*

4. SUPER GRID

2009.-e godine Evropska unija (EU) i G8 šefova vlada obavezali su svoje zemlje na 80% smanjenja gasova koji izazivaju efekat staklene bašte gasova do 2050. Da bi postigla ove ciljeve Evropska unija mora da nađe način da proizvede enegiju sa skoro nula emisije ugljenika. Slanje

energije koja je proizvedena sa nula emisije ugljenika u domove i preduzeća širom EU će zahtevati otvoreno tržište električne energije, podržano od strane unapređene i nove trans-nacionalne prenosne mreže. Nova generacija prenosnog sistema ne samo da će omogućiti predaju značajne količine energije dobijene na farmama vetrenjača na moru tamo gde je potrebna, nego će i fizički međusobno povezati evropski energetski sektor što će dovesti do veće sigurnosti snabdevanja, integracije obnovljivih izvora energije i veće energetske efikasnosti.

Za razliku od onshore mreža koje koriste naizmeničnu struju, offshore mreža bi koristila jednosmernu struju i stoga bi zahtevala nove vrste offshore trafostanica, sisteme kontrole, konvertore i prekidače [5].

Konačni izgled super grida će biti razmotren zajedno sa predviđenenim kapacitetima na kopnu i moru, poboljšanjem fleksibilnosti sistema i globalnog uticaja na životnu sredinu i troškove. Jedan od predloga za izgled Super grida je prikazan na slici 11.



Slika 11. *Predviđen izgled Super grida 2050.-e [5].*

5. ZAKLJUČAK

U proizvodnji električne energije nijedan izvor nije imao tako dinamičnu ekspanziju u poslednjih dvadesetak godina. Off-shore vetroelektrane daju čistu i ekonomičnu energiju. Savremene jedinice dostižu snagu od 5 MW, sa razvojnim planovima do 8MW (pa i 10MW). Po ekonomičnosti su izjednačeni sa klasičnim izvorima, tako da se u narednom periodu može očekivati da će energija vetra zauzeti značajno mesto u ukupnom svetskom energetskom bilansu. Time neposredno doprinose i odluke Vlade Nemačke od 31.05.2011. o zatvaranju svih nuklearnih elektrana do 2024. godine.

Najveći problem izgradnje farmi vetrenjača na moru je visoka cena. Pored troškova izgradnje (transport komponenti, osnova, električne mreže) i prenosni energetski sistem do kopna mnogo skuplji nego kod farmi na kopnu. Specijalno dizajnirane i veoma skupe konstrukcije, kao i off-shore platforme (transformatori, konvertor stanice) su potrebne. Sva postavljena oprema je projektovana da zadovolji visoke standarde kako bi bila efikasna u visoko korozivnom off-shore okruženju. Održavanje, kako same farme tako i prenosnog sistema zahteva visoko obučen kadar i specijalnu opremu i transportna sredstva (helikopteri, brodovi specijalno dizajniran za ovu svrhu). U mnogim slučajevima, održavanje se može značajno odložiti zbog loših vremenskih prilika, što dovodi do dodatnih troškova. Međutim, smatra se da će se narednih godina svi ovi troškovi smanjivati sa većom proizvodnjom i korišćenjem proverenih tehnologija.

Predviđene off-shore farme u budućnosti će zahtevati fleksibilniju električnu mrežu od ove danas u pogledu rukovanja fluktuirajućom snagom. Predviđeni super grid će omogućiti dostavljanje jeftine energije bilo kom delu Evrope i tako učiniti sve evropske države mnogo energetski stabilnijim.

6. LITERATURA

[1] Blue H Usa, prezentacija proizvoda: www.bluehusa.com

[2] The Beatrice Windfarm Demonstrator, prezentacija projekta: all-energy.co.uk

[3] **"**Massive U.K. Offshore Wind Farm Fires Up" Nino Marchetti, Septembar 2010.

[4] "Hywind floating wind turbine goes live" Ross Dulmaine, Septembar 2009.

[5] "Position paper on the EC Communication for a European Infrastructure Package", discussion: www.friendsofthesupergrid.eu

[6] "Offshore pedestal LTR11200", November 2010. Source www.vertikal.net

[7] BEATRICE WIND FARM, www.scaldis-smc.com

[8] "Far shore wind farm Thorntonbank, Belgium", www.deme.be

[9] "Barrow Offshore Wind Farm", www.warwickenergy.com

[10] T. Feld, J.L. Rasmussen, P.H. Sorensen, "Structural and Economic Optimization of Offshore Wind Turbine Support Structure and Foundation" , Feotechnical engineering group Aalborg university Denmark, Jul 1999.

[11] Gilbert M. Masters, "Renewable and Efficient Electric Power Systems" , Stanford University.

Kratka biografija:



Jelena Kartal rođena je u Beogradu 1983. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Mikroračunarska elektronika odbranila je juna 2011.god.



Vladimir Katić je doktorsku disertaciju odbranio je na Univerzitetu u Beogradu 1991. Redovan profesor Fakulteta tehničkih nauka u Novom Sadu je od 2001. za oblast Energetska elektronika, mašine i pogoni. Prodekan je FTN-a i šef katedre za energetsku elektroniku i pretvarače. Oblasti interesovanja su energetska elektronika, kvalitet električne energije i obnovljivi izvori električne energije.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.311.16:004.4

UTICAJ PROMENE NAPONA NA SNAGU POTROŠAČA

VOLTAGE ALTERNATION INFLUENCE ON CONSUMER POWER

Bojan Cvetinov , *Fakultet Tehničkih Nauka , Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu su prikazani osnovni principi kontrole potrošnje potrošača na osnovu promene vrednosti napona na njima. Pritom, osnovni cilj te kontrole je smanjenje (redukcija) ukupne aktivne snage injektirane u distributivnu mrežu (DM). Smanjenjem aktivne snage koju distributivna mreža preuzima iz prenosne mreže, uz poštovanje tehničkih kriterijuma, rasterećuju se proizvodni i prenosni kapaciteti i smanjuje zagađenje okoline. Smanjenje gubitaka u prenosu i distribuciji, koje se takođe postiže regulacijom napona, dovodi do povećanja efikasnosti samog sistema.*

Abstract – *In this paper basic principles of consumer power control by changing voltage on them are shown. Here, the main goal of this control is to reduce (reduction) total active power injected into distribution network. Reducing active power that distribution network is taking from transmission network, with respecting technical criteria, relives production and transmission capacity and reduces environmental pollution. Reducing transmission and distribution losses ,witch is also achieved by voltage regulation, leads to increasing efficiency of the system itself.*

Ključne reči – *Regulacija napona, kontrola potrošnje, smanjenje gubitaka*

1.UVOD

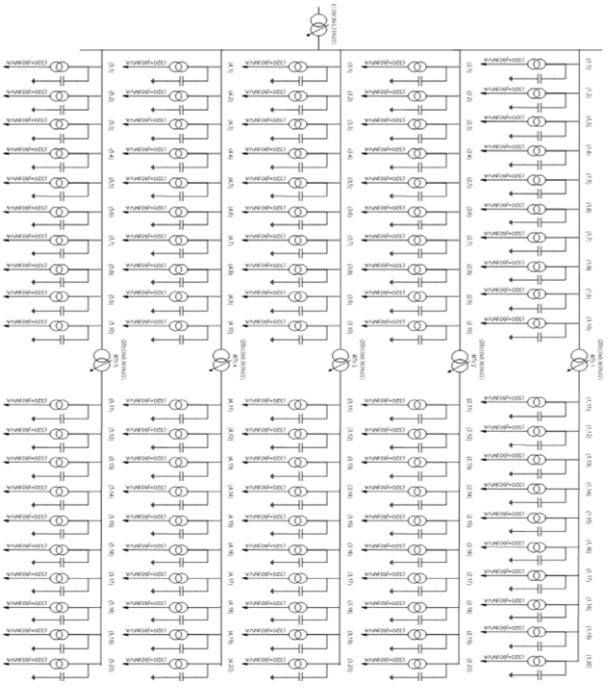
Kontrolom napona moguće je smanjiti snage potrošnje samih potrošača. Takođe ravnanjem naponske slike,odnosno smanjenjem pada napona mogu se smanjiti i gubici snage duž voda. U ovom radu su razmotrena dva načina regulacije napona koji se koriste u te svrhe.

Prvi način je ubacivanje kondenzatorskih baterija (KB) u sistem. One podižu napon u tačkama u kojima su postavljene i menjaju naponsku sliku duž voda. Njima se smanjuje pad napona a sa njime i gubici snage.

Drugi način je redukcija napona regulacionim transformatorima (RT). Smanjenje napona na potrošaču,kod potrošača kojima je snaga zavisna od napona (konstantne struje i konstantne impedanse) dovodi do smanjenja ukupne injektirane aktivne snage u sistem.Ovo smanjenje kvantitativno zavisi od karakteristika potrošača.

Naravno, postoje ograničenja koje utiču na upotrebu ovih metoda. Pre svega, primenom KB profil napona ne može u potpunosti da se izravna. Osnovni razlozi za to su: 1) KB su regulacioni resursi sa diskretnom mogućnošću regulacije; 2) kod naponsko zavisnih potrošača povećanjem napona više se poveća potrošnja aktivne snage nego što se smanje gubici aktivne snage; 3) redukcija napona je ograničena tehničkim minimumom napona [1].

Za realizaciju osnovnog cilja ovog rada formiran je poseban softver. Uticaj RT i KB kvantifikovan je na primeru radijalne distributivne test mreže koja je prikazana na slici 1. Mreža se sastoji od pet paralelnih srednjenaponskih vodova sa po dvadeset potrošača. Na početku mreže postoji jedan napojni RT i po jedan RT na sredini svakog od vodova. KB će se priključivati u mrežu u potrošačke čvorove sa VN strane distributivnih tranformatora. Po dve KB će se,po potrebi,postaviti na svaki vod a njihovo mesto i veličina zavise od konfiguracije DM. (U slučaju jedne KB dobijaju se lošiji rezultati dok je postavljanje KB u svaki čvor neekonomično.) Simulacije za ove slučajeve su takođe urađene i rezultati su dati u radu.



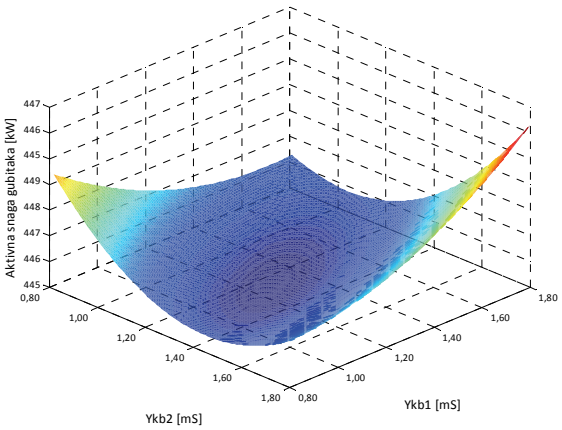
Slika 1. Izgled distributivne test mreže

Softver je realizovan u programskom paketu Matlab, a za proračun tokova snaga korišćen je algoritam sumiranja struja [2].

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je prof. dr Goran Švenda.

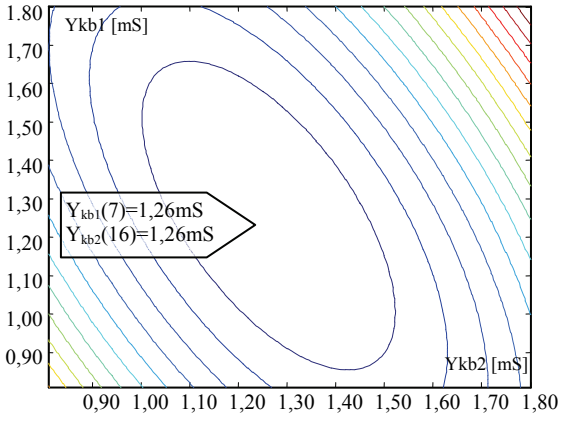
2.UTICAJ KONDENZATORSKIH BATERIJA

Uticaj kondenzatorskih baterija (KB) na naponske prilike u sistemu zavisi od mesta na koje su postavljene, od njihove količine i veličine. Najbolji rezultati smanjenja gubitaka aktivne snage za sve tipove potrošača se dobijaju ako se baterije nalaze u svim potrošačkim čvorovima mreže. Međutim to rešenje je i najmanje ekonomski isplativo,odnosno ima najduži period isplaćivanja.
Zadovoljavajući rezultati uz poštovanje ekonomskih faktora se dobijaju kada se postave po dve KB na svakom vodu.
Na slici 2. je data zavisnost aktivne snage gubitaka u odnosu na veličinu dve KB koje su priključene na vod. Ova slika se odnosi na slučaj kada su u mreži potrošači konstantne struje i simulacijom se dobija minimum gubitaka kada su ove KB postavljene u 7. i 16. čvor.



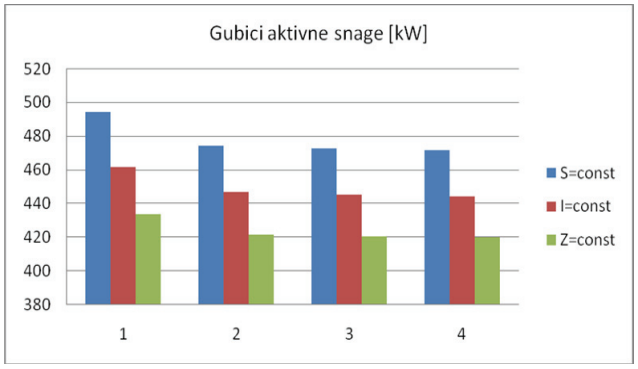
Slika 2. Rezultat simulacije sa dve KB i potrošačima tipa konstantne struje

Rezultati ove simulacije se bolje vide na grafu kontura koji je dobijen iz slike 2 i prikazan na slici 3. Svaka kontura predstavlja smanjenje gubitaka aktivne snage za 420W.



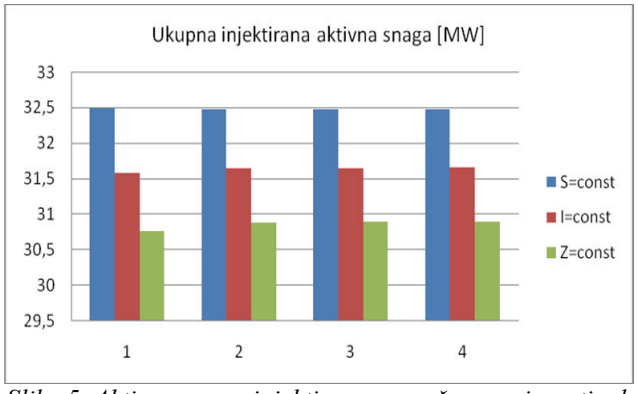
Slika 3. Rezultat simulacije prikazani konturom grafika

Na slici 4 je prikazan uticaj priključenih KB na vrednost gubitaka aktivne snage za sva tri tipa potrošača i to za primere kada je u mreži priključena:
1) ni jedna KB, 2) po jedna KB na vodu, 3) po dve KB na vodu i 4) po jedna KB u svakom potrošačkom čvoru.



Slika 4. Gubici aktivne snage u mreži u zavisnosti od ubačenih KB

Na slici 5 je za iste primere dato poređenje za vrednost aktivne snage injektirane u razmatranu mrežu.



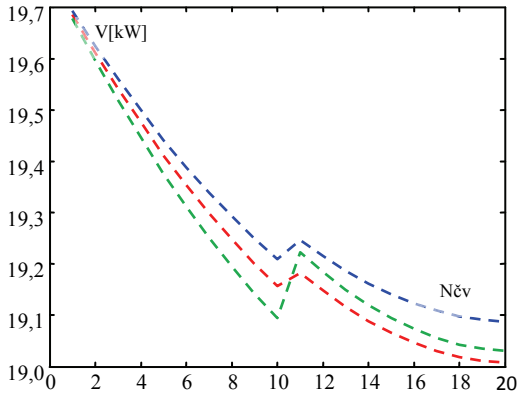
Slika 5. Aktivna snaga injektirana u mrežu u zavisnosti od uključenih KB

Kao što se vidi sa slika 4 i 5 ovaj metod regulacije napona donosi smanjenje gubitaka, ali ne i smanjenje vrednosti aktivne snage injektirane u mrežu. Razlog: povećanjem napona vrednost aktivne snage naponski zavisnih potrošača će porasti više nego što će se smanjiti gubici aktivne snage [3].

3.REDUKCIJA NAPONA

Redukcija napona se koristi kao jedan od načina kontrole napona i potrošnje. Pomoću RT spušta se napon u sistemu a sa njim i potrošnja aktivne snage potrošača koji imaju naponsko zavisnu karakteristiku.
Ova metoda regulacije je ograničena tehničkim minimumom napona koji je dozvoljen u sistemu. Primenom RT koji se nalaze na sredini voda može se podići naponski profil u drugoj polovini voda (najkritičniji deo voda, sa najnižim naponima). Simulacijom mreže dobijaju se dozvoljene kombinacije pozicija teretnih menjača svih RT u sistemu pri kojima imamo minimum injektirane aktivne snage [4].
Kod potrošača konstantne snage redukcija napona ne dovodi do smanjenja injektirane aktivne snage.Ono se kod ove vrste potrošača može postići samo smanjenjem gubitaka aktivne snage koje RT na vodovima,zbog sopstvenih gubitaka, povećavaju. Zato treba uzeti u obzir i varijante potpunog isključivanja ovih transformatora iz sistema .
Na slici 6 je prikazan profil napona na vodu koji primenom RT može minimalno da se realizuje. Može se

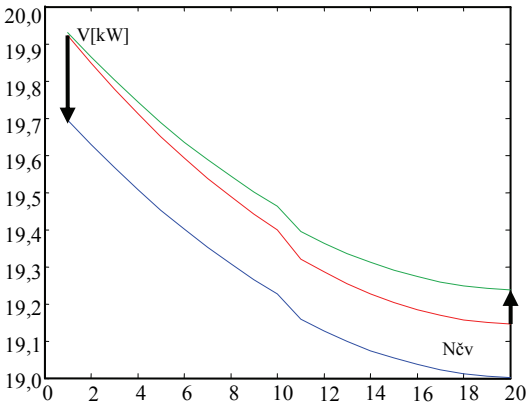
uočiti da je za mrežu sa potrošačima konstantne snage (zelena linija) potrebno postaviti teretni menjač RT na viši nivo.



Slika 6. Redukcija napona primenom RT, za sva tri tipa potrošača: (S) zelena, (I) crvena i (Z) plava

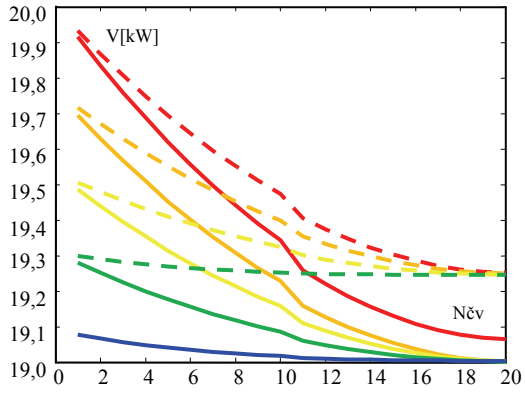
4. UTICAJ REGULACIONIH TRANSFORMATORA I KONDENZATORSKIH BATERIJA

Dodatna redukcija napona, na početku voda, moguća je samo ako se u regulaciju uključe i mogućnosti (efekti) KB. Njihovim uključenjem podižu se vrednosti celokupnog naponskog profila. To je pokazano na slici 7.Crvenom linijom je prikazan početni napon na vodu a zelenom povišeni napon dobijen posle priključenja KB. U trenutku kada se uključenjem KB naponski profil "dovoljno" podigne, kontrolu preuzima RT i naponski profil se smanjuje promenom pozicije teretnog menjača, plava linija na slici 7. (Dovoljno podignut naponski profil znači da promenom pozicije teretnog menjača za jednu vrednost naponi na svim potrošačima ostaju u tehničkim granicama).



Slika 7. Redukcija napona primenom RT i KB

Na slici 8 se vidi kako se pomoću KB može delovati na naponski profil u toj meri da se pomoću RT napon na vodu može spustiti do tehničkog minimuma. U ovom slučaju KB su priključene u svaki čvor.
Na slici 8 crvenom linijom je prikazan napon kada nemamo priključenih KB. Potom se podiže napon pomoću KB do te vrednosti (crvena isprekidana linija) pri kojoj je moguće spustiti poziciju teretnog menjača RT za jedan nivo (narandžasta puna linija). Minimalni dozvoljeni napon u svim čvorovima (plava linija) se dobija ponavljanjem ovog ciklusa nekoliko puta.

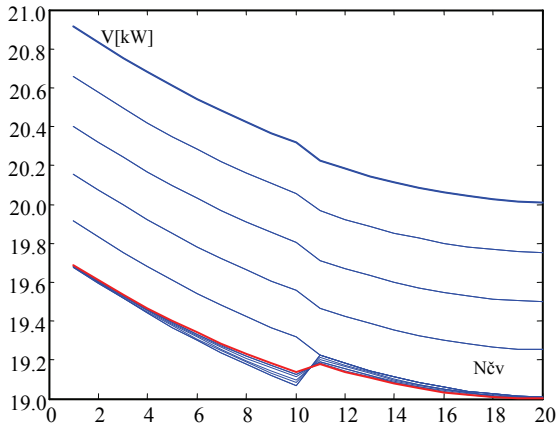


Slika 8. Profil napona prilikom redukcije napona do tehnički minimalne dozvoljene vrednosti

Ovom metodom može se uticati na napon do te mere da se on spusti na minimalni nivo u svim potrošačkim čvorovima. To će dovesti do minimalne potrošnje svih potrošača sa naponski zavisnom karakteristikom ali usled povećanja gubitaka neće rezultovati minimum injektirane aktivne snage u mrežu. Iz tog razloga minimum napona ne donosi optimalnu konfiguraciju mreže.

5.OPTIMALNA KONFIGURACIJA SISTEMA

Ako se uzmu u obzir sve navedene metode, softverom se može simulirati rad bilo koje mreže, sa nesimetričnim opterećenjima voda i različitim tipovima potrošača, i doći do optimalne konfiguracije. Optimalna konfiguracija obezbeđuje minimum injektirane aktivne snage u DM uz poštovanje kriterijuma.
Na slici 9 se vidi kako se menja napon na jednom od vodova tokom jedne od tih simulacija. Gornja linija predstavlja početnu vrednost napona. Crvena linija pokazuje napon pri kojem je aktivna snaga preuzeta iz prenosa minimalna.



Slika 9. Jedna simulacija pronalaženja optimalne konfiguracije

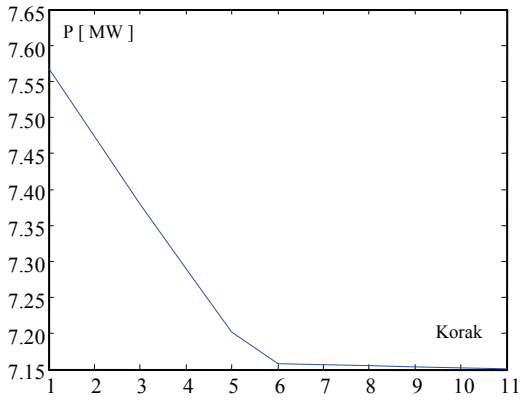
Koraci simulacije za ovaj vod su dati u tabeli 1. U zavisnosti od tipa potrošača u sistemu optimalna konfiguracija može podrazumevati nijednu, jednu ili dve KB na vodu,

uticati na njihovu poziciju i veličinu, položaj teretnog menjača napojnog RT, kao i na to da li će se na vod postaviti RT, a ako hoće kako će biti podešeni.

Tabela 1 Rezultati simulacije jednog voda

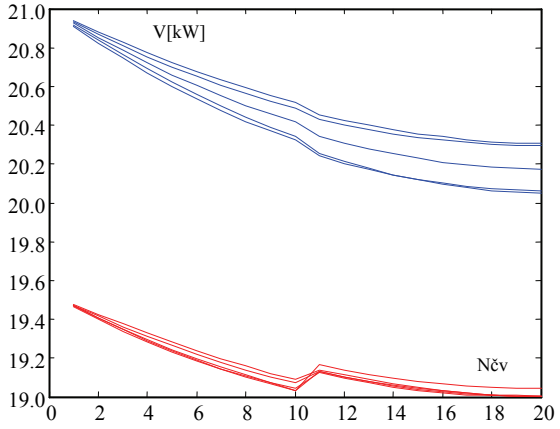
Korak	Napojni RT	RT	KB1 [čv.]	KB1 [mS]	KB2 [čv.]	KB2 [mS]
1	0	-	-	-	-	-
2	-1	-	-	-	-	-
3	-2	-	-	-	-	-
4	-3	-	-	-	-	-
5	-4	-	-	-	-	-
6	-5	5	-	-	-	-
7	-5	4	-	-	11	0,5
8	-5	3	-	-	11	1,0
9	-5	2	-	-	11	1,5
10	-5	2	2	0,5	20	1,5
11	-5	1	2	0,5	20	1,5

Na slici 10 se vidi kako prilikom ove simulacije svakim korakom opada injektiranje aktivne snage u razmatrani vod.



Slika 10 Injektirana aktivna snaga u vod tokom pretrage optimalne konfiguracije

Ako se radi sa DM sa pet vodova na kraju simulacija se dobijaju naponske slike svakog od njih. To se vidi na slici 11 gde su plavim linijama prikazani početni naponi a crvenim naponi posle optimizacije.



Slika 11. Profili napona pre i posle optimizacije

Ovaj primer simulacije je urađen na DM sa potrošačima konstantne struje. Oni su modelovani tako da im potrošnja linearno zavisi od napona. Iz tog razloga smanjenje injektiranja aktivne snage koje je dobijeno direktno zavisi od smanjenja napona. Optimalna konfiguracija sa ovakvim tipom potrošača donosi smanjenje potrošnje aktivne snage za približno 6% u odnosu na potrošnju pre optimizacije, što u ovom slučaju iznosi oko 1,89MW. Kod potrošača konstantne impedanse naponska slika posle optimizacije je slična kao kod potrošača konstantne struje. Oni su modelovani tako da im potrošnja kvadratno zavisi od napona. Zbog toga se smanjenjem napona kod njih dobija najveći pad preuzete aktivne snage. To smanjenje iznosi i do 11,7% u odnosu na injektiranu snagu pre optimizacije, odnosno nešto više od 3,59MW. Količina injektirane aktivne snage u DM ovde se može smanjiti samo smanjenjem gubitaka i zato iznosi do 0,07% ukupne snage pre optimizacije ili oko 22,5kW.

6.ZAKLJUČAK

Rast potražnje za električnom energijom iz godine u godinu postaje sve veći globalni problem. Povećanje proizvodnih i prenosnih kapaciteta predstavlja jedno od mogućih rešenja. Usled ekonomskih i ekoloških razloga to rešenje je sve više podređeno rešenjima koja se okreću ka efikasnijem iskorišćenju postojećih resursa.

U ovom radu je pokazano kako se statičkim uređajima kao što su regulacioni transformatori i kondenzatorke baterije može uticati na vrednost potrošnje aktivne snage i smanjenje gubitaka aktivne snage. Za razmatranu distributivnu test mrežu injektiranje aktivne snage smanjeno je čak i do 11,7% što je veoma značajan rezultat kada su u pitanju veličine reda MW.

7.LITERATURA

[1] V.Strezoski: *Osnovi elektroenergetike*, Stilos, Novi Sad, 2002.

[2] D.Popović, D.Bekut, V.Treskanica: *Specijalizovani DMS algoritmi,* Novi Sad, 2004.

[3] N.Dervišević: *Osetljivost vrednosti snage potrošača na promenu vrednosti napona*, diplomski – master rad, FTN Novi Sad , 2010.

[4] IMO: *Voltage Reduction Test July 2. 2004,* IMO_REP_0215, Issue 1.0, October 2004.

Kratka biografija



Bojan Cvetinov rođen je u Novom Sadu 1983. godine. Srednju elektrotehničku školu je završio u Bačkoj Palanci 2002. godine kad se i upisuje na Fakultet Tehničkih Nauka u Novom Sadu na smer Elektrotehnika i Računarstvo. Od 2010. godine stalno je zaposlen u Osnovnom sudu u Novom Sadu na održavanju IT sistema.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.3

JEDNO REŠENJE GRAFIČKE APLIKACIJE ZA PRIKAZ I ANALIZU ISTORIJATA SCADA SISTEMA

ONE SOLUTION OF GRAPHICAL APPLICATION FOR PLOTTING AND ANALYZING HISTORY DATA OF SCADA SYSTEM

Risto Obradović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Ovaj rad predstavlja opis realizacije jednog rešenja za prikaz i analizu istorijata SCADA sistema GAUS. Nakon opisa osnovnih elemenata SCADA sistema, predstavljeni su alati i tehnologije korištene za izradu samog rešenja. Rešenje je u potpunosti implementirano korištenjem .NET tehnolologija.*

Abstract – *This paper presents the description of development of one solution of graphical appcitation for plotting and analyzing history data of SCADA system GAUS. After the description of basic elements of SCADA system, the tools and tehnologies used to design the solution are presented. Solution is completely implemented using .NET technologies.*

Ključne reči: *SCADA, GAUS, WPF, CLR*

1. UVOD

Za razliku od industrijske ere gde su glavni resursi bili fizički i finansijski, današnje vreme poznato je kao informatička era, koja kao ključne resurse karakteriše znanje i informacije. Razvojem računarskih tehnologija, i blagovremenoj dostupnosti znanja i informacija, razvijaju se namenske računarske strukture koje povećavaju mogućnosti nadzora i upravljanja u industrijskim sistema.

Ovakvi sistemi, nazvani SCADA *(engl. Supervisory Control And Data Acquisition)* ili AUS *(Akviziciono upravljački sistemi)*, kontunalno mere stanje, vrše nadzor i upravljaju industrijskim postrojenjima. Zato je istorijat SCADA sistema najtačnija slika ponašanja postrojenja u dužem vremenskom periodu, i kao takav nezaobilazan u svakoj tehnološkoj analizi. Sistem GAUS (*Generalizovani AUS*), korišćen pri izradi rada, jedno je rešenje SCADA sistema opšte namene, razvijeno na Odseku za računarsku tehniku i računarske komunikacije.

Cilj opisane korisničke aplikacije je upravo olakšani pristup istorijskim podacima sistema GAUS, kao i njihov kasniji prikaz i analiza. Izabrane programske tehnologije obezbedile su pristup iz lokalnog ili iz WEB okruženja.

Pre opisa primenjenog rešenja i same realizacije, u radu su u nastavku date osnove akviziciono upravljačkih – SCADA sistema, kao i sažeti prikaz korišćenog razvojnog okruženja.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Branislav Atlagić, vanredni prof.

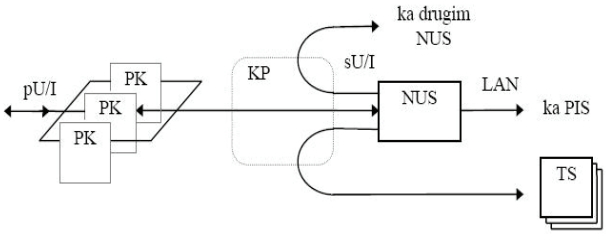
2. AKVIZICIONO UPRAVLJAČKI SISTEMI

Termin SCADA ili AUS, podrazumeva skup distribuiranih računarskih modula, koji zajedno upravljaju nekim tehnološkim postrojenjem. Svaki od njih ima svoju specifičnu namenu, i određeni stepen autonomije.

SCADA sistemi obuhvataju širok spektar opreme, podsistema i tehničkih rešenja koji omogućavaju prikupljanje i obradu podataka o procesima (udaljenim sistemima), i reagovanje na adekvatan način. Upravljanje procesima, u opšem slučaju, može biti automatsko ili inicirano od strane operatera.

Osnovna funkcija AUS sistema je ciklična akvizicija digitalizovanih odmeraka različitih fizičkih veličina koje određuju stanje proizvoljnog fizičkog procesa. Zahtevi koje AUS sistem mora da obezbedi su rad u realnom vremenu, distribucija računarskih resursa u okviru AUS-a i postizanje maksimalne pouzdanosti i raspoloživosti.

Na slici 1 prikazana je struktura AUS-a, koju čine dve osnovne komponente. To su procesni kontroler (PK) i nadzorno upravljačka stanica (NUS) koji su međusobno povezani komunikacionim podsistemom (KP).



Slika 1. *Konfiguracija AUS*

PK je mikroprocesorska stanica koja služi prvenstveno za obuhvat mernih signala sa senzora, kao i za generisanje pobude izvršnih elemenata. KP u okviru AUS-a čine komunikacioni kanali i komunikacioni kontroler stanice NUS, koja je centralni čvor AUS-a. Njen cilj je objedinjavanje i obrada svih procesnih podataka dobijenih sa ulaza.

Generalizovani akvizicioni upravljački sistem (**GAUS**) predstavlja sistem čiju arhitekturu definiše objekat upravljanja i upravljačka jedinica realizovana jednim ili skupom PC računara. Generalizacija u GAUS-u se odnosi na mogućnost konfigurisanja programskih komponenti u upravljačkom bloku, u cilju prilagođavanja objektu

upravljanja. Upravljanje objektima realizuje se adekvatnim izborom skupa upravljačkih funkcija.

GAUS obuhvata sve nivoe akviziciono upravljačkog sistema, od procesne sprege sa mernom i regulacionom opremom u postrojenju, do uklapanja u poslovni informacioni sistem posredstvom lokalne mreže. Suštinska razlika GAUS sistema u odnosu na AUS sistem je dosledna primena IT komponenti i programskih rešenja na svim nivoima akviziciono – upravljačkog sistema sto obezbeđuje jedinstvenost fizičke i programske platforme.

3. RAZVOJNO OKRUŽENJE

U ovom poglavlju dat je opis tehnologija korišćenih za realizaciju rada. U njemu je opisano *Microsoft .NET,* CLR i WPF tehnologija.

3.1 Microsoft .NET okruženje

.NET Framework okruženje je računarska platforma koju je projektovao Microsoft da bi pojednostavnio razvoj savremenih aplikacija kao sto su:

- Aplikacije koje koriste Internet;
- Aplikacije koje su distribuirane preko više računara;
- Aplikacije koje koriste baze podataka;
- Aplikacije koje korise GUI klijente.

Ovo okruženje dizajnirano je tako da se može koristiti iz bilo kog jezika: C#, C++, Visual Basic, Jscript, pa čak i iz starijih jezika kao što je COBOL. Da bi sve to funkcionisalo, pojavile su se i posebne verzije ovih jezika za .NET: Managed C++, Visual Basic .NET, Jscript .NET, kao i razni drugi koji imaju pristup .NET okruženju i mogućnost međusobne komunikacije. Sasvim je moguće prilikom programiranja u jeziku C# koristiti kod napisan u Visual C++ .NET-u, kao i obrnuto.

.NET okruženje se najvećom delom sastoji od ogromne biblioteke koda koja se upotrebljava preko klijentskih jezika koristeći tehnike objektno orijentisanog programiranja. Pored ove biblioteke najvažnija komponenta u .NET okruženju je zajednički jezički mehanizam izvršavanja ili skraćeno CLR (*engl. Common Language Runtime*) koji je odgovoran za izvršavanje svih aplikacija razvijenih uz pomoć .NET biblioteke.

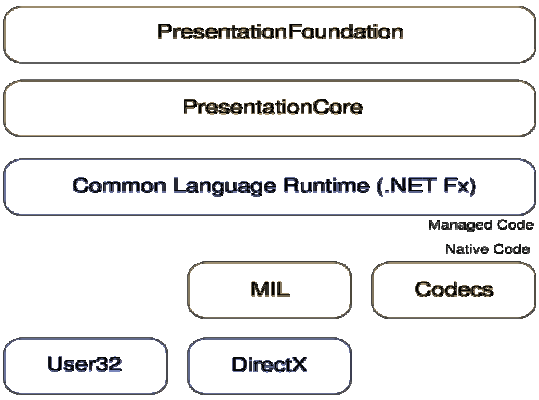
3.2 CLR (Common Language Runtime)

Zajednički izvršni jezik (CLR) obezbeđuje okruženje koje upravlja izvršavanjem koda i obezbeđuje usluge kao što su bezbednost, upravljanje memorijom i komunikacija između objekata. Kod koji CLR mehanizam izvršava naziva se kontrolisani kod, a kod koji se izvršava izvan kontrole CLR-a je nekontrolisani kod. Celokupan kod u jeziku C# je kontrolisan, ali je moguće napisati i kontrolisan i nekontrolisan kod u jeziku Microsoft Visual C++ i tako imati oba tipa koda koji zajedno rade u istom programu.

Svi .NET jezici se prevode u posredan oblik koji se naziva međujezik ili IL (*engl. Intermediate Language*). IL kod je sličan međukodu jezika Java po tome sto predstavlja posredan oblik koda koji proizvodi prevdilac i koji se direktno ne može izvršiti na ciljnom sistemu. Međutim, za razliku od međukoda jezika Java, IL kod se uvek konvertuje u mašinski kod pre nego što se izvrši, a to obavlja JIT (*engl. Just-In-Time*) prevodilac. Ova konverzija može da se dogodi na zahtev funkcija za vreme izvršavanja programa ili prilikom instaliranja programa.

3.3 WPF (Windows Presentation Foundation)

WPF je grafički podsistem za prikaz korisničke sprege u aplikacijama zasnovanim na operativnom sistemu Windows. Umesto oslanjanja na stariji GDI (*engl. Graphics Device Interface*) podsistem, WPF koristi DirectX tehnologiju koju čini skup API funkcija za efikasno manipulisanje fizičkom arhitekturom grafičkog kontrolera. Zadatak WPF-a je da obezbedi konzistentan model programiranja za izradu aplikacija i nudi jasnu razliku između korisničke sprege i poslovne logike.



Slika 2. *Arhitektura WFP-a*

Na slici 2 prikazana je arhitektura WPF-a, gde su plavom bojom označene Windows komponente a drugom bojom WPF komponente. Arhitektura WPF-a se može podeliti i u dve grupe sa aspekta koda, pri čemu jednu grupu čine komponente pod kontolisanim (*engl. Managed*) kodom dok drugu grupu obrazuju pod upravom lokalnog (*engl. Native*) koda.

Najzančajnija osobina WPF-a je XAML (*engl. Extensible Application Markup Language*), izveden iz XML-a, i predstavlja alternativni način za definisanje grafičkih komponenti i njihovog odnosa sa ostalim komponentama.

4.TEHNIČKI OPIS PROBLEMA

U ovom poglavlju dat je kratak opis podsistema istorijata sistema GAUS, formata zapisa istorijskih podataka u binarne datoteke, kao i način na koji ih aplikacija koristi.

Zapisivanje istorijata *GAUS* sistema je relativno jednostavna, ali u eksploataciji izuzetno važna funkcija. Cilj istorijata je obezbeđenje pouzdanih podataka o stanju procesnog sistema, kao i o funkcionisanju samog akviziciono-upravljačkog sistema. Istorijat se zapisuje na magnetni medijum SCADA servera podataka i obuhvata periodičan zapis skupa tehnološki značajnih procesnih

promenljivih, u formatu pogodnom za pristup i izvan sistema AUS. Period zapisa se određuje u skladu sa dinamikom fizičkog procesa. Zapisom podataka u binarne datoteke dobijamo istorijat AUS sistema, koji je osnovno polazište za brojne aplikacije čiji je cilj analiza rada u proteklom periodu, kontrola tehnoloških parametara, održavanje i sl.

Zapis podatka u binarne datoteke je opisan strukturom prikazanom na slici 3.

```
typedef struct
{
   BYTE rectype;
   union
   {
      struct
      {
         long timestamp; /* hh:mm */
         short no_data;
         short scan_c;
      } vreme;
      struct
      {
         BYTE ime[NAME_LEN];
         float vrednost;
         BYTE tip;
         DWORD Status;
      } data;
   } slog;
} STAT_SLOG;
typedef struct
```

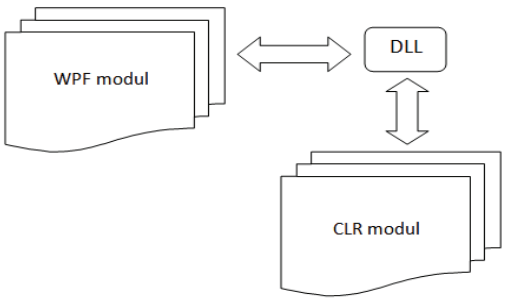
Slika 3. Struktura binarne datoteke

U okviru strukture *STAT_SLOG* nalazi se unija dve strukture:

- *vreme* (vremenski trenutak sat:minut) u kome su validni podaci *data* za svaku učitanu procesnu veličinu
- *data* sadrži sve relevantne informacije o zapisanoj procesnoj veličini (*ime, vrednost, status* – stanje-kvalitet, i *tip*).

5. OPIS REALIZACIJE

Celokupan rad na projektovanju i izradi ove aplikacije je realizovan u *MS Visual Studio 2008* razvojnom okruženju. Tehnologija na kojoj je zasnovano rešenje je kombinacija WPF i CLR tehnologija koje se nalaze unutar korištene verzije .NET Framework-a 3.5.
Takođe, i programski jezik na kome je napisana ova aplikacija je kombinacija C# i CLI/C++ programskog jezika.

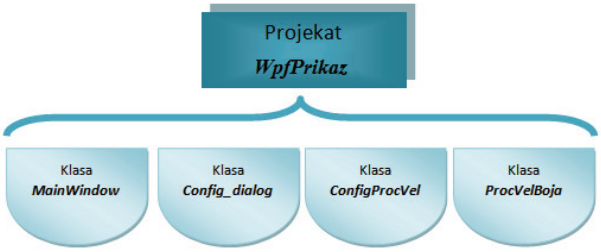


Slika 4. *Struktura sistema*

Iz slike 4 vidimo da se aplikacija za prikaz i analizu sastoji od WPF i CLR modula, koji međusobnu komunikaciju ostvaruju preko DLL (*engl. Dynamic-link library*) komponente. Programerski gledano, ovi moduli predstavljaju dva posebna projekta povezana u zajedničku celinu gde CLR modul generiše DLL komponentu, a WPF modul koristi tu komponentu.

5.1 WPF modul

WPF modul čini projekat sastavljen od više klasa napisanih u jeziku C#, koje su međusobno spregnute, i služe za obradu i prikaz rezultata na samoj aplikaciji. U ovom modulu, koga reprezentuje projekat pod imenom ***WpfPrikaz***, najznčajnije su četiri klase koje su prikazane na slici 5.



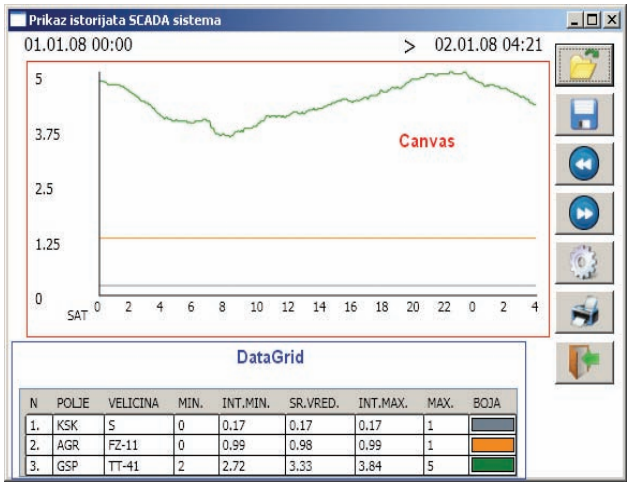
Slika 5. *WPF modul*

Na slici 6 prikazan je glavni prozor na kome su iscrtane procesne veličine i njihovi parametri. Osnovne kontrole na glavnom prozoru su ***Canvas***, ***DataGrid*** i ***Button-i*** koje se nalaze uz desnu ivicu prozora.

Canvas je osnovna kontrola glavnog prozora na koju se vrši iscrtavanj
e grafika i ispis vrednosti po x i y osi. Sva iscrtavanja i ispise na ovu kontrolu obavlja funkcija *Draw()* koja je definisana u klasi ***MainWindow***.

DataGrid kontrola predstavlja tabelu u koju su smeštene osnovne inf
ormacije o prikazanim pocesnim veličinama. Ažuriranje podataka u DataGrid-u obavlja ObservabileCollection lista *dataGrid_list.*

Button kontrole ozačavaju funkcijske tastere pomoću kojih se vrši upravljanje celom aplikacijom.



Slika 6. *Izgled glavnog prozora*

Klasa ***Config_dialog*** grafički predstavlja dijalog iz koga je moguće izvršiti obradu nad procesnim veličinama koje su prikazane na glavnom prozoru. Klikom miša na taster "Konfiguracija" u glavnom prozoru kreira se objekat klase ***Config_dialog,*** na osnovu čega se pomoću funkcije *ShowDialog* prikazuje dijalog za konfiguraciju grafikona, koji je dat na slici 7.
Funkcionalnosti ovog prozora podeljene su u četiri grupe.

Prvu grupu čine dve *RadioButton* kontrole čija je namena da sa odgovarajućim vrednostima označe y-osu glavnog prozora.

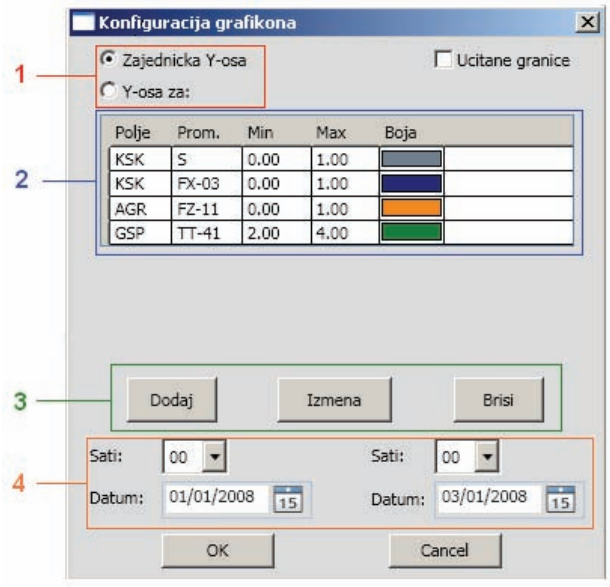
Drugu grupu čini *DataGrid* kontrola koja služi za prikaz imena polja, promenljive, njihovih minimalnih i maksimalnih vrednosti u odabranom intervalu i boje svake procesne veličine.

Treća grupa se sastoji od tastera Dodaj, Izmeni i Brisi. Ovi tasteri vrše obradu nad globalnom CLR strukturom *SWEEP* u koju su smešteni podaci o procesnim veličinama.

Četvrtu grupu kontrola čine dva *ComboBox*-a za izbor vremena i dva *DatePicker*-a za izbor datuma.

Klasa ***ConfigProcVel*** opisuje dijalog koji se aktivira pritiskom na taster Dodaj i Izmena.

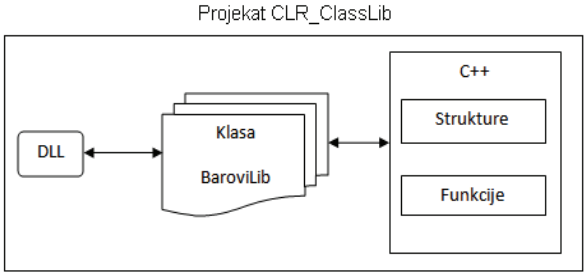
Posto u ovoj verziji WPF-a ne postoji ugrađena komponenta za izbor boje, potrebno je napraviti ručno kontrolu sa paletom boja. To je realizovano pomoću klase ***ProcVelBoja.***



Slika 7. *Konfiguracija grafika*

5.2 CLR modul

CLR modul je deo aplikacije u kome se izvršava određen deo funkcionalnosti, kao što su učitavanja binarnih i konfiguracionih datoteka i smeštanje podataka u odgovarajuće strukture. Projekat koji čini ovaj modul zove se CLR_ClassLib i prikazan je na slici 8.
Glavni delovi ovog projekta su klasa BaroviLib i određene funkcije i strukture. Klasa BaroviLib je realizovana u paru datoteka *CLR_ClassLib.h* i *CLR_ClassLib.cpp.* Ove datoteke imaju isti naziv kao projekat.



Slika 8. *CLR modul*

Sve funkcije i strukture koje su realizovane u C++ programskom jeziku vidljive su samo unutar klase BaroviLib. Unuar ove klase se vrši obrada podataka posredstvom CLI koda, nakon čega se vrši njihovo pakovanje u DLL datotaku koja ima naziv *CLR_ClassLib.dll.* Sve strukture definisane u CLR klasi će prevodilac transformisati u "metadata" datoteku odakle će struktura biti vidljiva WPF okruženju.

6. ZAKLJUČAK

Ovim radom je realizovano jedno rešenje grafičke aplikacije za prikaz i analizu istorijata SCADA sistema. Cilj ovog rada je bio da se objedine nove tehnologije sa starim funkcionalnostima, što je uspešno urađeno. Pojavom novih tehnologija razvijaju se poboljšana rešenja, na osnovu kojih se nove aplikacije brže i jednostavnije razvijaju.

Razvojem DLL komponente u .NET okruženju je omogućeno da se funkcionalnost aplikacije, odnosno CLR komponenta, izvršava na svim .NET jezicima. Na osnovu ovoga, ovaj rad je moguće realizovati na bilo kojoj .NET tehnoligiji korišćenjem gotove CLR komponente.

7. LITERATURA

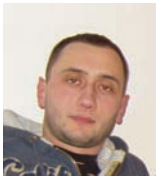
[1] B.Atlagić, GAUS – Generalizovani akviziciono-upravljački sistem, Novi Sad, 2001.

[2] J.Templeman; A.Olsen, Visual C++ .NET, Microsoft edition, 2005.

[3] S.Lippman, C# izvornik, Don box series editor, 2003.

[4] Adam Nathan, WPF, Sams, 2007.

Kratka biografija:



Risto Obradović rođen je u Trebinju 09.03.1985. god. Apsolvent je na Fakultetu tehničkih nauka, smer računarstvo i automatika, odsek računarska tehnika i računarske komunikacije.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.316.925

ANALIZA RADA DISTANTNE ZAŠTITE U REALNOJ DISTRIBUTIVNOJ MREŽI

DISTANCE PROTECTION WORK ANALYSIS IN REAL DISTRIBUTION NETWORK

Saša Stojanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Cilj rada je da se na primeru realne distributivne mreže ispita uticaj distributivnih generatora na funkcionalnost relejne zaštite. Testiranje je vršeno pomoću ditributivnog menadžment sitem (DMS) paketa – modulom za relejnu zaštitu koji je razvijen na Fakultetu tehničkih nauka. Rezultati testiranja dati su na podlozi realne mreže Hydro One (Kanada).*

Abstract – *The aim of this paper is to be an example of real distribution network examine the effect of distributed generators on the functionality relay protection. Testing was performed by distribution management system (DMS) package - module for relay protection which was developed on Faculty of technical sciences. The results of testing are given on example of real Hydro One network (Canada).*

Ključne reči: *Relejna zaštita, Distributivni generatori*

1. UVOD

U ovom radu se razmatra uticaj distributivnih generatora na funkcionalnost relejne zaštite na podlozi realne distributivne mreže. U pitanju je Hydro One distributivna mreža koja se nalazi u Kanadi. Hydro One je veoma posvećena povezivanju obnovljive energije na distributivni sistem uz očuvanje sigurnosti i pouzdanosti sistema. Za ovu mrežu je karakterističan veki broj distributivnih generatora. Ovaj rad se takođe fokusira na veoma važan aspekt vezan za relejnu zaštitu, a to je podešenje i koordinacija u slučaju prisustva distributivnih generatora.

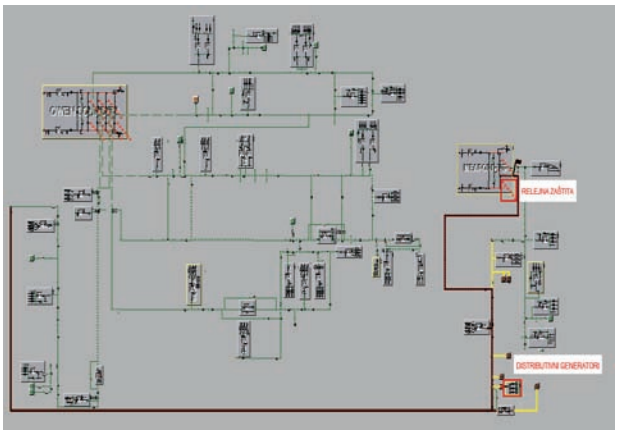
U leto 2009. godine, kada su prvi relativno veliki distributivni generatori bili povezani na Hydro One distributivni sistem, Hydro One je postao svestan efekata i problema koje donosi priključivanje distributivnih generatora u distributivnu mrežu. Analize su pokazale da tradicionalna ograničenja vezana za priključenje distributivnih generatora kao što su kapacitet, maksimalno opterećenje, nisu bili dovoljna da spreče narušavanje performansi sistema. Ovo je posebno bilo izraženo kada su veliki distributivni generatori priključivani na relativno duge distributivne izvode. Hydro One je tada nastavio da postavlja nove standarde vezane za priključenje distributivnih generatora, zaštite distributivnog sistema...

2. PROVERA FUNKCIONALNOSTI RELEJNE ZAŠTITE U DMS SOFTVERU

Test mreža koja se koristi za analizu funkcionalnosti relejne zaštite je prikazana na slici 1.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je prof. dr Duško Bekut.

Za verifikaciju rezultata izabran je deo Hydro One distributivne mreže. Test mreža sadrži srednjemaponski 44 kV izvod koji polazi iz VN/SN transformatorske stanice Meaford i koji je označen na slici 1. Izvod je obeležen braon bojom dok su laterali (otočni delovi) označeni žutom bojom. Zaštitni releji nalaze se na početku izvoda i uokvireni su crvenim kvadratom.



Slika 1. - Test mreža

Na izvodu se nalaze tri distributivna generatora koja se nalaze u postrojenju Eugenia, takođe uokvirena crvenom bojom.

Posmatraju se slučajevi sa i bez priključenja distributivnih generatora za različite tipove kvara. Podešenja distantne zaštite (oznaka 21) u transformatorskoj stanici Meaford su sledeća:

- 21 (Zona 1) = 1.54 [Ω], RCA=90°, left blinder=1.54, right blinder=1.35, DCA=30°, kvadriteralna karakteristika, t=0 [s];
- 21 (Zona 2) = 12.35 [Ω], RCA=70°, Mho karakteristika, t=0.05 [s];
- 21 (Zona 3) = 25.74 [Ω], RCA=80°, Mho karakteristika, t=3 [s];

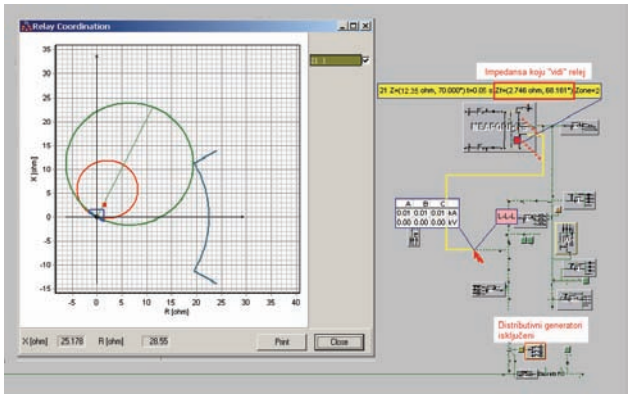
2.1. Uticaj distributivnih generatora na funkcionalnost distantne zaštite za slučaj kvara na izvodu između zaštite i distributivnih generatora

Za potrebe ove analize, posmatra se fazna distantna zaštita koja se nalazi na početku izvoda. Ova zaštita ima tri zone rada. Odgovarajuća vrednost podesenja za svaku zonu je data na početku ove glave. Analiza rada distantne zaštite vrši se za dva tipa kvara (metalni kvar, kvar sa lukom) na izvodu, sa i bez priključenja distributivnih generatora.

Rezultati analize

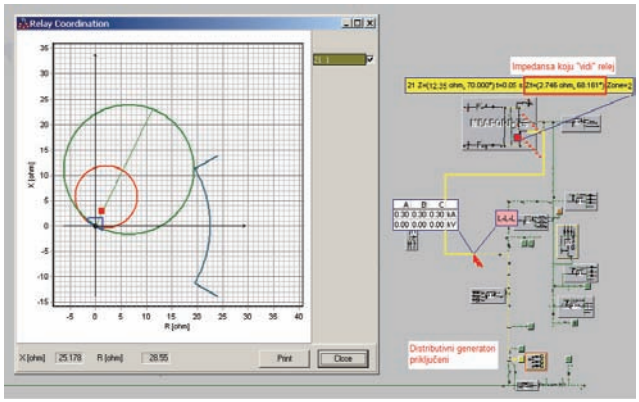
U prvom slučaju, simuliran je metalni kvar na izvodu bez priključenja distributivnih generatora. Pošto distributivni

generatori nisu priključeni, mesto kvara se napaja samo iz transformatorske stanice. Na slici 2 predstavljena je kvadriteralna i dve Mho karakteristike fazne distantne zaštite kao i linija voda sa radnom tačkom izmerenom za ovaj slučaj kvara.



Slika 2. – Radna karakteristika distantnog releja sa radnom tačkom za metalni kvar na izvodu bez priključenja distributivnih generatora

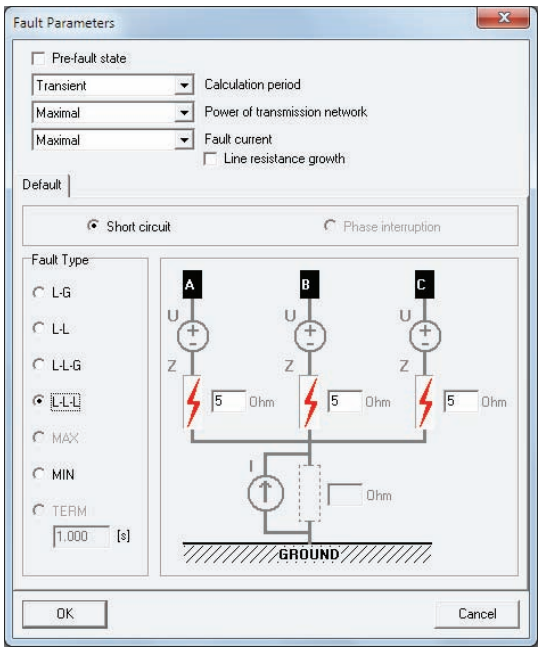
Crvenim kvadratom u okviru prvog kruga označena je radna tačka koja pokazuje merenu vrednost impedance u slučaju razmatranog kvara. Veličina izmerene impedanse je 2.746 [Ω], a izmereni ugao je 68.161°. Kvar se nalazi u zoni 2 distantnog releja. Ukoliko je kvar metalni, merena impedansa (radna tačka) nalazi se na liniji voda. Uopšteno govoreći, linija voda nije prava linija ako se odnos R/X menja duž izvoda.
Neophodno je istaći da ne postoji uticaj distributivnih generatora na merenu impedansu za slučaj metalnog kvara što dokazuje i sledeća slika.



Slika 3. – Radna karakteristika distantnog relaja sa radnom tačkom za metalni kvar na izvodu sa priključenim distributivnim generatorima

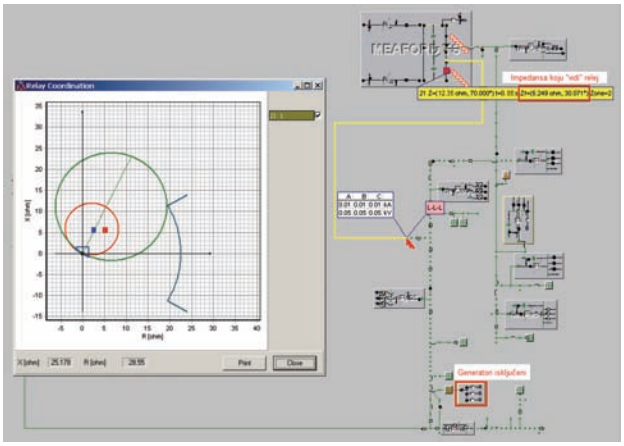
U ovom slučaju impedansa koja se meri relejom je jednaka impedansi kvara kao u slučaju bez distributivnih generatora na izvodu. Dakle, zaštita se ponaša isto u oba slučaja kvara sa i bez generatora na vodu. Jedina razlika koja se javlja u ovom slučaju je vrednost struje na mestu kvara koja je sada veća jer se mesto kvara istovremeno napaja iz transformatorske stanice i distributivnih generatora.

Međutim, postojanje luka na mestu kvara značajno menja uticaj distributivnih generatora na rad distantne zaštite. U navedenom slučaju smatra se da je otpornost luka 5 [Ω], što se može videti i na sledećoj slici gde je za parametre otpora na mestu kvara zadata ta vrednost.



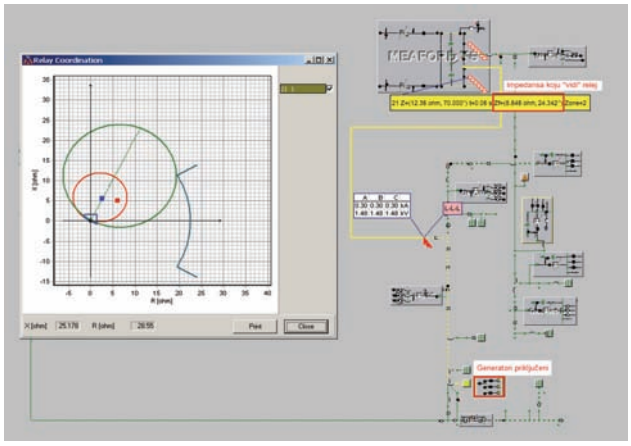
Slika 4. – Parametri kvara sa lukom

Bez prisustva distributivnih generatora na fideru, u slučaju kvara sa lukom, dolazi do pojave napona na mestu kvara zbog postojanja otpornosti i rezultantna struja je u fazi sa merenom vrednošću na lokaciji distantnog releja. To govori da je impedansa luka čisto termogena. Dakle, radna tačka će biti premeštena iz linije voda, paralelno sa R osom, kao što je prikazano na slici 5. (plavi kvadrat označava metalni kvarm a crveni kvar sa lukom). Impedansa koja se meri relejom je 6.249 [Ω] i ugao je 30.071 stepeni. Kvar se i dalje nalazi u zoni 2 distantnog releja.



Slika 5. – Radna karakteristika distantnog releja sa radnom tačkom za kvar sa lukom na fideru bez priključenja distributivnih generatora

Kada su distributivni generatori priključeni na izvod, struja na mestu kvara se razlikuje od struje koju se meri relejom i radna tačka će biti pomerena paralelno sa R osom. Fazni pomeraj struje kvara sa strane distributivnih generatora se razlikuje od struje koja se meri distantnim relejom. Ovaj efekat dovodi do pojave dodatnih impedansi koje pomeraju radnu tačku na R-X ravni i ukupna merena impedansa distantnog releja može biti manja ili veća u odnosu na slučaju bez prisustva distributivnih generatora.
Za razmatrani slučaj kvara sa lukom impedansa koja se meri relejom je 8.545 [Ω], dok je njen ugao 24.342°. Ovaj slučaj prikazan je na slici 6.
Iz ovih primera može se zaključiti da je uticaj distributivnih generatora u slučaju kvara sa lukom dosta izražen i to može predstavljati problem u funkcionisanju distantne zaštite. Takođe, potrebno je posvetiti posebnu pažnju prilikom određivanja podešenja ovih zaštita.



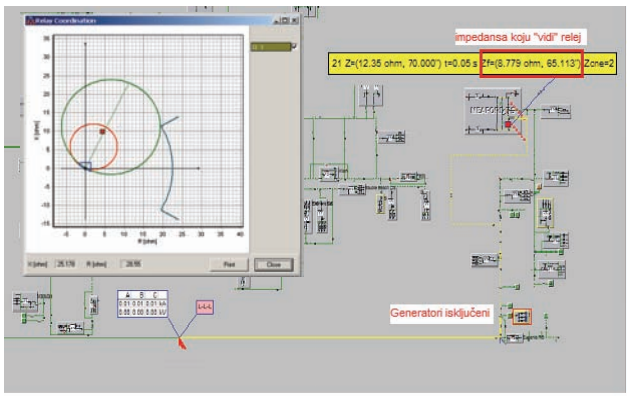
Slika 6. – Radna karakteristika distantnog releja sa radnom tačkom za kvar sa lukom na izvodu sa priključenim distributivnim generatorima

2.2. Uticaj distributivnih generatora na funkcionalnost distantne zaštite za kvar na fideru iza distributivnih generatora

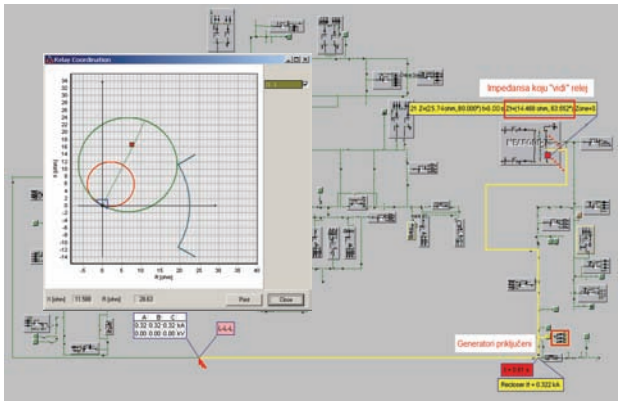
Efekat distributivnih generatora za kvar simuliran na fideru iza distributivnih generatora je veoma velik. Za potrebe ove analize, kvar je simuliran na kraju izvoda, iza mesta gde su priključeni distributivni generatori što je prikazano na sledećim slikama.

Rezultati analize
U slučaju metalnog kvara, kada distributivni generatori nisu priključeni na izvod, impedansa koja se meri distantnim relejom ima vrednost 8.779 [Ω], dok je ugao 65.113°. Kvar se nalazi u zoni 2 distantnog releja. Ovaj slučaj prikazan je na slici 7.

Međutim, kada su distributivni generatori priključeni na izvod impedansa koja se meri relejom ima dosta veću vrednost. To dovodi do toga da će se kvar sada nalaziti u zoni 3 distantnog releja.
Impedansa koja se meri relejom u ovom slučaju ima vrednost 14.468 [Ω], a ugao impedanse je 63.552°.
Može se oceniti da impedansa koju meri distantni relej na početku izvoda za kvar iza distributivnih generatora na kraju izvoda je veća od one koja se ima kada distributivni generatori nisu povezani na izvod.



Slika 7. – Slučaj metalnog kvara iza tačke priključenja distributivnih generatora pri kraju izvoda kada su distributivni generatori isključeni



Slika 8. – Slučaj metalnog kvara iza tačke priključenja distributivnih generatora pri kraju izvoda kada su distributivni generatori priključeni

Ako je u pitanju metalni kvar, napon koji meri relej biće:

$$V_R = I_{tr} \cdot Z_{S1} + (I_{tr} + I_{g1}) \cdot Z_{S2}\,, \qquad (1)$$

ili:

$$\frac{V_R}{I_{tr}} = Z_{S1} + (1 + \frac{I_{g1}}{I_{tr}}) \cdot Z_{S2}\,, \qquad (2)$$

gde je:

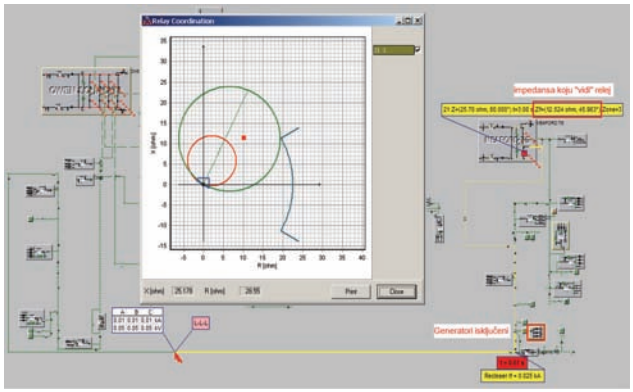
V_R – mereni napon na lokaciji releja,

Z_{S1} – impedansa prvog dela fidera (do tačke priključenja distributivnih generatora),

Z_{S2} – impedansa drugog dela fidera (od tačke priključenja distributivnih generatora).

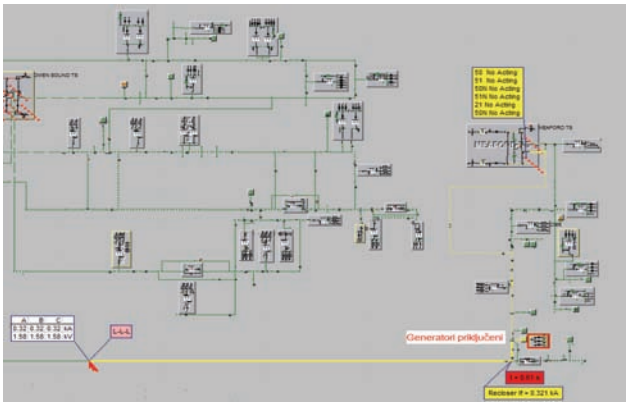
Dakle, relejom se mere dodatne impedanse, izračunate u izrazu (2), umesto prostog zbira impedansi Zs_1+Zs_2. Imajući u vidu da je numerička vrednost u zagradi u izrazu (2) viša od one u (1), krajni efekat je značajno povećanje merene impedanse kao što je ilustrovano na slici 8 za kvar na posmatranom izvodu.
U slučaju kvara sa lukom iza tačke priključenja distributivnih generatora, kada generatori nisu priključeni, na mestu kvara će se pojaviti napon zbog postojanja otpornosti na mestu kvara. Impedansa koja se meri distantnim relejom u ovom slučaju ima vrednost od

12.524 [Ω], a izmereni ugao je 45.936°. Kvar se nalazi u zoni 3. Ovaj slučaj prikazan je na sledećoj slici.



Slika 9. – Slučaj kvara sa lukom iza tačke priključenja distributivnih generatora pri kraju fidera kada su distributivni generatori isključeni

U slučaju kvara sa lukom kada su distributivni generatori priključeni struja koja se meri relejom nije ista kao i struja na mestu kvara. Uticaj distributivnih generatora je u ovom slučaju toliko velik da je ugrožena funkcionalnost distantne zaštite, tj. radna tačka će se nalaziti izvan zone štićenja releja.
Dakle, u slučaju kvara sa lukom pri kraju izvoda, kada su distributivni generatori priključeni, distantna zaštita ne reaguje, što je i prikazano na slici 10.



Slika 10. - Slučaj kvara sa lukom iza tačke priključenja distributivnih generatora pri kraju izvoda kada su distributivni generatori priključeni

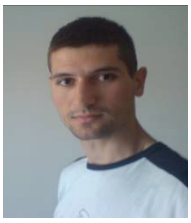
3. ZAKLJUČAK

U radu je vrlo detalno ukazano na problem uticaja distributivnih generatora na rad zaštite. Zbog efekta dodatnog napajanja koje nastaje zbog priključenja pomenutih generatora značajno se uložnjava rad distantne zaštite. Kroz primere je ukazano u kojim slučajevima treba obratiti posebnu pažnju na rad zaštite i koji su slučajevi od posebnog interesa za razmatranje.
Radom je takođe ukazano da je praktično nemoguće zaobići detaljnu analizu pogona mreže sa distributinim generatorima vezano za aspekte zaštite.

4. LITERATURA

[1] A. Sarić, M. Ćalović: ,,*Planiranje EES I deo: principi i metodologija planiranja EES*“, Beograd, 2000.
[2] Duško Bekut: ,,*Relejna zaštita*“, FTN Izdavaštvo, Novi Sad 1999.
[3] ,,*Distribution System Protection*“, Univerzitet Western Ontario, 2008.
[4] J.M.Gers, E.J.Holmes: *,,Protection of electricity distribution networks“*, IEE, London, 1998.
[5] DMS Group: ,,*DG Solution Proof of Concept Report Part II*“, Hidro One ADS Program, Jun 2010.
[6] DMS Group: ,,*Protection and Control Functional Requirements Standard for DG Enablement: Fider Protection (Phase2)*“, Hydro One Networks Inc, April 2009.
[7] DMS Group: ,,*Distributed Generation Interconnection Protection Principles & Information Exchange*“, Hydro One Networks Inc, Jul 2010.
[8] DMS Group: ,,*Case Study Relay Protection Real Network*“, Maj 2010.

Kratka biografija:



Saša Stojanović rođen je u Novom Sadu 1985. godine. Diplomski–master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetike odbranio je 2011. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

JEDNO REŠENJE METODOLOGIJE ISPITIVANJA SLOŽENIH SISTEMA PROFESIONALNE ELEKTRONIKE

ONE SOLUTION OF TESTING METHODOLOGY FOR COMPLEX PROFESSIONAL ELECTRONIC SYSTEMS

Nemanja Paunović, Jelena Kovačević, Ivan Rešetar, *Fakultet tehničkih nauka, Novi Sad*

Kratak sadržaj – U ovom radu biće prikazan jedan od pristupa ispitivanju složenih sistema profesionalne elektronike zasnovan na automatizaciji i na primeni metode crne kutije, a podeljen u 4 faze ispitivanja. Cilj je da se proces ispitivanja u potpunosti automatizuje, da bude što pouzdaniji, a vreme potrebno za izvršavanje testova i uticaj ljudskog faktora svedu na minimum. Ispitivanje se vrši tokom proizvodnje, sa akcentom na ispitivanju funkcionalnosti uređaja. Sve provere moraju biti izvršene pre nego što uređaj izađe na tržiše.

Ključne reči: *metoda crne kutije, automatizacija, ispitivanje*

1. UVOD

U današnje vreme razvoj složenih sistema profesionalne elektronike je u stalnoj ekspanziji. Konkurencija među proizvođačima složenih elektronskih uređaja je ogromna. Potreba svakog proizvođača je da svoj proizvod što brže plasira na tržište i ponudi ga kupcima, ali s druge strane ta brzina ne sme da utiče na kvalitet i ispravnost proizvoda. Savremeni uređaji poseduju veliki broj funkcionalnosti, počev od različitih sprežnih podsistema, do kombinovanja više uređaja u samo jednom uređaju. Ispitivanje i testiranje ovako složenih sistema postaje ozbiljan problem. Prilikom razvoja uređaja koji spadaju u domen složenih sistema profesionalne elektronike više od 40% vremena potroši se na verifikacije i provere [1]. Usled težnje proizvođača da obezbede konkurentnost na tržištu, razvoj sistema u ovoj oblasti je u velikoj meri uslovljen cenom sa jedne strane i opravdanom potrebom za skraćenjem vremena pojavljivanja proizvoda na tržištu, sa druge. Zajedno, ova dva faktora teže smanjenju ukupnog vremena razvoja sistema. Skraćenje vremena razvoja se u praksi najčešće postiže skraćivanjem procesa ispitivanja. U uslovima sve izraženijeg širenja funkcionalnosti sistema, smanjenje troškova, tj. cene na ovaj način dovodi do kratkoročnih rezultata, nedovoljno ispitan proizvod vodi ka degradaciji kvaliteta, što negativno utiče na konkurentost. Da bi se obezbedilo kraće vreme proizvodnje i pružio bolji kvalitet, potrebno je više pažnje posvetiti analizi kvaliteta sistema i unaprediti proces ispitivanja, istovremeno održavajući/poboljšavajući njegov kvalitet. [2]

NAPOMENA:

a) Ovaj rad proistekao je iz diplomskog-master rada Nemanje Paunovića. Mentor je bila dr Jelena Kovačević.

b) Rad je prethodno publikovan na konferenciji ETRAN, Banja Vrućica, juni 2011.

Razlikujemo dva tipa ispitivanja:

- Ispitivanje kao završna faza tokom projektovanja uređaja i izrade prototipova (QTP – qualification test procedure), ovim ispitivanjem se potvrđuje ispravnost dizajna i programske podrške uređaja.
- Ispitivanje uređaja u procesu proizvodnje (PTP - production test plan)

Rad je fokusiran na PTP ispitivanje. Predložena je metoda ispitivanja složenih sistema profesionalne elektronike podeljena u 4 faze koje su deo PTP ispitivanja. Akcenat je stavljen na drugu fazu koja je zasnovana na metodi crne kutije (Black Box Testing) [3]. Faze ispitivanja ciljnog uređaja (DUT – device under test):

- ispitivanje fizičke platforme
- ispitivanje funkcionalnosti
- ispitivanje izdržljivosti
- ispitivanje u uslovima rada van predviđenih uslova

Faze se izvršavaju sekvencijalno uz uslov da je za prelazak na sledeću fazu potrebno da izveštaj prethodne faze ispitivanja prosledi pozitivne rezultate. Prednost ovakvog pristupa ispitivanju je u tome što sekvencijalni pristup ispitivanju uz dobar plan ispitivanja omogućava struktuisano pronalaženje i otklanjanje grešaka i daje precizne rezultate.

Ispitivanja uređaja u vezi sa sertifikacijama za pojedina tržišta, uskladjenosti sa zahtevima i standardima, deo su QTP testiranja, nisu u fokusu ovog rada.

Predložena metoda je praktično primenjena prilikom ispitivanja uređaja za prikupljanje audio i video uzoraka u realnom vremenu pod nazivom RT-AV100.

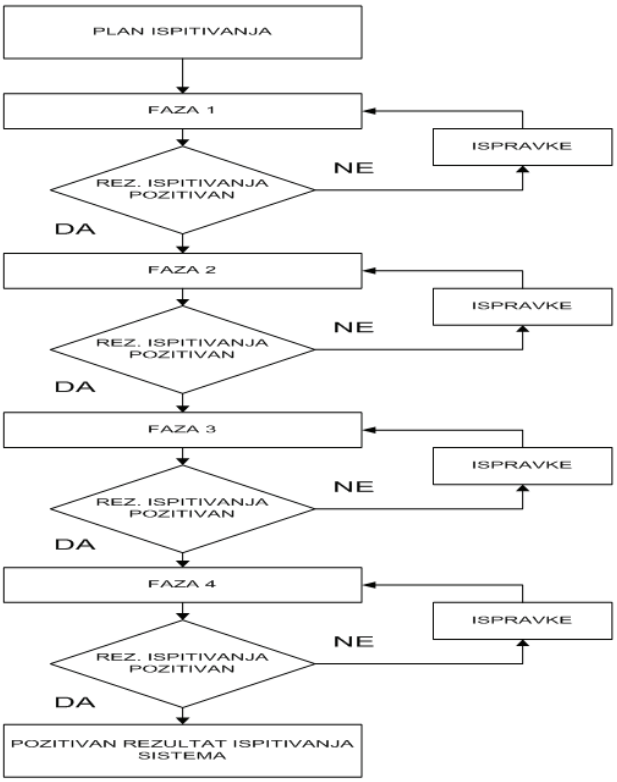
U nastavku rada biće opisana svaka od 4 faze uz detaljan opis faze dva i primer kako je predložena metodologija primenjena pri ispitivanju uređaja RT-AV100.

2. PREDLOG METODOLOGIJE ISPITIVANJA

Na slici 1. prikazan blok dijagram toka ispitivanja predložene metode.

U okviru projektovanja uređaja potrebno je definisati detaljan ispitni plan. Plan ispitivanja sadrži zahteve koji se postavljaju na osnovu specifikacije DUT-a, spisak potrebne opreme za formiranje ispitnog okruženja, opis svih predviđenih ispitivanja i očekivane rezultate ispitivanja. Kao ulaz u svaku od faza koristimo ispitni plan, kao izlaz iz svake od faza dobijamo izveštaj o tome koja ispitivanja su završena pozitivnim rezultatom, a koja moramo ponoviti ili prijaviti grešku. Za slučaj da neki od

zahteva nije prošao ispitivanje važno je koristiti sistem za kontrolisanje grešaka (eng. BUG tracking system) preko kojeg test inženjer prijavljuje grešku koju je pronašao. U zavisnosti da li se izvrsava QTP ili PTP izveštaj o grešci se adresira na tim za istraživanje i razvoj (Research & Development) (za QTP) ili na tim iz proizvodnje (tehničari za PTP). Kada tim na koji je adresirana greška ispravi grešku prijavljuje da je nepravilnost otklonjena i tada test inženjer ponavlja fazu na kojoj je obustavio ispitivanje kada je ustanovio grešku.



Slika 1. *Blok dijagram toka ispitivanja predložene metode*

2.2. Ispitivanje fiziške platforme

Prva faza je ispitivanje ispravnosti fizičke platforme uređaja. Ispituju se štampane ploče, veze na njima i ugrađene komponente u cilju otkrivanja, verifikacije i ispravke eventualnih grešaka nastalih prilikom proizvodnje istih.

Kod proizvodnje štampane ploče ispituju se postojanje električnih veza (očekivana vrednost otpora 0 oma u tačkama istog potencijala) i beskonačna otpornost u različitim čvorovima (bez montiranih elektronskih komponenata). Druga merenja koja se vrše mogu biti: debljina slojeva metalizacije, gabariti i poprečni presek ploče, karakteristična impedansa vodova ako je zahtevana za pojedine interfejse i drugo.

Sledeća faza je asembliranje (lemljenje) i tada se koristi optičko ispitivanje, i x zraci, da se proveri da li su prave komponente na pravom mestu, i da li su pravilno orijentisane, i da li ima loših lemova, kratkih spojeva itd.

U savremenim uslovima proizvodnja štampanih ploča obicno izmeštena iz kuća koje se bave razvojem, a samo fizičko ispitivanje ploča se naručuje kao usluga, a ploče koje su isporučene su ispitane.

Prilikom preliminarne kontrole funkcionalnosti sklopova koristi se sprege, Dongle i JTAG.

Jedan od postupaka ispitivanja za koji je oprema relativno jeftina i koji se može obaviti u laboratorijama bez nekih posebnih zahteva je ispitivanje sa ivice. Ispitivanje sa ivice (Boundary scan) je predmet standarda IEEE 1149.1 [5] **,** poznato je i pod nazivom Joint Test Action Group (JTAG). Ovaj standard se danas masovno koristi za ispitivanje rada integrisanih kola. Pomoću ove sprege moguće je pristupiti svim digitalnim podsistemima u uređaju i ona predstavlja osnovni način ispitivanja ugnježdenih računarskih sistema. Potrebna je programska podrška na računaru za uspostavu komunikacije sa predmetom ispitivanja, potrebna je programska podrška koja će biti učitana u DUT i potreban je uređaj kojim će računar i DUT biti povezani (eng. DONGLE). Učitana programska podrška obezbeđuje samodijagnostiku funkcionalnosti sistema i po završetku simulacije na računaru dobija izveštaj o tome koje funkcionalne celine su prošle ispitivanje, a koje nisu.

2.3. Ispitivanje funkcionalnosti

Druga faza je provera funkcionalnosti uređaja. Posle pozitivnog izveštaja o rezultatima ispitivanja fizičke platforme prelazi se na proveru funkcionalnosti uređaja. Kompleksnost ove faze zavisi od konstrukcije uređaja. Najkomplikovaniji su slučajevi u kojima se proverava uređaj koji je sklop više podsistema u jednom, poseduje više različitih sprežnih podsistema i različite načine povezivanja.

Ispitivanje funkcionalnosti se definiše kao aktivnost koja se sprovodi radi ocene kvaliteta programske podrške i njenog unapređenja, identifikacijom grešaka i problema. [6].

Ispitni plan u ovoj fazi sadrži zahteve koji se sastavljaju na osnovu specifikacije uređaja, opis ispitnog okruženja koje će simulirtati realno radno okruženja DUT-a, detaljno uputstvo toka ispitivanja koje obuhvata sve funkcije koje DUT podržava i očekivane rezultate ispitivanja. U zavisnosti od namene uređaja definiše se dozvoljena tolerancija na odstupanja rezultata ispitivanja.

Nakon analize sličnih rešenja na tržištu kao rešenje za programsku podršku ispitivanju odabran je RT-Executror, program razvijen u istraživačko razvojnom institutu „RT-RK“, koji podržava ispitivanje po principu „crne kutije“. Program RT-Executor daje podršku za korišćenje velikog broja uređaja koji imaju različite namene, npr. uređaj za obradu audio uzoraka, za rad sa datotekama/direktorijumima, za kontrolu spoljašnih uređaja, raznih generatora audio i video signala itd. [7]. RT-Executor aplikacija je u potpunosti prilagodljiva okruženju. RT-Executor omogućava jednostavnu implementaciju novih programskih uređaja. Metoda crne kutije podrazumeva ispitivanje funkcionalnosti programske podrške i potrebno je znati samo šta prosleđujemo na ulaz sistema i šta očekujemo na njegovom izlazu. Na osnovu specifikacije DUT-a napišu se testovi na osnovu kojih će se ispitati da li sve predviđene operacije uređaj ispravno izvršava. Testovi se pišu ili u BBT_SCRIPT jeziku (namenski skript jezik) ili

u jeziku „Pyhton“. Testovi trebaju biti potpuno automatski jer svako uključivanje test inženjera povećava šanse za eventualnu grešku i produžava vreme ispitivanja. Ono što je prvo potrebno uraditi je postaviti referentne datoteke koji će se koristiti za poređenje sa testnim datotekama. Kada su ispisani testovi koji pokrivaju kompletnu funkcionalnost uređaja i kada se proveri da su ispravno napisani od njih se pravi projekat koji se pokreće u RT-Executor-u. Testovi se automatski izvršavaju i nakon njihovog završetka može se videti koji od testova su pozitivni, koji nisu, a koji su nerazjašnjeni i tada se po potrebi testovi koji nisu pozitivni ponavljaju. Da bi dobili rezultate ispitivanja potrebno je postaviti opseg tolerancije u okviru kojeg se smatra da je ispitivana funkcionalnost zadovoljila postavljene zahteve. RT-Executor prikazuje pregledan izveštaj o rezultatima testova koji se može prebaciti u html datoteku radi slanja, čuvanja ili radi vođenja evidencije. Na slici 2. dat je primer izgleda jednog test izveštaja u html formatu.

Testing Results

No	Date & Time	Test case	Result
1	CVBS_mode_tv_0_83_480_NTSC	PASS	16.02.2011,11:51:45
2	HDMI1_11_83_576p50	FAIL	16.02.2011,11:52:51
3	S-VIDEO_mode_tv_0_83_480_NTSC	PASS	16.02.2011,11:53:27
4	CVBS_mode_tv_1_83_576_PAL	FAIL	16.02.2011,11:54:01
5	CVBS_mode_tv_0_89_480_NTSC	PASS	16.02.2011,11:54:30
6	S-VIDEO_mode_tv_0_89_480_NTSC	PASS	16.02.2011,11:55:07
7	HDMI2_14_83_1080i50	FAIL	16.02.2011,11:57:34
8	HDMI2_162_89_1080p30	INCONCLUSIVE	16.02.2011,11:57:45

Tested cases:	8	
Passed:	4	50%
Failed:	3	38%
Inconclusive:	1	13%

Slika 2. *Primer test izveštaja*

Za testove koji su “FAIL“ ili “INCONCLUSIVE“ potrebno je proveriti ispravnost testova i ponoviti ih. Da bi proglasili da je uređaj završio ispitivanje uspešno, svi testovi moraju biti “PASS“. U slučaju da posle provere testova i njihovog ponavljanja rezultati ponovo nisu pozitivni, test inženjer prijavljuje grešku.

2.4. Ispitivanje izdržljivosti

Ispitivanje izdržljivosti je treća faza koja za cilj ima da potvrdi stabilnost sistema kada ambijentalni i drugi uslovi odstupaju od nominalnih, a u opsegu koji je definisan specifikacijom uređaja. Podrazumeva ispitivanje izvan normalnog režima rada često i do namerno izazvanog otkaza rada uređaja u interesu sagledavanja rezultata [8].

Ispitnim planom definišu se okruženje u kojima uređaj treba da se ispituje, ambijentalni uslovi, trajanje ispitivanja u zavisnosti od namene uređaja i očekivani rezultati ispitivanja.

RT-Executor može biti upotrebljen prilikom ove faze ispitivanja na taj način što postoji mogućnost da se napisani testovi izvršavaju automatski neprekidno veliki broj sati. Prednost je što smanjuje angažovanost ljudskih (skupih i nepreciznih) resursa.

Ova ispitivanja se zapravo urade u toku QTP, a u proizvodnoj fazi se vrši uzorkovanje i pojedini kritični testvi se ponavljaju na propisanom uzorku iz svake serije.

2.5. Ispitivanje u uslovima rada van predviđenih uslova

Završna faza ispitivanja uređaja je ispitivanje uslovima rada van predviđenih uslova, prilikom nepravilnog korišćenja uređaja, kao i posledice takvih situacija. Ispitivanje u uslovima rada van predviđenih uslova definisano je kao stepen do kojeg sistem ili komponenta mogu pravilno funkcionisati u ekstremnim uslovima rada. [9].

Sadržaj ovih ispitivanja se utvrđuje na osnovu izvršene analize rizika i posledica rada u okruženju van predviđenih uslova (FMEA - failure mode and effects analisys).

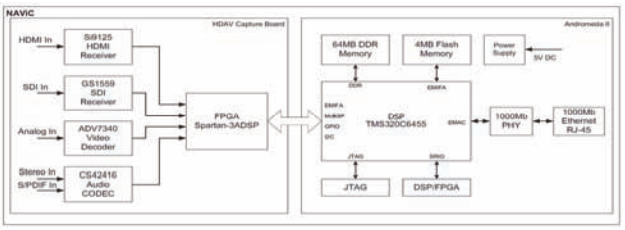
Ispitnim planom definišu se okruženje u kojima uređaj treba da se ispituje, ambijentalni uslovi, trajanje ispitivanja u zavisnosti od namene uređaja i očekivani rezultati ispitivanja.

Zavisno od namene uređaja predviđaju se najekstremnije i neočekivane situacije u kojima se DUT može naći npr. pad uređaja, prolivanje tečnosti, naglo rashlađivanje ili grejanje i sl.

Ova ispitivanja se zapravo urade u toku QTP, a u proizvodnoj fazi se vrši uzorkovanje i pojedini kritični testvi se ponavljaju na propisanom uzorku iz svake serije.

3. ISPITIVANJE UREĐAJA RT-AV100

Predložena metodologija je primenjena kod ispitivanja uređaja RT-AV100. RT-AV100 koristi se u TV industriji za ispitivanje STB (set top box – uređaj koji se koristi kao prijemnik signala digitalne televizije) uređaja. Uređaj je zasnovan na digitalnom signal procesoru i poseduje mnogobrojne audio i video sprežne podsisteme (HDMI (High-Definition Multimedia Interface), YpbPr, S-video (Separate Video), S/Pdif (Sony/Philips Digital Interconnect Format) optical, S/Pdif coaxial, CVBS (Composite Video, Blanking, and Sync)). Uređaj ima mogućnost povezivanja na 1Gb mrežu putem LAN (Local Area Network) priključka koji poseduje [4].



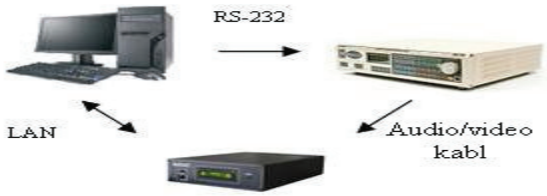
Slika 3. *Blok sema RT-AV100 uređaja*

Na slici 3. je prikazana blok šema na kojoj se mogu videti najvažniji delovi uređaja. RT-AV100 spada u komleksne sisteme profesionalne elektronike, poseduje na jednoj ploči računarski sistem koji se sastoji od digitalnog signal procesora (DSP), operativne memorije (RAM), memorije za skladištenje (FLASH memory) i mrežnog priključka, dok je na drugoj ploči glavni element programabilno integrisano kolo FPGA (Field-programmable Gate Array).

Ispitivanje je podrazumevalo je zasebno ispitivanje 2 štampane ploče. Ova ispitivanja obavio je dobavljač po narudžbini firme. Ispitivanje sa ivice za proveru ispravnosti funkcionisanja komponenti obavljena su u laboratorijama firme. Zbog realizacije uređaja pomoću digitalnog signal procesora proizvođača *Texas Instruments* kao programska podrška na računaru

korišteno je programsko okruženje *Code Composer Studio* takođe proizvod ove firme, dok je programska podrška korištena za učitavanje na DUT napisana u programskom jeziku C. Računar je povezan sa DUT-om USB (Universal Serial Bus) JTAG sprežnim priključkom. Rezultati ispitivanja na računaru prikazani su u formi liste komponenti sa naznakom "PASS" za komponentu koja je pozitivno završila ispitivanje i "FAIL" za onu koja nije.

Prilikom ispitivanja funkcionalnosti RT-AV100 zahtev je bio ispitivanje funkcionisanja svih audio (HDMI, S/Pdif, LineIN) i video (HDMI,YpbPr, S-Video, CVBS) sprežnih podsistema. Ispitno okruženje sastoji se od generatora video i audio signala koji podržava sve rezolucije slike i formate audio uzoraka i poseduje sve sprežne podsisteme koje poseduje i uređaj. Generator povežemo sa uređajem audio i video kablovima, dok uređaj i računar međusobno komuniciraju preko LAN-a, okruženje ja zatvoreno kada se računar i generator povežu putem RS-232 kabla.



Slika 4. *Ispitno okruženje*

Na slici 4. je prikazan izgled ispitnog okruženja. Primer provere HDMI video srežnog podsistema: test inženjer sastavlja testove koji će se izvršavati preko RT-Executor-a. Pokrenuti test daje signal video generatoru da prosledi sliku određene rezolucije preko HDMI kabla DUT-u, DUT dobija komandu takođe preko RT-Executor-a da prikupi sliku, poslednji zadatak RT-Executor-a je da prikupljenu sliku uporedi sa prethodno postavljenom referentnom slikom na računaru uz toleranciju od 10 piksela (koji ne smeju biti grupisani u jednom bloku slike) koliko prikupljena slika sme da odstupa od referentne. RT-Executor upoređuje testne uzorke sa referentnim i daje rezultat poređenja u formatu izveštaja. Trajanje jednog testa je oko 30 sekundi. Ista procedura ponavlja se za sve ostale sprežne podsisteme.

Pri ispitivanju izdržljivosti RT-AV100 uređaja zahtevi su bili da u laboratorijskim uslovima uređaj ponavlja cikluse prikupljanja audio i video uzoraka neprekidno tokom perioda od 12 časova. Rezultat ispitivanja izdržljivosti je bio pozitivan jer je uređaj i posle 12 časova neprekidnog rada pravilno funkcionisao tj. rezultati u izveštaju RT-executor-a su bili pozitivni (PASS).

Zbog konstrukcije i namene RT-AV100 uređaja nije bilo zahteva za ispitivanja uređaja u uslovima van predviđenih uslova rada. Uređaj je na predlog test inženjera ispitan tako što mu je dok je bio u procesu rada prosleđen audio umesto video signala i obrnuto, konekcija LAN kablom je fizički prekinuta u toku rada, zatim ponovo uspostavljena nakon čega je DUT nastavio ispravno da funkcioniše.

Ispitivanje jednog RT-AV100 uređaja trajalo je približno 20 sati i izvršeno je preko 2000 testova u RT-Executor-u. Pri ispitivanju neki od problema bili su da uređaj za neke formate nije prikupljao sliku koja je identična referentnoj ili da uređaj nije mogao da funkcioniše neprekidno više sati, ali svaki od uočenih problema je bio odmah prijavljen i efikasno rešen.

4. ZAKLJUČAK

U ovom radu prikazana je jedna od metodologija ispitivanja sistema profesionalne elektronike podeljena u 4 nezavisne faze koje obuhvataju ispitivanje uređaja u PTP fazi. Ovakav pristup ispitivanju je vrlo pouzdan jer pozitivan rezultat ispitivanja garantuje potpunu ispravnost i pouzdanost uređaja. Metoda se bazira na što većoj automatizaciji procesa ispitivanja što doprinosi bržem ispitivanju i smanjenju broja grešaka. Glavna prednost prikazane metodologije je precizna putanja otkrivanja i otklanjanja grešaka.

Praktična primena ove metode prilikom ispitivanja uređaja RT-AV100 pokazala se kao vrlo pouzdana. Za 1 sat ispitivanja RT-Executor-om moguće je izvršiti i do 120 testova što garantuje temeljnu proveru. Provera izdržljivosti uređaja oduzima najviše vremena,tako da se ukupno vreme ispitivanja može skratiti smanjivanjem vremena predviđenog za ovu fazu.

Mana prikazane metodologije je što sekvencijalna organizacija testiranja za posledicu ima da se u procesu proizvodnje sekvencijalno premešta i opterećenje sa resursa ne resurs (kako ljudski tako i tehnički), sto onemogućava njihovo optimalno iskorišćavanje.

Zahvalnica:

Ovaj rad je delimično finansiran od Ministarstva prosvete i nauke Republike Srbije, projekat 32014, od 2011. godine.

5. LITERATURA

[1] B.korel, „Automated software test data generation", IEEE *Transaction on Software Engineering,* vol.16, no.8, pp. 870-879, Aug '90.

[2] D.Marijan, "Razvoj metodologije testiranja softvera u multimedijalnim sistemima", 2011

[3] D.Marijan, N.Teslić, M.Temerinac, V.Peković, „On the Effectiveness of the System Validation Based on the Black Box Testing Methodology", *Journal of Electronic Science and Technology of China,* vol. 7, no. 4, 2009.

[4] "RT-AV100" http://www.rt-rk.com/audiovideo-capture-devices/navic [pristupljeno: mart 2011.]

[5] IEEE 1149.1-1990 (Standard Test Access Port and Boundary-Scan Architecture)

[6] A.Abran, J.Moore, P.Borque, R.Dulpis, L.Tripp, "Guide to the software engineering body of knowledge ", IEEE, 2004

[7] Peković Vukota, Teslić Nikola, Rešetar Ivan, Tekcan Tarkan, „Test Management and Test Execution System for Automated Verification of Digital Television Systems"

[8] Nelson, Wayne B., (2004), Accelerated Testing - Statistical Models, Test Plans, and Data Analysis, John Wiley & Sons, New York

[9] A. Kövi, Z. Micskei: Robustness Testing of Standard Specifications - based HA Middleware (IEEE Std 610.12.1990)

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.311.064.1(075.8)

CYME APLIKACIJA ZA PRORAČUN KRATKIH SPOJEVA – OSNOVNE KARAKTERISTIKE

CYME APPLICATION FOR CALCULATION OF SHORT CIRCUIT – BASIC PROPERTIES

Siniša Božić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Za potrebe dimenzionisanja električne instalacije i opreme, kao i za potrebe određivanja sredstava neophodnih za zaštitu ljudi i materijalnih sredstava, neophodno je proračunati struje kratkog spoja za svaku tačku u mreži. Osnovni zadatak ovog rada je bio da se detaljnije prouče mogućnosti jednog od standardnih programa CYME i da se na jednom primeru potvrdi metoda koja se u ovom programu koristi.*

Abstract – *For the purpose of dimensioning of electrical installations and equipment, as well as for determining the resources necessary to protect people and material resources, it is necessary to calculate fault currents for each point in the network. The main purpose of this paper is to get more detailed insight in capability one of software applications for short circuit calculation CYME. In addition it was idea to confirm applied methodology on one example.*

Ključne reči: Kratki spojevi, CYME

Key words: Short circuit analysis, CYME

1. UVOD

Za proračun struja kratkih spojeva u sistemu, razvijene su metode koje mogu da se koriste i pre same izgradnje sistema, a kojima se pomoću matematičkog modela sistema vrši proračun struja kvara. Upotreba ovih metoda bez upotrebe računara je ograničena na sisteme sa malim brojem elemenata, jer zahtevaju veliki broj složenih proračuna, što sa porastom sistema koji se posmatra postaje nepraktično. Upotrebom računara i specijalizovanih softverskih rešenja kojima se modeluju elementi razmatranih sistema i proračunava struja kvara, postupak se ubrzava i proširuje njegovo područje primene.

Jedan od programa koji može da se koriste u tu svrhu je CYMDIST5.0 kompanije CYME International. Korišćena je free trial verzija za upoznavanje sa ovim programom. Ovaj softverski paket sadrži simulacione alate koji se koriste za rešavanje širokog spektra problema sa kojima se susreću elektroinženjeri u prenosu, distribuciji i industrijskim mrežama.

U ovom radu opisan je simulacioni alat koji se koristi za proračun struja kratkih spojeva, dok su razmatranja vezana za ostale mogućnosti ovog programskog paketa izostavljena.

Ciljevi upotrebe programa za proračun kratkih spojeva mogu se podeliti u sledeće kategorije:

- Dizajniranje i izbor opreme za prekidanje strujnih krugova (prekidača, sklopki, itd.).
- Određivanje zaštitnih uređaja u sistemu (osigurača, releja, itd.).
- Određivanje efekata struja kvara na različite elemente sistema kao što su kablovi, vodovi, sabirnice, transformatori, itd.
- Procena efekata uticaja različitih struja kvara na napon u celokupnom sistemu.

2. NAČINI PRORAČUNA STRUJA KVARA

Ovaj program podržava proračun struja kvara prema sledeće tri metode:

1 – Konvencinalna analiza kratkih spojeva

Konvencionalna analiza se ne ograničava strogo na poštovanje nekog standarda i ne vrši podešavanje reaktanse motora, ali dozvoljava korišćenje tranzijentne impedanse generatora da bi se proračunala vrednost struje nakon nekoliko perioda od trenutka nastanka kvara.

2 – ANSI analiza kratkih spojeva

Ovaj metod prati standarde za primenu prekidača C37.010 (simetrična struja), C37.5 (ukupna struja) i C37.13 (niskonaponski prekidači), Američkog Nacionalnog Instituta za Standarde.

3 – IEC analiza kratkih spojeva

Ova metoda prati smernice evropskog standarda IEC 60909 za analizu kratkih spojeva primenljivog na trofazne naizmenične sisteme koji rade i na 50 Hz i na 60 Hz.

Na ovaj način se proračunavaju početna, vršna i vrednost prekidanja struja kvara u mrežama kako radijalne tako i upetljane konfiguracije.

3. OSNOVNE KARAKTERISTIKE MODELA ZA PRORAČUN

CYME u svom sastavu ima bazu podataka opreme. Baza podataka opreme sadrži skup ugrađenih modela opreme koja se koristi u distributivnim mrežama.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio prof. dr Duško Bekut.

3.1. Određivanje impedanse izvora

Izvor (odnosno njegov ekvivalent) je početna tačka mreže. On predstavlja impedansu proizvodnje i prenosa, a definisan je sledećim podacima:

- Nominalna proizvodnja u MVA
- Napon
- Fazni ugao
- Ekvivalent izvora
- Sprega

3.2. Određivanje impedanse transformatora

Transformatori mogu biti dvonamotajni ili tronamotajni, kao i regulacioni ili sa fiksnim prenosnim odnosom.

3.2.1. Trofazni dvonamotajni transformator

Parametri potrebni za proračun impedanse dvonamotajnih transformatora su:

- Nominalno opterećenje
- Primarni napon. Linijski napon.
- Sekundarni napon. Linijski napon.
- Gubici praznog hoda
- Impedanse u domenu simetričnih komponenti (pozitivnog i nultog redosleda)
- Impedansa uzemljenja
- Sprega

3.2.2. Trofazni tronamotajni transformator

Parametri potrebni za proračun impedanse tronamotajnih transformatora su:

- Nominalno opterećenje
- Nominalni naponi. Linijske vrednosti napona za primarni, sekundarni i tercijarni namotaj
- R_g, X_g. Impedanse uzemljenja, rezistivni i reaktivni deo
- Sprega
- Z_1. Vrednost impedanse pozitivnog redosleda u %.
- Z_0. Vrednost impedanse nultog redosleda u %.
- X_0/R_0, X_1/R_1. Odnosi reaktansi i rezistansi.

3.2.3. Regulacioni transformator

Parametri potrebni za proračun impedansi regulacionih transformatora su:

- Tip. Jednofazni ili trofazni.
- Nominalni napon
- Gornja granica regulacionog opsega
- Donja granica regulacionog opsega
- Broj položaja regulacione sklopke
- Sprega
- Regulacioni opseg

Kod regulacionih transformatora postoji mogućnost zadavanja set point-a, ali to za razmatranje proračuna struja kratkih spojeva nije od interesa.

3.3. Određivanje impedanse generatora

Generatori mogu biti sinhroni ili asinhroni.

3.3.1. Sinhroni generator

Napomena: Dozvoljeni su samo trofazni sinhroni generatori.

Parametri koje korisnik podešava i koji utiču na impedansu kratkog spoja sinhronog generatora su:

- Nominalni napon
- Aktivna snaga
- Faktor snage
- Sprega
- Z (R, X). Ustaljena impedansa kratkog spoja.
- Z' (R' X'). Tranzijentna impedansa kratkog spoja.
- Z'' (R'', X''). Subtranzijentna impedansa kratkog spoja.
- Z_0 (R_0, X_0). Nulta impedansa kratkog spoja.
- Z_g (R_g, X_g). Impedansa uzemljenja.

Opcija za procenu parametara: Klikom na ovu opciju otvara se prozor u kome se mogu proceniti vrednosti subtranzijentne (Z''), tranzijentne (Z') i impedanse nultog redosleda (Z_0) kao i odnosa X/R.

3.3.2. Asinhroni generator

Kod asinhronih generatora, koristi se samo subtranzijentna impedansa kratkog spoja, kada je u opcijama analize ona uključena.

Subtranzijentna impedansa Z'' (reaktansa X'' i rezistansa R'') asinhronog generatora može biti uneta od strane korisnika, ukoliko je poznata ili proračunata od strane programa na osnovu ekvivalentne šeme generatora. Vrednosti impedanse mogu se uneti u relativnim jedinicama na naponskom nivou generatora ili u omima.

3.4. Određivanje impedanse motora

Motori mogu biti asinhroni ili sinhroni.

3.4.1. Asinhroni motor

Parametri koji se koriste pri proračunu impedanse asinhronog motora su sledeći:

- Nominalna snaga
- Nominalni napon
- ANSI grupa 2,3,4 ili 5.
- R'', X''. Subtranzijentna rezistansa i reaktansa.

Vrednosti subtranzijentne impedanse mogu biti i procenjene na osnovu NEMA (National Electrical Manufacturers Association) koda i drugih podataka o motoru. Slovo NEMA koda označava odnos polazne struje motora i nominalne struje pri maksimalnom opterećenju.

3.4.2. Sinhroni motor

Parametri koji se koriste pri proračunu imppedanse su:

- Nominalna snaga
- Nominalni napon
- Z'', Z0, Xd zasićenja. Subtranzijentna impedansa (Z''), impedansa nultog redosleda (Z0) i unutrašnja impedansa zasićenja (Xd)
- Zg. Impedansa uzemljenja, uvijek se daje u omima.

Opcija za procenu parametara: Klikom na ovu opciju otvara se prozor u kome se mogu proceniti vrednosti subtranzijentne (Z'') i impedanse nultog redosleda (Z_0)

3.5. Određivanje impedanse nadzemnih vodova

Za uravnotežene vodove, potrebno je svakom vodu definisati samo impedanse (Z1 i Z0), susceptanse (B1 i B0), i uslove (leto i zima). Za vodove koji nisu

uravnoteženi potrebno je dati impedanse i susceptanse po fazi i fizički raspored provodnika.
Ukoliko su poznate, ove vrednosti korisnik unosi direktno, a ukoliko su nepoznate CYME može da ih proračuna na osnovu tipa provodnika i prostornog rasporeda na stubu.

3.6. Određivanje impedanse kablova

U CYME-u je moguće podešavati parametre, impedansu i susceptansu tri tipa kablova: višežilni kružni sa nultim provodnikom, sa plaštom ili bez plašta.
Zajedničke osobine za sve tipove, koje definiše korisnik su:

- Z1. Impedansa pozitivnog redosleda (u Ω/km ili Ω/stopi).
- Z0. Impedansa nultog redosleda (u Ω/km ili Ω/stopi).
- Susceptansa. μS/km ili μS/stopi.
- Uslovi. Ljetnje i zimske vrednosti struje opterećenja.
- Nivo opterećenja. Pored proračuna struja kratkih spojeva CYME ima i proveru opterećenja koja se za zadati tip kratkog spoja radi na sledeći način:
  - 3 - fazni kratak spoj: $I_{MaxRadno} - K_{max} \cdot V_{LN}/Z_1$
  - 3 - fazni kratak spoj sa zemljom: $I_{MaxRadno} - K_{max} \cdot V_{LN}/(Z_1+Z_f)$
  - 2 - fazni kratak spoj: $I_{MaxRadno} - K_{max} \cdot V_{LL}/(2 \cdot Z_1+Z_f)$
  - 2 - fazni kratak spoj sa zemljom: $I_{MaxRadno} - K_{max} \cdot V_{LL} \cdot Y$ gde je:
    $$Y = \frac{a^2 \cdot (z_1 + z_0) - a \cdot z_0 - z_1}{z_1 \cdot (z_1 + z_0) + z_1 \cdot z_0}; \quad a = e^{j2\pi/3}$$
  - 1 - fazni kratak spoj: $I_{MaxRadno} - K_{max} \cdot (3 \cdot V_{LN})/(2 \cdot Z_1+Z_0+3 \cdot Z_f)$

Opcija za procenu parametara: Prikazuje se prozor za proračun impedanse kabla gde se definiše tip kabla na osnovu parametara koji se unesu u ovom prozoru.

Parametri provodnika koji su isti i za nadzemne vodove i za kablove su:

- Spoljni prečnik. Ukupni prečnik provodnika.
- Geometrijski srednji poluprečnik GMR. (Geometric Mean Radius) Može obično da se nađe među karakteristikama provodnika. Definisan je kao N^2-ni koren proizvoda N^2 rastojanja između N pod-provodnika (žica) provodnika, ukoliko su žice identične.

$$GMR = N^2 \cdot \sqrt{\prod_{k=1}^{N} \prod_{k=1}^{N} D_{km}} \qquad (1)$$

Gde je $D_{mm} = e^{-1/4} \cdot r = 0{,}7788 \cdot r$, za okrugle žice

- R 25°C i R 50°C. R 25°C i R 50°C su dve različite vrednosti otpornosti na proizvoljnim temperaturama (najčešće na 25°C i 50°C).
- Nominalno opterećenje.

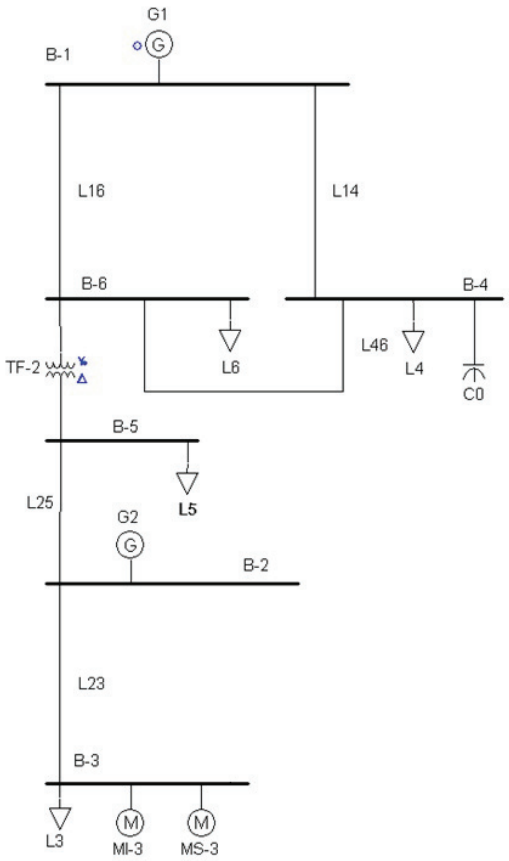
3.7. Redne induktivnosti

Redne induktivnosti služe da ograniče stuju kvara pri kratkom spoju. Vrednosti koje mogu da se podešavaju pri njihovom postavljanju u kolo su:

- Nominalna struja,
- Reaktansa – unosi se u omima.

4. PRIMER

Kao ilustracija postupka odrađen je proračun struja kvara na test mreži. Mreža u svom sastavu ima 6 sistema sabirnica (označene sa B-1 – B-6; eng. Bus), generatore G1 i G2 priključene na sabirnice B-1 i B-2 respektivno, asinhroni motor MI-3 snage 20000 KS (konjskih snaga) i sinhroni motor MS-3 snage 10250 KS. Oba motora su priključena na sabirnice B-3 na koje je još priključeno i opterećenje L3 (eng. Load). Preostala dva opterećenja L5 i L6 priključeni su na sabirnice B-5 i B-6 respektivno. Mreža ima dva naponska nivoa a vezu između njih obezbeđuje transformator TF2. Izgled mreže prikazan je na slici 4.1.



Slika 4.1 – *Izgled test mreže za ilustraciju rada CYME-a*

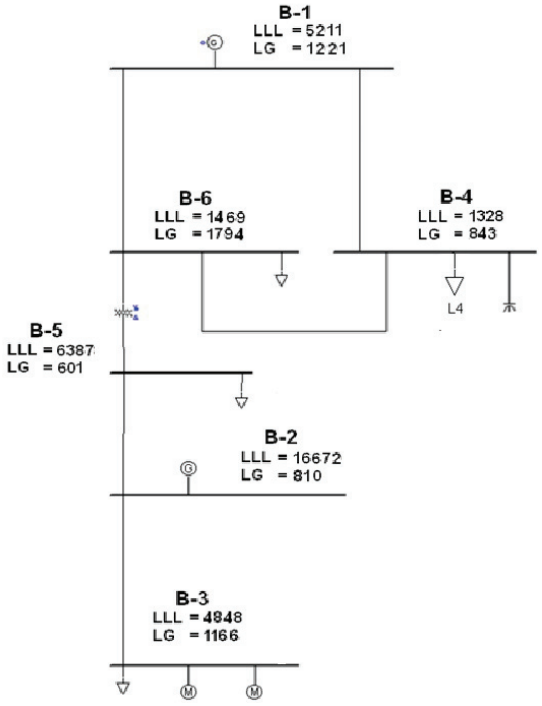
U svrhu ilustracije postupka podešeni su sledeći parametri:

a) Zahtevan je proračun struja na svim sabirnicama sistema.
b) Impedansa kvara postavljena je na nula, što znači da se proračunava metalni kratak spoj.
c) Napon pre kvara podešen na baznu vrednost.
d) Impedanse mašina postavljene da budu subtranzijentne.
e) Koriste se srednji položaji regulacionih transformatora.
f) Uvažavaju se doprinosi motora i susceptansi nultog redosleda vodova.

Nakon izvršenja programa i odabira opcije za prikaz rezultata dobija se izgled mreže prikazan na slici 4.2, gde brojevi označavaju vrednosti struja kvara u amperima na

odgovarajućim pozicijama a oznake LLL, i LG imaju sledeća značenja:

- LLL – trofazni kratak spoj,
- LG – jednofazni kratak spoj.



Slika 4.2 – *Izgled prikaza rezultata u CYME-u*

5. VERIFIKACIJA POSTUPKA PRORAČUNA

Kao verifikacija postupka proračuna struja kvara odrađen je proračun struja kvara na istoj test mreži kao i pri proračunu pomoću CYME-a, prikazanoj na slici 4.1. Proračun je urađen pomoću programskog paketa MATLAB pri čemu je određivanje ekvivalentne impedanse kvara odrađeno ručno za svako mesto kvara posebno.

Pri proračunu razmotrene su metode proračuna struja kratkih spojeva za izbor i proveru prekidača kao što je propisano u standardu ANSI Std C37.010-1999. Ovaj standard se koristi pri upotrebi prekidača za napone iznad 1kV, i predviđa korišćenje prekidača pri promenljivim uslovima upotrebe. Uporedni prikaz rezultata dobijenih ručnim proračunom i rezultata dobijenih korišćenjem CYME-a dati su u tabelama 5.1 i 5.2. U tabeli 5.1 prikazani su rezultati za trofazni kratak spoj a u tabeli 5.2 su prikazani rezultati za jednofazni kratak spoj. U prvoj koloni date su kompleksne vrednosi struje kvara, u drugoj njihovi moduli, u trećoj vrednosti dobijene korišćenjem CYME-a, u četvrtoj je razlika rezultata u procentima.

Tabela 5.1 – *Poređenje rezultata pri trofaznom kratkom spoju*

	I_3 [kA]	$\|I_3\|$ [A]	I_3 CYME	Δ [%]
B1	0.6818-j5.1688	5213	5211	0.03838
B2	1.9214-j16.560	16671	16672	-0.006
B3	0.9899-j4.7460	4848	4848	0
B4	0.2842-j1.2989	1329	1328	0.0753
B5	1.5304-j6.2101	6396	6387	0.1409
B6	0.3561-j1.4278	1471	1469	0.1361

Tabela 5.2 – *Poređenje rezultata pri jednofaznom kratkom spoju*

	I_1 [kA]	$\|I_1\|$ [A]	I_1 CYME	Δ [%]
B1	0.3968-j1.1550	1221	1221	0
B2	0.2475-j0.7710	810	810	0
B3	0.1357-j1.1582	1166	1166	0
B4	0.2456-j0.8068	843	843	0
B5	0.2248-j0.5574	601	601	0
B6	0.3572-j1.7606	1796	1794	0.1114

Očigledno je da su razlike vrednosti proračuna male, čime je potvrđeno slaganje modela korišćenog u CYME-u i onog korišćenog u ovom radu, što je i bio jedan od osnovnih zadataka rada.

6. ZAKLJUČAK

U radu su predstavljene osnovne karakteristike CYME programa za proračun struja kratkih spojeva. Rezultati dobijeni ručnim proračunom na test mreži slažu se sa rezltatima dobijenim korišćenjem CYME-a, uz razlike koje su manje od 0,15% što se može pripisati greškama u zaokruživanju pri proračunu. To pokazuje da su korišćene metode proračuna identične.

7. LITERATURA

[1] „IEEE Application Guide for AC High-Voltage Circuit Breakers Rated on a Symmetrical Current Basis.“ *IEEE Std C37.010*™-1999 (R2005)

Kratka biografija:



Siniša Božić rođen je u Bijeljini 1983. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetika – Elektroenergetski sistemi odbranio je 2011.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 620.93

SISTEM ZA ODREĐIVANJE KARAKTERISTIKA FOTONAPONSKIH PANELA

PHOTOVOLTAIC PANELS CHARACTERIZATION SYSTEM

Branislav Rodić, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Ovaj rad opisuje sistem dizajniran i realizovan u cilju određivanja karakteristika i parametara fotonaponskih (FN) panela, pri radu u realnim uslovima, sa funkcijama za merenje i prikazivanje rezultata merenja. Sistem koristi elektronsko opterećenje za određivanje I-U karakteristike. Merenje i analiza signala se vrši primenom koncepta tzv. virtuelne instrumentacije. Za upravljanje sistemom, kao i obradu i prikaz podataka razvijen je grafički korisnički interfejs u programskom okruženju LabVIEW.*

Abstract – *This paper describes a data acquisition system, designed and implemented for tracing photovoltaic (PV) panels characteristics and parameters, which operate under real field conditions, with facilities for measuring and results displaying. The system uses an electronic load for I-U characterization. Measuring and signal analysis is based on the so-called virtual instrumentation. For system control, data processing and displaying, a graphical user interface is developed in LabVIEW software.*

Ključne reči: *Fotonaponski panel, I-U karakterizacija, elektronsko opterećenje, LabVIEW*

1. UVOD

Proizvođači fotonaponskih (FN) panela pružaju podatke o I-U karakteristici i glavnim parametrima pod standardnim uslovima testiranja, koji iznose: 1000 W/m^2 vertikalnog sunčevog zračenja, 25°C temperature ćelije sa odstupanjima ± 2°C i definisanim svetlosnim spektrom (spektralnom raspodelom solarnog zračenja prema standardu IEC 60904-3) sa vazdušnom masom AM = 1.5 [1]. Ovi podaci se dobijaju u laboratorijskom okruženju, pod kontrolisanim uslovima.

Međutim, stvarni uslovi rada FN panela znatno odstupaju od laboratorijskih. Pojedini FN paneli, uprkos njihovoj dobro poznatoj pouzdanosti, degradiraju, pa čak i otkazuju posle dužeg perioda rada u spoljašnjim uslovima. U dijagnozi stepena degradacije FN panela koriste se njihove strujno-naponske (I-U) karakteristike, kao i parametri koji se iz ovih karakteristika dobijaju kao što su: struja kratkog spoja $I_{ks,}$ napon praznog hoda V_{ph}, maksimalna snaga P_{max}, faktor ispune FI i efikasnost η.

Sa druge strane, I-U karakteristika se može pratiti, a rezultati iskoristiti da bi se ispitala i uporedila stvarna snaga, koju panel proizvodi pod realnim radnim uslovima.

NAPOMENA:
Ovaj rad je proistekao iz master rada čiji je mentor bio prof. dr Vladimir Katić.

I-U karakteristike nisu značajne samo za projektovanje pretvarača, već i za dizajniranje FN sistema kako bi se oni učinili efikasnijim.

Na tržištu postoji više komercijalnih uređaja za testiranje FN panela pod realnim radnim uslovima. Oni se zasnivaju na upotrebi elektronskih DC opterećenja, koji simuliraju potrošače i koji mogu da menjaju otpornost veoma brzo. Takvi uređaji često su veoma skupi. Međutim, koristeći prosta i mnogo jeftinija kola, takođe se može napraviti elektronsko DC opterećenje, koje koristi prednosti rada MOSFET-a (metal-oksidnog tranzistora sa efektom polja) [2].

U ovom radu prikazan je sistem za određivanje karakteristika ili karakterizaciju FN panela u obliku laboratoriske edukativne makete, zasnovan na upotrebi MOSFETA-a u aktivnom režimu. Sistem se koristi za simulaciju potrošača, a samim tim i za određivanje I-U karakteristike.

2. OPIS SISTEMA

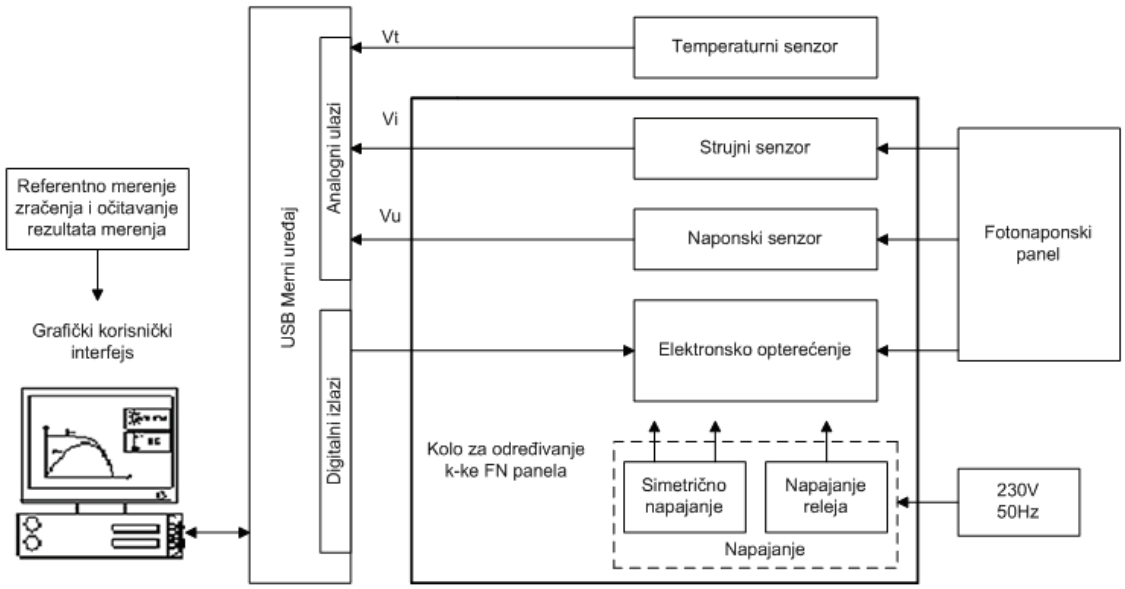
Na slici 1. prikazan je blok dijagram sistema za karakterizaciju FN panela. Osnovni element sistema predstavlja kolo za određivanje I-U karakteristike. Ono u sebi sadrži elektronsko opterećenje, napajanje, kao i senzore struje i napona.

Merni signali uvode se u multifunkcionalni uređaj za akviziciju podataka, koji pored analognih i digitalnih ulaza, sadrži i digitalne izlaze iskorišćene za upravljanje elektronskim opterećenjem. Uređaj za akviziciju podataka komunicira sa računarom opšte namene preko univerzalne serijske magistrale. Upravljanje mernim uređajem, akvizicija signala, matematička obrada prikupljenih podataka, kao i prikaz rezultata merenja i obrade vrše se preko LabView–baziranog grafičkog korisničkog interfejsa (GUI - Graphical User Interface). Omogućeno je prikazivanje I-U i P-U karakteristika kao i parametra koji se iz ovih karakteristika dobijaju.

3. ELEKTRONSKO OPTEREĆENJE

3.1. Princip rada

Elektronsko opterećenje je uređaj koji može da simulira statičko ili dinamičko opterećenje. Ono zamenjuje klasično rezistivno opterećenje i dopušta elektronsku kontrolu struje (ili ponekad napona) na bazi rada u linearnom (aktivnom) režimu. Nasuprot bipolarnom tranzistoru, MOSFET se mnogo lakše upravlja i nudi mnoge prednosti kao što su nepodložnost fenomenu sekundarnog proboja, mala vrednost drejn-sors otpornosti $R_{DS(ON)}$ (čime se postiže mali pad napona kod velikih struja), i dr. [3]. Osnovno kolo elektronskog opterećenja za testiranje FN panela, prikazano je na slici 2.



Slika 1. Blok dijagram sistema za karakterizaciju FN panela

V_{GS} je napon izmedju gejta i sorsa, V_{DS} je napon između drejna i sorsa, I_D jestruja drejna MOSFET-a, V_P je izlazni napon FN modula, I_P je izlazna struja. Sa slike možemo zaključiti sledeće:

$$I_P = I_{DS} \tag{1}$$

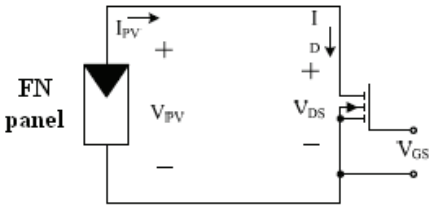
$$V_P = V_{Ds} \tag{2}$$

Pošto su struje I_D i I_{PV} jednake, radna tačka se dobija izjednačavanjem struje panela date izrazom (1) sa strujom MOSFET-a, koja je data sa:

$$I_D = K\left(2(V_{GS} - V_{TH})V_{DS} - V_{DS}^2\right) \tag{3}$$

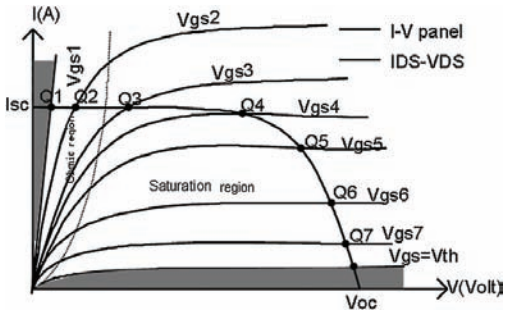
$$I_D = K(V_{GS} - V_{TH})^2(1 + \lambda V_{DS}) \tag{4}$$

Jednačina (3) važi u linearnoj (omskoj) oblasti gde je $V_{DS} < V_{GS}$-V_{TH}, dok jednačina (4) važi u aktivnom režimu gde je $V_{DS} > V_{GS}$-V_{TH}. K i λ su parametri uređaja dok je V_{TH} napon praga provođenja.



Slika 2. Osnovno kolo za testiranje FN panela, koristeći MOSFET kao elektronsko opterećenje

Dakle, radna tačka FN modula odgovara preseku njegove karakteristike i karakteristike MOSFET-a za dati napon V_{GS} (slika 3). Tada se može izmeriti par vrednosti struje i napona (I, U) modula. Menjajući napon V_{GS} radna tačka MOSFETA-a će prebrisati I-U karakteristiku modula između napona praznog hoda i struje kratkog spoja (na slici V_{OC} i $I_{SC,}$ respektivno). Promena napona V_{GS} se obezbeđuje punjenjem i pražnjenjem kondenzatora. U ovom slučaju koordinate radne tačke određene su promenljivom vrednošću napona na kondenzatoru. Može se primetiti da je odgovarajući rad MOSFET-a ograničen sa dve sive oblasti. U horizontalnoj oblasti MOSFET ne vodi, jer je napon V_{GS} manji od napona praga provođenja



Slika 3. *Karakteristika FN panela i karakteristike MOSFETA-a za različite vrednosti napona V_{GS}*

V_{TH}, tj. MOSFET je isključen. Povećavanjem napona V_{GS} iznad napona $V_{TH,}$ ulazi se u aktivni režim, gde struja drejna I_D raste približno linearno sa naponom $V_{GS.}$ Što se tiče vertikalne sive oblasti na slici 3, ona je posledica činjenice da, čak i u saturaciji, MOSFET predstavlja otpornost $R_{DS(ON)}$ i time ograničava oblast pravilnog rada.

3.2. Dimenzionisanje MOSFET-a

Maksimalna snaga, koju mora da se oslobodi na MOSFET -u, odgovara maksimalnoj snazi koju FN panel može da isporuči $P_{max.}$ Izabrani MOSFET mora da bude u stanju da disipira datu snagu. Za dimenzionisanje MOSFETA-a korišćen je FN panel Solar module 5W-12V proizvođača „Conrad“, dimenzija 235 x 170 mm. Podaci dobijeni od proizvođača, pod standardnim uslovima testiranja su: [4]

I_{KS} = 0.3 A
U_{PH} = 21.6 V
P_{max} = 5 W

MOSFET koji je upotrebljen je IRF520 i ima sledeće karakteristike: [5]

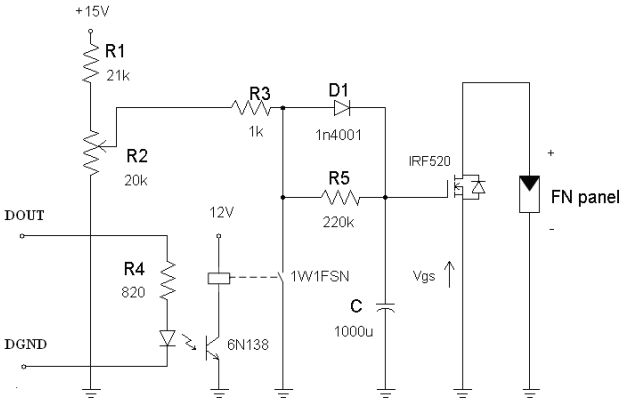
$R_{DS(ON)}$ = 0.23 Ω
BV_{Dss} = 100 V > U_{PH}
I_{DS} = 10 A > I_{KS}
P_D = 70 W pri 25° C > P_{max}
Vth = 4 V

3.3. Upravljanje MOSFET-om

Na slici 4 prikazan je deo kola za određivanje I-U karakteristike FN panela koji služi za upravljanje MOSFET-om, odnosno naponom V_{GS}. Kao što je već ranije istaknuto, menjajući napon V_{GS} radna tačka MOSFETA-a će prebrisati I-U karakteristiku panela. Ovaj deo kola sadrži relej (1W1FSN) i optokapler (6N138) koji je povezan na digitalni izlaz uređaja za akviziciju podataka. Ukoliko je vrednost digitalnog izlaza uređaja za akviziciju podataka na logičkoj nuli, optokapler ne vodi, relej ne dobija napon, pa se kondenzator C puni preko otpornika R_3 i diode D_1 na napon određen otpornicima R_1 i R_2. Kada se napon na kondenzatoru ustali, MOSFET provodi, imamo FN panel u kratkom spoju i merimo struju kratkog spoja I_{KS}.

Postavljanjem vrednosti digitalnog izlaza na logičku jedinicu, tj. 5 V, optokapler provede, relej dobije napon od 12 V i dolazi do zatvaranja njegovih kontakata i pražnjenja kondenzatora C preko otpornika R_5 na masu. Napon na kondenzatoru, tj. napon V_{GS} opada, MOSFET se koči i njegova radna tačka će prebrisati karakteristiku FN panela od struje kratkog spoja do napona praznog hoda. Kada se MOSFET u potpunosti zakoči, FN panel radi neopterećen i tada se meri napon praznog hoda V_{PH}.

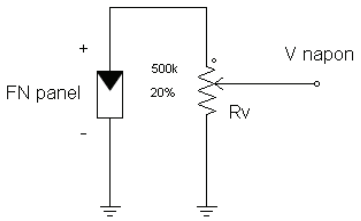
Upravljanje digitalnim izlazom uređaja za akviziciju podataka vrši korisnik u grafičkom korisničkom interfejsu.



Slika 4. Upravljanje MOSFET-om

4. MERENJE NAPONA

Za merenje napona FN panela koristi se običan trimer potenciometar od 500kΩ, kao što je to prikazano na slici 5. Vrednost otpora se postavi na 20% ukupne otpornosti. Naponski signal merenog napona V_{napon} se uvodi u uređaj za akviziciju podataka.



Slika 5. Merenje napona FN panela

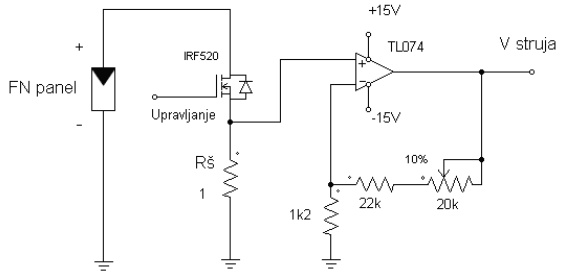
5. MERENJE STRUJE

Merenje struje fotonaponskog panela vrši se kao što je to prikazano na slici 6. Strujni signal se na mernom šantu pretvara u naponski i uvodi u neinvertujući pojačavač Napon na izlazu pojačavača je:

$$V_{struja} = I_{FN} \cdot R_{šant} \cdot A_{poj} \quad (5)$$

Kako je za date otpornosti u kolu pojačavača, pojačanje A=17.5, a otpornost šanta $R_{šant}$ = 1 Ω, naponski signal merene struje iznosi:

$$V_{struja} = I_{FN} \cdot 17.5 \quad (6)$$



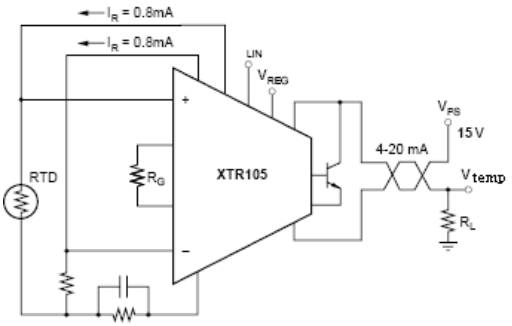
Slika 6. Merenje struje FN panela

Naponski signal merene struje V_{struja} se sa izlaza pojačavača uvodi u uređaj za akviziciju podataka.

6. MERENJE TEMPERATURE

Temperatura panela meri se pomoću kola prikazanog na slici 7, koje koristi PT100 temperaturni senzor i XTR105 strujni pretvarač. XTR105 je integrisani strujni pretvarač, koji radi sa dvožičnom konfiguracijom, struje 4-20mA, sa dva strujna izvora [6]. Zavisnost izlazne struje od temperature data je sledećom jednačinom:

$$I = \left[\frac{16}{400} \cdot (T + 50)\right] + 4 \quad [\text{mA}] \quad (7)$$



Slika 7. Kolo za merenje temperature FN panela

Strujni signal se na otporniku R_L pretvara u naponski signal $V_{temp.}$ Za pretvaranje stujnog signala u naponski uzet je otpornik otpornosti R_L = 150 Ω. Kolo za merenje temperature realizovano je na štampanoj ploči, koja se postavlja sa zadnje strane FN panela. Na ovu ploču dovodi se napon napajanja od 15V sa kola za određivanje I-U karakteristike. Naponski signal merene temperature V_{temp} uvodi se u uređaj za akviziciju podataka.

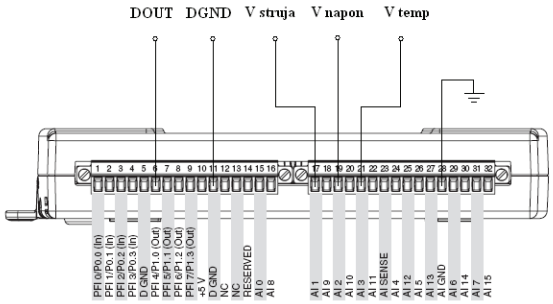
7. MERENJE ZRAČENJA

Merenje zračenja vrši se pomoću kalibrisanog fotonaponskog uređaja *Sunny SensorBox*-a. Rezultati merenja se snimaju u Excel tabelu i unose u LabVIEW-baziran grafički korisnički interfejs.

8. AKVIZICIJA PODATAKA

Za realizaciju merenja putem virtuelne instrumentacije, koristi se National Instruments USB-6210 DAQ (Data

acquisition) uređaj za akviziciju podataka, optimizovan za brzo sempliranje podataka (250 kS/s) i visoku tačnost merenja, koji se napaja i komunicira sa računarom preko USB – interfejsa. Poseduje 16 analognih ulaza (opsega ±0.2 do ±10 V), četiri digitalna ulaza, četiri digitalna izlaza [7]. Na slici 8 prikazana je šema priključivanja NI USB-6210 mernog uređaja sa kolom za određivanje I-U karakteristike kao i sa temperaturnim senzorom.



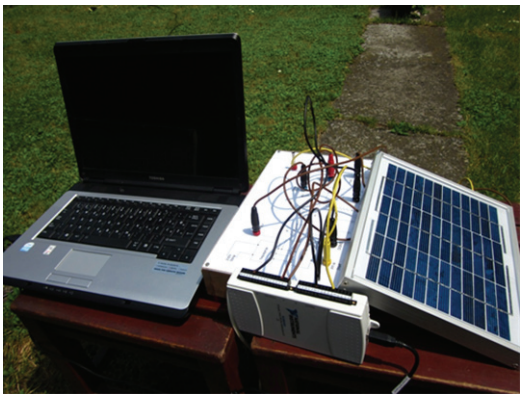
Slika 8. Povezivanje NI USB-6210 mernog uređaja

9. GRAFIČKI KORISNIČKI INTERFEJS

U programskog okruženju LabVIEW realizovan je grafički korisnički interfejs koji se sastoji iz dva potprograma – virtuelna instrumenta. Prvi potprogram služi za upravljanje kolom za određivanje I-U karakteristike, merenje veličina FN panela (struje, napona i temperature) i snimanje rezultata merenja. Drugi potprogram služi za obradu rezultata merenja, prikazivanje I-U i P-U karakteristika kao i parametara koji se iz ovih karakteristika dobijaju: napona praznog hoda, struje kratkog spoja, maksimalne snage, napona pri maksimalnoj snazi, struje pri maksimalnoj snazi, temperature panela, faktora ispune i efikasnosti. Indikatori zračenja i površine panela su upravljački, tj. korisnik unosi vrednosti ovih veličina.

10. TESTIRANJE SISTEMA

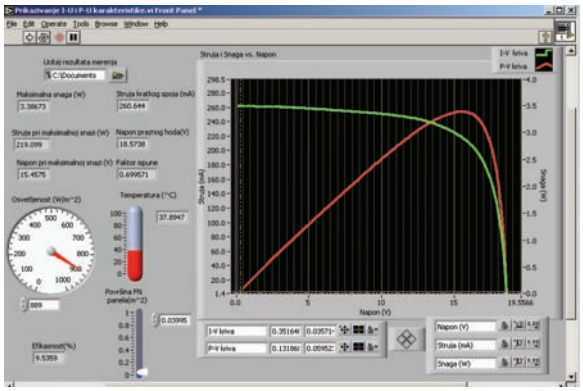
Sistem je razvijen u obliku laboratorijske edukativne makete prikazane na slici 9. Testiranje sistema za karakterizaciju izvršeno je za gore pomenuti FN panel Solar module 5W-12V proizvođača „Conrad“, površine ćelija 0.04 m^2. Rezultat karakterizacije dat je na slici 10.



Slika 9. Eksperiment za karakterizaciju FN panela

11. ZAKLJUČAK

Ovaj rad je opisao sistem za karakterizaciju fotonaponskih panela pod realnim radnim uslovima, koji koristi elektronsko opterećenje, uz merenje primenom virtuelne instrumentacije. Korišćenjem MOSFET-a za realizaciju elektronskog opterećenja i određivanje I-U karakteristike, dobijen je dobar rezultat. Svi važni parametri radnog režima su merljivi, sa prihvatljivom preciznošću. Upotreba uređaja za akviziciju podataka omogućila je jednostavnu kontrolu i merenje veličina sistema, a LabVIEW baziran grafički korisnički interfejs daje pregledan prikaz rezultata merenja i obrade podataka.
Predloženi sistem je pogodan za analizu uticaja zračenja, temperature i delimičnog zamračenja na performanse FN panela, kao i detekciju uslova degradacije i otkazivanja.



Slika 10. I-U i P-U krive Solar module 5W-12V,

$T = 38$ °C, $E = 889$ W/m^2 , $V_{PH} = 18.57$ V, $I_{KS} = 260.6$ mA
$P_{max} = 3.39$ W, $I_{Pmax} = 219.1$ mA, $U_{Pmax} = 15.45$ V,
$F = 0.69$, $\eta = 9.54$ %

12. LITERATURA

[1] ****, *“Planning and installing photovoltiac systems*”, The German Energy Society (Deutsche Gesellshaft fur Sonnenenergie), Berlin, 2008.

[2] V. Leite, F. Chenlo: *“An improved electronic circuit for tracing the I-V characteristics of photovoltaic modules and strings*”, International Conference on Renewable Energies and Power Quality, Granada (Spain), 2010.

[3] H. Belmili, S. M. A. Cheikh, M. Haddadi, C. Larbes: *“Design and development of a data acquisition system for photovoltaic modules characterization”,* Renewable Energy, 2010.

[4] Solar module 5W-12V, www.conrad.com, [Available On-line], Last accessed 15.06.2011

[5] MOSFET IRF520, www.datasheetcatalog.com, [Available On-line], Last accessed 15.06.2011

[6] Current transmitter XTR105, Texas Instruments, www.ti.com. [Available On-line], Last accessed 15.06.2011

[7] USB-6210 DAQ, National Instruments, www.sine.ni.com [Available On-line], Last accessed 15.06.2011

Kratka biografija:



Branislav Rodić je rođen 1985. god. u Novom Sadu. Diplomski - Master rad na Fakultetu tehničkih nauka u Novom Sadu, iz oblasti Elektrotehnike i računarstva, na smeru Energetska elektronika i mašine, odbranio je u julu 2011. god.



Vladimir Katić je doktorirao na Univerzitetu u Beogradu 1991. Redovni profesor Fakulteta tehničkih nauka u Novom Sadu je od 2002. za oblast Energetska elektronika, mašine i pogoni. Prodekan je FTN-a i šef Katedre za energetsku elektroniku i pretvarače. Oblasti interesovanja su energetska elektronika, kvalitet električne energije i obnovljivi izvori električne energije.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621

ELEKTRONIKA ZA UPRAVLJANJE HIDRAULIČNOM PRESOM ZA BRIKETIRANJE

ELECTRONIC CONTROL FOR HIDRAULIC BRIQUETTE PRESS

Jovan Brkljača, Laslo Nađ, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U okviru ovog rada je osavremenjen stari tip hidraulične prese za briketiranje. Rad opisuje dve verzije realizacije automatskog upravljanja briketirke.*

Abstract – *Old type of hydraulic press for briquetting has been modernized in this paper. The paper describes two versions of the implementation of automatic control for hydraulic briquette press.*

Ključne reči: *briketiranje piljevine, hidraulični cilindar, relejna logika*

1. UVOD

Briketiranje biomase je proces u kojem se smanjuje zapremina i povećava specifična masa. Ideja o proizvodnji briketa nastala je iz sopstvenih potreba. Pri proizvodnji parketa se stvara drvni otpad, suva piljevina. Godinama se manji deo piljevine koristio za loženje u kotlu koji služi za zagrevanje sušare za drvo, a veći deo je prodavan ili je bio odvožen na deponiju. Za potrebe zagrevanja kotla nije neophodno koristiti suvu piljevinu, može se koristiti i drugi drvni otpad. Odvoženje piljevine na otpad je dodatni trošak koji treba izbegavati. Takođe, prodaja piljevine kao biomase je neisplativa. Iz prethodno navedenih razloga najbolje rešenje je proizvodnja briketa. Samo se suva piljevina može briketirati. Tokom pomenute proizvodnje se upravo stvara samo suva piljevina, što predstavlja ogromnu prednost, jer su izbegnuti troškovi energije za sušenje. Piljevina iz proizvodnje parketa je isključivo od tvrdih vrsta drveta i visoke je kalorične vrednosti.



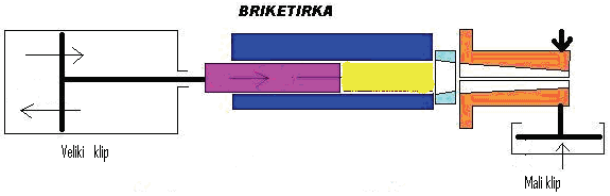
Slika 1. *Izgled briketirke*

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Laslo Nađ, vanr. prof.

Idealne je granulacije za hidrauličnu presu za briketiranje (u daljem tekstu briketirka).
U okviru ovog rada je urađeno automatsko upravljanje realizovano na dva načina za briketirku Sipa K-60, starije proizvodnje. Takođe je neophodno bilo realizovati i ručno upravljanje. Izvršen je remont hidrauličnih sklopova. Mašina nije imala prateću dokumentaciju, niti je imala automatsko upravljanje u originalnoj izvedbi. Na slici 1. dat je izgled mašine.

2. OPIS RADA BRIKETIRKE

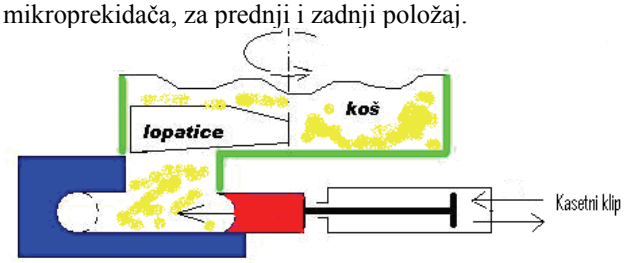
Briketirka Sipa K-60 je namenjena za presovanje usitnjene biomase, koja se javlja kao nusproizvod u poljoprivrednoj proizvodnji, drvnoj industriji i dr. Ona nalazi primenu u savremenim poljoprivrednim gazdinstvima sa ciljem korištenja biomase kao obnovljivog izvora energije, u industriji prerade drveta korišćenjem otpada i piljevine i slično [1].
Mašina treba da predstavlja pouzdan uređaj koji se sastoji od postolja, prihvatnog koša, radnog dela i hidraulične instalacije. Na postolju je pogonski elektromotor koji pokreće hidrauličnu pumpu. Prihvatni koš sa dozatorom materijala u radni cilindar je takođe na postolju. Prilikom briketiranja prethodno pripremljen materijal (adekvatno usitnjen i optimalne vlažnosti) se dovodi u prihvatni koš. Dozator na dnu koša ubacuje materijal u radni cilindar. Proces presovanja se odvija pod dejstvom sila hidrauličnih cilindara. Materijal prolazi kroz cilindar i odlazi na uvrećavanje. U sastavu mašine su tri hidraulična cilindra sa klipom i klipnjačom i to dva dvosmerna i jedan jednosmerni. (U nastavku će se koristiti termin klip umesto hidraulični cilindar.) Na slici 2 prikazan. je veliki i mali klip, a na slici 3. kasetni klip. Veliki klip je dvosmerni i služi da izbaci komad briketa.



Slika 2. *Veliki klip i mali klip*

Kasetni klip je takođe dvosmerni i služi da sabije piljevinu. Dejstvo kasetnog klipa je normalno na ravan slike 2. Mali klip je jednosmerni i pod pritiskom je samo kad kasetni klip sabija piljevinu, a vraća se u početni požaj kada veliki klip izbacuje briket. Operacije se ponavljaju periodično sledećim redosledom: veliki klip se kreće nazad (oslobađa radni prostor cilindra), kasetni klip napred (sabija piljevinu), veliki klip napred (izbacuje se briket) i na kraju kasetni klip nazad. Kada se kasetni klip

kreće nazad, nova količina piljevine se ubacuje u radni prostor cilindra. Veliki klip i kasetni klip se kreću do graničnih mikroprekidača. Svaki klip ima po dva granična mikroprekidača, za prednji i zadnji položaj.



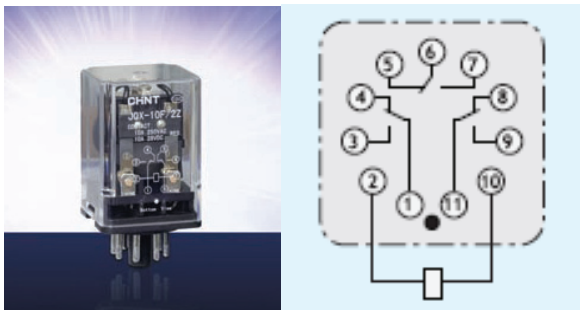
Slika 3. *Kasetni klip*

3. REALIZACIJA UPRAVLJANJA BRIKETIRKE

Upravljanje briketirkom se obavlja uključivanjem elektromagnetnih hidrauličnih razvodnika što ima za posledicu kretanje hidrauličnih klipova do graničnih mikroprekidača. U narednim potpoglavljima su opisana dva načina realizacije.

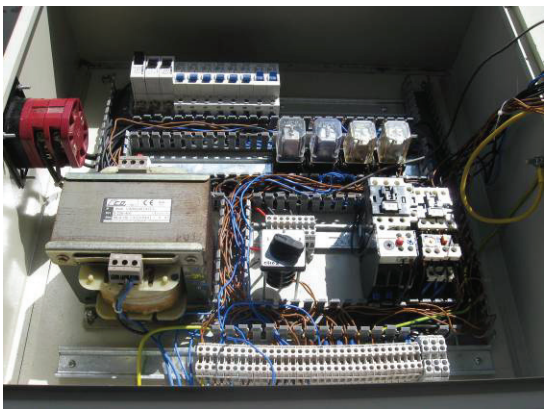
3.1. Realizacija upravljanja pomoću releja

Automatsko upravljanje bazirano na relejnoj logici koristi tri releja sa oznakom IJQX-10F proizvođača *CHNT* (slika 4.).



Slika 4. *Relej za automatsko upravljanje*

Elektrokomandni ormar je prikazan na slici 5. U elektrokomandnom ormaru se nalaze osigurači, prekidači, kontaktori, bimetali, signalizacione sijalice...



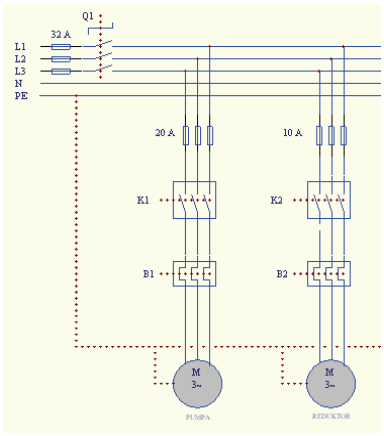
Slika 5. *Elektrokomandni ormar*

Na slici 6. je data električna šema sa dva elektromotora :

- elektromotor koji pogoni hidrauličnu pumpu i
- elektromotor dozatora piljevine iz koša.

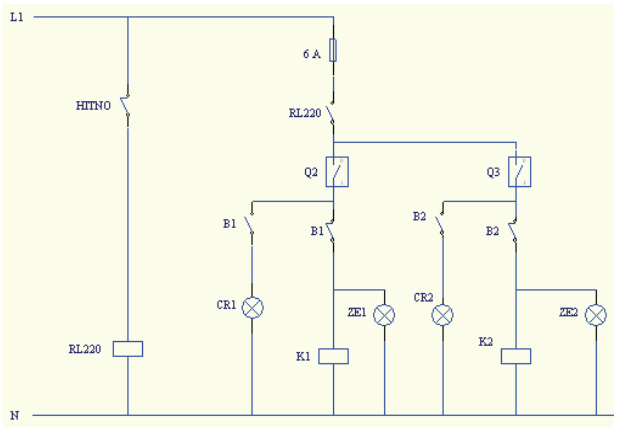
Grebenasti prekidači služe za zatvaranje strujnih kola. Napajanje iz mreže je dovedeno preko glavnog grebenastog prekidača Q1 i osigurača 32 A.

Kontaktori se koriste za upravljanje, odnosno uključivanje elektromotora. Zajedno sa bimetalnim zaštitnim relejom i osiguračima upotrebljavaju se kao električna zaštita elektromotora. Predviđeni su za trajni i isprekidani rad. Pomoćni kontakti smešteni su sa strane kontaktora i mogu pretstavljati mirni ili radni kontakt. Ove kombinacije omogućuju rešavanje raznih zahteva kod automatizovanog pogona [2].



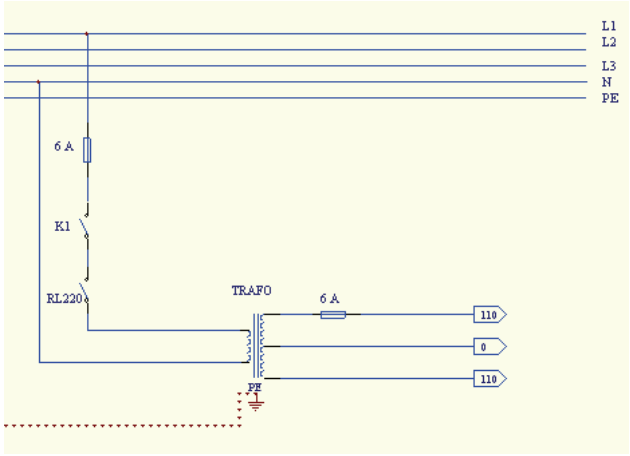
Slika 6. *Elektromotori sa električnom zaštitom*

Elektromotor pumpe se uključuje preko kontaktora K1, sa bimetalnom zaštitom B1 i osiguračima od 20 A, dok se elektromotor dozatora piljevine iz koša uključuje preko kontaktora K2 sa bimetalnom zaštitom B2 i osiguračima 10 A . Na slici 7. je prikazano pod kojim uslovima je moguće uključiti elektromotor pumpe i elektromotor dozatora piljevine iz koša. Ako sigurnosni stop prekidač (u daljem tekstu hitni prekidač) nije u radnom položaju, nije uključen ni relej RL220. Relej RL220 ima tri radna ili mirna kontakta. Da bi bila uključena pumpa, odnosno kontaktor K1, potrebno je da relej RL220 bude uključen, da prekidač Q2 bude uključen i da ne radi bimetalna zaštita B1. U slučaju kada ne radi bimetalna zaštita, zelena sijalica ZE1 svetli, a ukoliko proradi bimetalna zaštita, isključuje se pumpa preko kontaktora K1 i imamo signalizaciju na crvenoj sijalici CR1. Slično kao i za pumpu je za elektromotor dozatora piljevine iz koša sa kontaktorom K2, prekidačem Q3, bimetalom B2 i sijalicama (crvenom CR2 i zelenom ZE2).



Slika 7. *Uslovi uključivanja kontaktora*

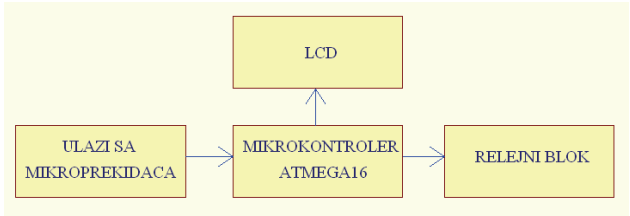
Transformator 230/110 VAC je neophodan za napajanje elektromagnetnih klipnih razvodnika. Na slici 8. je prikazan način povezivanja transformatora. Uslovi da transformator bude uključen, je uključen kontaktor K1 i aktivan relej RL220. Ukoliko nisu uključeni kontaktor K1 (pumpa) i relej RL220 (hitni prekidač), nema potrebe za napajanje elektromagnetnih klipnih razvodnika, odnosno uključivanje mrežnog transformatora.



Slika 8. *Uslovi uključivanja transformatora*

3.2. Realizacija upravljanja pomoću mikrokontrolera

Mikrokontroler ATMEGA16 na osnovu stanja mikroprekidača na ulazu zadaje stanje releja u relejnom bloku na izlazu i na taj način upravlja mašinom. Mikrokontroler prikazuje podatke na displeju (LCD). Blok šema je prikazana na slici 9.

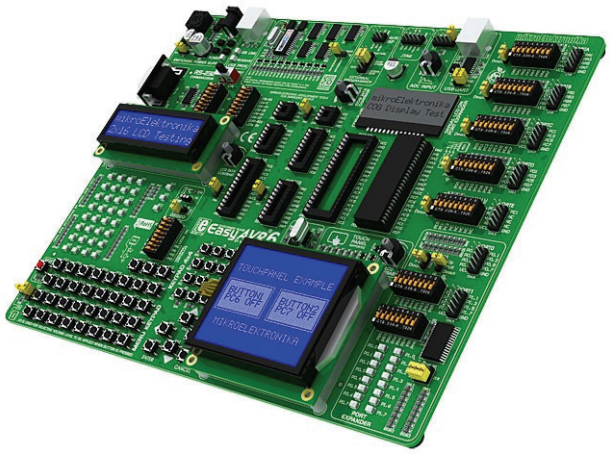


Slika 9. *Blok šema*

Napajanje treba da obezbedi stabilisane napone od 5VDC i 12VDC sa strujama ne većim od 1A. Korišćen je transformator sa dva izvoda na sekundaru i naponski stabilizatori LM7805 i LM7812. Napon od 12VDC se koristi za releje na izlazu, a napon od 5VDC za napajanje mikrokontrolera sa svim pratećim elementima.

Mikrokontroler ATMEGA16 je osmobitni mikrokontroler kompanije *ATMEL*. U osnovi se nalazi moderni 8-bitni mikroprocesor koji sadrži 32 registra opšte namene, 16K bajta fleš memorije koja služi za čuvanje programa koji procesor izvršava, 512 bajta EEPROM memorije koja služi za čuvanje podataka koji ne smeju biti izgubljeni po nestanku napajanja i 1K bajta interne SRAM memorije za čuvanje podataka. Performanse ovog AVR mikrokontrolera su dosta iznad prosečnih. AVR mikroprocesore odlikuje velika brzina izvršavanja, do 16 MIPS-a, koja je oko 12 puta veća nego kod standardnih 8-bitnih mikroprocesora, dok su sa druge strane i memorijske performanse znatno poboljšane, kombinovanim korišćenjem fleš memorije i EEPROM memorije. ATMEGA16 poseduje tri tajmer/brojač jedinice. Dva tajmer/brojača su osmobitna, dok je jedan tajmer/brojač šesnaestobitni [4].

Za potrebe ovog rada korišćena je razvojna ploča Easyavr6 proizvođača *Mikroelektronika*, Beograd (slika 10.). Releji koji se koriste su proizvod firme *Rayex elec.* Oni su odabrani jer ih je moguće postaviti na štampanu ploču manjih dimenzija.



Slika 10. *Razvojna ploča Easyavr6 [3]*

Napon napajanja releja je 12VDC. Za upravljanje relejima je korišćen odgovarajući tranzistorski drajver, koji se nalazi u kolu ULN2803.

Ulazi sa graničnih mikroprekidača su povezani na mikrokontroler. Galvansko odvajanje je realizovano optokaplerima.

4. TESTIRANJE PROTOTIPA

Nakon projektovanja i realizacije upravljanja, urađeno je testiranje prototipa. Prototip je testiran pre povezivanja sa mašinom. Kako automatsko upravljanje pomoću releja, tako i upravljanje pomoću mikrokontrolera su testirani u laboratorijskim uslovima. Takođe su testirani i energetski sklopovi, koliko je to bilo moguće u laboratoriji.

Prilikom testiranja upravljanja sa relejima, povezani su granični mikroprekidači na elektrokomandni ormar i dobijeni su očekivani rezultati. Testiran je rad u zavisnosti od početnih stanja mikroprekidača i rad kada je za početne uslove uzeto da su svi mikroprekidači isključeni. U okviru testiranja upravljanja varijante sa relejima, testiran je i ručni način upravljanja. Od energetskih sklopova je ispitano sledeće: uključivanje i isključivanje pumpe i motora sa reduktorom, uključivanje i isključivanje transformatora, signalizacija kada radi i kada ne radi bimetalna zaštita i signalizacija stanja u kojima se nalaze klipovi.

Za varijantu sa relejima se koriste samo tri releja koja imaju po tri mirna ili radna izlaza. Na svakom releju se koriste svi izlazi, što govori da se koristi minimalni broj komponenti da bi se obavljale potrebne funkcije. Održavanje je veoma jednostavno, otklanjanje kvara podrazumeva samo zamenu neispravnog releja. Cena ovog rešenja je niska. Ukoliko se želi napraviti bilo koja, čak i najmanja izmena u procesu rada mašine ovim načinom upravljanja, neophodno je projektovanje od početka, što predstavlja najveću manu ovog rešenja. S obzirom da je prilikom projektovanja upravljanja pomoću mikrokontro-

lera korišćena razvojna ploča, na koju su bili spojeni releji na izlazima, potrebni u ovoj varijanti upravljanja, u svakom koraku projektovanja je obavljano ispitivanje.
Glavna prednost ovog rešenja je fleksibilnost, lakoća i jednostavnost izmene koda programa po potrebi, odnosno procesa rada mašine. To ima smisla raditi ukoliko unapred nemamo jasno definisan redosled operacija koje treba da obavlja mašina. Korišćenje displeja nije neophodno, s obzirom na male dimenzije istog. Velika mana je što bilo koji kvar vezan za upravljanje varijante sa mikrokontrolerom mora da otkloni stručno lice (a ne obučeni tehničar za održavanje), koje je radilo razvoj hardvera i softvera. Izrada i održavanje su dosta skupi. Ipak i pri odabiru ovog rešenja se ne sme izostaviti ručni način upravljanja čija realizacija nije preporučljiva da bude bliska ovom rešenju, zbog uslova rada (prašina i mogućnosti mehaničkih oštećenja). Tu se pre svega misli na korišćenje tastera na štampanoj ploči u uslovima proizvodnje.
Sledeći korak vezan za testiranje je povezivanje upravljačkog bloka sa mašinom i ispitivanje rada mašine u uslovima u kojima je predviđeno da se proizvodnja obavlja. To će se obaviti u prilikom prvog sledećeg remonta mašine.

5. ZAKLJUČAK

U okviru ovog rada je prikazan značaj i potreba za korišnjenjem biomase u oblasti briketiranja. Opisan je način rada hidraulične briketirke i osnove hidrauličnih sistema. Realizovano je automatsko upravljanje pomoću releja i upravljanje pomoću mikrokontrolera za hidrauličnu briketirku Sipa K-60. Za projektovanje štampane ploče je korišćen softverski paket Protel99. Štampana ploča je napravljena na Katedri za elektroniku na Fakultetu tehničkih nauka. Komponente su lemljene ručno. Elektrokomandni ormar je sklopljen samostalno. Za pisanje programskog koda u jeziku C za mikrokontroler Atmega16 je korišćen kompajler Avrstudio. Programiranje mikrokontrolera je urađeno na razvojnoj ploči Easyavr6, proizvođača *Mikroelektronika*, Beograd. Testiranje prototipa je urađeno u laboratorijskim uslovima.

Za razvoj upravljanja pomoću mikrokontrolera je potrebno više vremena nego za razvoj upravljanja pomoću releja. Održavanje varijante sa relejima je lakše od varijante sa mikrokontrolerom. Odgovor na pitanje koji je način upravljanja pouzdaniji dobiće se nakon dužeg vremena testiranja sa jednim i sa drugim rešenjem, u normalnom režimu eksploatacije mašine.

6. LITERATURA

[1] Dr Miladin Brkić**,** *"Potencijali i mogućnosti briketiranja i peletiranja otpadne biomase na teritoriji pokrajine Vojvodine", studija,* Poljoprivredni fakultet, Novi Sad, 2007.
[2] Đorđe B.Kovačević, *"Električno pokretanje sa automatikom" ,* Zagreb, 1971.
[3] -, *"Uputstvo za Easyavr6"*, http://www.mikroe.com/eng/downloads/get/348/easyavr6_manual_v101.pdf , Mikroelektronika Beograd, jun 2011.
[4] - , Atmega16, http://www.atmel.com/dyn/resources/prod_documents/doc8154.pdf , Atmel, 2011

Kratka biografija:



Jovan Brkljača rođen je u Ljubinju, BiH 1984. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektronika odbranio je 2011.god.



Laslo Nadj rođen je u Subotici 1953. Doktorirao je na Fakultetu tehničkih nauka 1992. god. Oblast interesovanja su elektronika u robotici, mehatronici / klasična elektronika (impulsna i analogna elektronika, digitalna elektronika).

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK 621.397.13

REALIZACIJE DVB–T2/SFN MREŽE PREDAJNIKA/REPETITORA

REALISATION OF THE DVB –T2/SFN TRANSMITTERS/REPEATERS NETWORK

Nebojša Dragić, *Fakultet tehničkih nauka, Novi Sad*

Oblast–ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj–*MPEG kompresija, FEC prevencija greške i OFDM modulacije predstavljaju tri osnovna elementa zemaljskog prenosa i emitovanja digitalnog TV signala u DVB–T standardu. Ako se tome doda mogućnost rada u SFN mrežama, čini da se zemaljski prenos i emitovanje mogu ravnopravno takmičiti sa kablovskim prenosom i distribucijom.*

Abstract–*MPEG compression, FEC error prevention and OFDM modulation are three basic elements of digital TV signal terrestrial transmission in the form of DVB–T standard. Expanded capability of that reality to work in SFN networks seems to terrestrial transmission and broadcasting can equally compete with cable transmission and distribution.*

Ključne reči: MPEG, FEC, OFDM, DVB–T, DVB –T2, SFN

1. UVOD

Prenos i distribucija digitalizovanog TV signala mogući su jedino u uslovima kompresije i optimalnog korišćenja spektra prenosnog kanala. Kompresijom signala TV slike i pratećeg audio signala u vidu MPEG formata bitski protok digitalizovanog TV signala dovodi se na opseg vrednosti kojim se prenos i distribucija digitalnog TV signala čine realno mogućim.

Primenom QAM (kvadraturne aplitudske modulacije) nad međusobno ortogonalnim prostoperiodičnim nosiocima dobija se OFDM modulacija.

Na ovaj način postiže se najveća moguća spektralna efikasnost koja postojeće analogne prenosne kanale čini pogodnim za prenos multipleksnih video i audio sadržaja (programa).

Posebno svojstvo OFDM modulacije da se upotrebom posebnog zaštitnog intervala može rešiti problem interferencije čini prijemni signal otpornim na smetnje i otvara mogućnost istovremenog prijema istog signala na istoj frekvenciji iz više različitih izvora što je osnova rada jednofrekvencijskih radiodifuznih mreža (SFN-Single Frequency Networks).

Mada radiodifuzne mreže digitalnog multipleksnog sadržaja mogu da rade i na klasičan način kao i postojeće analogne mreže pokrivanjem određene zone servisa većim brojem kanala u vidu višefrekvencijskih MFN (Multi Frequency Network) mreža, SFN mreže svakako predstavljaju efikasan način da se ekonomično i optimalno koristi raspoloživi spektar.

NAPOMENA:

Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Vojin Šenk, red. prof.

2. DVB–T/DVB–T2 STANDARD PRENOSA I RADIODIFUZNOG EMITOVANJA

Diskretna struktura digitalnog TV signala omogućava primenu aparata diskretne matamatike u digitalnoj obradi signala kako u domenu kompresije tako i u domenu mehanizma sprečavanja greške–FEC **(**Forward Error Corection).

Ulaz DVB–T/DVB–T2 sistema radiodifuznog prenosa i emitovanja čini digitalizovani multipleksni sadržaj MPEG formata kompresije. MPEG–2 format kompresije bitski protok TV signala svodi na protok reda veličine 3-5 Mb/s. Time je omogućen prenos i distribucija 4-7 TV programa SDTV kvaliteta za bitski protok DVB–T standarda od 20 Mb/s.

U međuvremenu je razvijen efikasniji standard kompresije MPEG–4.10 kao i DVB–T2 standard kod koga su mogućnosti prethodnog standarda proširene i poboljšane. Primenom MPEG–4 formata potreban bitski protok video i audio sadržaja SDTV rezolucije svodi se na oko 2,5 Mb/s, a HDTV rezolucije na 7-11 Mb/s čime postaje realno moguć prenos i distribucija HDTV sadržaja.

Istraživanja pokazuju da DVB–T2 standard omogućava i do 88% povećanja bitskog protoka u odnosu na DVB–T standard korišćenjem istog MPEG–4 formata [1]. DVB–T2 standard u kombinaciji sa MPEG–4 standardom kompresije postiže više nego duplo veći dobitak u broju programa jednog multipleksnog paketa u odnosu na prethodni DVB–T/MPEG–2 standard [1].

2.1 Sprečavanje greške (FEC)

Sprečavanjem greške postiže se efekat prividnog prenosnog kanala bez greške–QEF (Quasy Error Free) kanala. Da bi se izbegli dugački nizovi uzastopnih 0 ili 1 digitalizovanog TV signala (nizovi uzastopnih jedinica sadrže značajnu DC komponentu), vrši se predobrada.

Postupak kojim se obezbeđuje podjednaka raspodela energije u RF kanalu i koji ne pripada prevenciji greške jeste energetska raspodela (randomizacija).

Reorganizacija signala ostvaruje se skremblovanjem signala pseudo-slučajnom binarnom sekvencom (PRBS) generisanom polinomom $1 + X^{14} + X^{15}$.

Osnov sprečavanja greške predstavlja zaštitno kodovanje kombinovano sa odgovarajućim interliving postupcima. Interliving postupci obezbeđuju prekompoziciju vremenskog redosleda bitskih sekvenci odgovarajuće veličine u funkciji cepanja i preraspodele greške u sigurnije intervale vremena na osnovu kojih će primenom zaštitnog kodovanja greške verovatnije biti otklonjene.

Interliving predstavlja svojevrsno proširenje zaštitnog kodovanja u smislu povećanja korektibilne sposobnosti kodova. Spoljašnji kod DVB–T standarda jeste RS (Reed-

Solomon) kod 204,188,8. On je u potpunosti pilagođen veličini MPEG–2 transportnog paketa veličine 188 bajta.
Unutrašnji kod DVB–T standarda predstavlja konvolucioni kod osnovnog kodnog količnika 1/2. Kodni količnik konvolucionog koda se zbog smanjenja redundantnosti uvećava punktiranjem na neku od standardnih vrednosti: 2/3, 3/4, 4/5, 5/6, 7/8.
Punktiranje odgovarajućih bita vrši se na osnovu kodnog količnika i prema odgovarajućem ključu (ETS 300 421).
Fornijev konvolucioni interliving proširuje RS kod dok se za unutrašnje proširenje koda koristi bitski interlivig zasnovan na permutacionoj funkciji definisanoj preporukom ETS 300 744.
Primena spoljašnjeg BCH koda i unutrašnjeg LDPC blok koda primerenijih dužim kodnim sekvencama (64 Kb) u kombinaciji sa višestrukim interliving postupcima sa rotiranom konstelacijom predstavljaju osobenosti DVB–T2 FEC postupka. BCH kod u odnosu na RS kod doprinosi umanjenu ukupne redundantnosti za približno 8%.
LDPC kod ne doprinosi umanjenju redundantosti pošto su kodni količnici: 1/2, 3/5, 2/3, 3/4, 4/5 i 5/6 sličnih vrednosti kodnim količnicima konvolucionog koda. Bitski, vremenski, ćelijski i frekvencijski interliving relativno složenih permutacionih funkcija [4] u kombinaciji sa rotiranom konstelacijom DVB–T2 signal čine efektivno otpornijim za oko 3 dB u potrebnom nivou prijemnog polja, što rezultira nižim potrebnim kodnim količnikom a time i povećanim bitskim protokom [3].

2.2 OFDM modulacija

Ulazni binarni bitski niz OFDM modulatora dužine n bita prevodi se postupkom mapiranja u M-arni digitalni reprezent ($M = 2^n$).
Ove vrednosti OFDM modulator interpretira kao diskretne Re(k) i Im(k) spektralne komponente.
Primenom IFFT (Inverse Fast Fourier Transform) proračuna, OFDM modulator generiše niz sukcesivnih prostoperiodičnih međusobno ortogonalnih nosilaca koje moduliše odgovarajućim Re(k) i Im(k) M- arnim digitalnim reprezentima kvadraturnom amplitudskom modulacijom (N–QAM, $N = M^2$).

$$S(k) = \sum_{n=0}^{N-1} S(n)\, e^{-\frac{j2\pi kn}{N}} \quad k = 0,1,\dots,N-1 \qquad (1.1)$$

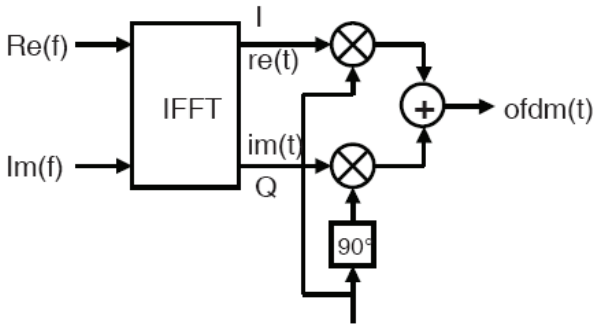
$$S(n) = \sum_{k=0}^{N-1} S(k)\, e^{\frac{j2\pi kn}{N}} \quad n = 0,1,\dots,N-1 \qquad (1.2)$$

$$S(k) = Re(k) + Im(k) \qquad (1.3)$$

$$Re\big(S(n)\big) = \sum_{k=0}^{N-1}(Re(k)\cos(\omega_k t_n) - Im(k)\sin(\omega_k t_n)) \qquad (1.4)$$

$$Im\big(S(n)\big) = \sum_{k=0}^{N-1}(Im(k)\cos(\omega_k t_n) - Re(k)\sin(\omega_k t_n)) \qquad (1.5)$$

Rezultat IFFT jeste modulacija odgovarajućeg nosioca u skladu sa vrednostima Re i Im komponente, njihovog vremenskog redosleda (određuje frekvenciju nosioca). Krajnji rezultat predstavlja sumiranje modulisanih nosilaca po Re grani kao i po Im grani u okviru IFFT proračuna (slika 1).



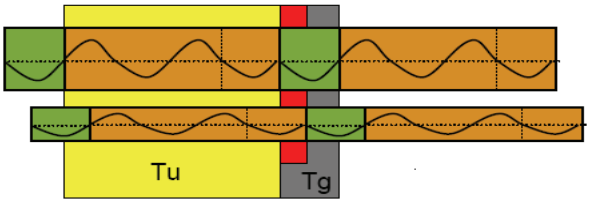
Slika 1. *OFDM modulator*

Jednostavnom transformacijom (Ojlerovom formulom) analitički izraz koji opisuje IFFT (1.2) može dekomponovati u formi jednačina (1.4) , (1.5) na osnovu kojih se može videti da OFDM modulacija predstavlja amplitudno - faznu modulaciju većeg broja prostoperiodičnih nosilaca. Njihova frekvencija korespondira sa vremenskim rasporedom modulišuće komponente.
Mada se za OFDM modulaciju koristi IFFT proračun, na osnovu analititičkih izraza koji opisuju FFT i IFFT (1.1), (1.2) može se uočiti da se razlikuju samo u predznaku eksponenta, što upućuje na to da se po istom principu za OFDM modulaciju može koristiti i FFT.
U opštem slučaju, ako se neka od ovih funkcija koristi u postupku modulacije, potrebno je da se demodulacija na prijemnoj strani zasniva na inverznoj funkciji [2].
Nakon D/A konverzije i NF filtracije produkti Re i Im grane IFFT procesa amplitudno modulišu RF nosioca u kvadraturi. OFDM modulacija omogućava da se simboli serijskog digitalnog niza prenose paralelno. Time se efikasno koristi raspoloživi spektar prenosnog kanala.
Produženo vreme trajanja OFDM simbola omogućava uvođenje zaštitnog intervala koji otklanja uticaj intersimbolske interferencije (ISI).
Zaštitni interval popunjava se cikličnim prefiksom koji predstavlja kopiju završnog dela OFDM simbola. Ciklični prefiks OFDM simbol čini prividno periodičnim tako da se u prijemu autokorelacijom može tačno detektovati početak OFDM simbola[3].
Broj OFDM nosilaca iskazan FFT modom zasniva se na konkretnoj nameni DVB –T mreže. Viši FFT modovi koje karakteriše veći broj nosilaca i duže trajanje OFDM simbola a time i mogući duži zaštitni interval i manji frekvencijski razmak između nosilaca pogodniji su za stabilan prijem i SFN mreže dok su niži modovi usled većeg frekvencijskog razmaka otporniji na efekat Doplerovog pomeraja frekvencije pa time pogodniji za uslove mobilnog prijema pri većim brzinama.
Osetljivost na frekvencijsku i faznu sinronizaciju predstavlja osnovni nedostatak OFDM modulacije. Svaka razdešenost frekvencije ili faze u procesu prenosa i obrade (mešanja) dovodi do gubitka ortogonalnosti odnosno pogrešne demodulacije.
Proširene mogućnosti DVB–T2 standarda (uključujući 2k, 4k i 8k FFT modove DVB–T standarda) u izboru dodatnih 1k, 16k, 32k FFT modova kao i dodatnog N=256 stepena modulacije (N=4, 16, 64, 256, N-QAM) i kodnog količnika obezbeđuju kvalitetan kompromis

bitskog protoka, zaštitnog intervala i nivoa signala u zoni servisa.

2.3 SFN mreže

Zaštitni interval dovoljne dužine koji prethodi odgovarajućem OFDM simbolu obezbeđuje da prethodno emitovan OFDM simbol (usled kašnjenja zbog višestruke refleksije) ne uđe u interval vremena prijema aktuelnog OFDM simbola (slika 3. *Zaštitni interval*, ISI–crveno polje).
Detektovanje tačnog početka OFDM simbola daje mogućnost da se u uslovima odgovarajućeg sinhronizma istovremeno primi aktuelni OFDM simbol iz više različitih izvora (slika 3. *Zaštitni interval*, intrasimbolska interferencija doprinosi prijemnom signalu–žuto polje), što predstavlja osnovu rada SFN mreža.



Slika 2. *Zaštitni interval*

Maksimalna moguća razlika pređenih puteva signala sa najbližeg i najudaljenijeg predajnika SFN mreže određeni su
- dužinom zaštitnog intervala i
- brzinom prostiranja elektromagnetnih talasa.

U praksi, međusobna udaljenost predajnika (separacija) koje prijemnik može da primi treba da obezbedi da ne dođe do ISI. Iz perspektive korisnika (prijemnika) udaljenost najudaljenijeg predajnika čij signal može da primi ne sme da premaši maksimalnu moguću udaljenost koju dozvoljava zaštitni interval:

$$\Delta d = c \cdot T_g \qquad (1.6)$$

Uvođenjem dodatnih viših 16k i 32k FFT modova u DVB–T2 standard proširuju se mogućnosti u domenu dužine i izbora odgovarajućeg zaštitnog intervala u odnosu na DVB–T standard. Modovi 16k i 32k saglasno sa odgovarajućim zaštitnim odnosima povećavaju vrednosti maksimalne razlike putanja signala sa 67,2 km (8k) na 79,8 km, 134,4 km (16k, 32k) i 159,6 km (32k). Vremenski i frekvencijski sinhronizam predajnika SFN mreže zasnivaju se na GPS sistemu kao zajedničkom globalnom izvoru vremenske i frekvencijske reference.
GPS sistem predstavlja stabilan izvor referentnog 1PPS signala (1 Pulse Per Second) vremenske reference i signala frekvencije 10 MHz (referentni takt signal).
Drugu vremensku referencu predstavlja vreme prenosa. Ovaj podatak je statičan, pažljivo proračunat i zasnovan na šemi raspodele predajnika u mreži.
Vreme prenosa svakog predajnika je različito i jedinstveno za svaki predajnik (1.7). Potreban uslov jeste da vreme prenosa ne dostigne maksimalno kašnjenje u mreži da bi SFN mreža korektno radila.

$$T_{TRANS} = (STS + \max\ \_network\_delay) mod 10^7 \qquad (1.7)$$

Vremensku referencu DVB–T2 standarda predstavlja T2 MI paket (Modulator Interface Packet) tipa DVB–T2 timestamp indiciran 8-bitnim poljem packet_type u okviru T2 MI zaglavlja (T2 MI header).
T2 MI paketi prenose se u vidu MPEG–2 TS (Transport Stream–transportni niz) paketa od 188 bajta. Polje second_since_2000 veličine 40 bita sadrži apsolutno vreme emitovanja izraženo ukupnim brojem sekundi u odnosu na trenutak 2000–01–01 T 00:00:00 UTC.
Polje sub_seconds veličine 27 bita iskazuje relativno merenje vremena u odnosu na vreme odnosno vrednost iz polja second_since_2000.
Kada vrednost second_since_2000 nije jednaka 0 vreme emitovanja je:

$$T_E = \text{second\_since\_2000} + \text{sub\_seconds} \cdot T_{SUB} \qquad (1.8)$$

$T_{SUB} = 1/64\ \mu s$ - za kanal širine 8 MHz

Nulta vrednost polja second_since_2000 indicira samo relativno merenje vremena definisano sledećim poljem sub_seconds. Vreme emitovanja (1.8) definisano je u odnosu na graničnu SI (Service Information) sekundu. Granična SI sekunda može biti definisana 1PPS signalom [5].

$$T_E = \text{sub\_seconds} \cdot T_{SUB} \qquad (1.9)$$

Na sve T2 ramove jednog super rama odnosi se ista vrednost vremenske reference.

2.4 Realizacija jedne DVB–T2 mreže predajnika/repetitora

Planom raspodele radio frekvencija GE06 za teritoriju Republike Srbije definisano je 9 zona raspodele frekvencija za VHF opseg, 15 zona raspodele frekvencija za UHF opseg, i u Beogradu po jedna posebna zona raspodele za VHF opseg i UHF opseg.
U VHF zonama raspodele frekvencija definisana je po jedna SFN mreža (frekvencija), dok su u UHF zonama raspodele frekvencija definisane 7–9 SFN mreža (frekvencija) po zoni raspodele. Pošto se planom raspodele radio frekvencija predviđa izuzimanje korišćenja 61. ch–69. ch. efektivno u UHF zonama raspodele ostaje mogućnost za 5 do najviše 8 SFN mreža (u zavisnosti od samih zona raspodele).
Realizacija jedne (prve) DVB–T2 mreže predajnika/ repetitora može se ostvariti u tri faze. Prva faza obuhvata sve matične predajnike mreže (koji su istovremeno matični predajnici analogne mreže).
Realizacijom odgovarajuće RR mreže za prenos signala multipleksnog paketa, emisionih antenskih sistema odgovarajućeg dobitka i dijagrama zračenja, sistema besprekidnog napajanja i instalacija vodenog i/ili vazdušnog hlađenja ostvarili bi se svi preduslovi za puštanje u rad primarne mreže DVB–T predajnika.
Ovom fazom postigao bi se najveći moguć efekat radiodifuznog pokrivanja korisnika DVB–T2 signalom pošto

se lokacije ovih predajnika uglavnom nalaze u blizini najvećih urbanih centara i oblasti najveće koncetracije stanovništva (rečne doline, putni pravci).
Sledeća faza realizacije se sprovodi da bi se izvršilo pokrivanje korisnika koji nisu obuhvaćeni prvom fazom usled nepovoljne konfiguracije terena ili senki koje su posledica strogih zahteva redukovanja dijagrama zračenja nekih predajnika primarne mreže (slabljenje po nekim pravcima dijagrama zračenja može dostići i do 15 dB).
Druga faza realizacije DVB–T mreže predajnika/repetitora zasniva se na postojećoj analognoj mreži repetitorskih objekata. Samo neke od ovih lokacija su uključene u sistem RR mreže za prenos signala multipleksnog paketa (veći urbani centri, lokacije sa kojih se napaja određeni broj repetitora ili lokacije za koje je to jedini siguran način za kvalitetan prijem multipleksnog paketa), dok se sve ostale lokacije napajaju repetitorski.
Dijagrami zračenja emisionih antenskih sistema strogo su definisani prema pravcima nepokrivenih zona servisa, sa što manjim preklapanjem u pokrivenim zonama servisa primarne mreže. Emisioni antenski sistemi su relativno jednostavni i sastoje se od antena usmerenih ka zonama servisa. Pošto predajnici i repetitori ove faze imaju značajno manje snage, zahtevi u domenu besprekidnog napajanja i hlađenja značajno su skromniji.
Treća faza ne mora biti vremenski ograničena pošto predstavlja deo korišćenja i modelovanja mreže prema razvoju konkretne situacije. U okviru ove faze otklanjaju se nedostaci prethodnih faza u domenu loše rešenog pokrivanja određenih delova zone servisa.
Faze realizacije mreže DVB–T2 predajnika/repetitora ne moraju biti strogo ograničene vremenom i prostorom unutar granica projektovanih fazama. Mogu se preklapati u skladu sa prioritetom i ispunjenošću tehničkih uslova svake emisione lokacije. Prioritet i važnost svake lokacije određeni su njenim značajem u domenu veličine pripadajuće zone servisa, uključujući i repetitore koje napaja signalom.
Podrazumevajući da je sprovedeno odgovarajuće planiranje i priprema usaglašenim radom više službi i izvođača radova realizacija jedne DVB–T2 mreže jednog multipleksnog paketa više TV programa sa 15 UHF SFN mreža moguća je za dve godine.
Realizacija sledećih mreža odvija se bržom dinamikom po istom modelu pošto se većim delom koristi ista infrastruktura (isti emisioni antenski sistemi, isti sistemi besprekidnog napajanja i sl.).

3. ZAKLJUČAK

Rast kapaciteta prenosa kao posledica svojstava DVB–T2 standarda i MPEG - 4 formata kombinovan sa više SFN mreža u istoj zoni servisa obezbeđuju raznovrsnost i kvalitet sadržaja uporedljivih sa onim koje obezbeđuju kablovska ili satelitska distribucija.
Loše stanje sistema emisione tehnike i veza i nedovoljna sredstva u najvećoj meri usporavaju realizaciju DVB–T2/SFN mreže u Srbiji.
Moguća rešenja koja nisu optimalna u domenu emisionih antenskih sistema i snaga predajnika primarne mreže mogu značajno produžiti vreme potrebno za realizaciju DVB–T2/SFN mreže.

4. LITERATURA

[1] T. Vieracker, “Planning DVB–T2”, LS telcom White Paper, June 2010.
[2] C. Langton, “Orthogonal Frequency Division Multiplex (OFDM) Tutorial”, www.complextoreal.com, 2004.
[3] W. Fischer, “Digital Video and Audio Broadcasting Technology”, Springer, August 2009.
[4] DVB Document A122, “Frame structure channel coding and modulation for second generation digital terrestrial television broadcasting system (DVB–T2)”, June 2008.
[5] DVB Document A136, “Digital Video Broadcasting (DVB); Modulation Interface (T2–MI) for second generation digital terrestrial television broadcasting system (DVB–T2)”, September 2010.

Kratka biografija:



Nebojša Dragić rođen je u Beogradu 1969. god. Zaposlen je u JP Emisiona tehnika i veze, Beograd, diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva–Telekomunikacije odbranio je 2011.god.

UDK: 621.397.13

DLNA PLEJER ZA OPERATIVNI SISTEM WINDOWS

DLNA PLAYER FOR OPERATING SYSTEM WINDOWS

Borislav Polić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Sadržaj – *Ovaj rad prikazuje jedan od načina portovanja postojećeg DLNA plejera sa operativnog sistema Linux na operativni sistem Windows. Sama programska celina DLNA plejer organizovana je kao skup tri manje celine programskih celina, UPNP, DLNA i pthread bibiloteka, napravljenih za dinamičko linkovanje (.dll), kao i izvršne datoteke DLNA-player.exe, koja je pisana kao MFC aplikacija, dialog zasnovana. Kao jedna celina oni predstavljaju DLNA plejer za Windows*

Abstract – *This paper presents one way of porting the existing DLNA player with the Linux operating system to Windows operating system. DLNA player is organized as a set of three smaller units, UPnP, DLNA and pthread library, designed for dynamic linking (.Dll) and executable files DLNA-player.exe, which is written as dialog based MFC application. Together, they represent a unit called DLNA player for Windows.*

Ključne reči: *DLNA, UPNP, pthread, plejer*

1. UVOD

Sertifikovani DLNA uređaji prepoznatljivi su po svom logu, čiji je izgled prikazan na slici 1.1.



Slika 1.1. *DLNA logo*

Digital Media Player (DMP) uređaji pronađu sadržaj izložen na DMS ili na M-DMS serverima i onda izvlače izabrani sadržaj sa servera i omogućavaju reprodukciju. Neki primeri DMP uređaja uključuju televizore, stereo i kućne bioskope, multimedijalni mobilni telefoni, bežični monitori i konzole za igru [1,2].

Postojeći DLNA plajer koji je osposobljen da radi pod operativnim sistemom Linux, treba da se portuje, tj osposobi za normalan rad pod operativnim sistemom Widows. Sam programski paket sastoji se iz tri manja programska paketa:

- UPNP, pisan u C/C++ programskom jeziku, prilagođen radu na Linux operativnim sistemom,
- DLNA, pisan u C/C++ programskom jeziku, prilagođen radu na Linux operativnim sistemom i
- fslib testa, koji je inače izvršni program.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Vojin Šenk, red. prof.

Kako je projekat je napisan u C/C++ programskom jeziku, prilagođen radu na Linux operativnim sistemom, zadatak je portovati ga za ispravan i pouzdan rad na operativnom sistemu Windows.

Portovanje treba realizovati tako da se sama sintaksa programske implementacije ne menja, nego da ako neke funkcije nisu članice standardnih Windows biblioteka, jednostavno pod istim imenom implementiraju. Ovim postižemo da ista programska implementacija, uključivanjem ili isključivanjem nekih makroa bude funkcionalna na oba operativna sistema.

2. DLNA ARHITEKTURA

Digitalni dom se sastoji od mreže uređaja potrošačke elektronike, personalnih računara, i mobilnih uređaja, koji sarađuju transparentno, pružajući jednostavnu, besprekornu interoperabilnost koji poboljšava i obogaćuje korisnička iskustva. To je komunikacija i kontrolni oslonac za kućne mreže.

Gradivni blokovi potrebni da bi se olakšala ova interoperabilnost su:

- Transparentno povezivanje između uređaja u digitalnom domu
- Jedinstven pristup za otkriće uređaja, konfigurisanje i kontrolu
- Interoperabilni medija formati i prateći protokoli
- Usaglašeno upravljanje medijima i kontraola
- Kompatibilan kvalitet uslužnih mehanizama
- Kompatibilni autentifikacioni i autorizacioni mehanizmi za korisnike i uređaje.

2.1. DLNA proces

Vrlo važno je da krajnji korisnici imaju jasan osećaj za nove mogućnosti i razumevanje karakteristika koji će obezbediti brz unos i širenje proizvoda na tržište.

Tek posle prerade, dokumentovanja i određivanja prioriteta, za ove kritične slučajeve, može se pristupiti kreiranju detaljne tehničke smernice bazirane na priznatim industrijskim standardima.

2.2. Medija format i model saobraćaja

DLNA media format je namenjen da postigne osnovnu mrežnu usaglašenost, dok se podstiču inovacije dalje na media kodek tehnologiju.

DLNA zahtevi za podršku medijskih formata, primenjuju se na medijske sadržaje koji prolaze kroz kućnu mrežu iz DMS ili M-DMS uređaja do DMP ili M-DMP uređaja. Model DLNA media formata (slika 2.1) definiše skup potrebnih media formata i skup opcionih formata za svaku od tri klase media i obe kategorije uređaja: slika, zvuk i video sa zvukom (AV).

Media Formats	Required Formats Set	Optional Formats Set
Imaging	JPEG	GIF, TIFF, PNG
Audio	LPCM (2 channel)	MP3, WMA9, AC-3, AAC, ATRAC3plus
Video	MPEG2	MPEG1, MPEG4, WMV9

Slika 2.1. DLNA media format za kućne uređaje

Uređaji koji su izvor medijskog sadržaja kao i uređaji koji reprodukuju medijski sadržaj preko kućne mreže, moraju da podrže HTTP kao obavezan prenos.

2.3. Otkrivanje usluga i uređaja i njihova kontrola

Upravljanje medijima i kontrola uređajima i programima omogućuje da pronađu, upravljaju i distribuiraju medijski sadržaj kroz stacionarnu kućnu mrežu, ili da ga prenesu na mobilne uređaje.
UPnP Audio/Video (AV) arhitektura omogućava uređajima da prihvate zabavni sadržaj bilo kog formata i preko bilo kog transfer protokola. UPnP AV specifikacija definiše sledeće četiri usluge čiji su domaćini media serveri ili media rendereri:
- Sadržaj imenika
- Rukovodilac veze usluga
- Usluga AV saobraćaja
- Reprodukcija kontrolne usluge

3. UPNP(UNIVERSAL PLUG AND PLAY)

UPNP Forum definiše tehnologiju čiji su ciljevi kućne mreže, mreže sa proksijem (engl. proximity networks) i mreže u malim kompanijama i poslovnim zgradama. Ovim se omogućava saradnja između bilo koja dva sertifikovana UPnP uređaja pod kontrolom bilo kojeg komandnog uređaja na mreži.
UPnP uređaji su prepoznatljivi po svom logu, koji je prikazan na slici 3.1.

Slika 3.1. *UPnP logo*

3.1. Tehnički pregled-Jednostavno povezivanje i uokvirivanje

UpnP Forum tehnologija je široka industrijska inicijativa koja se transformiše i pojednostavljuje umrežavanje. UpnP sertifikovani uređaji besprekorno i trensparentno lociraju jedni druge. UpnP Forum arhitektura uređaja je distribuirana otvorena mreža koja usklađuje postojeće i proširuje internet protokole kao što su TCP/IP (pruža end-to-end povezanost i precizira kako bi podaci trebalo da budu formatirani, adresirani, preneti, usmereni i primljeni na odredište), HTTP i UDP da bi omogućilo jednostavnije umrežavanje, sve u svrhu prenosa podataka između mrežnih uređaja u kući i na poslu.
Ona nudi povezivanje na mrežu pod komandom bilo kakvog kontrolnog uređaja na mreži. UpnP sertifikovani uređaji pružaju sledeće mogućnosti:
- Nezavisnost medija i uređaja
- Platformska nezavisnost
- Tehnologije zasnovane na internetu
- Programska kontrola
- Proširivost.
- Diferenciranje proizvoda.

3.2. Poboljšana podela i kontrola AV sadržaja

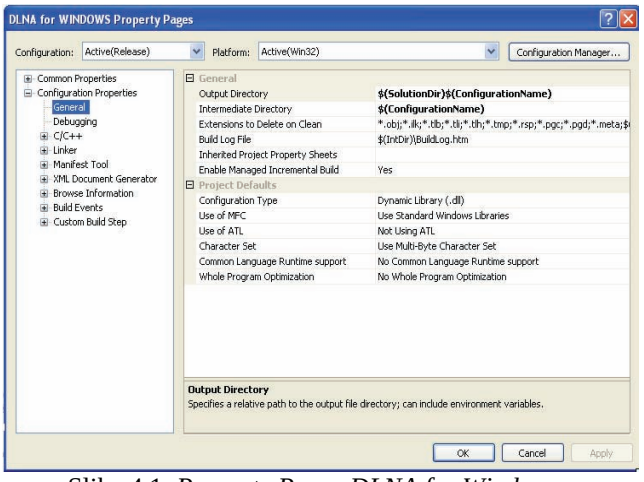
Sertifikovani UPnP AV server i renderer uređaji u ovoj DCP formi predstavljaju jezgro komponenata modernog kućnog sistema sa zabavnim sadržajem. UPnP sertifikovani AV server uređaj obezbeđuje skladištenje, organizovanje i ponudu ovog sadržaja, dok sa druge strene, dok UPnP sertifikovani AV renderer uređaj obezbeđuje konzumiranje (gledanje, slušanje) ovog sadržaja.

4. RAD SA DINAMIČKIM BIBLIOTEKAMA

Programska biblioteka predstavlja skup funkcija i klasa koje su dostupne programeru prilikom izrade programa. Da bi koristio određenu funkciju iz biblioteke, programer ne treba da zna kako je ona realizovana, već samo njen prototip. Zahvaljujući tome, izrada biblioteke je nezavisna od izrade programskog koda. Postoje dva tipa programskih biblioteka: statičke i dinamičke.
Dinamičke biblioteke imaju sve funkcionalnosti kao i statičke. Ekstenzija im je .dll. Osnovna prednost u odnosu na statičke jeste mogućnost da više programa istovremeno koriste jedan .dll fajl.

4.1. Visual studio 2008

Visual Studio 2008 je jedinstven skup alatki koji omogućava programerima i razvojnim timovima da naprave kvalitetne aplikacije na Microsoftovim platformama. Podrška za razvoj operativnog sistema Windows Vista i sistema Microsoft Office pomaže programerima da kreiraju efektne i bogate klijentske aplikacije.
Na slici 4.1 je prikazana vlasnička strana (engl. Property Pages) projekta, gde je sa leve strane odabrana opcija Configuration Properties->General i sa desne strane vrlo bitna stavka pod opcijom Project Default->Configuration Type, gde se uzme u obzir u ovom slučaju Dynamic Library (.dll).



Slika 4.1. *Property Pages DLNA for Windows*

Ovim se omogućuje da je novonastali projekat organizovan kao dinamička biblioteka.

5. PTHREAD BIBLIOTEKA

POSIX(engl. **P**ortable **O**perating **S**ystem **I**nterface for Uni**x**) biblioteka niti je osnovni standarad niti za API(eng. application programming interface) za C/C++. Ovo omogućava novim procesima da se istovremeno izvršavaju, i vrlo je efikasan alat na multiprocesorskim ili

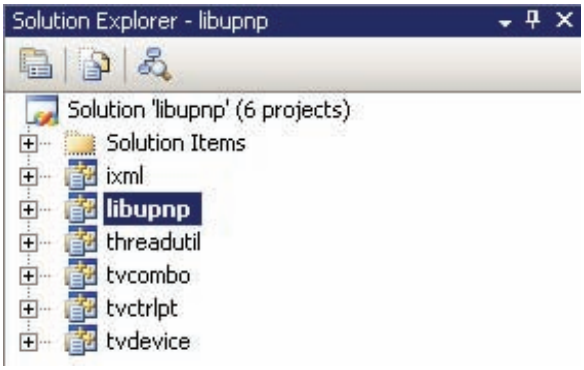
multijezgarnim sistemima, gde tok procesa može biti zakazan za rad na dugom procesoru, i tako se dobija na brzini kroz paralelnu ili distribuiranu obradu. Vrlo je efikasan alat na multiprocesorskim sistemima, isto tako je jako koristan i na jednoprocesorskim, koristeći kašnjenja u I/O i ostalih sistemskih funkcija koje mogu zaustaviti izvršenje procesa.
Sve niti u okviru procesa dele isti adresni prostor. Svrha korišćenja biblioteke niti je prosta: višenitno programiranje omogućuje složenim programima da se mnogo brže izvršavaju nego sekvencijalni programi.
Operacije sa nitima obuhvataju stvaranje niti, njegovo gašenje, sinhronizacije (pridruživanje, blokiranje), planiranje, upravljanja podacima i međusobno dejstvo između procesa.

5.1. Pthread biblioteka za operativni sistem Windows

Jedna od stvari na koje programmer mora da misli prilikom pravljenja programa jeste i portabilnost, ili prenosivost programa. Da bi višenitni program, bildovan na operativnom sistemu Linux bio portabilan, jedan od potrebnih rešenja jeste i pravljenje pthread biblioteke za oprativni sistem Windows, jer nju (pthread biblioteku) u prirodom obliku ne podržava a verovatno neće podržavati ni u budućnosti. U suprotnom, isti program bi se mogao realizovati sa funkcijama koje podržava Windows operativni sistem, ali onda to ima za posledicu da je napravljen program iste funkcionalnosti ali različito kodovan, u ovom slučaju baš zbog razlike u funkcijama koje operišu sa nitima. Pravljenjem pthread biblioteke za Windows operativni sistem, rešava se u dobrom delu ovaj problem, tako da se samo uključivanjemi i isključivanjem nekih makroa, dobija program koji je funkcionalan i na Windows i na Linux operativnom sistemu.
Razni pojedinci su radili nezavisno na implementaciji i dobrom dokumentovanju ovog standardizovanog API-ja niti, ali većina njih ne ugleda svetlost dana. Na sajtu http://sourceware.org/pthreads-win32/, pronađena je pouzdana implementacija pthread biblioteke za Windows operativni sistem. Testirana je na više manjih višenitnih programa i na svim testiranim programima radila je besprekorno, kao i na samom projektu, DLNA plejeru za Windows.

6. UPNP ZA OPERATIVNI SISTEM WINDOWS

Kod velikih i složenih programa, da bi se osigurala, čitljivost programa i olakšalo njegovo razumijevanje i održavanje, modularnost postaje nužna. Modularnost u programiranju je jedan od osnovnih principa.
NA slici 6.1. prikazan je Solution Explorer, koji daje oragnizovan prikaz projekta i njegovih datoteka, kao i lak pristup komandama koje se odnose na njih.
Ako se u Solution Explorer-u klikne na bilo koji +, otvoriće se lista modula (.c fajlova), kao i njima istoimenih zaglavlja (.h fajlova) koji su organizovani po grupama u zavisnosti od njihove funkcionalnosti (postignuta modularnost). Kako je UPnP višenitni program, i ovde je bilo neophodno dodavanje pthread biblioteke.
UPnP projekat je pisan od strane Intel Corporation, a koriste ga svi sa manjim ili većim modifikacijama, a modifikovan je tako da se prilagodi ispravnom radu na Windows XP platformi.



Slika 6.1. *Prikaz modularnosti*

7. DLNA BIBLIOTEKA ZA WINDOWS

Prilikom portovanja DLNA biblioteke sa Operativnog sistema Linux na operativni sistem Windows, javila su se sledeća tri problema:

- Prvi i najlakši, funkcije koje imaju istu namenu, imaju iste parametre, a imaju različito ime.
- Drugi, funkcije koje imaju istu namenu, nemaju iste parameter, i imaju različito ime.
- Treći, funkcije, kojim je kodovan DLNA plejer na operativnom sistemu Linux, ne postoje na operativnom sistemu Windows i moraju se implementirati.

7.1. Određivanje ekstenzije fajla

U DLNA projektu, modulu didl.c definisane su dva konstantna niza, pn2ext_map[] i mime2ext_map[].
U modulu didl.c opisan je i način prepoznavanja ekstenzija, tako što DLNA prvo ispituje mapu mime tipa, pa ako ne uspe, onda se ispituje dlna profil (pn2ext_map).
Funkcije koje operišu sa ovim mapama su char* get_ext_by_pn (const IXML_Node* node) i char* extract_extension (const IXML_Node* node).

7.2. Pretraživanje

Osnove pretraživanja (engl. browse) nalaze se u modulu file_dlna.c, gde je na početku definisana struktura koje predstavlja keš za trenutni direktorijum predmeta (engl. items).(typedef struct browse).
Sama akcija pretrage vrši se pomoći skupa funkcija koje su implementirane u istom modulu, a neke od njih su i:

- char* set_cur_dir (char *URI) – postavlja trenutni direktorijum na prošlu putanju(engl. passed path)
- char* get_cur_dir (char *URI) – vraća apsolutnu putanju trenutnog direktorijuma
- char* get_first (char *URI) – vraća prvi dete predmet(engl. first child item) u trenutnom direktorijumu
- char* get_next (char *URI) – vraća sledeći predmet u trenutnom direktojimum. Ako nema sledećeg, vraća se NULL

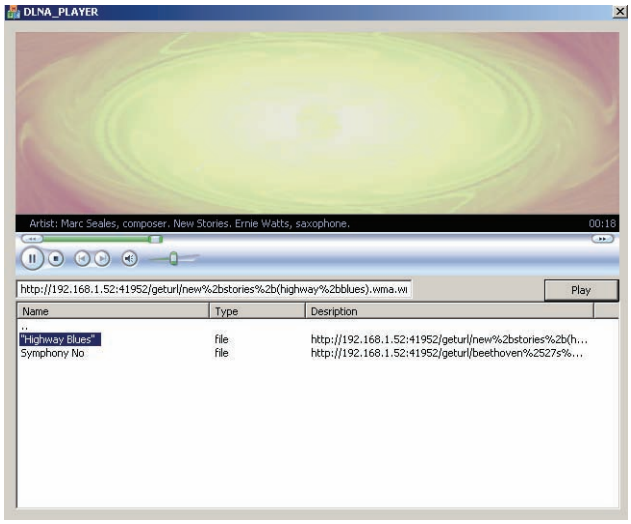
8. REALIZACIJA DLNA PLEJERA

Dialog je pravougaoni prozor čija je glavna uloga da bude domaćin drugim Windows kontrolama. Iz tih razloga, dialog se naziva i kontejner. To je primarna sprega korisnika sa računarom. Sam po sebi, dialog ne znači ništa [3,4,5].

Dialog sadrži sledeće kontrole:

- Edit kontrola: to je kontrola pravougaonog prozora koja se obično koristi u dialogu da korisnik može da unosi i uređuje tekst preko tastature.
- Dugme (engl. button) kontrola: predstavlja standardno komandno dugme koji je renderovano kao HTML ulazni element. Kada korisnik napravi klik na dugme, oblik se dostavlja na server.
- Microsoft Windows Media Player kontrola: je ActiveX kontrola koja se koristi od strane programera za dodavanje mogućnosti multimedijalne reprodukcije na Web stranicama ili aplikacijama.
- List kontrola: obezbeđuje način da organizuje grupu stavki (engl. items) i omogućava korisniku da izabere jednu ili više tih stavki. List kontrola ima labava ograničenja koji tip dete elementa može da sadrži.

Na slici 8.1 prikazan je pokrenut media fajl pronađen na DLNA serveru.



Slika 8.1. *Pronalaženje media datoteke i puštanje u rad*

9. ZAKLJUČAK

U radu je opisan način kako je DLNA plejer implementiran pod operativnim sistemom Linux, portuje, tj osposobi za normalan rad pod operativnim sistemom Widows.

Sam programski paket sastoji se iz tri manja programska paketa: UPNP, pisan u C/C++ programskom jeziku, portovan radi prilagođenja rada na Windows operativnim sistemu, DLNA, pisan u C/C++ programskom jeziku, portovan radi prilagođenja rada na Windows operativnim sistemu i fslib testa, koji je inače izvršni program, takođe pisan u C/C++ programskom jeziku, portovan radi prilagođenja rada na Windows operativnim sistemu, koji su sklopljeni u jednu funkcionalnu celinu.

Kao predlog za moguće unapređenje ovog DLNA plejera jeste dodavanje ostalih DLNA uređaja, kontrolera i renderera u sam projekat.

Može se napraviti modifikacija koda da DLNA plejer (sa dodacima renderera i kontrolera) radi i na Windows mobilnim platformama.

Takođe, može se napraviti ActiveX kontrola DLNA plejera i ostalih DLNA uređaja, sličnu Microsoft Windows Media Player ActiveX kontrolu.

10. LITERATURA

[1] Vladimir Kovačević, Miroslav Popović: *Sistemska programska podrška u realnom vremenu*, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Novi Sad, 2002.

[2] Digital Living Network Alliance: *DLNA Overview and Vision Whitepaper, 2007.*

[3] Contributing Members of the UPnP Forum, *UPnP™ Device Architecture 1.1*, Document Revision Date 15 October 2008

[4] UPnP Forum, *UPnP CERTIFIED TECHNOLOGY-YOUR SIMPLE SOLUTION FOR HOME, OFFICE AND SMALL BUSINES INTEROPERABILITY*

[5] Microsoft, *MSDN Library*

Kratka biografija:



Borislav Polić rođen je u Kikindi 1982. god.. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti elektronike, energetike i telekomunikacija, iz predmeta Televizija i digitalna obrada slike, odbranio je 2011.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.391

NEKI PRAKTIČNI ASPEKTI PRENOSA IPTV SIGNALA

SOME PRACTICAL ASPECTS OF IPTV SIGNAL TRANSMISSION

Karolina Valentović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Rad se prevashodno bavi prenosom IPTV signala preko ADSL2+ sistema, a u okviru toga se razmatraju parametri kao što su margina šuma, INP i interleave kašnjenje, kako njihove teorijske vrednosti, tako i eksperimentalno utvrđene vrednosti.*

Abstract – *Main theme in this paper is about transmission of IPTV signal via ADSL2+, and parameters such as noise margin, INP and interleave delay are analyzed, their theoretical and experimental values.*

Ključne reči: *IPTV, ADSL2+, margina šuma, INP, interleave delay.*

1. UVOD

Internet Protokol Televizija ili kraće IPTV počela je da osvaja svet uz integraciju sa već postojećim telekomunikacionim kablovskim, satelitskim, zemaljskim i drugim tehnologijama i ostalim Internet pristupima za prenos video sadržaja preko IP zasnovanog servisa. IPTV opisuje mehanizam za prenos strima video sadržaja putem mreže koja koristi mrežni Internet protokol.

Zahtevi pri prenosu IPTV-a preko ADSL-a mnogo su veći nego oni koji se odnose na *best-effort* prenos Interneta. Ako pretpostavimo da su upravljanje kapacitetima i QoS inženjering na jezgru mreže adekvatni, tada je obično glavni izvor oštećenja pri TV prenosu ustvari DSL tehnologija i/ili kućne mreže.

Razni šumovi i vremenska promenljivost kanala na bakarnoj parici mogu rezultovati gubitkom paketa i samim tim izazvati video artefakte. Ipak, uticaj šuma i drugih promena pri prenosu signala preko ADSL-a mogu se ublažiti pametnim izborom vrednosti parametara pri formiranju DSL profila.

U nastavku ćemo videti koje su to preporučljive vrednosti određenih parametara koji omogućuju uspešan prenos IPTV signala preko ADSL2+

2. IPTV PREKO ADSL2+

Kao što već znamo, ADSL koristi već postojeću mrežnu infrastruktuju na kojoj postoji i POTS. Međutim, da bi se preneli sofisticiraniji servisi koje prenosi ADSL (koristeći frekvencijske opsege iznad onog koji koristi klasična analogna telefonija), moraju se izvršiti određene adaptacije. Razni šumovi mogu da prouzrokuju greške pri prenosu signala ili čak i da dovedu do prekida komunikacije. Postoje mnoge vrste i klasifikacije šumova, od kojih su najznačajniji:

- preslušavanje,
- aditivni beli Gausov šum,
- radio šum i
- impulsni šum,

a posebno tzv. ponavljajući električni impulsni šum – REIN (*Repetitive Electrical Impulse Noise*).

Kako DSL tehnologija koristi već pomenute bakarne parice, može biti puno izvora šumova na putu gde se one prostiru, što naravno utiče na kvalitet signala koji se prenosi.

Imajući to u vidu, treba predvideti i pokušati izbeći te šumove koji prete da degradiraju transmisiju.

2.1. Teorijska osnova

2.1.1. Pristupi DSL profilu

Pre svega, u radu sa ADSL2+ postoji više mogućnosti kada je u pitanju pristup tom sistemu prenosa. To znači da, bez obzira na podešavanje parametara, moramo da vidimo kakve nam mogućnosti pruža određeni ADSL2+ profil. Postoje tri vrste pristupa ADSL2+ profilu:

- profil sa fiksnom brzinom
- profil sa adaptivnom brzinom bez upotrebe SRA (*Seamless Rate Adaptation*)
- profil sa adaptivnom brzinom sa upotrebom SRA.

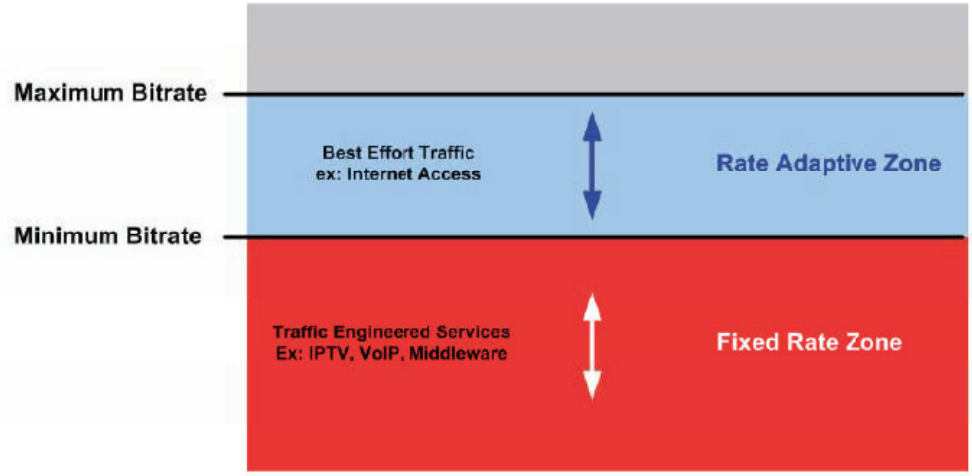
Profili sa fiksnom brzinom podrazumevaju obezbeđivanje neke fiksne, sigurno zagarantovane brzine u prenosu signala. Loša strana ovakvog pristupa jeste da početni uslovi sistema moraju biti dobro podešeni pre bilo kakve ponude određene usluge, dok je dobra strana dobra stabilost sistema.

Profili sa adaptivnom brzinom bez SRA ne fiksiraju brzinu, već u zavisnosti od mogućnosti linije i zahtevane usluge mogu da naznače maksimalnu dostižnu brzinu. Jedan način da se ovo ostvari jeste da se garantuje sigurna brzina prenosa sa mogućnošću da se poveća do određene granice po potrebi. To je prikazano na slici 1.

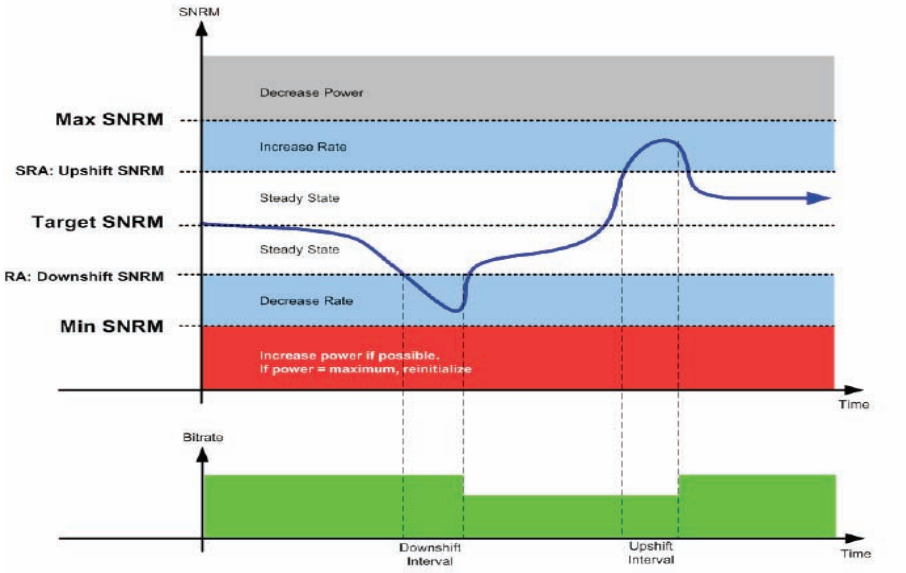
Profili sa adaptivnom brzinom sa upotrebom SRA dozvoljavaju ADSL2+ pristupnim čvorovima da ostvaruju kontinualne (bešavne, bez prekida) promene brzine prenosa podataka da bi se izbegao prekid veze.

SRA dinamički ostvaruje promene u brzini prenosa podataka da bi se prilagodilo promenama šuma na liniji i takođe da bi se sprečilo prekidanje i gubitak veze. Na slici 2 vidimo kako se to ostvaruje detekcijom promene margina.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Vojin Šenk, red.prof.



Slika 1. *Profil sa adaptivnom brzinom*



Slika 2. *Promena brzine prenosa usled promene margina*

2.1.2. Značajni parametri

Parametri koji igraju značajnu ulogu u prenosu IPTV signala putem ADSL2+ sistema su SNR margina, *interleave* kašnjenje i INP parametar.

SNR predstavlja odnos snage korisnog signala i snage šuma. Da bi sistem radio pri određenoj brzini, neophodan je određeni SNR (što je veća brzina, potreban je veći SNR). SNR margina predstavlja razliku između SNR-a na liniji i neophodnog SNR-a za tu brzinu prenosa.

Interleaving je proces skremblovanja određenih sekvenci podataka.

To se radi tako što se uzastopne oštećene sekvence „razbijaju“ na više kodnih reči. Sa jedne strane, *interleaving* obezbeđuje zaštitu od dugih, uzastopnih grešaka, ali istovremeno i unosi kašnjenje.

Dubina *interleave*-a je broj blokova koji se istovremeno skrembluju, odnosno koji se zadržavaju na baferu dok se skremblovanje ne izvrši. Ta zadrška, tj. kašnjenje ima svoju dozvoljenu granicu.

INP (*Impulse Noise Protection*) je broj uzastopnih DMT simbola koji mogu da se oporave nakon bursta impulsnog šuma, a računa se kao:

$$INP = D\,\frac{R}{2}\frac{1}{L/8} \tag{1}$$

gde je D dubina *interleave*-a, R je broj redudantnih bita u kodnoj reči, a L je broj simbola koje DMT kao tehnika linijskog kodovanja obuhvata.

2.2. Preporučene vrednosti parametara

U tabeli koja sledi biće prezentovani opsezi koji predstavljaju neke osnovne vrednosti karakteristične za izvorni ADSL2+ profil za IPTV. Ove preporuke mogu predstavljati polaznu tačku za servisne provajdere koji žele da distribuiraju IPTV preko ADSL2+.

2.3. Praktično izmerene vrednosti parametra

Prvi parametar koji smo analizirali je bila margina šuma (SNR margina). Videli smo da veća SNR margina obezbeđuje bolju zaštitu linije, ali istovremeno smanjuje brzinu i izaziva preslušavanje. Preporučljivo je da se ovaj parametar postavlja oprezno. U našem eksperimentu dužina petlje bila je 2550 m. Primetili smo da SNR margina od 180 dB toliko smanjuje brzinu na liniji, da je skoro nemoguće kvalitetno sprovesti SDTV kanal do korisnika.

Bitrate Attributes	Downstream		Upstream
Maximum	Total bitrate required for all services offered, including overhead.		Total bitrate required for all services offered, including overhead.
Minimum	Minimum bitrate required for IPTV and supporting applications.		Minimum upstream bitrate required for IPTV control signaling.
Margin Attributes	**Downstream**		**Upstream**
Target	**8 dB**		**8 dB**
Maximum	**16 dB**		**16 dB**
Minimum	**0 dB**		**0 dB**
SRA downshift	**<target-min>/2**		**<target-min>/2**
SRA Upshift	**<max-target>/2**		**<max-target>/2**
Error Protection Attributes	**Downstream**		**Upstream**
INP	**2** **(increase to 4 if equipment permits without overly degrading throughput)**		**0.5**
Max-delay	**8 ms**		**2 ms**

Tabela 1: *Preporučene vrednosti parametara za ADSL2+ prenos IPTV-a*

Ako uzimamo vrednosti od 10 dB do 14 dB, to je sasvim dovoljno da se obezbedi adekvatna zaštita, a istovremeno je dovoljno širok propustni opseg za slanje video sadržaja. Preporučljivo je da SNR margina bude 10 dB ili 12 dB kada podešavamo ADSL liniju koja treba da primi SDTV kanal.

Drugi parametar koji smo analizirali jeste *interleave* kašnjenje. Posmatrajući rezultate, zaključili smo da je *interleave* kašnjenje konfiguracije uvek oko 16 ms, bez obzira što smo dopuštali da maksimalno kašnjenje bude do 128 ms. Takođe, videli smo da kada je INP jednako 16 i maksimalno *interleave* kašnjenje takođe 16, linija ne može da se sinhronizuje. Zato je preporučljivo da *interleave* kašnjenje bude 32 iz prethodno navedenih razloga i zbog toga što kad povećavamo target SNR marginu, interleave kašnjenje raste.

Treći parametar koji smo analizirali bio je INP. INP nije radio kako smo očekivali, a on je veoma važan kada treba da projektujemo željenu zaštitu linije. S druge strane, target SNR margina je odigrala značajniju ulogu od INP-a u našem eksperimentu, diktirajući ponašanje *interleave* kašnjenja i INP-a pri postavci. Zaključak je da je opremi bitnija SNR margina nego *interleave* kašnjenja i INP. To se jasno vidi u činjenici da minimalna INP nikad nije bila ispoštovana kada je SNR margina bila fiksirana ili je imala niske vrednosti. INP vrednost konfiguracije je uvek bila 2 DMT simbola. Stoga, postavkom INP vrednosti na 2 bi trebalo da se rešavaju problemi koji se mogu podvesti pod šum. Drugim rečima, za situacije sa manje agresivnim šumom, INP vrednost bi trebalo da bude 2, a za situacije sa agresivnijim šumom 4.

3. ZAKLJUČAK

Cilj rada bio je da se unapredi kvalitet prenosa slike putem ADSL2+. Primetili smo da parametri kao što su INP, SNR margina i i*nterleave* kašnjenje imaju značajnu ulogu u konfigurisanju ADSL linije i ako oni nisu adekvatno podešeni i prilagođeni okruženju, sofisticirani servisi kao što je IPTV se ne mogu na adekvatan način preneti, tj. kvalitet servisa ne može da se unapredi. Naravno, svaka implementacija IPTV-a je drugačija: servisi koji se nude, oblast koja se opslužuje, vrste šumova, karakteristike opreme i mogućnosti provajdera se kombinuju i čine specifično okruženje. Stoga, ovo je bio samo primer kako jedno specifično okruženje reaguje na određena podešavanja.

4. LITERATURA

[1] Gustavo Gentil Barreto Santos, "*Video Streaming Optimization in ADSL Architecture*" , 2007. god

[2] Technical report TR-176 "*ADSL2Plus Configuration Guidelines for IPTV* ", 2008. god

[3] http://www.kitz.co.uk/

Kratka biografija:



Karolina Valentović rođena je u Novom Sadu 1984. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacije, odbranila je 2011.god.



Vojin Šenk rođen je 1958. godine u Beogradu. Od 1960. živi u Novom Sadu, gde se i školovao. Doktorsku disertaciju odbranio je na Elektrotehničkom fakultetu u Beogradu 1992. godine a od 2003. stiče zvanje redovni profesor. Uža naučna oblast mu je Telekomunikacije i Obrada signala.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9

IMPLEMENTACIJA WMS-WFS KLIJENTA ZASNOVANA NA PRISM PATERNU I WPF TEHNOLOGIJI

WPF BASED WMS-WFS CLIENT IMPLEMENTATION WITH PRISM PATTERN

Luka Komatina, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu je opisana izrada WMS-WFS klijenta u WPF tehnologiji sa primenom PRISM paterna. Za implementaciju softvera korišćeno je Microsoft Visual Studio 2010 okruženje sa .NET Framework 3.5 platformom, programski jezik C# i WPF tehnologija.*

Abstract – *This paper describes WMS-WFS client implementation in WPF technology with usage of PRISM pattern. Microsoft Visual Studio 2010 enviroment with .NET Framework 3.5, C# programming language and WPF technology were used for software implementation.*

Ključne reči: *WPF, WMS, WFS, PRISM*

1. UVOD

Cilj ovog rada je kreiranje klijentske aplikacije bazirane na WPF (*Windows Presentation Foundation*) tehnologiji, koja treba da omogući korisniku komunikaciju sa WMS (*Web Map Service*) i WFS (*Web Feature Service*) serverima, radi dobavljanja i rukovanja geoinformacionim podacima. U radu je opisan WMS standard, funkcionalnost WMS servera i strukture podataka koji se razmenjuju tokom klijent-server komunikacije. Takođe je opisan WFS standard, funkcionalnost WFS servera i strukture podataka koji se razmenjuju tokom klijent-server komunikacije. Radi dobijanja modularne aplikacije koja poseduje razdvojene slojeve za prikaz i logiku korišćen je *PRISM* patern.

2. OPIS WMS STANDARDA

WMS dinamički stvara geoprostorne mape iz geografskih podataka. To je internacionalni standard koji definiše mapu kao digitalnu sliku na kojoj su prikazane geografske informacije. WMS [1] produkovane mape su generalno prikazane u PNG (Portable Network Graphics), GIF (Graphics Interchange Format) ili JPEG (Joint Photographic Experts Group) formatu, a ponekad i kao vektor-bazirani grafički elementi u SVG (Scalable Vector Graphics) ili WebCGM (*Web Computer Graphics Metafile*) formatu. WMS operacije mogu biti pozvane iz standardnog web browser-a tako što mu prosledimo pravilno kreiran URL (Uniform Resource Locator). U zavisnosti od sadržaja URL-a poziva se određena informacija. Npr. kad unesemo GetMap zahtev,
u URL-u se može videti koje informacije će biti prikazane na mapi, koji deo Zemlje će biti prikazan na mapi, koji se koordinatni sistem koristi, dimenzije slike u kojoj će mapa biti smeštena. Standard podržava i kompoziciju mapa. Kompozicija se dobija prilikom slanja zahteva za dve ili više mapa sa istim geografskim podacima i izlaznim dimenzijama slike. Korišćenjem GIF ili PNG formata slike, koji poseduju transparentnost, daje nam mogućnost za stvaranje kompozitnih slika. Pojedinačne mape mogu biti zahtevane od različitih servera. WMS standard se oslanja na HTTP (*Hypertext Transfer Protocol*) protokol. Ovaj internacionalni standard definiše tri operacije: *GetCapabilities, GetMap, GetFeatureInfo.*

2.1 Opis *GetCapabilities* zahteva

Svrha *GetCapabilities* zahteva je da isporuči metapodatke o servisu koji su čitljivi kako za mašinu tako i za čoveka. Metapodaci sadrže opis sadržaja servera i dozvoljene vrednosti parametara u zahtevu. Osnovni parametri *GetCapabilities* zahteva su: *format* (opcioni parametar sa kojim određujemo željeni format metapodataka o servisu), *version* (opcioni parametar koji govori koja se verzija koristi), *service* (obavezni parametar koji indicira koji će od dostupnih tipova servisa na određenom serveru biti zatraženi) i *request* (obavezan parametar, koji se koristi se za pozivanje *GetCapabilities* operacije). Odgovor na *GetCapabilities* zahtev je XML dokument koji sadrži metapodatke o servisu**.**

2.2 Opis *GetMap* zahteva

GetMap operacija vraća mapu*. Prilikom prihvatanja GetMap* zahteva*, WMS* servis će ili zadovoljiti zahtev ili vratiti izuzetak. Osnovni parametri *GetMap* zahteva su: *version* (obavezan parametar koji predstavlja željenu verziju servisa), *request* (obavezni parametar kojim se navodi naziv željenog zahteva), *layers* (obavezan parametar koji predstavlja listu mapa-slojeva koji mogu biti vraćeni klijentu prilikom *GetMap* zahteva), *styles* (obavezan parametar koji nam govori u kojem stilu će svaki *layer* biti renderovan), *crs* (željeni koordinatni sistem), *bbox* (uređen par koordinata koji predstavljaju donji levi i gornji desni ugao mape), *format* (obavezni parametar kojim se navodi željeni format mape), *width*, *height* (obavezni parametri koji specificiraju dimenzije mape u pikselima koja treba da bude vraćena), *transparent* (opcioni parametar koji govori da li će pozadina mape biti transparentna ili ne) i *bgcolor* (opcioni parametar *string* tipa koji specificira pozadinsku boju mape).

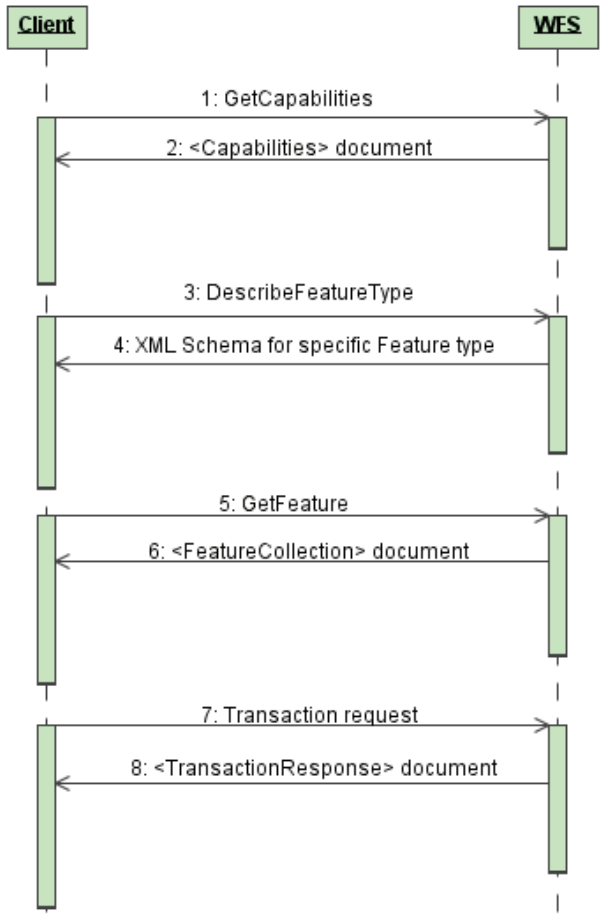
NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Hajduković, red.prof.

2.3 Opis *GetFeatureInfo* zahteva

GetFeatutreInfo je opcion zahtev. Podržan je samo za one *Layer*-e kod kojih atribut *queryable* ima vrednost "1" (*true*). Klijent neće moći da zatraži *GetFeatureInfo* za ostale *Layer*-e. U suprotnom WMS će vratiti izuzetak *(code=OperationNotSuported*). *GetFeatureInfo* operacija je napravljena tako da omogući WMS klijentima dodatne informacije o mapi koja je dobijena prethodnim *GetMap* zahtevom. Osnovni parametri *GetFeatureInfo* zahteva su: *version* (obavezan parametar koji predstavlja željenu verziju servisa), *request* (obavezni parametar kojim se navodi naziv željenog zahteva), *query_layers* (obavezan parametar koji navodi za koje *Layer*–e će biti vraćene informacije), *info_format* (obavezan parametar koji navodi format povratne vrednosti), *feature_count* (opcioni parametar koji navodi maksimalan broj *Feature*–a po *Layer*–u za koje će informacije biti vraćene), *I,J* (obavezni parametri *integer* tipa koji navode željenu tačku na mapi za koju su nam potrebne informacije).

3. OPIS WFS STANDARDA

Web Feature Service (WFS) [2] definiše interfejse za pristup i manipulaciju geografskim podacima preko HTTP protokola. Klijent-server komunikacija je prikazana na slici 1.



Slika 1. - *Pojednostavljen dijagram klijent–**WFS** server komunikacije*

Slično kao što WMS omogućava prikaz mapa sa različitih servera, tako i WFS omogućava korisniku da dobije i ažurira geografske podatke kodirane u GML-u saviše različitih WFS servera.
Klijent-server komunikacija se sastoji iz sledećih koraka:

1. Klijentska aplikacija vrši zahtev za *capabilities* dokumentom nad WFS serverom. U dokumentu je sadržan opis svih operacija koje WFS server podržava i lista svih *feature*-a koje može da vrati.

2. Na osnovu *capabilities* XML fajla klijentska aplikacija pravi zahtev i šalje ga serveru.

3. Server obrađuje zahtev.

4. Po završetku obrade zahteva, server generiše fajl i vraća ga klijentu. U slučaju da je došlo do greške server šalje klijentu izveštaj o grešci.

WMS server podržava sledeće operacije: *GetCapabilities*, *GetFeature*, *DescribeFeatureType*.

3.1 Opis *GetCapabilities* zahteva

Svaki *Open Geospatial Consortium Web Service* (OWS), uključujući i *Web Feature Service*, mora imati sposobnost da opiše svoje mogućnosti tako što će vratiti metapodatke o servisu preko *GetCapabilities* zahteva. Povratna vrednost *GetCapabilities* zahteva je XML fajl u kojem su opisane mogućnosti WFS servisa. Parametri *GetCapabilities* zahteva su dati u tabeli 1.

Tabela 1. - *Parametri GetCapabilities zahteva*

Parametar zahteva	Obavezan parametar?	Opis
Version	jeste	verzija zahteva
Service	nije	tip servisa
Request	jeste	ime zahteva

XML dokument koji je rezultat *GetCapabilities* zahteva sastoji se od sledećih sekcija:

1. *Service Identification section* – obezbeđuje informacije o samom WFS servisu.

2. *Service Provider section* – obezbeđuje metapodatke o organizaciji koja upravlja WFS serverom.

3. *Operation Metadata section* – obezbeđuje metapodatke o operacijama koje su definisane u specifikaciji i implementirane od strane posebnog WFS servera.

4. *Feature Type list section* – definiše listu tipova *feature*-a i operacije za svaki tip *feature*-a koji su omogućeni od strane WFS servisa. Takođe definiše sve koordinatne sisteme koji su podržani od strane WFS servera za svaki *feature*.

3.2 Opis *GetFeature* zahteva

GetFeature operacija omogućava dobavljanje *feature*-a od strane WFS servera. WFS server obrađuje *GetFeature* zahtev i ako *outputFormat* parametar ima vrednost *text/xml; subtype=gml/3.1.1*, vraća se GML instanca dokumenta koja sadrži traženi set podataka. Osnovni parametri *GetFeature* operacije su dati u tabeli 3.

Tabela 2. - *Parametri GetFeature zahteva*

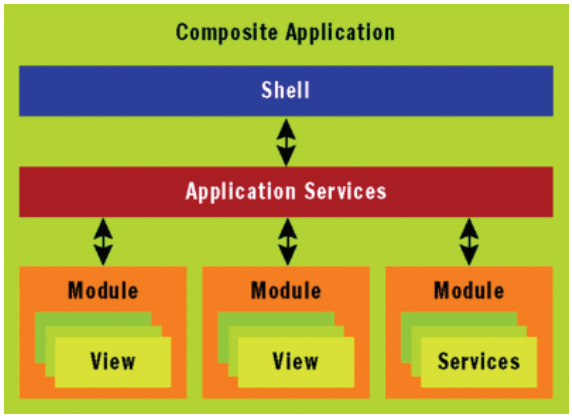
Parametar zahteva	Obavezan parametar?	Opis
Version	jeste	verzija zahteva
Service	jeste	tip servisa
Request	jeste	ime zahteva
TypeName	jeste	Nazivi ***feature***-a koji se traže
MaxFeatures	nije	Maksimalan broj ***feature***-a koji želimo od servera

3.3 Opis *DescribeFeatureType* zahteva

DescribeFeatureType operacija generiše XML šemu koja opisuje *feature*-e koji su ponuđeni od strane WFS servera. Ova XML šema definiše strukturu WFS *feature*-a koja je potrebna prilikom *insert* i *update* zahteva kod ulaza podataka, kao i pri *GetFeature* i *GetGmlObject* zahtevima prilikom vraćanja podataka. *DescribeFeatureType* element sadrži nula ili više *TypeName* elemenata kojim se navodi za koje će *feature*-e WFS server vratiti opis. Ako je sadržaj *DescribeFeatureType* elementa prazan, onda će to biti interpretirano kao zahtev za sve *feature*-e koje WFS server nudi.

4. OPIS PRISM PATERNA

Windows Presentation Foundation (WPF) [5] obezbeđuje programerima da na jednostavan način kreiraju aplikaciju sa naprednim korisničkim interfejsom. Iako ova tehnologija obezbeđuje dobru separaciju između korisničkog interfejsa i programske logike, ona ipak ne rešava problem kreiranja lako održive aplikacije. U manjim projektima na kojima radi manji broj ljudi, održavanje i dalji razvoj ne predstavlja neki veći problem. Međutim kako se aplikacija i broj ljudi koji na njoj rade uvećava, održavanje i razvoj se eksponencijalno usložnjavaju. Kompozitne aplikacije predstavljaju rešenje za ovaj problem. Za kreiranje kompozitnih aplikacija koristimo CAL (*Composite Application Library*, *code-name Prism*) [6]. Kompozitne aplikacije sastoje se od slabo povezanih modula koji se dinamički uključuju u aplikaciju tokom *run time*-a. Moduli sadrže vizualne i nevizualne komponente koje reprezentuju različite delove sistema po vertikali. Vizualne komponente (*views*) su sadržane u zajedničkom *shell*-u koji se ponaša kao domaćin za čitav sadržaj aplikacije. Kompozitnost obezbeđuje servise koji povezuju module zajedno. Struktura kompozitne aplikacije je predstavljenja na slici 2.



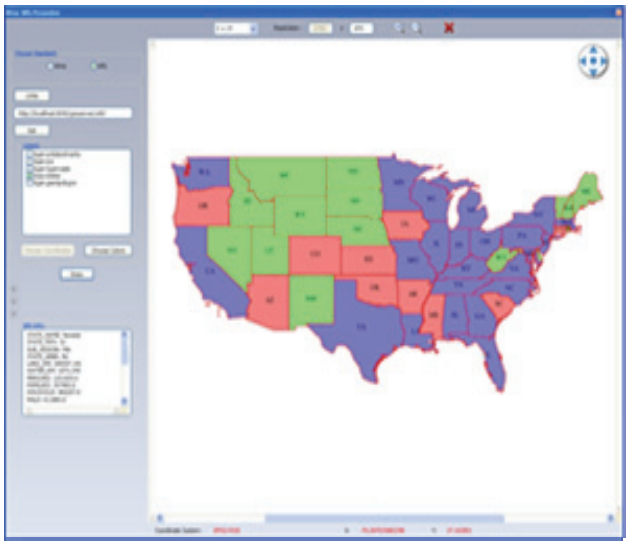
Slika 2. - *Struktura kompozitne aplikacije*

Moduli sadrže *view* iz kojih se kompozicijom dobija kopozitni *view* (koji se još naziva *shell*) tj. celokupan izgled aplikacije. Moduli nikad direktno ne referenciraju jedan drugog, niti direktno referenciraju *shell*. Umesto toga oni koriste servise za komunikaciju između sebe i sa *shell*-om kako bi odgovorili na korisničke zahteve. Kompozitni sistem pruža nekoliko beneficija. Sistem može mnogo lakše da evoluira tokom vremena. Kako se korisnički zahtevi menjaju, novi moduli mogu biti dodati u sistem sa mnogo manje problema nego što bi to bilo u nemodularnom sistemu. Postojeći moduli se mogu nezavisno razvijati i testirati. Na kraju, moduli se mogu razvijati, testirati i održavati od strane različitih timova.

5. OPIS REŠENJA PROBLEMA

Aplikacija *Wms-Wfs Presenter* je zasnovana na *Prism* paternu i sadržana je od sledećih modula: *CanvasModule*, *NavigatorModule*, *StatusBarModule*, *ToolBarModule*, *WfsInfoModule*, *WfsLayerModule*, *WmsLayerModule*, *WmsWfsChooserModule* i *WmsWfsPresenter*. Svrha aplikacije je prikaz geoinformacionih podataka koji se preuzimaju od WMS i WFS servera. Svi moduli se sastoje od *model* i *view* dela. U *model delu* se nalazi funkcionalnost komponente dok se u *view delu* nalazi prikaz komponente u WPF-u. Komponente međusobno komuniciraju preko *CompositePresentationEvent* događaja koji se nalaze *GlobalEvents* klasi u *Infrastructure* paketu. Takođe moduli sadrže i klasu koja služi za prijavljivanje i inicijalizaciju modula. *WmsWfsPresenter* je glavni modul koji objedinjava ostale module. Poseduje *BootStrapper* klasu u kojoj se vrši dodela modula katalogu modula. *WmsWfsPresenter* takođe poseduje i *Shell.xaml* fajl u kojem je definisan okvir aplikacije i regioni u koje se kasnije smeštaju

moduli. *CanvasModule – CanvasView.xaml* sadrži *image* element na kojem se iscrtava slika i *canvas* element na kojem se isrtavaju podaci dobijeni od WFS servera, koji mogu biti tipa: tačka, prava linija i poligon. U *CanvasPresentationModel.cs* klasi se nalazi logika zadužena za iscrtavanje slike, za iscrtavanje WFS *feature*-a, *zoom* na klik, laso *zoom* (zumiranje selekcije), *scroll zoom*. *NavigatorModule* sadrži logiku koja omogućava pomeranje mape i iscrtanih WFS *feature* elemenata na levo, desno, gore, dole. Mapa se pomera klikom na korespodentne strelice. Klikom na centralni „*home*“ kružić vraćamo mapu u *default* stanje. *StatusBarModule* prikazuje koordinatni sistem koji je korisnik odabrao kao i x,y koordinate miša na trenutno iscrtanoj mapi. *ToolBarModule* – poseduje komande za *zoom in* i *zoom out* kao i komandu za brisanje iscrtanih WFS elemenata na kanvasu. Takođe omogućava korisniku da podesi odnos x:y da budu 2:1, 1:2 ili 1:1, kao i da podesi rezoluciju slike. *WfsInfoModule* služi za prikazivanje dodatnih statističkih podataka o iscrtanim WFS elementima. Podaci se upisuju u *text box* na desni klik miša na željeni poligon, pravu ili tačku. *WmsLayerModule* služi za prikaz naziva svih iscrtanih WMS *layer*-a. Nakon iscrtavanja WMS *layer*-a *text* polje se popunjava sa nazivima iscrtanih WMS *layer*-a. *WmsWfsChooserModule* služi za odabir standarda (WMS ili WFS) i za odabir URL-a do željenog servera. Ovaj modul služi i za odabir *layer*-a koje želimo da iscrtamo. On je zadužen i za odabir koordinatnog sistema u kojem će *layer*-i biti iscrtani, kao i za odabir boja u kojima želimo da budu iscrtani WFS *feature*-i. Izgled *Wms-Wfs Presenter* aplikacije je prikazan na slici 3.



Slika 3. - *Izgled Wms-Wfs Presenter aplikacije*

Aplikacija radi tako što se prvo poziva *GetCapabilities* zahtev nad WMS serverom i na osnovu povratnog XML fajla, koji opisuje mogućnosti servera, popunjava lista *layer*-a. Nakon odabira *layer*-a i koordinatnog sistema vrši se iscrtavanje mape. Zatim se poziva *GetCapabilities* zahtev nad WFS serverom i na osnovu njega popunjavalista *feature*-a. Nakon odabira *layer*-a od strane korisnika vrši se njihovo iscrtavanje na kanvasu u željenoj boji.

Na slici 3 je prikazana mapa Sjedinjenih Američkih Država dobijena preko *GetMap* zahteva od WMS servera. Na mapi su iscrtane granice država crvenom bojom. Koordinate za iscrtavanje granica dobijene su od WFS servera. Desnim klikom miša na neku od država dobijamo dodatne podatke o njoj u *WfsInfoModule* modulu, tj. u tekst polju u levom uglu aplikacije. Dodatni podaci o *feature*-u u ovom primeru su npr. naziv države, broj stanovnika, broj muškaraca, broj žena, broj porodica itd. Server kojem se pristupa u ovom primeru je lokalno podignut *GeoServer* koji podržava WMS i WFS standard.

6. ZAKLJUČAK

U ovom radu je opisana geoinformaciona klijentska aplikacija u WPF tehnologiji. Takođe su opisani WMS i WFS standardi koji služe za komunikaciju sa geoinformacionim serverom. Opisan je i *Prism* šablon koji je korišćen radi dobijanja kompozitne strukture. Ova aplikacija korisniku omogućava brzu i laku manipulaciju geoinformacionim podacima baziranim na WMS i WFS standardu. Omogućen je prikaz geografskih mapa kao i iscrtavanje geoinformacionih podataka na njima. Takođe je omogućen i prikaz statističkih podataka o iscrtanim *feature*-ima. Dobra strana ove aplikacije je ta što poseduje mogućnost da komunicira sa bilo kojim serverom na svetu koji poštuje WMS ili WFS standard, kao i jednostavna manipulacija geoinformacionim podacima.

7. LITERATURA

[1] *Open Geospatial Consortium, Inc. : OpenGIS® Web Map Server Implementation Specification*, March 2006.
[2] *Open Geospatial Consortium, Inc. : OpenGIS® Web Feature Service Implementation Specification*
[3] *Apress Pro C Sharp 2008 and the dot NET 3.5 Platform 4th Edition*, Nov 2007.
[4] *Wikipedia, Windows Presentation Foundation*, 2011, Wikipedia
[5] *MSDN Library, Windows Presentation Foundation*, Dec 2010.
[6] *MSDN Library, Microsoft Prism*, Nov 2010

Kratka biografija:



Luka Komatina je rođen 1985. godine u Bačkoj Palanci. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo odbranio je 2011. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9

KOMPARACIJA ŠEMA HEŠIRANJA

HASHING SCHEMES COMPARISON

Andraš Čizik, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu su predstavljeni najčešće korišćeni načini implementacija heš tabela, realizovani u C# programskom jeziku. Analizirane su teorijske karakteristike implementacija. Testirane su performanse najčešće korišćenih operacija nad heš tabelama i merena su memorijska zauzeća heš tabela.*

Abstract – *This paper describes the implementation of different hashing schemes, using C# programming language, analyzes the theoretical characteristics of the implementations and tests the performance of the most common operations on hash tables and measures the memory usage of the hash tables.*

Ključne reči: *heširanje, struktura podataka, performansa heš tabele*

1. UVOD

Heš tabela je struktura podataka, koja u sebi čuva skup ključeva sa pridruženim vrednostima. Omogućuje ***lookup*** operaciju, koja za zadati ključ vraća pridruženu vrednost. Postoje statičke i dinamičke varijante, u zavisnosti od toga, da li je skup ključeva u heš tabeli unapred zadat i ne može da se menja ili ključevi mogu da se dodaju i brišu po potrebi.

Heš tabela koristi heš funkciju, da bi se ključ pretvorio u indeks. Ovaj indeks se dalje koristi za indeksiranje tabele, gde bi trebalo da bude vrednost pridružena ključu. Formalna matematička definicija je:

,,Heš funkcija h je funkcija, koja preslikava elemente iz U na elemente u Z_m, gde je U skup mogućih ključeva, a m je veličina heš tabele. [1]''

$$h : U \rightarrow Z_m$$

U idealnom slučaju heš funkcija bi svaki element iz U jednoznačno preslikavala u element iz Z_m. Naravno ako je $|U| > |Z_m|$ (broj mogućih ključeva je mnogo veći, nego veličina heš tabele), dođe do slučaja da se dva ključa preslikavaju u isti indeks. Ovo se naziva kolizijom i u većini slučajeva ne može se izbeći. Zbog toga heš tabela mora da ima odgovarajuću strategiju za rešavanje kolizije. Broj kolizija direktno zavisi od faktora popunjenosti heš tabele (***load factor***), odnos između n/m, gde je n broj elementa u heš tabeli, a m veličina heš tabele. Kako faktor popunjenosti raste prema 1, verovatnoća kolizije postaje sve veća i veća. Iznad 1 sigurno postoji kolizija u tabeli.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Miroslav Hajduković, red.prof.

Očekivano vreme ***insert***, ***lookup*** i ***delete*** operacije je O(1) [2], konstantno u smislu da ne zavisi direktno od broja elementa u heš tabeli. Vreme izvršavanja operacija zavisi od popunjenosti heš tabele, tačnije od broja kolizija. Da bi održali performansu operacija, često korišćene vrednosti za faktor popunjenosti su od 0,1 do 0,7 [2]. Iznad toga performanse jako opadaju, ispod 0,1 memorijsko zauzeće tabele raste, bez vidljivog poboljšanja performanse. Da bi heš tabela održala faktor popunjenosti između određenih granica, tabela se povremeno povećava ili umanjuje, u zavisnosti od broja elementa u heš tabeli.

Jedna moguća primena heš tabela je unutar ***SmartCache*** komponente, razvijena sa strane ***TelventDMS*** firme. ***SmartCache*** komponenta je ***in-memory*** baza podataka, koja ima mogućnosti za publikovanje promena, replikacije promena između ***hot*** i ***standby*** keša, snimanje sadržaja u ***SQLite*** bazu, itd.. Zbog zahteva za brzo čitanje i pisanje velike količine podataka u ***SmartCache***, treba izabrati strukturu podataka, koja ima odgovarajuću brzinu, ali sa najmanjom memorijskom potrošnjom. Heš tabela predstavlja pravi izbor za takvu komponentu, zbog prosečne O(1) brzine izvršavanja operacija i malog ***overhead***-a, koji se dodaje za svaki element u heš tabeli.

Heš tabela poseduje i neke loše osobine, zbog kojih u nekim slučajevima nije primenljiva:

- Iteracija elemenata u heš tabeli je moguća, ali samo u pseudo-random redosledu, zbog same definicije heš tabele, koja elemente čuva u nesortiranom redosledu.
- Heš tabela omogućuje ***lookup*** za određeni ključ, ali ne dopušta upite tipa manje, veće ili između (***range query***).
- Neke implementacije heš tabela su jako loše za keširanje sa strane procesora (***locality of reference***), zbog pseudo-random načina pristupu heš tabeli, koja prouzrokuje veliki broj promašaja u procesorovom kešu (***cache miss***).
- Iako je očekivano vreme operacija O(1), u slučaju lošeg izbora heš funkcije ili pojave patološke raspodele ključeva, performanse mogu da degradiraju na O(n). Verovatnoća za takvu pojavu je jako mala.

2. HEŠ FUNKCIJA

Heš funkcija preslikava ključ u indeks, koji određuje gde bi element trebao da bude u tabeli. U ovom radu za potrebe ***SmartCache*** komponente posmatrane su heš funkcije koje preslikavaju ***Int32*** u ***Int32***.

Sledeće osobine su očekivane od dobre heš funkcije:

- Uniformnost – dobra heš funkcija trebalo bi da sa jednakom verovatnoćom preslikava moguće vrednosti ključeva u indekse u heš tabeli. Razlog za to je da performanse ***lookup*** operacije jako opadaju, kako se broj kolizije povećava.

- ***Low cost*** – da bi rešenja bazirana na heširanju bili primenljiva, cena računanja heš funkcije mora da bude niska. Korišćenje kompleksne heš funkcije, koja prouzrokuje manji broj kolizija od jednostavne heš funkcije, možda nije isplatljivo, ako se mnogo više vremena potroši na računanje heš funkcije, nego što bi se trošilo na rešavanje kolizije.
- Determinističnost – heš funkcija za zadati ključ uvek mora da vraća isti indeks. Ovaj zahtev isključuje heš funkcije, koje se baziraju na pseudo-random generatoru ili na memorijskoj adresi objekta, pošto to može da se menja prilikom automatskog oslobađanja memorije (***garbage collection***).

Ove osobine moraju da važe samo nad intervalom od *0* do *m-1*, gde je *m* veličina heš tabele. U većini slučajeva za veličinu heš tabele biraju se stepeni dvojke ili prosti brojevi. Takav izbor omogućuje zamenu skupe ***mod*** operacije za računanje indeksa sa ***and*** operacijom, u slučaju ako je veličina tabele stepen dvojke. U ovom radu implementirane heš tabele koriste istu multiplikativnu metodu za računanje indeksa:

$$h(x) = (p * x) \bmod m$$

gde je *m* veličina heš tabela i *p* prost broj. Razlog za korišćenje iste heš funkcije je da prilikom testiranja, to daje isti broj kolizija, što omogućuje upoređivanje strategija za razrešavanje kolizije.

3. REŠAVANJE KOLIZIJE

Dobar izbor heš funkcija, može da smanji broj kolizija, ali nikako ne eliminiše kolizije. Zato heš tabele moraju da imaju svoju šemu za razrešavanje kolizije. U zavisnosti od toga kako rešavaju koliziju, heš tabele mogu biti podeljeni u dve grupe ***closed addressing*** (***open hashing***) i ***open addressing*** (***closed hashing***).

Kod ***closed addressing*** šema, kolizija se rešava van tabele u posebnoj strukturi, koja se zove ***bucket***. Svi elementi, koji imaju isti indeks, smeštaju se u isti ***bucket***. ***Lookup*** operacija nakon računanja indeksa, treba samo da uporedi ključeve unutar ***bucket***-a.

Kod ***open addressing*** šema, elementi se uvek smeste u ***slot***-ove tabele. Heš funkcija određuje samo početnu poziciju elementa. U slučaju ako je početna pozicija zauzeta, kolizija se rešava tako, da se element smesti u neku drugu slobodnu lokaciju.

3.1 *Chained hash table*

Najzastupljenija varijanta heš tabela je ***chained hash table*** (***hash table with separate chaining***). Ova ***closed addressing*** varijanta koristi jednostruko spregnutu listu za ***bucket*** i jednostavne algoritme. Svaki ***slot*** tabele sadrži samo referencu na prvi element liste. Moguće je prvi element liste smestiti u samu tabelu (***list head***). To bi ubrzalo pristup prvom elementu, ali uz trošenje više memorije, naročito, ako su ključ i/ili pridružena vrednost dosta veći od veličine reference. Implementacija ispitivana u ovom radu koristi prvu varijantu sa referencama. Sami elementi (parovi ključ/vrednost) drže se u čvorovima liste (***node***). Čvor liste, osim elementa, sadrži još i referencu na sledeći čvor liste.

Insert algoritam prvo proverava, da li ključ već postoji u heš tabeli i da li je ***load factor*** tabele postigao određenu granicu. U slučaju ako je ***load factor*** postigao granicu, alocira se nova tabela dvostruke veličine u odnosu na staru i svi elementi iz stare tabele prebacuju se u novu tabelu, pre nego što bi novi element bio dodat u tabelu. ***Load factor***-i korišćeni u ovom radu za implementaciju heš tabela su preporučene vrednosti iz literature za datu šemu. Računanjem heš funkcije, određuje se u koji ***bucket*** treba smestiti element. Nakon toga alocira se novi čvor, koji se dodaje na početak liste. Moguće je listu sortirati po rastućem redosledu ključeva, to bi smanjio vreme za ***lookup*** na pola, naravno usporavajući ***insert*** operaciju. Isto tako moguće je prilikom ***lookup*** operacija čvor, koji sadrži traženi ključ, smestiti na početak liste, ako je očekivano da će korisnik heš tabele ponovo pristupiti tom elementu. Implementacija korišćena u ovom radu uvek dodaje element na početak.

Lookup operacija nakon računanja heš funkcije, vrši linearnu pretragu jednostruke spregnute liste, da bi našla traženi ključ. U najgorem slučaju, ako su svi ključevi imali isti indeks i završili u istom ***bucket***-u, performanse će pasti na nivo jednostruke spregnute liste.

Delete operacija je slična ***lookup*** operaciji. Kad je ključ pronađen, čvor se izvezuje iz liste i ostavlja ***garbage collector***-u.

Iako je jednostavna za implementaciju, ***chained hash table*** ima mnogobrojne mane:

- Jednostruko spregnuta lista je loša za keširanje sa strane procesora
- U slučaju snimanje heš tabela na disk, čvorovi mogu da završe u različitim blokovima
- Reference na sledeći čvor troše memoriju, naročito ako je veličina ključa i/ili vrednosti mala.

3.2 *Linear probing*

Najjednostavnija varijanta ***open addressing*** šeme je ***linear probing***. ***Insert*** algoritam, ako je početna pozicija zauzeta, jednostavno nastavlja da traži slobodan ***slot***, povećavajući početni indeks sa konstantom (***step***). Tako dobijena sekvenca ***slot***-ova se naziva probna sekvenca (***probe sequence***). Funkcija za računanje probne sekvence bila bi sledeća:

$$h(x, i) = (h(x) + c * i) \bmod m$$

gde je *h(x)* vrednost početne heš funkcije, *c* je step, *m* je veličina heš tabele, a *i* je *i*-to mesto u probnoj sekvenci. Prilikom pretrage za slobodno mesto u heš tabeli, algoritam mora da poseti sve ***slot***-ove u heš tabeli barem jednom, pre nego što se vrati u početnu poziciju. U suprotnom moglo bi da se desi da algoritam preskoči slobodan slot, iako on postoji u tabeli i ***insert*** bude neuspešan. Ako je $c=1$, to je očevidno zadovoljeno i linearni pristup tabeli biće efikasno keširan. Ako je $c>1$, to je zadovoljeno samo u slučaju ako je c neparan broj i veličina heš tabele stepen dvojke [3].

Lookup operacija bi imala istu probnu sekvencu, kao ***insert***, i pregledala bi ***slot***-ove dok ne nađe traženi ključ ili slobodno mesto. U drugom slučaju traženi ključ ne postoji u heš tabeli. Zato u tabeli uvek mora da postoji barem jedno slobodno mesto, inače operacija se nikad ne bi završila.

Delete operacija bi koristila isti algoritam da nađe traženi ključ. Kad ga nađe, ne može jednostavno da ga izbriše, pošto je moguće da u nastavku probne sekvence postoje još elementi, koji bi jednostavno nestali, ako se ***slot***

označi praznom, jer bi ***lookup*** prerano zaključio da oni ne postoje u tabeli. Da bi ovo izbegli, moguće je uraditi dve stvari, pomerati elemente u probnoj sekvenci iza brisanog elementa za jedan unapred i pregaziti nepotreban element. Druga mogućnost je uvođenje posebne oznake da je element izbrisan (***tombstone***). ***Delete*** operacija bi samo označila da je element izbrisan. ***Insert*** operacija bi tretirala takve ***slot***-ove, kao slobodne i popunjavala ih novim elementima. ***Lookup*** operacija bi jednostavno preskočila takve ***slot***-ove i nastavljala sa pretragom. Implementacija u ovom radu koristi način sa oznakama za izbrisani element.

Parovi ključ/vrednost su organizovani u jedan niz struktura. Bilo bi moguće dodati još jedan atribut u strukturu za označavanje, da li je neki ***slot*** trenutno slobodan ili je u njemu bio element, koji je izbrisan, ali zbog štednje memorije u ovom radu pomenuti pristupe nije korišćen. Pošto implementacija u ovom radu koristi ključeve koji su tipa ***Int32***, rezervisane su dve vrednosti za označavanje praznog ***slot***-a i ***tombstone***-a. Vrednost 0 će značiti da je ***slot*** prazan, -1 će označavati ***tombstone***. U slučaju, ako korisnik heš tabele proba da dodaje ključeve 0 ili -1, vrednost pridružena tim ključevima neće biti smeštena u heš tabelu, nego u posebnim atributima u klasi heš tabele. Sve operacije nad heš tabelom će prvo proveriti, da li je ključ 0 ili -1. Ako jeste, uopšte neće raditi sa heš tabelom, nego će samo pristupiti atributima u klasi, gde su smeštene vrednosti pridružene tim ključevima.

Poželjno je s vremena na vreme izbaciti ***tombstone***-e iz tabele, da probne sekvence ne bi postale predugačke. Zato se, prilikom povećavanja heš tabele, kad ukupan broj elementa i ***tombstone***-a preraste određenu granicu, izbacuju ***tombstone***-i. Isto tako oni se izbacuju i u slučaju, ako broj ***tombstone***-a preraste ukupan broj upisanih elemenata u tabeli.

Linear probing je jednostavna za implementaciju i efikasna za keširanje (ako je ***step*** = 1), ali ima jako veliku manu, koja se naziva primarna klasterizacija (***primary clusterning***). Ona se javlja u slučaju, ako heš funkcija vraća indekse, koji su međusobno blizu. Tako upisani elementi će napraviti klaster u heš tabeli i proizvoditi dugačke probne sekvence. Svaki novi upis u tabelu će povećati klaster i dodatno degradirati performanse.

3.3 *Quadratic probing*

Da bi izbegli primarnu klasterizaciju, za različite početne pozicije treba generisati različite probne sekvence. Jedan način za to je ***quadratic probing***, koji koristi rezultate polinoma drugog stepena za računanje probne sekvence. Funkcija je sledeća:

$$h(x, i) = (h(x) + c_1 * i + c_2 * i_2) \bmod m,\ c_2 \neq 0$$

gde je $h(x)$ vrednost početne heš funkcije, m je veličina heš tabele, a i je i-to mesto u probnoj sekvenci. Često korišćene vrednosti za konstante su $c_1 = c_2 = ½$, $c_1 = c_2 = 1$, i $c_1 = 0$, $c_2 = 1$. Heš tabela implementirana u ovom radu koristi konstante $c_1 = c_2 = ½$ iz dva razloga. Prvo, ako je veličina tabele stepen dvojke i $c_1 = c_2 = ½$, algoritam će posetiti svaki ***slot*** tačno jednom, pre nego što bi se vratio na početnu poziciju [3]. Drugi razlog je da se adrese u takvoj sekvenci uvek povećavaju sa *1, 2, 3.. m-1* [2]. To omogućuje optimizaciju računanja probne sekvence na sledeći način:

$$h(x, 0) = h(x) \bmod m$$
$$h(x, i) = (h(x, i - 1) + i) \bmod m$$

Quadratic probing u nekoj meri može efikasno da se kešira, naravno ne toliko kao ***linear probing***. Kod ***quadratic probing***-a postoji drugi problem, sekundarna klasterizacija (***secondary clustering***), kad elementi sa istom početnom pozicijom imaju istu probnu sekvencu.

3.4 *Double hashing*

Problem sekundarne klasterizacije nastaje zbog toga što probna sekvenca jedino zavisi od početne pozicije i ne uzima u obzir ključ, koji se dodaje. Jedno rešenje za takav problem je korišćenje ***linear probing*** šeme sa konstantnim ***step***-om, ali koji se računa iz ključa koristeći drugu heš funkciju h_2. Ovakav način rešavanja kolizije se naziva ***double hashing***, zbog korišćenja dve heš funkcije. Funkcija za računanje probne sekvence bila bi sledeća:

$$h(x, i) = (h_1(x) + h_2(x) * i) \bmod m$$

gde je $h_1(x)$ vrednost početne heš funkcije, $h_2(x)$ vrednost druge heš funkcije, m je veličina heš tabele, a i je i-to mesto u probnoj sekvenci. Funkcije h_1 i h_2 treba da budu međusobno nezavisne, inače bi moglo da se desi da se dobije ista probna sekvenca za istu početnu poziciju. Implementacija u ovom radu jednostavno koristi više bitove heš funkcije, koje se ne koriste u h_1, za računanje h_2. Da bi obezbedio da probna sekvenca poseti sve ***slot***-ove pre nego što se vrati na početnu poziciju, funkcija h_2 mora da vrati neparan broj u slučaju, ako je veličina heš tabele stepen dvojke [3].

Mana ***double hashing*** šeme se javlja u slučaju, ako je $h_2 >> 1$. Tada će probna sekvenca sadržavati ***slot***-ove, koji su toliko udaljeni, da neće moći koristiti prednosti keša.

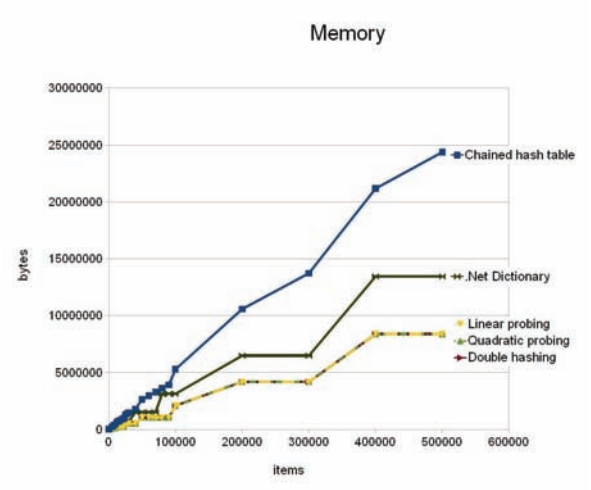
4. POSTOJEĆE HEŠ TABELE U *.NET BASE CLASS LIBRARY (BCL)*

.NET framework već sadrži dve implementacije heš tabela, ***System.Collections.Hashtable*** i ***System.Collections.Generic.Dictionary***. ***Hashtable*** klasa koristi ***double hashing*** strategiju za rešavanje kolizije, dok ***Dictionary*** koristi hibridno rešenje između ***open addressing*** i ***chained hashing*** [4]. ***Hashtable*** klasa je bila deo ***.NET framework***-a pre pojave generičkih tipova i u sebi može držati samo objekte. ***Dictionary*** je generička klasa i omogućuje definisanje tipa ključa i tipa vrednosti.

5. TESTIRANJE PERFORMANSE

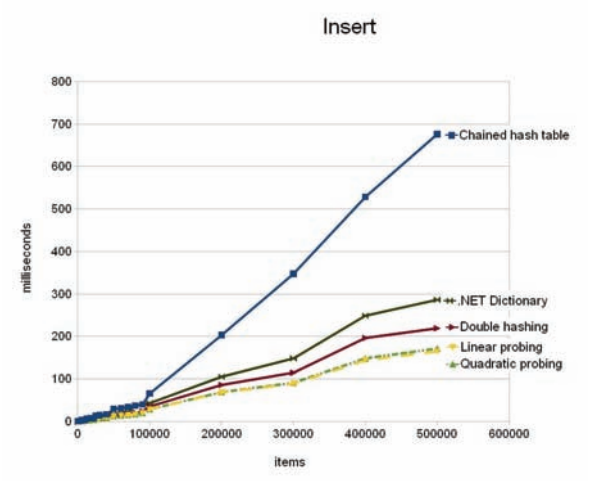
Za testiranje brzine izvršavanja ***insert***, ***lookup*** i ***delete*** operacija korišćeno je ***System.Diagnostics.Stopwatch***. Testovi su pokrenuti više puta i kao rezultat uzeto je prosečno vreme izvršavanja. Za merenje zauzeća memorije korišćen je ***CLRProfiler*** alat. U nastavku slede rezultati testova. ***Hashtable*** klasa iz ***BCL***-a nije prikazana, pošto je zbog ***boxing***-a imala jako loše performanse.

Na slici 1 je prikazano memorijsko zauzeće implementacija. ***Chained hash table*** zbog korišćenja jednostruke spregnute liste zauzima najviše memorije.



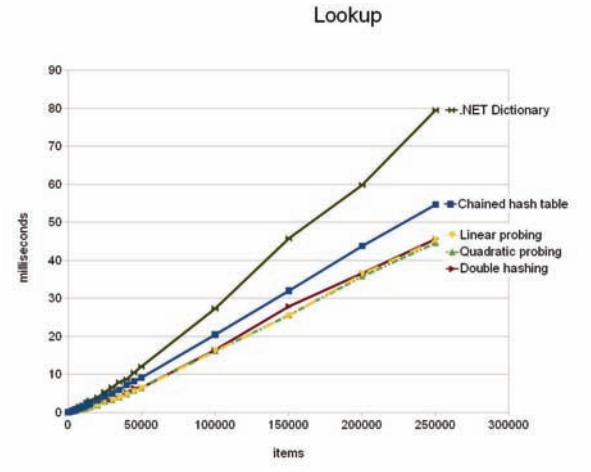
Slika 1. *Memorijsko zauzeće*

Na slici 2 je prikazana performansa ***insert*** operacija. ***Chained hash table*** je jedina, koja alocira memoriju prilikom ***insert***-a, pa je zato najsporija.



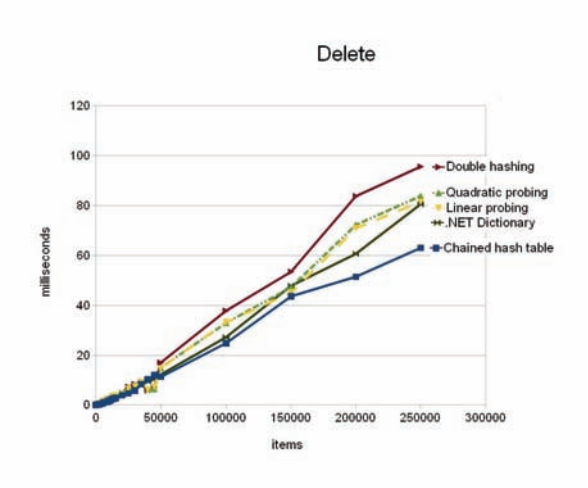
Slika 2. *Performansa insert operacija*

Na slici 3 je prikazana performansa ***lookup*** operacija. Ovde se vide prednosti implementacija, koje mogu efikasno da se keširaju u odnosu na ***chained hash table***.



Slika 3. *Performansa lookup operacija*

Na slici 4 je prikazana performansa ***delete*** operacija. Varijante koje s vremena na vreme izbacuju ***tombstone***-e iz heš tabele (***linear probing***, ***quadratic probing*** i ***double hashing***) su sporije od ***chained hash table***, koja jednostavno izvezuje čvor iz liste i ostavlja ga ***garbage collector***-u.



Slika 4. *Performansa delete operacija*

6. ZAKLJUČAK

U ovom radu su implementirane i testirane razne varijante heš tabela. Upoređene su performanse sa već postojećom heš tabelom u ***BCL***. Pokazano je da je za specifične slučajeve, poput SmartCache komponente, moguće napraviti bolja rešenja.

Sledeći korak u razvoju heš tabela za ***SmartCache*** komponentu bi bio izbor odgovarajuće heš funkcije. Moguće je heš funkciju izabrati na osnovu testova sa različitim distribucijama ključeva, ali najbolje bi bilo izbor doneti na osnovu testova, koje koriste ključeve iz realnog sistema.

Load factor-i korišćeni u implementacijama su uzeti iz literature, kao preporučene vrednosti za datu šemu heširanja. Pravi izbor bio bi da se heš tabele testiraju sa ključevima iz realnog sistema, i da se ***load factor***-i određuju na osnovu broja kolizija, koje se javljaju prilikom heširanja.

7. LITERATURA

[1] ***Advanced Data Structures***
http://courses.csail.mit.edu/6.851/

[2] ***Wikipedia artikles about the hash table***
http://en.wikipedia.org/wiki/Hash_table

[3] ***Virginia Tech Algorithm Visualization Research Group Website***
http://research.cs.vt.edu/AVresearch/hashing/

[4] ***An Extensive Examination of Data Structures Using C# 2.0***
http://msdn.microsoft.com/en-us/library/ms379571(VS.80).aspx

Kratka biografija:



Andraš Čizik rođen je u Senti 1986. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2011.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

AUTOMATIZACIJA RADA PEĆI NA BIOMASU PRIMENOM FUZZY LOGIKE

AUTOMATION OF BIOMASS FURNACES USING FUZZY LOGIC

Siniša Knežević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Predmet istraživanja u ovom radu je napredan princip upravljanja peći na biomasu. Peć je automatizovana primenom fuzzy logike. Simulacija rada sistema je prikazana u Matlab programskom paketu.*

Abstract – *The research presented in this paper is the advanced principle of control of furnaces on biomass. The furnace is automated using fuzzy logic. Simulation of the system is shown in the Matlab software package.*

Ključne reči: *Pelet biomase, Peć na biomasu, Fuzzy logika, Napredno upravljanje, Simulink.*

1. UVOD

Grejanje na biogena goriva postala je danas savremena i prema budućnosti orijentisana alternativa. Jednostavno opsluživanje, automatska regulacija, poluautomatsko punjenje i skladištenje goriva bez problema, odlikuju grejna postrojenja na biomasu poslednje generacije. Time ona nude komfor u opsluživanju i grejanju, koji nimalo ne zaostaje za onim kao kod savremenih načina grejanja na ulje ili gas. Vrednosti emisije dima grejnih postrojenja na biomasu su zahvaljujući naprednoj tehnologiji ispod svih relevantnih granica. Mala emisija produkta sagorevanja i primena obnovljivog goriva, drveta, su jaki argumenti za ekološki grejni pogon.

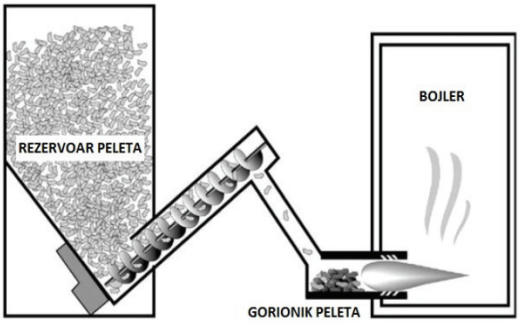
2. PEĆ ZA SAGOREVANJE PELETA

Osnovna funkcija peći je zagrevanje određene prostorije na željenu temperaturu. Peć kao gorivo koristi pelet. Pelet se stvara od presovane piljevine koja predstavlja nusprodukt pilana i drugih transformacionih aktivnosti [1]. Drugi način za dobijanja peleta je direktnim drobljenjem većih komada drveta putem masovnih mlinova. Pelet kuglice su vrlo guste i mogu biti proizvedene sa niskim sadržajem vlage (ispod 10%), što im omogućava da gore na vrlo visokoj temperaturi uz povećanu energetsku efikasnost. Prednost peleta kao biogoriva se ogleda i u tome što kao produkt sagorevanja ne ostavlja veće količine pepela. Količina pepela koja nastaje se može uporediti sa prašinom koja kroz dimnjak izlazi iz peći. Gledano sa ekološkog aspekta, zbog potpunijeg sagorevanja i većeg stepena iskorišćenja peleta ovaj sistem ima smanjenu emisiju štetnih gasova u odnosu na sagorevanje drugih čvrstih goriva, na primer drveta ili uglja [2].

Geometrija peleta omogućava da se lako upotrebljava za automatsko hranjenje peći pomoću transportera ili pneumatske trake.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je dr Velimir Čongradac, doc.

Peć za sagorevanje peleta, koja se koristi u ovom radu, se sastoji od: kotla sa vodenim cevima, gorionika, rezervoara peleta, trensportera peleta i pratećom automatizacionom opremom. Sam kotao je veoma sličan standardnom kotlu koji se koristi za grejanje na drva ili brikete. Najbitniji element ovog sistema je gorionik. Elementi gorionika su: pužni centrifugalni ventilator, elektrogrejač i prostor predviđen za loženje peleta. Rezervoar peleta treba da se nalazi u neposrednoj blizini peći, odnosno na dovoljnom rastojanju da transporter može nesmetano da doprema pelet do gorionika. Primer kompletnog sistema kojim treba upravljati je prikazan na slici 1[3].



Slika 1. *Peć na biomasu*

3. FUZZY LOGIKA

Cilj rada je da se izvrši napredno upravljanje gore pomenutog sistema, na takav način da se dobije optimalna vrednost odstupanja željene temperature od trenutne vrednosti temperature. U okviru naprednog upravljanja koristi se fuzzy logika[4].

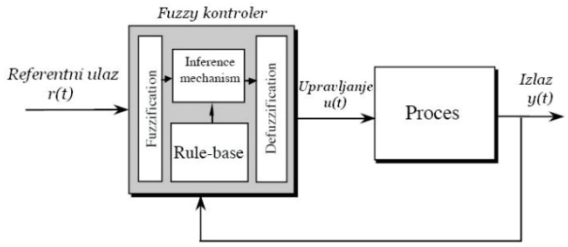
3.1. Fuzzy upravljanje

Fuzzy upravljanje obezbeđuje formalnu metodologiju za predstavljanje, manipulaciju i implementaciju ljudskog heurističkog predznanja o tome kako kontrolisati jedan, određeni sistem. Ovo ne isključuje razvoj modela procesa jer nam je ovaj model u svakom slučaju potreban za detaljniju simulaciju kao i za ispitivanje u cilju dobijanja zadovoljavajućih performansi, stabilnosti sistema, kao i za ispitivanje krajnih ograničenja samog dizajna. Cilj fuzzy pristupa je da umesto jezikom matematike pokušamo da što bolje rešimo problem upravljanja sistemom implementirajući inženjersko iskustvo, o procesu, u sam algoritam kontrolera.

3.2. Struktura fuzzy kontrolera

U osnovi na fuzzy kontroler treba gledati kao na veštačkog donosioca odluke koji radi u sistemu sa zatvorenom spregom u realnom vremenu. On „sakuplja“

podatke izlaza procesa, upoređuje ih sa referencom i onda na način svojstven fuzzy logici „odlučuje“ šta u tom trenutku da bude ulaz procesa, i to tako da se zadovolje željene performanse i zadati ciljevi specifikacije.
Da bi projektovali jedan fuzzy kontroler, potrebno nam je pre svega određeno predznanje o ponašanju procesa. Heurističke informacije o tome kako najbolje upravljati nekim procesom najčešće se dobijaju od operatera koji ima dovoljno iskustvenih podataka o tome kako na najbolji način upravljati procesom.
Ova pravila nam govore kakav bi trebalo da bude ulaz procesa ako se njegov izlaz ponaša na neki određeni način. Osnovna struktura fuzzy kontrolera prikazana je na slici 2.



Slika 2. *Struktura fuzzy kontrolera*

4. SIMULACIJA UPRAVLJANJA

Pre početka projektovanja fuzzy regulatora potrebno je prvo definisati i usvojiti logiku rada samog regulatora. Princip primene fuzzy regulacije na zadati sistem se zasniva na merenju razlike između trenutne temperature i željene temperature. Razlika koja se dobije, izražena u celziusima, se upoređuje sa razlikom koja je izmerena u prethodnom ciklusu. Na osnovu poređenja trenutnog odstupanja merene temeperature i željene, i odstupanja u prethodnom merenju, formira se look-up tabela kontrolne logike (Tabela 1)[5]. Vrednosti izražene u tabeli 1 predstavljaju procentualne vrednosti po kojima frekventni regulator upravlja transporterom peleta.

Tabela 1. *Fuzzy regulacija za transporter peleta*

Tren.M./Pret.M.	-4	-3	-2	-1	0	1	2	3	4
-4	100	100	100	100	100	100	100	100	100
-3	80	100	100	100	100	100	100	100	100
-2	60	60	80	80	80	80	100	100	100
-1	40	40	60	60	80	80	80	80	80
0	0	0	0	0	0	0	40	40	60
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0

Da bi se proverila ispravnost logike rada regulatora potrebno je izvršiti njegovu simulaciju rada na modelu sistema, pre njegove implementacije u realan sistem. Model sistema predstavljen u Matlab-ovom Simulink-u prikazan je na slici 3[6]. Matematički model grejanja:

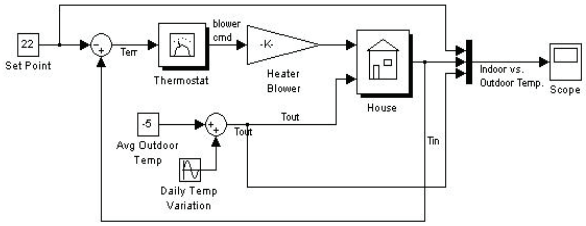
$$dQ_{heater} / dt = Mdot * c \quad (1)$$

Gde je: dQ_{heater}/dt - toplotni protok od peći ka prostoru grejanja, c - grejni kapacitet, $Mdot$ – količina protoka vode kroz potrošač (radiator). Model prostora koji se zagreva je opisan jednačinom:

$$dQ_{losses} / dt = (T_{room} - T_{out}) / R_{eq} \quad (2)$$

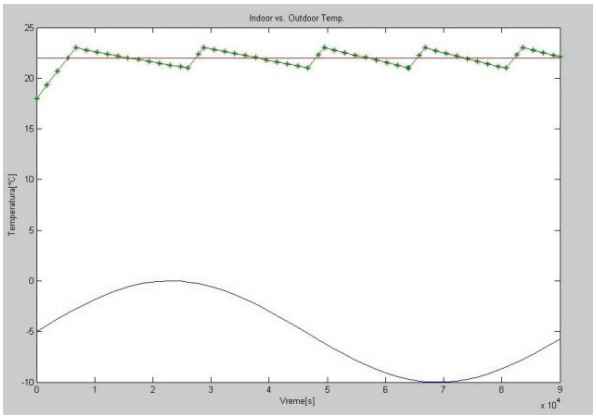
$$dT_{room} / dt = (dQ_{heater} / dt - dQ_{losses} / dt) / (M_{air} * c) \quad (3)$$

Gde je: M_{air} – količina vazduha unutar prostora grejanja, R_{eq} – termalna izolacija kuće.

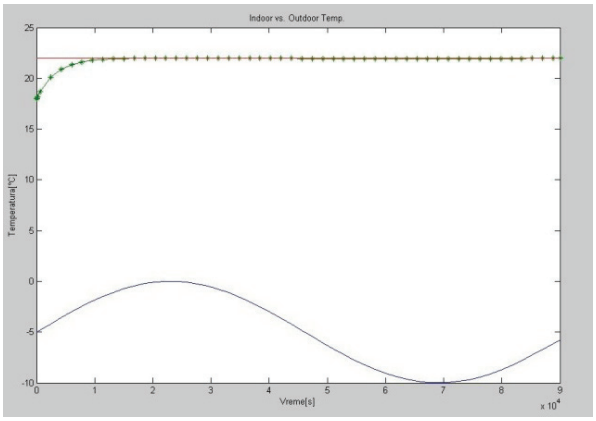


Slika 3. *Simulacioni model sistema za grejanje*

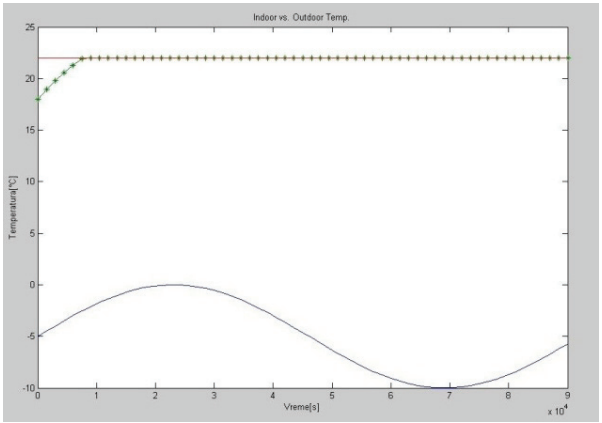
Radi poređenja rezultata simulacije prikazani su rezultati za on/off tip regulacije (Slika 4), PID tip regulacije (Slika 5) i na kraju fuzzy tip regulacije (Slika 6) [7].



Slika 4. *On/Off regulacija*



Slika 5. *PID regulacija*

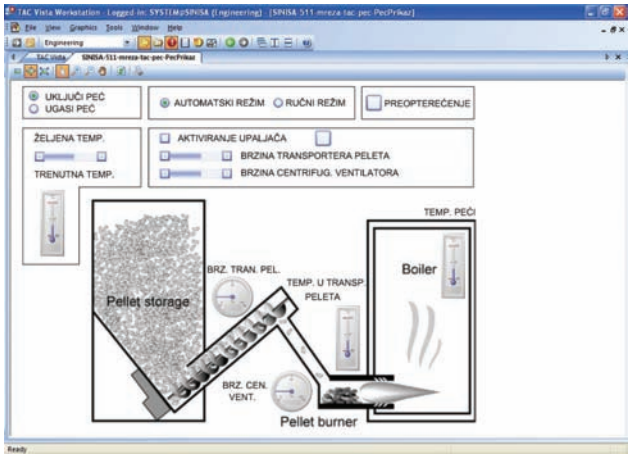


Slika 6. *Fuzzy regulacija*

Na grafiku simulacije žuta linija predstavlja željenu temperaturu, ljubičasta linija predstavlja trenutnu temperaturu i zelena linija predstavlja trenutnu temperaturu van objekta koji se zagreva. Rezultati dobijeni za fuzzy regulaciju i za PID regulaciju su gotovo identični [8][9]. Međutim, fuzzy logika je mnogo jednostavnija za primenu i razumevanje prilikom projektovanja regulatora za sistem peći na biomasu, a rezultat upravljanja je gotovo identičan sa PID tipom regulacije**.**

5. PROGRAMIRANJE PLC-A

Programiranje PLC-a je izvršeno primenom function block diagram-a. Centralna logika rada nalazi se u bloku pod nazivom Fuzzy blok. Vrednosti blokova unutar Fuzzy bloka su unapred definisane vrednosti iz look-up tabele (Tabela 1). Kompletna kontrola rada peći je zasnovana na upravljanju preko centralnog računara. Interfejs sistema predstavlja jedan od najbitnijih elemanata za korisnika sistema i njegovim posredstvom korisnik vrši kontrolu i nadzor peći na biomasu. Izgled interfejsa je prikazan na slici 7.



Slika 7. *Korisnički interfejs*

6. ZAKLJUČAK

Automatizacija danas zauzima veoma bitno mesto u tehnici i nauci [10]. Korišćenjem automatike dobijamo višestruke i značajne prednosti: Povećana je efikasnost u pogledu uštede energije, smanjenja troškova rada, i što je najvažnije u pogledu kvaliteta upravljanja sistemom. Predmet istraživanja u ovom radu je peć na biomasu. Peć kao gorivo koristi pelet, i koristi se za grejanje prostora u kom se nalazi termometar. Da bi se našao najjednostavniji i najefikasniji način regulacije izvršena je analiza i simulacija rada sistema sa više tipova upravljanja. Fuzzy logika je dala najbolje rezultate u pogledu jednostavnosti dok je kvalitet upravljanja fuzzy tipom regulacije identičan upravljanju sa PID tipom regulacije.

Iz ovoga proizilazi da upravo fuzzy tip regulacije omogućava optimalno upravljanje peći na biomasu. Primenom kontinualnog tipa upravljanja dobijamo umereno doziranje peleta uz čega se dobija kvalitetno praćenje željene temperature bez pojave oscilacija.

7. LITERATURA

[1] I. Obernberger, G. Thek, "*The Pellet Handbook*", Earthscan 2010, ISBN 978-1844076314.

[2] B. Wahlund, J. Yin, M. Westermark, "*A total energy of fuel upgrading by drying biomass feedstock for cogeneration*", Biomass and Bioenergy, vol. 23, page 271-281, 2002.

[3] L.S. Johansson, C. Tullin, B. Leckner, P. Sjovall, "*Particle emissions from biomass combustion in small combustors*", Biomass and Bioenergy, vol. 25, page 435-446, 2003.

[4] J. Yan, M. Ryan, J. Power, "*Using Fuzzy Logic*", Towards intelligent systems, 1994.

[5] Mr Branislav Karadžić, Dr Mirko Babić, "*Izbor koncepta fazi kontrolera za upravljanje mikroklimom*", Pregledni rad, strane 132-137, 2005.

[6] S.N. Sivanandam, S. Sumathi, S.N. Deep, "*Introduction to Fuzzy Logic using MATLAB*", 2010, ISBN 978-3542071447.

[7] Alexander Gegov, "*MATLAB & Simulink Based Books, Neural Networks and Fuzzy Logic*", University of Portsmouth, 2010, ISBN 978-3642155994.

[8] H. Bentarzi, R.A. Chentir, N.E. Mastorakis, "*A New Method Controlling Boiler of Thermal Power Plant Using Fuzzy Logic*", page 31-34, ISBN 978-9604740888.

[9] B. Zupančič, I. Škrjanc, A. Krainer, M. Atanasijecič-Kunc, "*Simulation and control system desing of thermal conditions in building using active and passive resources*", University of Ljubljana.

[10] Rohner Peter, "*Automation with programmable Logic Controllers*", 1996, ISBN 978-0868402871.

Kratka biografija:



Siniša Knežević rođen je u Novom Kneževcu 1986. god. Diplomski-bachelor rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Automatsko upravljanje odbranio je 2010.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

SISTEM ZA DIJAGNOZU KVAROVA NA KUGLIČNIM LEŽAJEVIMA

FAULT DIAGNOSIS SYSTEM FOR ROLLING ELEMENT BEARINGS

Martina Joševska, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu je predstavljen sistem za klasifikaciju stanja motornih ležajeva zasnovan na primeni dvoklasnog Support Vector Machine klasifikatora. Ulazni podaci za SVM klasifikator dobijeni su snimanjem signala vibracija. Na osnovu prikupljenih signala izvršen je proračun obeležja. Implementirane su tri tehnike za redukciju polaznog broja obeležja: LDA, PCA i ICA i izvršena je analiza rezultata klasifikacije.*

Abstract – *In this paper an intelligent system for condition classification of rolling element bearings is presented. The system is based on the application of two-class Support Vector Machine classifier. Input data for SVM classifier are obtained by recording vibration signals. In order to reduce the initial number of features three methods have been implemented: LDA, PCA and ICA and the analysis of classification results was carried out.*

Ključne reči: *Izdvajanje obeležja, redukcija dimenzionalnosti, LDA, ICA, PCA, Support Vector Machines*

Key words: *Feature extraction, Dimensionality reduction, LDA, PCA, ICA, Support Vector Machines*

1. UVOD

U industrijskim primenama, kuglični ležajevi smatraju se kritičnim mehaničkim komponentama i defekti takvih ležajeva, osim ukoliko nisu na vreme primećeni, izazivaju smetnje i mogu voditi otkazu mašine ili pogona. Defekti u ležajevima mogu nastati tokom upotrebe ili u toku procesa proizvodnje.

Različite metode primenjuju se u cilju detekcije oštećenih ležajeva. Među njima dominiraju one zasnovane na merenju signala vibracija, električnih (strujnih) signala ili akustičnih signala. U analizi stanja motornih ležajeva najčešće se primenjuju signali vibracija. Signali vibracija snimaju se primenom industrijskih akcelerometara opšte namene, koji se postavljaju u blizini ležaja duž x, y i/ili z ose motora. Akcelerometri snimaju pokrete (vibracije) indukovane delovanjem dinamičkih sila na ležaj. Analizom snimljenih signala vibracija mogu se ustanoviti abnormalnosti nastale kao posledica mehaničkih oštećenja ili električnih poremećaja.

U radu je prikazana implementacija inteligentnog sistema za klasifikaciju stanja motornih ležajeva. Klasifikacija se zasniva na primeni dvoklasnog Support Vector Machine klasifikatora. Postupak projektovanja sistema za klasifikaciju sastoji se iz nekoliko faza:

1. Akvizicija podataka merenjem signala vibracija,
2. Digitalna obrada prikupljenih signala,
3. Izdvajanje obeležja u vremeskom i frekventnom domenu,
4. Redukcija dimenzionalnosti vektora obeležja,
5. Projektovanje SVM klasifikatora,
6. Testiranje klasifikatora.

Sve navedene faze implementirane su u okviru softverske aplikacije opisane u ovom radu.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Zoran Jeličić, vanr. prof.

2. EKSPERIMENTALNI REZULTATI

Izvršeno je eksperimentalno snimanje signala vibracija sa 196 motornih ležajeva. Vibracije su prikupljene sa dva radijalno postavljena senzora. Od 66 neoštećenih ležajeva, 44 su novi, dok je 22 ležaja već korišćeno. Preostalih 130 motornih ležajeva je oštećeno. Za svaki snimljeni signal vibracija izdvojen je vektor obeležja dimenzija 18×1, koji karakteriše testirani ležaj. Obeležja su proračunata u vremeskom i frekvenstnom domenu. Kompletan skup obeležja preuzet je iz [1], gde je dat i detaljan opis posmatranih obeležja, kao i postupak njihovog proračuna i način obrade signala vibracija.

3. REDUKCIJA DIMENZIONALNOSTI

Kada smo suočeni sa situacijom koja podrazumeva visokodimenzionalne podatke, prirodno je razmotriti mogućnost projekcije ovih podataka na prostor manjeg broja dimenzija bez gubitka važnih informacija koje se tiču karakteristika originalnih promenljivih.

Klasifikacija u visokodimenzionalnom prostoru obeležja je neprihvatljiva, pa je pre projektovanja klasifikatora neophodno redukovati dimenzionalnost vektora obeležja. Postoje dva načina da se dimenzionalnost smanji: izdvajanje obeležja (*eng. feature extraction*) i selekcija obeležja (*eng. feature selection*). Izdvajanje obeležja podrazumeva transformaciju postojećeg prostora obeležja u prostor manjeg broja dimenzija, a selekcija obeležja označava selekciju podskupa postojećih obeležja bez ikakve transformacije. Većina tehnika za izdvajanje obeležja zasniva se na linearnim tehnikama izdvajanja kao što su *Principal Component Analysis* (PCA), *Linear Discriminant Analysis* (LDA) i *Independent Component Analysis* (ICA) [2].

3. 1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) se može smatrati tradicionalnim statističkim pristupom za redukciju dimenzionalnosti. Osnovni cilj LDA metode je da se izvrši redukcija dimenzionalnosti polaznog skupa podataka, a da se pri tome klasna diskriminatorna informacija očuva u što većoj meri. LDA je tehnika

linearne projekcije. Težimo da dobijemo skalarnu vrednost y projektovanjem uzorka $\boldsymbol{x}_i$ na liniju. Od svih mogućih pravaca najpodesniji je onaj kod koga su projekcije uzoraka maksimalno međusobno udaljene. Implementirani LDA algoritam sastoji se od nekoliko koraka. Najpre se izračuna unutarklasna i međuklasna matrica rasejanja. Unutarklasna matrica rasejanja može se dobiti na osnovu izraza:

$$\boldsymbol{S}_w = \sum_{i=1}^{2} \sum_{j=1}^{M_i} (x_j - \mu_i)(x_j - \mu_i)^T \qquad (1.1)$$

Izraz koji odgovara međuklasnoj matrici rasejanja za slučaj dve klase je:

$$\boldsymbol{S}_b = \sum_{i=1}^{2} (\mu_i - \mu)(\mu_i - \mu)^T \qquad (1.2)$$

U prethodnim jednačinama μ_i označava vektor srednjih vrednosti pojedinačne klase, dok je μ vektor srednjih vrednosti računat na čitavom skupu podataka ($\mu = \frac{1}{2}\sum_{i=1}^{2} \mu_i$). Vrednost M_i odgovara broju uzoraka u svakoj klasi pojedinačno. Konkretno, za klasu oštećenih ležajeva M_i uzima vrednost 130, dok za klasu zdravih ležajeva M_i iznosi 66.

LDA metoda izračunava transformaciju koja maksimizuje međuklasno rasejanje, dok unutraklasno rasejanje minimizuje. Drugim rečima maksimizuje se sledeći kriterijum optimalnosti:

$$\frac{det(\boldsymbol{S}_b)}{det(\boldsymbol{S}_w)} \qquad (1.3)$$

Linearna transformacija data je matricom **W** čije su kolone svojstveni vektori matrice $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$. Rešavanje LDA problema svodi se na rešavanje konvencionalnog problema svojstvenih vrednosti matrice $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ [3].

3.2 . Principal Component Analysis

Principal component analysis (PCA) je statistička tehnika kojom se početni skup promenljivih linearno transformiše u znatno manji skup nekorelisanih promenljivih. Nove promenljive se nazivaju glavne komponente i sadrže većinu korisnih informacija iz originalnih promenljivih. Zbog činjenice da je mali skup nekorelisanih promenljivih lakše razumeti i koristiti u daljoj analizi u odnosu na veliki skup korelisanih promeljivih, ova tehnika kompresije podataka je široko primenjena [2].

Geometrijski gledano, PCA se svodi na pronalazak novog koordinatnog sistema kao ortogonalne transformacije početnog sistema. Ose novog koordinantnog sistema usmerene su tako da prva osa odgovara maksimalnoj varijansi, druga odgovara maksimalnoj varijansi u pravcu ortogonalnom na prvu osu i tako redom.

Principal Component Analysis realizovana je upotrebom MATLAB-ove integrisane funkcije *princomp*. Ova funkcija se zasniva na SVD (Singular Value Decomposition) dekompoziciji kovarijansne matrice **C** ulaznih podataka. Osnovi postupak PCA transformacije ulazne matrice **X** može se opisati sledećim nizom aktivnosti: centriranje podataka oduzimanjem aritmetičke sredine, proračun kovarijansne matrice **C**, izračunavanje svojstvenih vrednosti i svojstvenih vektora kovarijansne matrice **C**, projektovanje originalnih podataka u prostor svojstvenih vektora ($\boldsymbol{y} = \boldsymbol{w}^T\boldsymbol{x}$) [3].

3.3. Independent Component Analysis

Idependent Component Analysis je metoda pronalaženja osnovnih komponenti u višedimenzionalnim statističkim podacima. Ono što ICA razlikuje od ostalih metoda je to što traži komponente koje su statistički nezavisne. Statistička nezavisnost leži u osnovi ICA tehnike Nezavisnost podrazumeva da informacija sadržana u jednoj komponenti ni na koji način ne utiče na informacije u ostalim komponentama mernog signala. Generički ICA model se može izraziti kao:

$$\boldsymbol{x} = \boldsymbol{As} \qquad (1.4)$$

gde je **A** nepoznata transformaciona matrica, **s** je matrica nezavisnih komponenti, a $\boldsymbol{x}$ je matrica snimljenih,posmatranih signala. Osnovi problem ICA algoritma je kako proceniti matricu nezavisnih komponenti **s** ili transformacionu matricu **A** na osnovu merenih podataka $\boldsymbol{x}$, ako prethodno ne znamo ništa o matrici **s** ili matrici **A** [4].

ICA algoritam pronalazi nezavisne komponente maksimizacijom neke mere nezavisnosti kao što je negativna entropija. Primenom centralne granične teoreme dokazano je da je negausijalnost ključ procene ICA modela i da nezavisne komponente moraju imati negausovu raspodelu kako bi estimacije ICA modela bila moguća. Jedna kvantitativna mera negausijalnosti je negativna entropija i definisana je izrazom:

$$J(y) = H(y_{gauss}) - H(y) \qquad (3.3)$$

pri čemu je y_{gauss} Gausova slučajna promenljiva koja ima istu kovarijansnu matricu kao y. Negantropija je uvek nenegativna i jednaka nuli ako i samo ako je raspodela promenljive y Gausova. Zadatak ICA metode je da maksimizuje funkciju cilja J(y), jer se na taj način maksimizuje i negausijalnost i pronalaze nezavisne komponente [4].

Implementirani ICA algoritam maksimizuje jednu aproksimativnu formu jednačine negativne entropije i naziva se FastICA algoritam. FastICA algoritam je poznat kao visokoefikasan algoritam za koji je nezavisnim eksperimentima utvrđeno da je deset do sto puta brža od konvencionalnih gradijentnih metoda često korišćenih u estimaciji ICA modela [4].

4. REZULTATI I DISKUSIJA

Programsko rešenje i prateći grafički interfejs realizovani su upotrebom softverskog paketa MATLAB. Za obuku i testiranje klasifikatora koriste se MATLAB-ove integrisane funkcije sadržane u *Bioinformatics* toolbox-u**.**

Polazi se od činjenice da su prikupljeni signali vibracija sa 196 različitih motornih ležajeva. Za svaki ležaj izdvojeno je 18 obeležja, od kojih prvih devet odgovara vremenskim karakteristikama signala, a preostalih devet frekventnim karakteristikama. Smatramo da je pojedinačni ležaj ili oštećen ili zdrav i ne razmatramo dalji uzrok oštećenja iako bi se projektovanjem višeklasnog SVM klasifikatora i to moglo učiniti.

Najpre se radi redukcija dimenzionalnosti putem tehnika za izdvajanje obeležja (LDA, PCA, ICA), a zatim klasifikacija. Implementiran je nelinearni dvoklasni SVM klasifikator. Rezultati klasifikacije su tabelarno prikazani. U svakoj tabeli navedene su upotrebljene kernel funkcije (linearna, kvadratna, polinomijalna i rbf), stopa klasifikacije u fazi obuke i testiranja, stopa greške za obuku i test, kao i broj *support* vektora. Stopa klasifikacije i stopa greške izražene su u procentima.

Rezultati klasifikacije sadržani su u tabelama 1-3 za slučaj kada se broj obeležja redukuje na tri. Efekati primene LDA, PCA i ICA metode na inicijalni skup podataka ilustrovani su slikama 1-3. *Linear Discriminant Analysis* je pokazala najbolje rezultate. Kada ova metoda prethodi klasifikaciji, stopa korektno klasifikovanih uzoraka iznosi 100%, nezavisno od upotrebljene kernel funkcije. Rezultati klasifikacije za ovaj slučaj prikazani su u tabeli 1 i na slici 1.

Procenat korektno klasifikovanih test podataka pri primeni PCA/SVM klasifikacije iznosi 94-97,33, za podatke sa prvog kanala, dok pri primeni ICA/SVM klasifikatora ovaj procenat varira u opsegu 96-97,33. Rezultati PCA/SVM klasifikacije prikazani su u tabeli 2 i na slici 2, a rezultati ICA/SVM klasifikacije u tabeli 3 i na slici 3.

Klasifikacijom kojoj prethodi ICA redukcija dimenzionalnosti, ostvaruju se nešto bolji rezultati. To se može objasniti time što ICA algoritam u potrazi za baznim vektorima ide korak dalje od PCA metode, i traži ne samo pravce koji su nekorelisani već pravce koji su međusobno statistički nezavisni. Ukoliko se posmatraju podaci sa drugog kanala, ostvaruju se još bolji rezultati klasifikacije u odnosu na ostvarenu stopu klasifikacije za podatke sa prvog kanala. Procenat korektno klasifikovanih podataka iznosi 100 i u slučaju kada se za redukciju dimenzionalnosti koristi PCA metoda, i u slučaju kada se primenjuje ICA tehnika.

Tabela 1 *Rezultati obuke i testiranja klasifikatora upotrebom LDA tehnike; Kanal I, 3 komponente.*

	Klasifikacija(%		**Greška (%)**		**SVs**
	Trening	Test	Trening	Test	
linearna	100	100	0	0	13
kvadratna	100	100	0	0	15
polinomijalna	100	100	0	0	10
rbf	100	100	0	0	39

Tabela 2 *Rezultati obuke i testiranja klasifikatora upotrebom PCA tehnike; Kanal I, 3 komponente.*

	Klasifikacija(%)		**Greška (%)**		**SVs**
	Trening	Test	Trening	Test	
linearna	89,13	86,67	10,87	13,3	32
kvadratna	97,82	97,33	2,18	2,67	21
polinomijalna	97,83	94,00	2,17	6,00	21
rbf	100	96,00	0,00	4,00	43

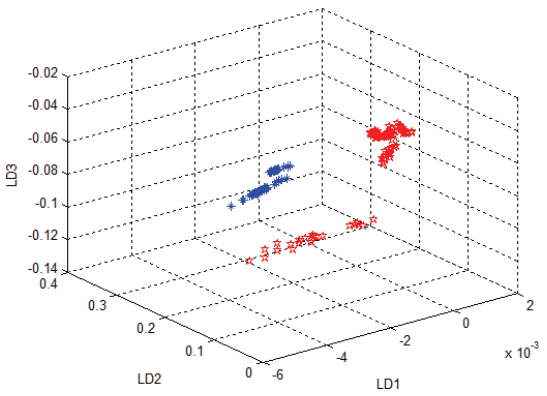
Tabela 3 *Rezultati obuke i testiranja klasifikatora upotrebom ICA tehnike* $g = u^3$*; Kanal I, 3 komponente.*

	Klasifikacija(%)		**Greška (%)**		**SVs**
	Trening	Test	Treni	Test	
linearna	89,13	86,67	10,87	13,3	32
kvadratna	100	97,33	0	2,67	24
polinomijalna	100	97,33	0	2,67	19
rbf	100	96	0	96	44

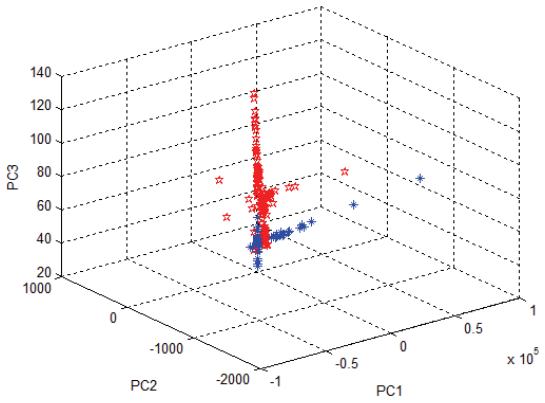
Prema tome, ukoliko samoj SVM klasifikaciji prethodi mapiranje sa osamnaest na tri dimenzije, dobri rezultati ostvaruju se upotrebom ma koje od tri implementirane tehnike za ekstrakciju obeležja. Pri tome najbolje performanse ispoljava LDA/SVM klasifikator. Za njim sledi ICA/SVM tip klasifikacije.

Međutim za razliku od LDA i PCA tehnike kod ICA metode računanje transformacione matrice ostvaruje se iterativnim algoritmom. LDA se pokazuje kao najefikasnija metoda za izdvajanje relevantnih obeležja. Za razliku od PCA i ICA metode, LDA je nadgledana tehnika tako da se podrazumeva da je za svaku opservaciju poznata i oznaka (labela) koja određuje pripadnost opservacije konkretnoj klasi.

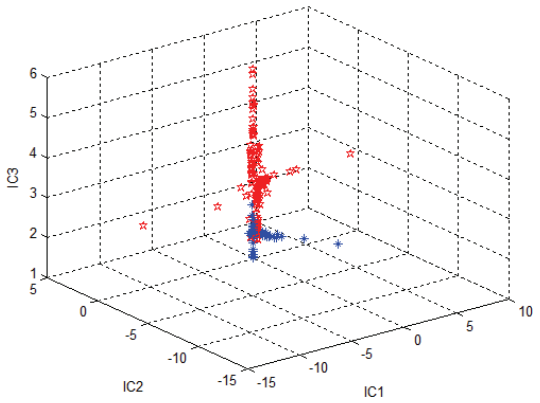
Rezultati klasifikacije u mnogome zavise od izbora obučavajućeg i test skupa. Predviđeno je da se ležajevi klasifikuju ili kao zdravi ili kao oštećeni. Međutim, ukoliko bi se analizirale vrednosti obeležja koja karakterišu stanje svakog pojedinačnog ležaja, uočilo bi se da se i u grupi oštećenih i u grupi zdravih ležajeva može izdvojiti nekoliko podgrupa.
Tako kod zdravih ležajeva, razlikujemo one koji su zdravi i još nekorišćeni i one koji su zdravi ali ipak korišćeni. Ove dve grupe zdravih ležajeva karakterišu dosta različite vrednosti obeležja.
Zato je primerke obe klase zdravih ležajeva potrebno uvrstiti u obučavajući skup. Na taj način izbegava se pogrešna klasifikacija zdravih ležajeva u fazi generalizacije (test fazi) i procenat pogrešno klasifikovanih uzoraka se smanjuje.



Sl. 1 *Rezultati primene LDA metode, kanal I, tri komponente*



Sl. 2 *Rezultati primene PCA metode, kanal I, tri komponente*



Sl. 3 *Rezultati primene ICA metode, kanal I, tri komponente*

5. ZAKLJUČAK

Predmet ovog rada bilo je projektovanje inteligentnog sistema za dijagnostiku oštećenih motornih ležajeva. Sistem za klasifikaciju defektnih ležajeva zasniva se na primeni dvoklasnog *Support Vector Machine* klasifikatora u kombinaciji sa metodama za ekstrakciju relevantnih i korisnih obeležja. *Linear Discriminant Analysis* (LDA), *Principal Component Analysis* (PCA) i *Independent Component Analysis* (ICA) su uspešno primenjene u postupku redukcije dimenzionalnosti polaznog prostora obeležja. Analizom rezultat uočeno je da je LDA najpodesnija metoda reprezentacije podataka i da njena primena vodi najuspešnoj klasifikaciji.

U osnovi ICA metoda je nešto uspešnija tehnika izdvajanja obeležja u odnosu na PCA. To je posledica činjenice da je kod PCA metode redundantnost određena postojanjem korelacija između podataka, dok je kod ICA tehnike primenjen mnogo potpuniji koncept statističke nezavisnosti. Gledano sa te strane, ICA omogućuje bolju reprezentaciju podataka, jer ide korak dalje u odnosu na PCA, ali ne znači da je efektivniji algoritam za postizanje same dimenzionalni redukcije. Redukcija dimenzionalnosti se kod ICA algoritma ostvaruje primenom PCA metode.

Osnovna PCA metode je upravo redukcija dimenzionalnosti, tj. smanjenje broja promenljivih posmatranog prostora. Dalje procesiranje obezbeđuje finiju predstavu i jasnije raspoznavanje uzoraka.

Za klasifikaciju uzoraka primenjen je koncept dvoklasne nelinearne SVM klasifikacije. Uzorci su klasifikovani ili kao oštećeni ili kao neoštećeni. Predviđena je upotreba jedne od četiri kernel funkcije: linearne, kvadratne, polinomijalne i Gausove rbf funkcije. U slučaju prethodne primene ICA ili PCA metode, odgovarajućom se pokazuje primena samo nelinearnih kernela, tako da se u tom slučaju linearna kernel funkcija može izopštiti iz razmatranja i dalje analize.

Kao pravac daljeg razvoja moguće je proširiti ovaj sistem klasifikacije tako da se umesto klasifikacije tipa oštećen–neoštećen primeni višeklasni tip SVM klasifikatora. U osnovi realizovani SVM klasifikator pripada grupi binarnih klasifikatora. Međutim u klasi oštećenih ležaja možemo uočiti nekoliko podklasa koje su posledica različitih tipova oštećenja. Iz tog razloga pogodovala bi višeklasna SVM klasifikacija, čime bi se omogućila ne samo detekcija oštećenja, već i detekcija uzroka oštećenja motornog ležaja.

6. LITERATURA

[1] Stepanić P., Latinović V. I., Đurović Ž (2009). A new approach to detection of defects in rolling element bearings based on statistical pattern recognition, Springer-Verlag, The International Journal of Advanced Manufacturing Technology, 45, 91-10

[2] Widodo Achmad, Yang Bo-Suk, Han Tian (2007). Combination of independent component analysis and support vector machines for intelligent faults diagnosis of induction motors. Science Direct, *Expert Systems with Applications*, 32, 299-312

[3] Martinez W.L., Martinez A.R.,Solka J.L. (2011). Exploratory Data Analysis with Matlab, Second Edition, CRC Press, 978-1-4398-1220-4

[4] Hyvärinen A.,& Oja E.(2000), Independent componenet analysis: algorithms and applications. Neural Networks, 13 (4-5), 411-430

Kratka biografija:



Martina Joševska, rođena je u Novom Sadu 1986. godine. Od oktobra 2010 godine je apsolvent na Fakultetu tehničkih nauka u Novom Sadu, na odseku za Računarstvo i automatiku.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.317:616-7

DIGITALNO MERILO BROJA UDISAJA

DIGITAL BREATH MEASURING DEVICE

Ivan Babić, *Fakultet Tehničkih Nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

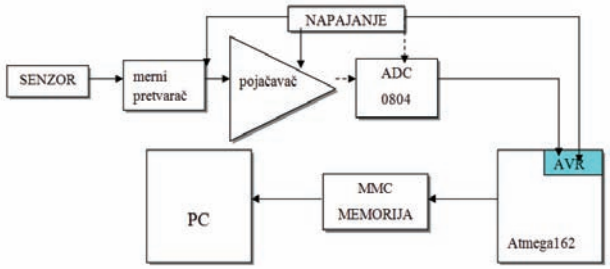
Kratak sadržaj – *U izlaganju se prikazuje digitalno merilo broja udisaja. Prikazano merilo udisaja je prenosivi uređaj sa baterijskim napajanjem, čiji je zadatak da u memoriju zabeleži 24 sata merenja broja udisaja. Udisaji se detektuju preko širenja i skupljanja grudi koje se prenosi na mernu traku. Signal sa merne trake se pojačava i dovodi na AD konvertor. Memorija korišćena u ovom prenosivom uređaju je flash memorija. Podaci iz flash memorije se kasnije preuzimaju i obrađuju u PC računaru. Na PC računaru je predviđena softverska aplikacija koja radi pod Linux operativnim sistemom.*

Abstract – *The presentation shows an evaluator of inspiration. Presented evaluator of inspiration is a portable device with battery power, whose task is to measure inspiration and record the inspiration detections to its memory during last 24 hours. Inspiration is detected through the expansion and shrinkage of the chest that is transmitted to a strain gage. The signal from the strain gage is amplified and drived to an AD converter. Memory used in this portable device is flash memory. The data from flash memory is later fetched and processed in a PC. PC software application runs on Linux operating system.*

Ključne reči: *merilo broja udisaja, senzor,merilo.*

1. UVOD

Merilo broja udisaja je prenosivi uređaj sa baterijskim napajanjem, čiji je zadatak da u flash memoriju zabeleži 24 sata merenja širenja i skupljanja pluća. Podaci iz flash memorije se kasnije preuzimaju i obrađuju u PC računaru pod Linux operativnim sistemom. Sastav uređaja prikazan je blok dijagramom na Slici 1.



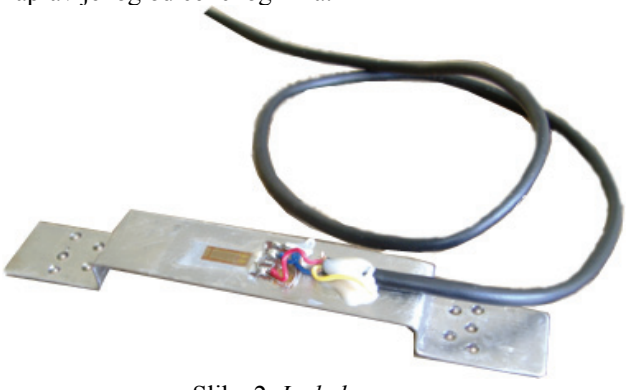
Slika 1. *Blok šema uređaja*

Senzor je specifičan – sastoji se od dve otporne merne trake koje su nalepljene na relativno čvrst materijal da bi se omogućila deformacija do 0.5%, jer je promena otpora pri takvoj deformaciji linearna. Otporne merne trake povezane su tako što se jedna traka veže u jednu a druga u drugu granu mernog mosta (Vitstonov most), a koje su na senzoru postavljene na različite načine (o čemu će biti reči kasnije). Zbog uravnotežavanja mosta će u jednoj grani biti promenljiv otpornik. Signal sa mernog mosta će biti nalik sinusoidi, odnosno signal ce se menjati u skladu sa disanjem. Pošto je signal sa mernog mosta reda mV on se mora dodatno pojačati, a to se postiže sa jednim instrumentacionim i dva operaciona pojačavača.

Za konvertovanje ovog analognog signala u digitalni, pogodan za upis u flash memoriju koristićemo A/D konvertor ADC0804 čiji je opseg ulaznog napona od 0V do 5V. Programiranje mikrokonrolera za odabiranje signala, merenje vremena i upisivanje u flash memoriju ovih podataka izvršićemo pomoću mikroPascal-a, a unošenje hex koda u mikrokontroler se vrši pomoću Atmelovog programatora - AVR Dragon-a. Takođe, program će uključivati kod za uštedu energije – na osnovu frekvencije odabiranja koju diktira eksterni oscilator i tako uštedeti 90% energije koja bi se trošila pri njegovom konstantnom radu. Takođe, kada se flash memorija prebaci u PC pod operativnim sistemom Linux, podaci se nalaze u txt formatu, gde će biti obrađeni pomoću C programa koji se pokreće iz *Shell*-a. Udisaji će biti izbrojani identifikovanjem maksimuma među odbircima.

2. SENZOR

Senzor (Slika 2.) se sastoji od mernih traka i postolja napravljenog od čeličnog lima.

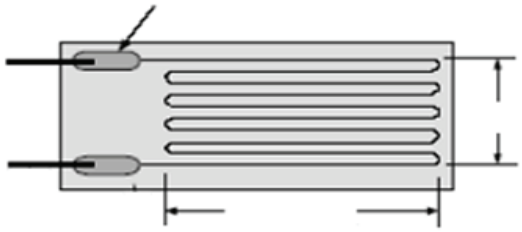


Slika 2. *Izgled senzora*

Merna traka spada u vrstu senzora koji rade na bazi promene otpora. Vrlo je osetljiva na milimetarsko ili čak mikronsko istezanje, savijanje i uvrtanje, pri čemu daje vrlo malu promenu otpora reda mili oma ($\pm$ 1 mΩ). To su homogeni provodni materijali koji omogućuju da

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-bečelor rada čiji mentor je bio dr Zoran Mitrović, van.prof.

procenom promene otpornosti odredimo deformaciju tela na koje je pričvršćena merna traka (u ovom slučaju postolje od čeličnog lima).Izgled merne trake je prikazan na Slici 3.



Slika 3. *Izgled merne trake*

Senzor sa Slike 2. se sastoji iz dve merne trake (Slika 3.) sopstvene otpornosti od $350\,\Omega$. Prva reaguje na deformisanje mateijala na koji je pričvršćena (postavljena horizontalno u odnosu na osu aluminijumske pločice) a druga ne reaguje na fizičke deformacije postolja (postavljena je vertikalno), već služi za kompenzovanje temperaturnih koeficijenata, kako ne bi uticali na tačnost merenja.
Na Slici 4. prikazan je princip rada predhodno opisanog senzora. Crnim isprekidanim linijama su prikazane elastične deformacije (reda milimetra,ali su na slici ta pomeranja prikazana u nešto većoj srazmeri radi boljeg uočavanja i veće jasnoće) materijala na koji je pričvršćena merna traka, a koje na nju direktno utiču.

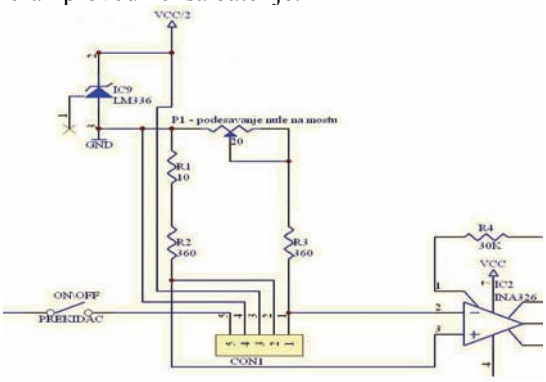


Slika 4. *Princip rada senzora*

Crne isprekidane linje obeležene slovom **a** označavaju pravac kretanja papučica postolja senzora, koje mogu da se približavaju ili udaljavaju jedna od druge. One se pomeraju pravolinijski, ali zbog same konstrukcije metalnog postolja dolazi do blage promene ugla papučice. Isprekidanom linijom sa oznakom **b** prikazana je elastična deformacija one površine senzora na kojoj se nalazi merna traka,a koja direktno utiče na promenu otpornosti merne trake (reda mΩ).
Ta promena može da bude sa pozitivnim i sa negativnim prirastom otpornosti, u zavisnosti da li deformacija navedene površine ima konveksan ili konkavan oblik, tj.u zavisnosti da li se papučice približavaju ili udaljavaju jedna od druge.
Merna traka se postavlja na sredinu limenog postolja, jer je pri vrlo malom pomeranju papučica po horizontali uvijanje najveće, i dovoljno da merna traka odreaguje. Međutim, za namenu ovog senzora to nije uslov,jer je pomeranje papučica nešto veće, a samim tim i uvijanje kompletnog postolja koje je sasvim dovoljno da merna traka ima bilo kakvu promenu otpornosti i ako se ne nalazi na sredini.
Za ovaj senzor korišćen je čelični lim debljine 1 mm, zbog svoje velike čvrstine, i potrebe da se pri deformisanju i savijanju blažeg tipa uvek vraća u približan početni položaj.

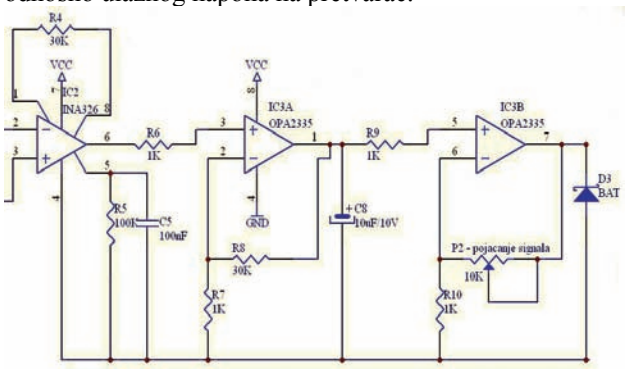
3. MERNI PRETVARAČ

Na Slici 5. je prikazan deo mernog mosta koji se nalazi na uređaju. Na konektoru CON1 se povezuju provodnici sa senzora i provodnici sa baterije.



Slika 5. *Prihvatanje signala sa mernog mosta*

Na Slici 3.1.2 data je električna šema mernog pretvarača koji na ulazu ima signal sa senzora čija je promena veoma mala (reda mV), a na svom izlazu daje pojačan signal u opsegu 0-5V u zavisnosti od promene napona na senzoru, odnosno ulaznog napona na pretvarač.



Slika 6. *Električna šema mernog pretvarača*

Instrumentacioni pojačavači se koriste tamo gde je potrebno pojačati ulazne signale malih nivoa. Nedostatak običnog diferencijalnog pojačavača je što su mu ulazne otpornosti relativno male i nisu u opštem slučaju jednake na oba ulaza. Ovaj nedostatak se može otkloniti složenijim kolima, odnosno primenom više operacionih pojačavača.
Merni pretvarač se sastoji od instrumentacionog pojačavača (INA326), operacionog pojačavača (OPA 2335), šotkijeve diode i nekoliko otpornika i kondenzatora. Signal sa mosta se dovodi na priključke 2 i 3 instrumentacionog pojačavača. Izlaz instrumentacionog pojačavača se veže na prvi operacioni pojačavač (neinvertujući) koji ima pojačanje 30, a zatim se signal vodi na drugi operacioni pojačavač koji ima promenljivo pojačanje. Dioda D3 sprečava dovođenje negativnog napona na ADC0804 jer se izlaz iz drugog operacionog pojačavača vezan na ADC0804.

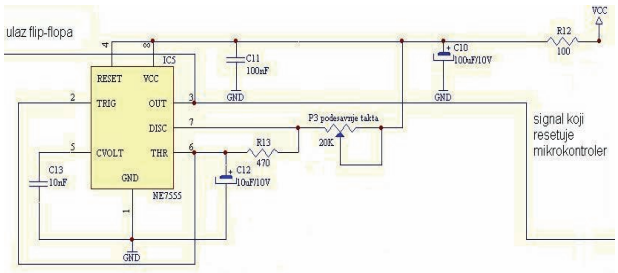
4. KOLO ZA SMANJENU POTROŠNJU BATERIJE

Za smanjenu potrošnju enegrije baterije korišćene su sledeće komponente:

- NE555
- 74HC02

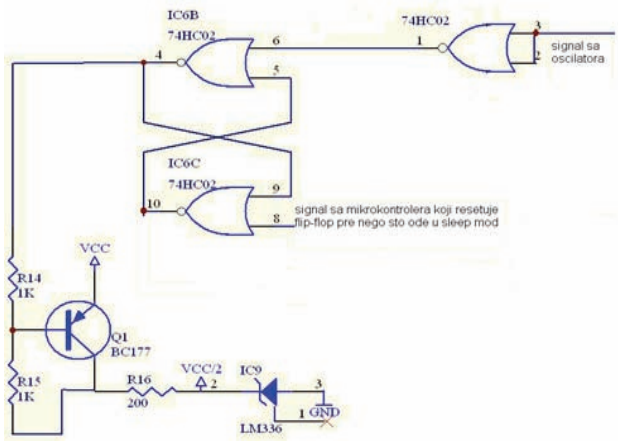
- LM336
- BC177

Princip rada je sledeći:
Eksterni oscilator (slika 7.) šalje impulsni signal kod koga visok napon (logička jedinica) traje 150 ms, a pauza 4 ms. Taj signal se dovodi na invertor koji je napravljen od NILI kola. Signal sa invertora se šalje na mikrokontroler i RS–flipflop. Signal koji dođe na flipflop setuje, a pošto tranzistor radi kao prekidač, merni most dobija napon od 2.5 V sa Zener diode. Signal koji dođe na mikrokontroler aktivira mikrokontroler, jer do tada je on bio u *sleep* modu. Kada mikrokontroler primi vrednost sa ADC-a on prvo resetuje flipflop, upiše vrednost u MMC karticu i ponovo odlazi u *sleep* mod. Električna šema je data na sledećim slikama.



Slika 7. *Eksterni oscilator*

Na sledećoj slici je prikazan flip-flop i referenca od 2.5 V.



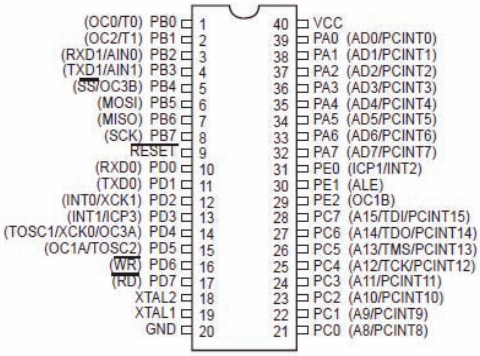
Slika 8. *RS flip-flop i referenca od 2.5 V*

5. A/D KONVERTOR

ADC0804 priprada familiji c-mos osmobitnih AD konvertora sa sukcesivnim aproksimacijama, i dizajniran je da radi sa 8080a izlazima kontrolne magistrale. Upravljanje AD konvertorom vrši mikrokontroler i to na sledeći način:
Sa prelaskom sa visokog na nizak nivo WR ulaza, resetuju se SAR unutrašnja kola i nivoi shift registra i INTR izlaz će biti postavljen na visoki nivo. Dokle god su CS i WR na niskom nivou AD konvertor će ostati u stanju reseta. AD Konverzija ce početi kada bar jedan ulaz (CS, WR) izvrši tranziciju sa niskog na visoki nivo. Posle završetka AD konverzije INTR pin će napraviti tranziciju sa visokog na niski nivo. Ovo se može iskoristiti da se blokira procesor. RD operacija će prebaciti INTR liniju ponovo na visoki nivo. Da bi se osiguralo paljenje pod svim mogućim uslovima, potreban je eksterni WR puls prilikom prvog paljenja kola. Stabilizator (LM336) koristimo radi dobijanja referentnog napona od 2.5 V za AD konvertor.
Frekvencija AD konvertora ADC0804 definisana je RC članom. Data linije AD konvertora su povezane na PORTD mikrokontrolera. Upravljački pinovi (CS, WR, RD, INTR) se nalaze na PORTC mikrokontrolera.

6. MIKROKONTROLER AVR ATmega162

Arhitektura svih mikrokontrolera familije AVR je identična. ATmega162 (slika 9) koristi koncept hardvarske arhitekture koja je karakteristična po tome da ima odvojen memorijski prostor za programske instrukcije i memorijski prostor za podatke (programska memorija i memorija podataka).
Postoji ugrađen pipeline mehanizam koji omogućava da se u toku izvršavanja jedne instrukcije preuzima sledeća instrukcija iz programske memorije. Pipeline mehanizmom i RISC arhitekturom omogućeno je da se jedna instrukcija izvrašava u svakom taktu.



Slika 9. *Raspored pinova*

Svi mikrokontroleri AVR familije imaju 32 registra opšte namene koji čine registar fajl. Neki od ovih registara imaju dodatne specijalne funkcije. Registri imaju imena od R0 do R31. Registar fajl je podeljen na 2 dijela po 16 registara i to prvi deo su registri od R0 do R15, a drugi deo su registri od R16 do R31.
Sve instrukcije koje vrše operacije nad registrima imaju direktni pristup u trajanju od jednog takta. Registar R0 i registri od R26 do R31 imaju dodatne funkcije.
Detaljniji prikaz mikrokontrolera nalazi se na zvaničnom Atmelovom sajtu [1].

7. MMC KARTICA

Ove multimedijalne kartice su visoko integrisane fleš memorije sa mogućnošću serijskiog i nasumičnog pristupa. Mala potrošnja snage i širok opseg napona napajanja zadovoljava mobilne telefone, aplikacije sa baterijskim napajanjem kao što su audio plejeri, agende, elektronske knjige, enciklopedije ili rečnici.
Korišćenjem veoma efektivne šeme za kompresiju podataka kao sto je MPEG, MMC ce ‘ostaviti’ dovoljno kapaciteta za bilo koji tip multimediajlnih podataka. U ovom projektu u MMC se kreira txt fajl i u njega se smeštaju odbirci na svakih 150 ms.

8. FIRMVER - mikroPascal for AVR

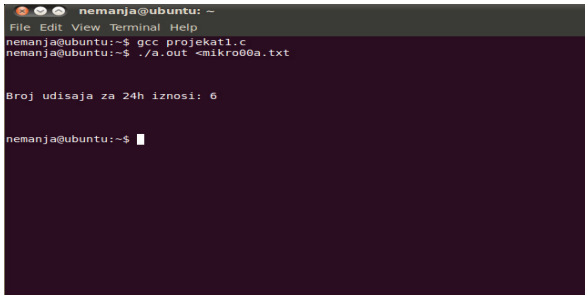
Razvojno okruženje mikroPascal for AVR je moćan i opcijama bogat razvojni alat za AVR mikrokontrolere. Dizajniran je tako da pruži korisniku najlakše moguće rešenje za razvoj aplikacije za ugrađene (*embedded*) sisteme, bez kompromisa na štetu performansi ili kontrole sistema. AVR mikrokontroleri i programski jezik Passcal predstavljaju savršenu kombinaciju, AVR mikrokontroler je najpopularniji 8-bitni čip u svetu i primenjuje se u velikom broju sistema, a programski jezik Passcal, kojem se odaje priznanje zbog svoje efikasnosti, je logičan izbor za razvoj *embedded* sistema.
Prilikom realizacije firmvera potrebno je proći kroz nekoliko faza. Prvu fazu predstavlja inicijalizacija mikrokontrolera. Mikrokontroler treba da prihvati digitalni signal sa AD konvertora i zatim upiše u MMC. Nakon toga mikrokontroler resetuje flip-flop, odlazi u *sleep* mod i čeka da ga signal sa oscilatora probudi. Kada se mikrokontroler probudi on ulazi u interapt funkciju gde ponovo očitava vrednost sa AD konvertora i tu vrednost upisuje u MMC. Ovaj ciklus se ponavlja sve dok se ne ugasi uređaj.

9. PC APLIKACIJA

Pod operativnim sistemom UNIX postoji vrlo elegantan način za *skretanje* glavnog ulaza i/ili izlaza. Time se postiže da se podaci čitaju iz neke datoteke umesto sa tastature. To je u ovom projetku iskorišćeno. U terminalu prilikom pokretanja C programa doda se jos <mikro00a.txt .
Pozivanje u terminalu izgleda ovako:

./a.out <mikro00a.txt

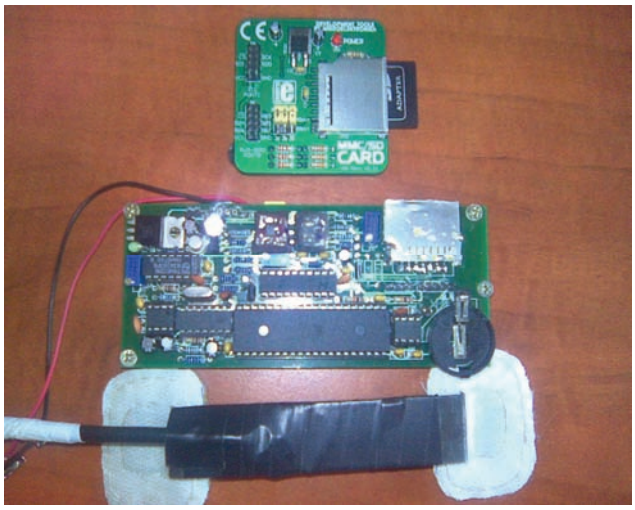
Na sledećoj slici je prikazan izgled aplikacije.



Slika 10. *Izgled aplikacije*

10. ZAKLJUČAK

U ovom radu je prikazana realizacija digitalnog merila broja udisaja. Merilo je napravljeno kao prenosivi uređaj sa baterijskim napajanjem, čiji je zadatak da u flash memoriju zabeleži 24-časovno merenje udisaja, zasnovano na detekciji širenja i skupljanja grudnog koša. Podaci iz flash memorije se kasnije preuzimaju i obrađuju u PC računaru pod Linux operativnim sistemom.
Osim što bi za praktičnu upotrebu bilo potrebno uraditi detaljniju analizu merne nesigurnosti uređaja, smernice za dalje usavršavanje realizovanog digitalnog merila mogu biti veoma različite.
Na sledećoj slici prikazan je izgled uređaja.



Slika 11. *Izgled uređaja*

11. LITERATURA

[1] sajtovi: www.mikroe.com, www.alldatasheet.com, www.kelm.ftn.uns.ac.rs, www.atmel.com/

Kratka biografija:



Ivan Babić rođen je u Šapcu 1986. god. Diplomski-bečelor rad na Fakultetu tehničkih nauka iz oblasti Elektroteh¬nike i računarstv Instrumentacija i merenje odbranio je 2011.god..

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.3

PROŠIRENJE OPERATERSKE KONZOLE SCADA OASyS DNA SISTEMA C#.NET PROGRAMSKIM SLOJEM ZA KONTROLU SLOŽENIH KORISNIČKIH PRIKAZA

EXTENSION OF OASyS DNA HMI CONSOLE WITH C#.NET SUPPORT FOR EDITING COMPLEX SCADA DISPLAYS

Marko Dojčinović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Cilj ovog rada jeste da se operaterska konzola (eXtended Editor – XE) SCADA OASyS DNA sistema proširi C#.NET programskim slojem sa ugledom na već postojeći VB.NET i OBEL programski sloj za kontrolu korisničkih prikaza (eng. display). Korišćeno je .NET (C# i C++) i OASyS DNA razvojno okruženje.*

Abstract – *The task of this paper is to extend HMI of SCADA OASyS DNA system in order to support C#.NET display editing in the same manner as already implemented VB.NET and Obel editing support.*

Ključne reči: *XE, C#.NET, Code generation*

1. UVOD

Cilj ovog rada jeste proširenje operaterske konzole SCADA OASyS DNA sistema (*eXtended Editor (XE)*) tako da podrži pisanje pozadinskog izvornog koda prikaza (*Display*) u C#.NET programskom jeziku. Pre implementacije C#.NET programskog sloja, XE je podržavao programiranje prikaza pomoću VB.NET programskog jezika i Obel skriptnog jezika (razvijenog od strane Telventa sa ciljem da pojednostavi programiranje prikaza).

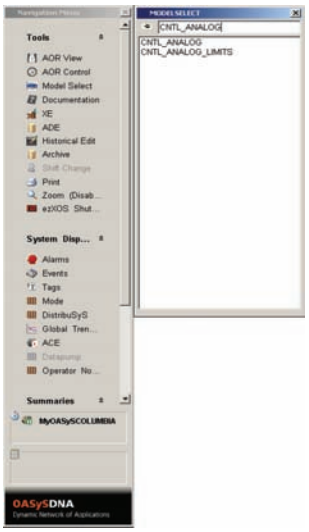
Operaterska konzola, kao deo HMI podsistema SCADA sistema, sadrži programsku podršku za crtanje i definisanje ponašanja prikaza, koju operateri koriste prilikom izmene postojećih ili dodavanja novih prikaza. Modeli prikaza mogu biti veoma jednostavni poput prikaza semafora, koji bojom predstavlja njegove stanje, ili složeni kao što je na primer prikaz svih pozicija liftova u nekom neboderu i njihova stanja, prikaz svih vozova na nekoj železničkoj mreži, i sl.

C#.NET programski jezik predstavlja relativno nov obejktno orijentisan programski jezik, nastao pojavom prve verzije .NET okruženja (.NET 1.0) u julu 2000. godine (aktuelna verzija je 4.0). On predstavlja evoluciju C++, C i Java programskih jezika i nastao je od strane Microsoft-a sa ciljem da zadrži njihove najbolje odlike, a istovremeno da izostavi ili ispravi njihove probleme. Razvijanje programske podrške korišćenjem C# programskog jezika je jednostavnije u odnosu na C++, jer se odlikuje jednostavnijom i jasnijom sintaksom.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je dr Branislav Atlagić, red. prof.

Proširenje XE-a C#.NET programskim slojem, treba da obezbedi generisanje C#.NET izvornog koda unutar editora koda (sa podrškom za C#.NET sintaksu), kao i njegovo prevođenje C#.NET prevodiocem.

Prilikom ispitivanja ponašanja i funkcionalnosti prikaza, definisanih C#.NET programskim jezikom, korišćena je ezXOS stanica (*Extended Operators Station*). ezXOS je OASyS DNA operatorska stanica zadužena za upravljanje komponentama sistema korišćenjem grafičke korisničke sprege (*Graphical User Interface (GUI)*). Iz njega se dakle pokreću i otvaraju prikazi (većinom definisani iz XE-a) i prozori vidljivi na operaterskom ekranu. Izgled ezXOS stanice sa otvorenom formom za izbor modela (prikaza) je dat na sledećoj slici:



Slika 1 *Izgled ezXOS stanice*

2. ANALIZA REŠENJA

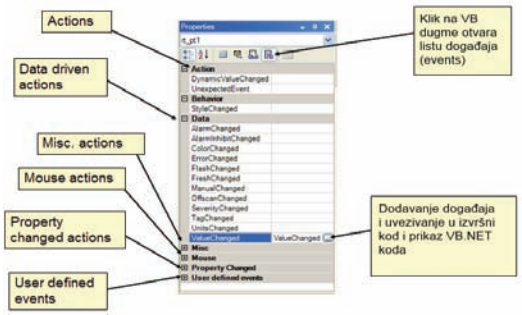
Rešenje, odnosno implementacija C# programske podrške za definisanje prikaza treba da poseduje funkcionalnost i karakteristike već postojećeg VB.NET programskog sloja za definisanje ponašanja prikaza unutar XE konzole.

XE predstavlja HMI aplikaciju, baziranu na Windows. NET formama, koja obezbeđuje "*What You See Is What You Get*" okruženje za konstruisanje i prepravljanje već postojećih prikaza. Prikaz može biti tipa Model ili Composite i njegovo ponašanje može biti opisano Visual Basic ili Obel programskim kodom. Cilj projekta jeste dodavanje C# sloja za prikaze tipa Model sa strukturom VB.NET programskog sloja.

Model predstavlja skup objekata, poput .NET kontrola (*button, text box, radio button, label...*) i generičkih

objekata (*Bar, Gauge, Text Clock*...). Svaki objekat je definisan zasebnim skupom svojstava (*Properties*) i događaja (*Events*), koji se mogu menjati preko liste svojstava (*property list*) i liste događaja (*VB* i *Obel events*). Ove liste se prikazuju unutar prozora svojstava (*property window*) izborom određenog objekata ili samog modela.

Događaje je moguće definisati u VB.NET programskom jeziku i u Obelu-u. Svaki objekat ima specifičnu listu događaja, grupisanih po kategorijama. Izborom nekog od događaja, određenog objekta, otvara se prikaz koda (programskog jezika za koji je izabran događaj) i generiše se kod funkcije za obradu izabranog događaja, kao i povezivanje iste sa izabranim objektom ili modelom. Na sledećoj slici je prikazana lista događaja za VB.NET programski jezik.



Slika 2 *Lista događaja objekta ili modela*

Jedan deo konačnog rešenja podrazumeva i dodavanje prethodno opisanog mehanizma za C#.NET, odnosno mogućnost definisanja događaja za C#.NET izvorni kod.

XE omogućuje prikaz modela sa tri aspekta:

- dizajna, odnosno grafički aspekt,
- pozadinski Visual Basic.NET kod,
- pozadinski Obel kod.

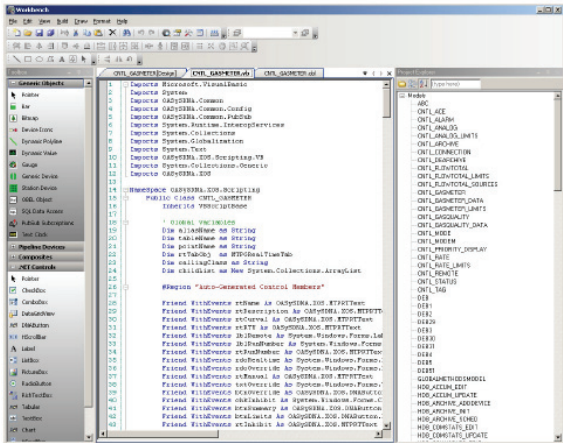
Svaki od prikaza se otvara u zasebnoj kartici (tab-u). Rešenje podrazumeva dodavanje opcije za prikaz pozadinskog C#.NET koda u zasebnoj kartici sa specifičnim osobinama i mogućnosti prevođenja C# izvornog koda.

Prikaz pozadinskog VB.NET koda modela (*Visual Basic view*) se koristi za pisanje VB.NET izvornog koda za određeni model. Preko VB.NET koda je moguće menjati, proveravati svojstva modela i objekata na njemu, pozivati metode i funkcije grafičkih objekata i globalnih objekata, definisanje određenih događaja, itd.. Otvaranje prikaza VB.NET koda se postiže iz Project Explorer-a (padajući meni) ili dodavanjem novog ili izborom već postojećeg događaja.

Prozor koji sadrži VB.NET kod, se sastoji od editora koda, u kome je prikazan kod, sa sledećim osobinama:

- definisana je sintaksa VB.NET programskog jezika,
- automatsko označavanje VB.NET rezervisanih reči,
- automatsko formatiranje koda (*Indent*), odnosno blokova koda,
- omogućene su standardne operacije tekst editora – cut, delete, copy, paste, undo, redo i sl,
- postojanje Intellisense-a (predikcija teksta razvijena od strane Microsofta za VS),
- prepoznavanje predprocesorskih direktiva,
- zatvaranje i otvaranje regiona, funkcija i blokova (*Collapse*),
- prikaz padajućeg menija na desni klik miša,
- mogućnost pretrage i zamene teksta (*find/replace dialog*).

Sve navedene osobine je potrebno implementirati i za C#.NET programski jezik, kako bi se omogućilo pisanje koda u C# programskom jeziku, a kasnije i njegovo provođenje. Izgled prikaza VB.NET izvornog koda modela je dat na sledećoj slici:



Slika 3 *Izgled VB.NET editora koda*

Neki od najbitnijih mehanizama, pored već pomenutih, koje je potrebno obezbediti, odnosno implementirati jesu:

- učitavanje modela i kreiranje novih modela sa C#.NET izvornim kodom,
- čuvanje modela sa dodatnim poljem, tj. C#.NET izvornim kodom,
- prevođenje (kompajliranje) C#.NET koda, bez narušavanja rada i redosleda izvršavanja već postojećih prevodilaca (VB i Obel).

Učitavanje modela se izvodi kroz dva koraka, a pokreće se prilikom zahteva za otvaranjem određenog modela. Nakon zahteva za otvaranje modela, prvo se otvara grafički prikaz modela, a on podrazumeva učitavanje svih grafičkih objekata koje sadrži kao i njihovih osobina i svojstava. Svi podaci vezani za određeni model se čuvaju u .XEM datotekama (XML datoteke za čuvanje modela), a pored grafičkih objekata i njihovih osobina, u .XEM datoteci se čuva i VB.NET i Obel izvorni kod. Pomoću ove datoteke je moguće rekonstruisati svaki model, i ona je dovoljna prilikom učitavanja modela. Kako bi se osiguralo upisivanje C#.NET izvorsnog koda u ovu datoteku, potrebno je napraviti dodatnu granu unutar .XML datoteke, koja će sadržati izvorni kod C#.NET programskog jezika. Drugi fazu učitavanja predstavlja učitavanje izvornih kodova modela iz datoteka izvornih kodova u editore koda. Datoteke izvornih kodova, potrebno je generisati i za C#.NET programski jezik i to na sledeći način: *ime_modela.cs*, unutar direktorijuma datoteka izvornog koda (../model/source). Pomoću ovih datoteka i .pdb datoteka (program database – Microsoftov format datoteke, u kojoj se čuvaju debug informacije programa) moguće je nadgledati tok izvršavanja programa (debugovanje) u Visual Studio-u.

Čuvanje modela se sastoji od čuvanja svih grafičkih objekata modela i njihovog rasporeda i svojstava, kao i

izvornog koda u .XEM datotekci, ali i čuvanja samog izvornog koda u datotekama izvornog koda.
Prevođenje izvornog koda se pokreće nakon čuvanja koda i izvodi se redom za VB.NET izvorni kod, Obel kod i na kraju C#.NET izvorni kod (potrebno implementirati). Prevođenje se sastoji od dve faze, od kojih jednu fazu čini standardno prevođenje C#.NET prevodiocem, a druga faza je revizija strukture samog koda. Revizija koda proverava se da li su definisani svi neophodni blokovi, deklaracije grafičkih objekata i događaja vezanih na njih, itd.. Obe faze prevođenja, ispisuju svoje rezultate (upozorenja i greške) u panel za ispis razultata prevodioca (*Syntax Results Window*), koji je prikazan na slici 4. C#.NET prevodilac nakon uspešnog prevođenja kreira izvršnu datoteku (assembly - .dll) i datoteku sa debug informacijama (.pdb). Kako VB.NET prevodilac kreira identične datoteke, sa identičnim nazivom, potrebno je i obezbediti poseban direktorijum za izvršne datoteke, kreirane od strane C#.NET prevodioca. Izvršne datoteke učitava operaterska stanica (ezXOS) i one predstavljaju i definišu ponašanje SCADA prikaza, odnosno krajnji rezultat i izlaze ovog rada.

Compiler Results

File Name	Line Number	Error Type	Message
_TEST1.vb	0		Success: 0 Errors, 0 Warnings
_TEST1.obl	0		Success: 0 Errors, 0 Warnings
_TEST1.cs	88	Error	Cannot implicitly convert type 'float' to 'int'. An explicit conversion exists (are you missing a cast?)
_TEST1.cs	0		Failed: 1 Errors, 0 Warnings

Slika 4 *Rezultati prevodioca sa dodatim rezultatima C#.NET prevodioca*

4. IMPLEMENTACIJA

Implementacija rešenja se može grubo podeliti u 3 faze, odnosno modula:

- proširenje tabele svojstava (*Property Grid*),
- proširenje mehanizma za upravljanje karticama,
- dodavanje editora koda za C#.NET.

4.1 PropertyGrid

Klasa PropertyGrid predstavlja implementaciju tabele svojstava grafičkih objekata i modela . Sastoji se od dve kontrole, odnosno instanci klasa:

- *System.Windows.Forms.ComboBox,*
- *XEPropertyGrid.*

ComboBox je .NET klasa, koja predstavlja padajuću listu, a *PropertyGrid* koristi njenu instancu za prikaz i izbor modela i svih grafičkih objekata unutar modela (prikaza). *XEPropertyGrid* klasa nasleđuje .NET klasu *System.Windows.Forms.PropertyGrid* , koja obezbeđuje korisničku spregu (*User interface*) za pregled svojstava (properties) i događaja (events) objekta. Prilikom promene objekta u *ComboBox*-u ili vrste događaja (VB, Obel, Action, C#), XEPropertyGrid se osvežava, odnosno popunjava svojstvima određenog objekta. Kako bi bilo moguće dodavati nove događaje za objekat u C#.NET programskom jeziku, potrebno je dodati dugme, čijim se pritiskom otvara lista događaja za C#.NET. Lista događaja je identična kao i za VB.NET programski jezik. Klasa *XEPropertyGrid* sadrži kolekciju *PropertyTab*-ova (kartice – tabulari), koji predstavljaju skup svojstava nekog objekta i oni popunjavaju tabelu svojstava ili događaja. Dodavanje dugmeta na kontrolu (instancu) klase PropertyGrid je obezbeđeno unutar funkcije *AddEventPropertyTabs()*. Ova funkcija je zadužena za dodavanje kartica događaja u PropertyGrid, u slučaju da je izabran nestatički objekat. Ovo se postiže dodavanjem određenog tipa kartice (tabulara) u *PropertyGrid* kolekciju *PropertyTab*-ova.
Klasa *XEPropertyEventsTabCS* predstavlja tip kartice za C#.NET listu događaja, i dodaje se nakon Obel i VB.NET kartice, u skladu sa zahtevima zadatka. Kartice svojstava se prikazuju unutar PropertyGrid kontrole i omogućava različite prikaze osobina objekata ili nekih drugih podataka. Funkcija *PropertyDescriptorCollection GetProperties()* vraća kolekciju svojstava objekta, odnosno događaja za trenutni tabular i nakon njenog poziva je moguće popuniti *PropertyGrid* C#.NET događajima. Promena vrednosti polja (događaja) unutar PropertyTab se definiše klasom *XEPropertyEventEditor*. Ova klasa nasleđuje .NET klasu *UITypeEditor*, koja predstavlja osnovnu klasu i koristi se prilikom menjanja vrednosti nekih tipova (u ovom slučaju događaja) kroz korisničku spregu. Editori polja (*type editors*) su veoma korisni u situacijama gde tekst polje (*textbox*) nije efikasno prilikom dodele ili menjanja vrednosti složenih tipova podataka. Editor polja treba da obezbedi sledeću funkcionalnost, u slučaju kada se radi o dodavanju C#.NET događaja:

- otvaranje kartice (tab) koja sadrži izvorni kod C#,
- ako događaj ne postoji, potrebno je pozvati funkciju za generisanje koda događaja,
- u slučaju da događaj postoji, potrebno je pozvati funkciju za lociranje koda vezanog za taj događaj.

4.2 Upravljanje karticama (tab control)

Predstave modela u XE-u, bilo da se radi o grafičkoj predstavi, prikazu VB.NET izvornog koda, Obel koda ili C#.NET koda su sadržane unutar kartica (tabulara), sa ugledom na Visual Studio, koji ima sličnu organizaciju. Kartica predstavlja instancu klase *TabObject*, a prateća kontrola i funkcionalnosti su definisane unutar klase *TabbedControl*.

Tipovi, odnosno vrste kartica koje mogu biti definisane unutar XE-a (nakon dodavanje i C# kartice) su:

- *Model* - grafički prikaz modela,
- Obel - sadrži Obel kod,
- VisualBasic - sadrži VB.NET kod,
- CSharp - sadrži C#.NET kod,
- Composite - sadrži XOS composite objekte,
- Symbol - sadrži XOS symbol objekte,
- Undefined - sadrži nepoznatu kontrolu.

Čuvanje sadržaja kartice u datoteku izvršava metoda *bool SaveTab(TabObject tab, bool checkForCompileErrors)*, a pored snimanja koda u datoteku, zadužena je i za pokretanje njegovog prevođenja. Prlikom snimanja kartice, pokreće se snimanje svih kartica koje su vezane za isti model. Pre samog snimanja, pokreće se prevođenje izvornih kodova za određeni model (za koji je pokrenuto snimanje kartice). Izvorni kodovi se prevode redom: VB.NET, Obel i na kraju C#.NET. Prevodilac je definisan unutar klase *CodeEditor* i on se eksplicitno pokreće. Po završetku prevođenja, snimaju se izvorni kodovi i model u sledećem redosledu: VB.NET, C#.NET, Obel na kraju sam Model. Snimanje modela mora se pokreće poslednje,

pošto ono snima sve informacija u .XEM datoteku. Snimanje kartice se pokreće pozivom metode *Save()* klase *TabObject*. Pored izvornog koda, čuva se i vreme poslednjeg snimanja.
Dodavanje editora koda i inicijalizaciju vrši funkcija *bool AddCSEditor(TabObject tab)*. Nakon kreiranja instance *CodeEditor* klase, poziva se funkcija *void InitCodeEditorComponents(CodeEditor codeEditor, TabObject tab)* koja vrši inicijalizaciju editora koda i uvezuje ga sa karticom.

4.3 Editor koda

Editor koda se sastoji od 4 modula:

- klasa CodeEditor,
- klasa ScriptEditor,
- klasa CodeEditorManager,
- LanguageManager klasa specifična za C#.NET.

CodeEditor predstavlja osnovnu klasu editora koda, koja sadrži instancu klase *ScriptEditor*, koja predstavlja sam tekst editor specifičan za određeni programski jezik. Pored instance *ScriptEditor* klase, jedno polje čini i instanca *TabObject* klase, pomoću koje se pristupa kartici koja sadrži ratzmatrani editor koda. Glavni zadatak ove klase je da poveže spoljne događaje sa *ScriptEditor* obejktom, odnosno editorom teksta, i na taj način se ponaša kao spoljni sloj *ScriptEditor*-a.
Klasa *CodeEditorManager* obezbeđuje statične funkcije pomoću kojih komponente editora komuniciraju i podižu (okidaju) događaje. CodeEditorManager sadrži sve potrebne definicije programskih jezika i obezbeđuje funkcije za prevođenje i druge poslove. Ova klasa je usko povezana sa rezultatima prevođenja i panelom za ispis rezultata prevođenja, kao i objektima CodeEditor klase (sadrži kolekciju svih prisutnih editora u sistemu).
Klasa *ScriptEditor* implementira kontrolu koja omogućuje integrisanje skriptovanja u korisničku spregu. Osnovna svrha ove kontrole je da obezbedi tekst editor koji omogućuje:

- obeležavanje rezervisanih reči prgramskog jezika,
- uvlačenje i formatiranje teksta (*Indent*),
- prevođenje izvornog koda,
- predikciju teksta, poznatiju kao Intellisense.

Ova klasa vrši prevođenje izvornog koda i upisuje rezultate u panel za ispis rezulata prevodioca. Ovakav mehanizam je implementiran unutar funkcije *bool Compile(bool suppressSuccessMessages)*.
XOSCSLanguageManager je klasa koja predstavlja rukovaoc programskog jezika za C#.NET i kontroliše mnoge stvari, poput generisanja izvornog koda, pretrage unutar izvornog koda u cilju pronalaženja događaja, konstruktora, klasa, definicija i deklaracija objekata, itd.
Jedan od važnijih zadataka ove klase jeste generisanje izvornog koda, odnosno teksta za prikaz (model). Generisanje koda se može klasifikovati u sledeće grupe:

- generisanje tela klase modela,
- generisanje koda funkcije za obradu događaja,
- generisanje deklaracija i definicija objekata modela,
- uklanjanje generisanog koda za objekat prilikom brisanja objekta iz grafičkog prikaza.
- Promena naziva objekata na svim mestima unutar izvornog koda u slučaju da se promeni naziv objekta unutar modela.

5. TESTOVI

Ispitivanje funkcionalnosti ovog proširenja podrazumeva sledeće:

- generisanje koda prilikom dodavanja novog prikaza,
- generisanje koda prilikom otvaranja prikaza, koji ne poseduje C#.NET izvorni kod,
- generisanje koda događaja i pozicioniranje na mesto umetanja prilikom dodavanja događaja iz tabela svojstava (*Property grid*),
- promena naziva objekta unutar koda, u slučaju njegove promene u tabeli svojstava i brisanje objekta i njegovih instanci unutar koda prilikom uklanjanja objekta iz modela,
- provera funkcionalnosti prevodioca,
- pokretanje modela unutar XE i pozivanje događaja, odnosno provera njihove ispravnosti,
- pisanje C# programske podrške za već postojeći i primenjeni prikaz, i njegovo testiranje unutar ezXOSa.

Svi test slučajevi (scenariji) su uspešno izvršeni.

6. ZAKLJUČAK

Zahtevi zadatka su uspešno ispunjeni i testirani kroz različite scenarije. Implementirana je podrška za programiranje funkcionalnosti modela u C#.NET programskom jeziku i proširene su odgovarajuće kontrole i dodate nove klase koje definišu programski jezik C#.NET.

Dalji tok razvijanja C#.NET programskog podrške u XE-u bi se odvijao u smeru pisanja podrške za prikaze tipa Composite, sa ugledom na VB.NET programski sloj. Pored ove funkcionalnosti trebalo bi deteljnije definisati proveru strukture izvornog koda koja se radi prilikom prevođenja.

7. LITERATURA

[1] OASyS DNA 7.5 Internals, Internals.chm, Telvent, 2009.
[2] OASyS DNA 7.5 XETutotials, Telvent, 2009.
[3] OASyS DNA 7.5 Fast Track Course, 7.5 Fast Track Course, Telvent, 2009.
[4] Andrew Troelsen: Pro C# 2010 and the .NET 4 Platform, apress, 233 Spring Street New York.
[5] ECMA International: C# Language Specification, Rue du Rhone 114 CH-1204 Geneva, 2006.
[6] Karli Watson, Christian Nagel, Jacob Hammer Pedersen, Jon Reid, Morgan Skinner: beginning Visual C# 2010, Wiley Publishing, 10475 Crosspoint Boulevard Indianapolis, 2010.

Kratka biografija:



Marko Dojčinović rođen je u Novom Sadu 1985. god. Apsolvent je na Fakultetu tehničkih nauka odsek elektrotehnika, smer računarstvo i automatika, usmerenje računarska tehnika i računarske komunikacije.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

INTEGRACIJA LON SERVERA SA SISTEMOM KUĆNE AUTOMATIKE

LON SERVER INTEGRATION WITH A HOME AUTOMATION SYSTEM

Perica Dejanović, *Fakultet tehničkih nauka*, *Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj –*U okviru rada je izvršena integracija Lon servera sa sistemom kućne automatike, realizovane namenske Web stranice i aplikacija za nadzor i kontrolu automatskog sistema (praćenje i izmena vrednosti promenljivih Lon sistema) sa testiranjem same aplikacije i proverom mogućnosti koje iLON server nudi u takvom sistemu.*

Abstract– *As part of this project Lon server integration with a home automation system was presented, designing custom Web pages and application for monitoring and automatic control system (Data points as variables of Lon systems), and testing applications and features that iLon server offers in such a system.*

Ključne reči: *iLON, SOAP poruka, LonWorks, "pametna" zgrada.*

1. UVOD

Današnje velike kompanije – sa mnogo rasejanih pogona po većem prostoru – posvećuju mnogo vremena rešavanju problema nepotrebne potrošnje energije i poboljšanja proizvodnje, sve to da bi smanjile svoje troškove. Uređaj kojim se to može postići je iLON 100 server. iLON 100 server se može koristiti kao samostalan server ili kao deo nekog većeg kontrolnog sistema. Sa ugrađenim drajverima za industriju, kao što su standardni protokoli za LonWorks tehnologiju, SOAP/XML, Modbus, Mbus, digitalnim U/I, brojačem i mogućnošću da se na njega poveže veliki broj različitih uređaja, iLON 100 server donosi veliku fleksibilnost u upotrebi. Još jedna od velikih prednosti ovog servera je mogućnost nadziranja i upravljanja procesima sa velikih udaljenosti preko internet mreže [1].

2. iLON 100 INTERNET SERVER

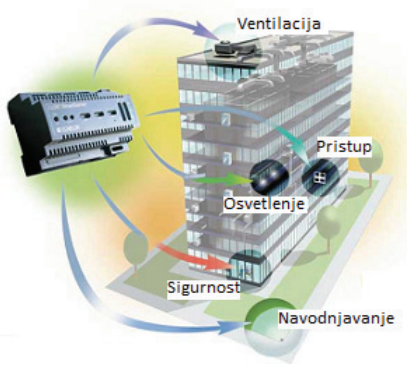
iLON 100 e3 internet server je kontroler, ruter i Web server preko kog se LonWorks, M-Bus i Modbus uređaji spajaju na zajedničke IP mreže ili na internet. iLON 100 server je svojevrstan server koji omogućava Web pristup svim podacima kojim upravlja, kao i aplikacijama za skladištenje i rad sa tim podacima.

Poseduje i mogućnost za povezivanje više LonWorks domena (Web binder), kao i mogućnost SOAP/XML Web servisnih interfejsa za upotrebu namenski urađenih Web stranica [2].

NAPOMENA:

Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Velimir Čongradac, docent.

Primena iLON servera sa LonWorks mrežnom platformom ima savršene rezultate u kontroli sistema unutar gradova, zgrada (Slika 1.), kuća i industrije. Omogućena je upotreba raznovrsnih elektronskih uređaja koji se prostim umrežavanjem preko LonWorks-a koriste za nadzor i kontrolu, u jednom sistemu. Primenom iLON servera sistem je potpuno nezavistan u radu od PC-a ili nekog drugog kontrolera.



Slika 1. *Prikaz sistema za kontrolu i upravljanje „pametnom" zgradom uz pomoć iLON 100 servera*

Da bi se omogućio bilo kakav rad servera, mora se uraditi njegova konfiguracija. Početna konfiguracija iLON 100 servera se, preporučljivo je, obavlja pomoću iLON 100 softvera kao što je LonMarker Integration Tool. Sa ovim softverom se odrađuju osnovne konfiguracije, kao što je određivanje IP adrese servera, preko koje će mu se moći direktno pristupati preko Web browser-a.

U ovom projektu nije toliko značajno koristiti se originalnim Web stranicama. Kroz dalji rad će biti posvećena pažnja izradi sopstvene aplikacije i Web stranice, koja će se koristiti samo za nadzor (kako promenljivih koje predstavljaju stanja na senzorima i uređajima tako i alarmnim promenljivim) i izmene njihovih vrednosti, bez mogućnosti ozbiljnijeg konfigurisanja iLON 100 servera.

3. PAMETNE ZGRADE

3.1. Upravljanje energijom

Proizvodnja, distribucija i korišćenje električne energije su od fundamentalnog značaja za društveni život i imaju najveći uticaj na okruženje vezano za bilo kakvu ljudsku aktivnost. Poslovno-stambenim objektima pripada udeo od 40% u ukupnoj potrošnji energije i 36% u emisiji ugljen-dioksida (CO_2). Glavni izvori energije u dvadeset-prvom veku su neobnovljivi izvori energije, a veliki problem kod njih je što ih ima u ograničenim količinama.

Zbog ovakvih ograničenja, došlo je do pojave razmišljanja u pravcu uštede energije, pri čemu je krajnji cilj smanjenje potrošnje energije u poslovnim i stambenim objektima a ujedno i redukcija emisije zagađujućih supstanci na ili ispod zakonom dozvoljenih vrednosti.

Energetska efikasnost se obično definiše kao skup aktivnosti kojima utičemo na smanjenje utroška energije prilikom njene transformacije. Ove aktivnosti formulišemo u strategije, kojima ciljano dejstvujemo na određene procese, odnosno podsisteme.[3]

Koncept „zelene zgrade“ čiji je smisao u energetskoj efikasnosti i zaštiti životne sredine, pruža sve neophodne informacije i parametre za uspešno upravljanje imovinom i servisima u poslovnim objektima. Na taj način se potpuno zaokružuje upravljanje modernom poslovnom zgradom, pri čemu pomenuti koncept predstavlja moćan poslovni alat i generator profita za vlasnike, a ujedno i izvor satisfakcije za korisnike prostora i odgovorno rešenje za ekološke potrebe zajednice.

3.2. Koncept pametnih zgrada

Nove tehnologije u segmentu poslovnih građevinskih objekata su dale efikasan odgovor na izazovni zadatak poboljšanja i optimizacije svih parametara i karakteristika uz istovremeno smanjenje troškova energenata i emisije ugljen-dioksida u okolinu. Energetska efikasnost i kontrola okruženja postaju ključni termini ze merenje performansi poslovnog objekta.

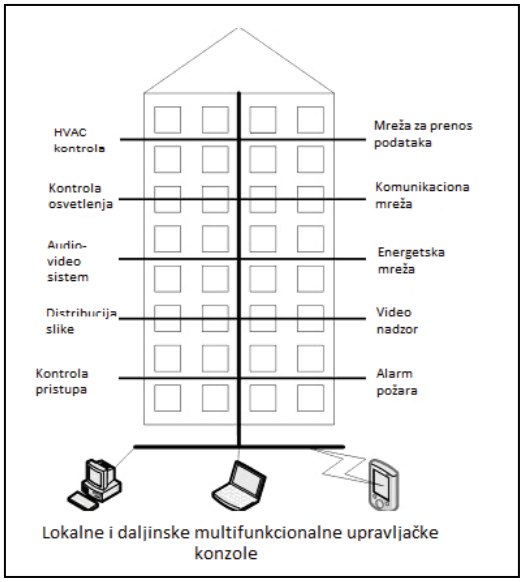
To je rezultovalo da postojanje sistema centralizovanog upravljanja poslovnom zgradom postane neformalni tržišni standard bez koga se ne može zamisliti moderan poslovni objekat.

3.3. Sistemi centralizovanog upravljanja „pametnim“ zgradama

Suština pametnih zgrada je posedovanje naprednih i integrisanih sistema za automatizaciju zgrada i telekomunikacionih sistema. Tradicionalni način dizajniranja i konstruisanja ovakve zgrade je u velikoj meri u dizajniranju, instaliranju i pokretanju svakog sistema zasebno.

Savremeni, integrisani centralni sistemi za nadzor i upravljanje u poslovno-stambenim objektima (u daljem tekstu *Integrisani BMS*), su bazirani na arhitekturi sa distribuiranim kontrolerima DCS (Distributed Control System) uz podršku svih standardih komunikacionih protokola.[4-5]

Integrisani BMS ima mogućnost nadzora i upravljanja nad zajedničkim podsistemima objekta. U tom poslu, on se oslanja na određene parametre dobijene merenjem ambijentalnih vrednosti u kancelarijama, sobama-apartmanima, zajedničkim prostorijama, server salama, itd., ukoliko su oni deo objekta. Dalje, BMS upravlja ambijentom u tim prostorijama i na taj način ostvaruje ciljane uslove koji su primereni za određeni prostor.
Integrisani BMS objedinjava sve podsisteme objekta, sa pružanjem mogućnosti za integracijom novih podsistema, putem standardnih komunikacionih protokola i razvijenom vezom ka servisnim službama koje vrše monitoring i održavanje objekta (Slika 2.).



Slika 2. *Integrisani sistem upravljanja*

4. SOAP PORUKE

SOAP (Simple Object Access Protocol) je protokol baziran na XML-u (Extensible Markup Language) kao svom formatu poruke. On definiše set pravila koja se upotrebljavaju za određivanje strukture poruka. Takve poruke se mogu koristiti za prosto jednosmerno slanje, ali se uspešno koriste i za zahtev-odgovor dialoge. SOAP nije vezan posebno ni za jedan transportni protokol, premda je HTTP (HyperText Transfer Protocol) najpopularniji za prenos poruka. Takođe, nije vezan ni za jedan operativni sistem ili programski jezik, tako da, teoretski, klijent i server (između kojih se vrši dialog) mogu biti pokretani na bilo kojoj platformi napisanoj na bilo kom jeziku. Sve ovo dok i jedan i drugi mogu da formulišu i razumeju SOAP poruke. Kao takav, SOAP protokol je veoma koristan pri razvoju aplikacija koje će se koristiti kao servis na Internet-u.

Ovaj XML bazirani protokol sastoji se iz tri dela:

- Envelop (omotnice) – koja definiše šta se nalazi u poruci i kako je treba procesirati
- Set pravila za enkodovanje
- Pravilo za predstavljanje „*call*“ i „*response*“ procedura [6]

Pošto je SOAP po prvi put predstavljen javnosti, vremenom je postao pozadinski nivo mnogo kompleksnijih Web servisa, baziranih na WSDL-u (Web Service Description Language). iLON 100 server poseduje WSDL fajl koji se koristi pri izradi Web aplikacija.

5. KREIRANJE NAMENSKIH WEB STRANICA I APLIKACIJA ZA ČITANJE I IZMENU VREDNOSTI PROMENLJIVIH

Jedan od načina kreiranja i modifikovanja sopstvene, namenski rađene Web stranice, za čitanje i upis novih vrednosti svojevrsnih promenljivih (data point-a) jeste upotreba iLON 100 softvera, tj. iLON Vision softvera.

Drugi način je koristeći SOAP/XML interfejsa koji je obezbedio iLON100 server. Konfiguracija svake iLON 100 aplikacije je smeštena u XML fajl. iLON 100 server čita ovaj fajl tokom svog periodičnog ciklusa, i setuje operativne parametre svake aplikacije u zavisnosti od podataka u XML-u koje ta aplikacija poseduje.

Treći način jeste koristeći SOAP interfejs kao Web servis. Aplikacija ovog tipa se može uraditi u nekom od programskih jezika, u našem slučaju Visual Basic .NET C-u.[7]

5.1. Izrada Web stranica koristeći SOAP/XML servis

iLON 100 server poseduje set SOAP funkcija koje se mogu iskoristiti da se kreira i provede u delo konfiguracija za svaku aplikaciju. Svaki put kada se pozove neka od ovih funkcija, SOAP poruka bude poslana iLON 100 serveru. Sadržaj SOAP poruke se zasniva na podacima kojim je snabdevena funkcija. iLON server čita sadržaj poruke, upisuje taj sadržaj u poseban XML fajl (WSDL fajl) i po njemu prilagođava operativne parametre aplikacije na koju se funkcija odnosi. Sve se ovo odvija dok je iLON 100 server operativan (u radu). Ovakav način konfigurisanja nije naročito popularan.

Pored ovih funkcija, za konfigurisanje, SOAP/XML interfejs takođe poseduje i set funkcija koje se mogu koristiti za pristup, čitanje i obradu podataka generisanih iLON 100 aplikacijama. Tako da se sa SOAP/XML interfejsom može i nadgledati proces.

Svaki iLON 100 server poseduje WSDL (Web Service Description Language) fajl. Ovaj fajl definiše iLON 100 SOAP/XML interfejs, i sadrži sve informacije koje će aplikaciji trebati za upotrebu SOAP/XML interfejsa.

Primer XML fajla za čitanje vrednosti promenljivih iLON 100 servera (zahtev za promenljivom po imenu i indeksu):

```
<iLONDataServer>
<DPType>
<UCPTindex>1</UCPTindex>
<UCPTname>NVL</UCPTname>
        <DP>
                <UCPTindex>0</UCPTindex>
        </DP>
        <DP>
                <UCPTpointName>NVL_nvo03Switch</UCPTp
                ointName>
        </DP>
…….
…….
</DPType>
</iLONDataServer>
```

Odgovor na ovaj zahtev je oblika:

```
<iLONDataServer>
<UCPTfaultCount>0</UCPTfaultCount>
<DPType>
<UCPTindex>1</UCPTindex>
<UCPTname>NVL</UCPTname>
<DP>
<UCPTindex>0</UCPTindex>
<UCPTpointName> NVL_nvo03Switch
</UCPTpointName>
                        ……
                        ……
                        <UCPTvalue>0.0 0</UCPTvalue>
                        ……
                        …….
                        <UCPTpriority>250
                        </UCPTpriority>
                </DP>
                …….
                …….
        </DPType>
</iLONDataServer>
```

<DPType> :specifikacija tipa promenljive (NVL, NVC,…),
<UCPTindex> : indeks dodeljen promenljivoj,
<UCPTpointName> :ime promenljive,
<UCPTvalue> :trenutna vrednost promenljive,
<UCPTpointStatus> : status promenljive. Njega setuje alarm generator ili dojava alarmnog stanja,
<UCPTpriority> :nivo prioriteta trenutno zadate promenljive (0-255).
On odlučuje koja aplikacija ima prvenstvo pristupa. Najviši nivo je 255, iako se eksplicitno ne navede, uzimaće se on.[7]

Kao što se vidi, ovakav način razmene poruka, ne samo da nije preporučljiv, nego je i vizuelno loš. U porukama koje se dobijaju kao odgovor ima mnogo linija sa različitim informacijama, koje su nepotrebne ako je pažnja usredsređena samo na vrednosti promenljive.

5.2. Korištenje SOAP interfejsa kao Web servisa

Da bi se iskoristila mogućnost da se radi sa SOAP interfejsom kao Web servisom, može se opredeliti za bilo koji razvojni alat koji ima mogućnost za pozivanje standardnog Web servisa sa iLON 100 SOAP interfejsom. Jedan od takvih alata je Microsoft Visual Studio .NET (u našem slučaju verzija 2010 Professional).

Kada se piše ovakva aplikacija, neki alati imaju mogućnost da "pokupe" iLON 100 WSDL fajl i da automatski naprave klasu za slanje i primanje SOAP poruka. Na ovaj način funkcioniše i Microsoft Visual Basic .NET alat. Sa njim se može konfigurisati iLON 100 server ili samo pratiti promenljive na njemu.

Visual Basic .NET aplikacija daje vizuelno lepši i pregledniji nadzor svih informacija nama potrebnih (u ovom slučaju vrednosti promenljive). Funkcije pisane u Visual Basic .NET-u su analogne onim u XML-u, tako da, recimo, funkcija za čitanje vrednosti promenljive izgleda ovako:

```
Dim dsRead As New ilon100.DS_Read
dsRead.DPType=Newilon100.DS_ReadDPType(1) {}
dsRead.DPType(0) = New ilon100.DS_ReadDPType
dsRead.DPType(0).UCPTname=moja
dsRead.DPType(0).DP=Newilon100.DS_ReadDPTypeDP(1) {}
dsRead.DPType(0).DP(0)=Newilon100.DS_ReadDPTypeDP
dsRead.DPType(0).DP(0).UCPTpointName=Data_Point.Text
Dim dsReadInfo As ilon100.DS_ReadInfo
dsReadInfo = New ilon100.DS_ReadInfo()

dsReadInfo = ilon.DataServer_Read(dsRead)

Data_Point_Info.Items.Add(dsReadInfo.DPType(0).DP(0).UCPTpointN
ame)
Data_Point_Info.Items.Add(dsReadInfo.DPType(0).DP(0).UCPTindex)
Data_Point_Info.Items.Add(dsReadInfo.DPType(0).DP(0).UCPTpriorit
y)
```

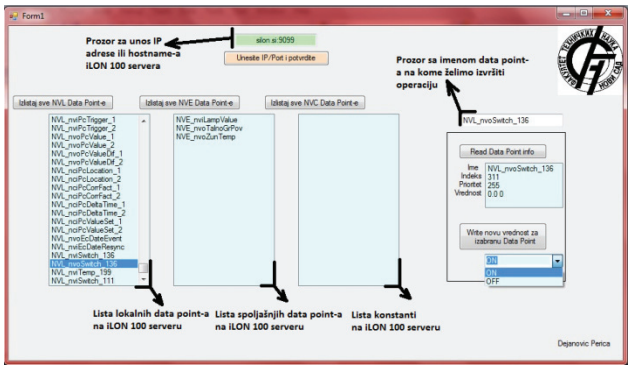
Data_Point_Info.Items.Add(dsReadInfo.DPType(0).DP(0).UCPTvalue.Value)

Ova funkcija će, za promenljivu sa zadatim tipom i imenom, isčitati sve njene podatke, s tim što će ispisati samo:

- UCPTpointName : ime promenljive,
- UCPTindex : indeks promenljive,
- UCPTpriority : prioritet promenljive,
- UCPTvalue.Value : trenutnu vrednost promenljive.

**vrednosti koje nam funkcija ispisuje su vrednosti koje su eksplicitno tražene, što ne znači da ne može ispisati i druge.

Naša aplikacija će biti u stanju da uradi sledeće: da isčitava vrednosti svih promenljivih koje se nalaze na iLON 100 serveru (posebno za svaki tip: NVL, NVE i NVC); da pročita zahtevane vrednosti promenljivih po našem izboru; i mogućnost izmene vrednosti promenljivih po našem izboru (Slika 3.). Za ovo je potrebna internet konekcija preko koje će se pristupati iLON 100 serveru koristeći njegovu IP adresu ili hostname.



Slika 3. *Izlistavanje promenljivih i izbor promenljive na kojoj se izvršava operacija*

Ovakav način nadgledanja i promene vrednosti željenih promenljivih je veoma brz i jednostavan. Kao što je rečeno u prethodnom tekstu, konfigurisanje iLON 100 servera je najbolje, najlakše i najsigurnije izvoditi pomoću originalnih (proizvođačevih) Web stranica, dok je za nadgledanje i promene vrednosti promenljivih, najjednostavnije kreirati sopstvene Web stranice ili, još bolje, aplikaciju.

6. ZAKLJUČAK

U narednom periodu nastaviće se rast cena energije i energenata i to će direktno uticati na porast troškova života i poslovanja. Stoga se prioritet daje održivoj potrošnji energije, kroz racionalno planiranje same potrošnje i implementaciju mera energetske efikasnosti.

Također, sve više će se posvećivati pažnja stvaranju bezbednog, komfornog, produktivnog i ekonomičnog radnog ili životnog okruženja, uz istovremenu štednju energenata, smanjivanje troškova održavanja i opšte društvenu korisnost zaštite životne sredine.

Jedna od mogućnosti dobijanja ovakvih sistema je korištenjem, tj. integracijom Lon servera sa sistemima kućne automatike. Zbog savršenih rezultata koje iLON server, sa LonWorks mrežnom platformom, ima u kontroli sistema unutar gradova, zgrada, kuća i industrije, u potpunosti zaslužuje poverenje korisnika. Omogućen pristup za upravljanje procesima sa velikih udaljenosti iLON serveru preko internet mreže, omogućena upotreba raznovrsnih elektronskih uređaja koji se prostim umrežavanjem preko LonWorks-a koriste za nadzor i kontrolu, u jednom sistemu, i nezavistan rad od PC-a ili nekog drugog kontrolera, sasvim su dovoljni razlozi za realizaciju jednog ovakvog sistema.

7. LITERATURA

[1] Save energy and improve operations with the i.LONSmartServer, ECHELON

http://www.echelon.com/solutions/eem/default.htm

[2] i.LON® 100 e3 User's Guide, ECHELON;

http://www.echelon.com/support/documentation/manuals/cis/078-0310-01B.pdf

[3] Energy management systems & direct digital control, Richard A. Panke, The Fairmont Press 2002.

[4] Joseph La Fauci: PLC or DCS: selection and trends, *ISA Transactions* vol. 36, Issue 1, strane 21-28, 1997.

[5] AM Seal: Distributed control systems (DCS), *Practical Process Control*, strane 95-105, 1998.

[6] What is SOAP,

http://www.soapuser.com/basics1.html

[7] i.LON® 100 e3 ProgrammersReference, ECHELON

http://www.echelon.com/support/documentation/manuals/cis/078-0250-01E.pdf

Kratka biografija:



Perica Dejanović rođen je 01.06.1983. u Sisku. Završio srednju tehničku školu "Buda Davidović" u Obrenovcu i Višu elektrotehničku školu u Beogradu. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva - Automatika i upravljanje sistemima odbranio je 2011. god.

The Use of Data Mining for Basketball Matches Outcomes Prediction

Dragan Miljković, Ljubiša Gajić, Aleksandar Kovačević, Zora Konjović
Faculty of Technical Sciences, Novi Sad, Republic of Serbia
dramilj@hotmail.com, ljubisa_gajic@yahoo.com, kocha78@uns.ac.rs, ftn_zora@uns.ac.rs

***Abstract* — Sport result prediction is nowadays very popular among fans around the world, which particularly contributed to the expansion of sports betting. This makes the problem of predicting the results of sporting events, a new and interesting challenge. Consequently systems dealing with this problem are developed every day. This paper presents one such system, which uses data mining techniques in order to predict the outcomes of basketball games in NBA (National Basketball Association) league. The problem of predicting the game result is formalized as a classification problem, where the Naive Bayes method is used. Besides actual result, for each game system calculates the spread, by using multivariate linear regression. The MVC Model 2 pattern based software system is implemented. The system was evaluated on the dataset comprising 778 games from the regular part of the 2009/2010 NBA season and it correctly predicted the winners of about 67% of matches.**

I. INTRODUCTION

Sport result prediction is nowadays very popular among fans around the world, which is particularly contributed to the expansion of sports betting. This is particularly evident for the most popular sports such as football and basketball. Many people have developed various systems with the aim of providing the best possible prediction of the winners of sporting events. The main problems with such systems are that users are often swayed by emotions or the systems do not work with the right set of data. One approach to this prediction problem is the use of data mining techniques, which will be demonstrated in this paper. This paper presents a system which predicts the outcomes and calculates the spread of NBA games. Prediction problem is formalized as a problem of classification, while for the calculation of the spread multivariate linear regression is used. Several classification methods were tested and among them, Naive Bayes method is selected because it provided the best results during testing phase. The system was evaluated on the collection of 778 games from 2009/2010 season and system has correctly predicted the winners of about 67% of matches.

The rest of the paper is organized in five sections describing related work, the problem and solution method, software implementation, system's evaluation and prediction results, and concluding remarks.

II. RELATED WORK

There are many papers dealing with the topic of predicting the results of sports events, and most of them are related to football. A variety of techniques are used, ranging from data mining to statistical methods. In [1] authors present the approach for predicting the results of football matches called FRES (Football Result Expert System). A combination of Bayes inference and rule-based reasoning is used. Each game is represented as a series of tides (flows). Probabilities of some events, like possible player substitute or change of formation, are calculated on the basis of previous data in each tide and are used to fire certain rules to determine the decisions for the next tide. The system was applied to the FIFA World Cup 2002 matches and it correctly predicted the champion and runner-up, and also 6 of 8 participants in the quarter-finals.

A common approach to this subject matter is also the use of artificial neural networks, which can be seen in [2] and [3]. The idea behind these works is the use of a number of statistical data to train the neural network (multi-layer perceptron in most of the cases). The system presented in [3] is evaluated in the international competition in result predictions - Top Tipper [4] and proved to be satisfactory by comparing it with other competitors. Average percentage of correct predictions for ranged from 54.6% for English Football Premier League to 67.5% of Super Rugby.

Naive Bayes classifiers are used to predict the Cy Young Award winners in American baseball in [5]. In the period since 1967 to 2006 this system successfully predicted over 80% of the prize winners.

A form of Elo rating system [6], modified is such way that to allow "home ground advantage", is used in [7] for generating prediction probabilities. The Minimum Message Length [8] is used as a parameter estimation technique.

In the basketball domain, data mining techniques are mostly used for purposes other than the prediction of match outcomes. Authors in [9] describe a system based on the statistical data and fuzzy evaluation which provides a performance evaluation of players and can also predict their performances in the upcoming matches. A solution for supporting the basketball coaches in making tactical/technical decisions during matches and also for pre-game and post-game analysis is presented in [10]. The system is based on the use of apriori algorithm and is tested on data from the Italian first division.

Lately a lot of work is done on the subject of processing and extracting relevant information from the video clips of sports matches. Solutions that address this topic are described in [11] and [12].

NAPOMENA:
a) Ovaj rad proistekao je iz diplomskog-master rada Dragana Miljkovića. Mentor je bila prof. dr Zora Konjović.
b) Rad je prethodno publikovan na konferenciji SISY, Subotica, 2010.

III. THE PROBLEM AND THE SOLUTION

The problem intended to be solved by the software proposed in this paper is the following one: predicting the outcomes of the NBA matches and calculating the spread for NBA games.

The outcome predicting problem is formalized as a classification problem, where the match outcomes belong to exactly one of the two categories (classes): those in which the host will win and those in which the visiting team will take the victory (unlike some other sports, in basketball there is no draws and teams are forced to play extra time until there a winner is determined).

Spread (which is used only in the sports betting) represents an advantage in the number of points given to one of the teams in order to equalize their chances for victory in the eyes of bookmakers.

Each game is described with record consisting of 141 attributes, which are related to the outcome of the match, and the participating teams. For each team there are 2 groups of attributes.

The first group consists of standard basketball statistics (field goals made, field goals attempted, 3 pointers, free throws, rebounds, blocked shots, fouls, etc). The attributes in the second group are composed of the information about league standings (number of wins and loses, home and away wins, current streak etc). For the first attribute group overall statistics are taken into account, as well as those from home games and those from away matches. Complete statistics attribute set with descriptions can be seen in Table I, while standings attribute set is described in Table II.

Multivariate linear regression is used for calculation of the spread for NBA games where the point difference was used as a target variable, while all other attributes were used as explanatory variables.

TABLE I.
STATISTIC ATTRIBUTE SET

Attribute	Description
FGM	Field goal made per game
FGA	Field goal attempted per game
FGP	Field goal percentage per game
3M	3-pointers made per game
3A	3-pointers attempted per game
3P	3-pointers percentage per game
FTM	Free throws made per game
FTA	Free throws attempted per game
FTP	Free throws percentage per game
OR	Offensive rebounds per game
DR	Defensive rebounds per game
TR	Total rebounds per game
AS	Assists per game
TO	Turnovers per game
ST	Steals per game
BL	Blocks per game
F	Fouls per game
P	Points per game

TABLE II.
STANDINGS ATTRIBUTE SET

Attribute	Description
Won	Total number of wins
Lost	Total number of losses
Pct	Percent of wins
Homewon	Number of games won at home
Homelost	Number of games lost at home
Roadwon	Number of games won away
Roadlost	Number of games lost away
Divwon	Number of games won in division
Divlost	Number of games lost in division
Confwon	Number of games won in conference
Conflost	Number of games lost in conference
Streak	Number of consecutive wins/losses
L10won	Number of wins in last 10 games
L10lost	Number of losses in last 10 games

In order to improve the performance of the classification in the system, feature selection and normalization were used. Experiments were carried out with the variety of classification techniques: decision trees, k nearest neighbors, Naive Bayes and support vector machines. The best results were obtained by the Naive Bayes classifier combined with normalization and feature selection.

IV. SOFTWARE IMPLEMENTATION

For development of a classification and regression models that are used in the system, RapidMiner [15] was selected as a tool. Also, RapidMiner's implementation of naive Bayes method and multivariate linear regression are used in application itself in the form of Java libraries.

All data used in this software are stored in MySQL relational database and accessed using JDBC (Java Database Connectivity) drivers. Database schema can be seen in Figure 1.

The complete system for result prediction was implemented in Java programming language using MVC Model 2 pattern [16] based on the synchronous work of three components: the model, visualization layer and controller. Figure 2 shows these components and how they interact together.

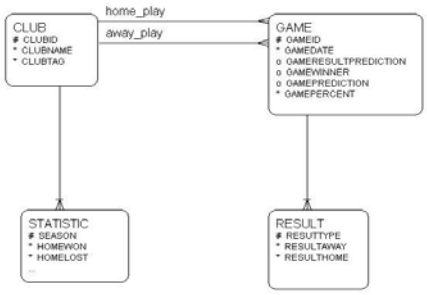


Figure 1. Database schema

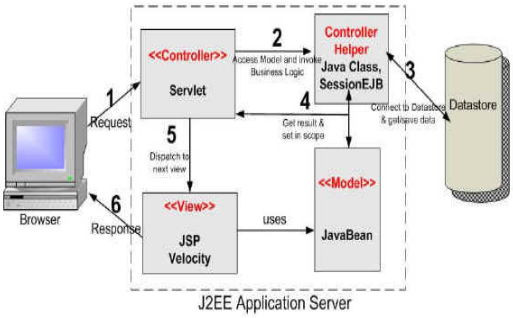


Figure 2. MVC Model 2 pattern

The model component models data and processes. This component interacts with the database and executes operations over data. Entity JavaBeans components, Java classes mapped to a corresponding table in relational database are used as model. For communication with the database there are Controller Helper components, including Session Enterprise JavaBeans classes and DAO (Data Access Object).

The view component (visualization layer) visualizes data provided by the model component. JSP (Java Server Pages) with JSTL (Java Server Pages Standard Tag Library), EL (Expression Language) and JavaScript is used in this software.

The controller manages the visualization layer and the model. All user actions go through this component which makes the decision on which application part will be executed and displayed. The roles of controller in this application perform servlets, more specifically Java Servlet classes.

The activity diagram is displayed on Figure 3.

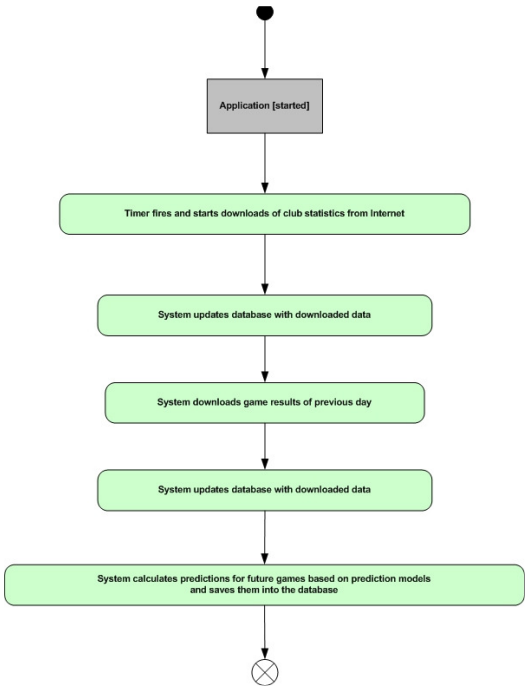


Figure 3. UML Activity diagram

From the Figure 3 one can see that the application operates in the following manner:

- Fresh information about the games that were played previous day, as well as new statistical data for all teams are downloaded.
- Based on these data and prediction models the system calculates the predictions for future games in the form of probabilities (for each team the likelihoods of winning a match are given).

The application allows user to view predictions for the upcoming matches, as well as the results of games already played and previous predictions. Some screenshots can be seen in Figures 4 and 5.

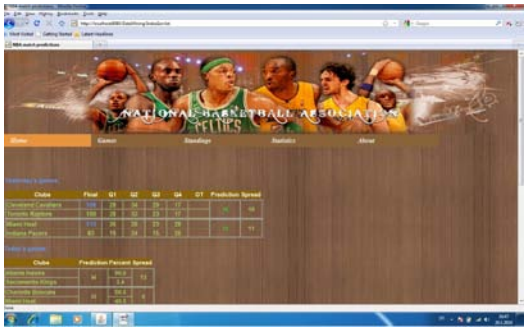


Figure 4. Home page

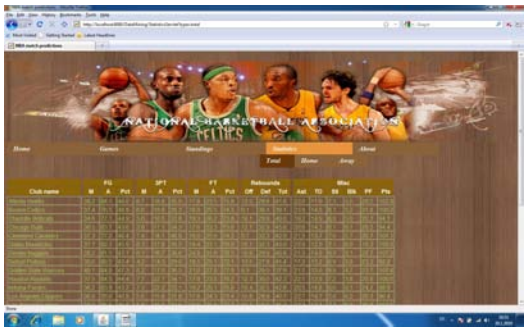


Figure 5. Statistics page

V. EVALUATION AND RESULTS

In order to evaluate the system, the statistical data of games that have already been played during a 2009/2010 season were used. The dataset consists of 778 games. All data were collected from the official NBA website [13] and part of the Yahoo website dedicated to NBA league [14]. One example from the dataset is given in Table III. This example represents the game between Sacramento Kings and Houston Rockets, played at April 12th, 2010 in Sacramento, California.

K fold cross-validation was used for evaluation of the system. In this validation method the dataset is divided into *k* subsets and each of them exactly once used as a test set, and in other *k-1* iterations as a training set. 10 fold cross-validation was used. Accuracy was used as the

performance measure, and it is calculated by the following formula:

$$accuracy = \frac{number\ of\ correct\ \ predictions}{total\ number\ of\ \ predictions} \quad (1)$$

TABLE III. DATASET EXAMPLE

Attribute	Sacramento Kings	Houston Rockets
Won	25	42
Lost	55	39
Pct	0.313	0.519
Homewon	18	23
Homelost	22	18
Roadwon	7	18
Roadlost	33	22
Divwon	5	9
Divlost	10	7
Confwon	16	27
Conflost	34	24
Streak	-1	-1
L10won	1	5
L10lost	9	5
FGM	38.3	37.7
FGA	84.1	84.4
FGP	45.6	44.7
3M	5.9	7.9
3A	16.9	22.4
3P	34.9	35.2
FTM	17.4	19.0
FTA	24.0	24.7
FTP	72.6	77.2
OR	11.9	11.8
DR	30.7	30.1
TR	42.6	42.0
AS	20.5	21.8
TO	14.2	13.8
ST	6.9	7.1
BL	4.5	3.9
F	22.3	20.9
P	100.0	102.4
Winning probabilities	42.8	57.2
Actual result	107	117

The system's achieved accuracy was of 67%, which means that it successfully predicted the outcome of more than two-thirds of the games in that period (a total of 778 games). This can be considered a very good result by comparing it with other systems. For example, sports journalists from CBS had the percentage accuracy of 68.3% in the previous season [17]. The overall results in terms of precision and recall are given in Table IV.

TABLE IV.
CONFUSION MATRIX

		Predictions		Recall
		Home win	Away win	
Actual results	Home win	354	100	77,97%
	Away win	156	168	51,85%
Precision		86,34%	62,69%	

As for the prediction of spread, the system has successfully provided predictions for 78 of 778 matches (~ 10%), which is the expected result, because it is very hard to predict the exact difference in which basketball game ends and the goal of this attempt was to provide only approximate information on the possible difference.

VI. CONCLUSION

This paper presents a system for predicting the outcomes of NBA league matches. The application provides predictions about the winner of the match (home or visiting team) and the value of spread for the match. Naive Bayes classification method and multivariate linear regression were used to perform these tasks. The achieved results are satisfactory and in line with expectations.

Future plans include experimenting with the system by applying it to other sports domains such as various football competitions, American football, hockey etc. and also experiments with some other methods of classification, such as neural networks. .

REFERENCES

[1] Byongho Min, Jinhyuck Kim, Chonghyoun Choe, Hyeonsang Eon, Robert Ian (Bob) McKay, "A Compound Framework for Sports Prediction: The Case Study of Football", *Knowledge-Based Systems*, vol. 21, issue 7, October 2008, pp. 551-562, Elsevier Science Publishers B. V. Amsterdam, The Netherlands

[2] Michael Baulch, "Using Machine Learning to Predict the Results of Sporting Matches", Department of Computer Science and Electrical Engineering, University of Queensland

[3] Alan McCabe, Jarrod Trevathan, "Artificial Intelligence in Sports Prediction", *Proceedings of the Fifth International Conference on Information Technology: New Generations*, 2008, pp. 1194-1197, IEEE Computer Society

[4] http://www.toptipper.com

[5] Lloyd Smith, Bret Lipscomb, Adam Simkins, "Data Mining in Sports: Predicting Cy Young Award Winners", *Journal of Computing Sciences in Colleges,* vol. 22, issue 4, April 2007, pp. 115-121, Consortium for Computing Sciences in Colleges

[6] http://en.wikipedia.org/wiki/Elo_rating_system

[7] Ryan Baird, "Probabilistic Sports Prediction Using Machine Learning Honors Thesis", School of Computer Science and Software Engineering, Monash University, Clayton, Vic.

[8] http://www.csse.monash.edu.au/~lloyd/tildeMML/

[9] Mourad Atlas, Yan-Qing Zhang, "Fuzzy Neural Agents for Online NBA Scouting", *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 58-63, IEEE Computer Society

[10] Guiseppe Polese, Massimiliano Troiano, Genoveffa Tortora, "A Data Mining Based System Supporting Tactical Decisions", *SEKE '02: Proceedings of the 14th international conference on Software engineering and knowledge engineering,* 2002, pp. 681-684, ACM

[11] Sebastian Lühr, Mihai Lazarescu, "A Visual Data Analysis Tool for Sport Player Performance Benchmarking, Comparison and Change Detection", *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence*, vol. 01, 2007, pp. 289-296, IEEE Computer Society

[12] Lawrence Y. Deng, Yi-Jen Liu, "Semantic Analysis and Video Event Mining in Sports Video", *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications – Workshops*, 2008, pp. 1517-1522, IEEE Computer Society

[13] http://www.nba.com

[14] http://sports.yahoo.com/nba

[15] http://www.rapid-i.com/

[16] Branko Milosavljević, Milan Vidaković, „Java i internet programiranje", 2007, FTN Izdavaštvo, Novi Sad

[17] http://www.cbssports.com/nba/exclus1ives

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.3

RAZVOJ OPC-DA MODULA ZA INTEGRACIJU HETEROGENIH SISTEMA SA RADOM U REALNOM VREMENU

REALIZATION OF OPC-DA BASED MODULE FOR HETEROGENOUS REAL-TIME SYSTEMS INTEGRATION

Dragan Krajnović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Rad obuhvata realizaciju integracionog modula za komunikaciju i dobavljanje podataka sa SCADA OPC poslužioca, opis realizacije, analizu, rezultate ispitivanja i opis testova vršenih nad realizovanim rešenjem.*

Abstract – *This paper describes implementation of module for communication and data acquisition from SCADA OPC servers, observations and analysis regarding designed solution, and specification and results of various tests aimed to verify its functionality and performances.*

Ključne reči: *SCADA, COM, OPC, CORBA, konfiguracija, komunikacija, veza*

Key words: *SCADA, COM, OPC, CORBA , configuration, communication, connection.*

1. UVOD

Jedan od izazova savremenog računarstva je razvoj integrisanih upravljačkih sistema namenjenih za nadzor i upravljanje složenim tehničkim sistemima. Suštinski zadatak ovakvih rešenja je povećanje efikasnosti tehničkog sistema kao rezultat potpunijeg/preciznijeg poznavanja njegovog stanja i primene složenih algoritama nadzora i upravljanja. Ovakvi sistemi se bez izuzetka odlikuju distribucijom fizičkih objekata, kao i različitih računarskih komponenti. Zato je problem standardizacije međusobne komunikacije između heterogenih računarskih podsistema na različitim nivoima hijerarhije vrlo važan.

Jedan od najboljih primera ovakvih tehničkih sistema je tzv. ***smart grid***, ili pametna mreža za isporuku električne energije. U suštini, smart grid podrazumeva primenu integrisanog sistema praćenja, analize, kontrole i komunikacije sistema za isporuku električne energije, u cilju efikasnijeg korišćenja propusne moći distributivnog sistema, uz istovremeno smanjenje gubitaka i potrošnje. Prvi sistemi ovog tipa su implementirani korišćenjem analognih tehnologija, ali je tek digitalizacija brojila i primena najnovijih računarskih tehnologija omogućila razvoj sofisticiranijih mreža. Povećanje kapaciteta prenosa podataka je učinila konceptualno mogućim da se očitavanje, merenje i kontrola uređaja u procesu proizvodnje, prenosa, distribucije i potrošnje električne energije podigne na viši nivo.

Ovi sistemi mogu razmenjivati informacije o stanju mreže sa korisnicima, operatorima i kontrolnim uređajima, omogućavajući potrošaču da dinamički reaguje na promene u stanju mreže.

SCADA sistem je namenjen nadgledanju, kontroli i prikupljanju podataka. Sistem vrši prikupljanje podataka sa senzora smeštenih u energetskim postrojenjima, fabrikama, na polju i na drugim udaljenim mestima, a zatim šalje podatke centralnom računaru koji na osnovu tih podataka vrši odgovarajuće upravljanje i kontrolu distribuiranog proizvodnog sistema. SCADA-e se koriste u mnogim industrijskim granama kao što su vodovodni, energetski, gasovodni sistemi, industrijski proizvodni sistemi, i mnogi drugi. U slučaju integracije korišćenjem OPC (*OLE for Process Control*) sprege, vezu sa SCADA-om predstavlja OPC DA server koji pruža standardne OPC sprege. Podaci predstavljaju stanja prekidača, vrednosti mernih uredjaja, obaveštenja i upozorenja o prekoračenju odredjene vrednosti i sl. OLE specifikacija predviđa DCOM (*Distributed Component Object Model*) kao tehnologiju za komunikaciju između programskih komponenti distribuiranih umreženih računara.

Vrh hijerarhije logičkog modela predstavlja OPC server objekat. To je COM objekat sa kojim aplikacije prvo uspostavljaju vezu. OPC server sadrži OPC grupe (*OPC Group*) i OPC tačke (*OPC Item*). OPC specifikacija opisuje OPC COM objekte i njihove sprege implementirane od strane OPC servera. OPC klijent uspostavlja komunikaciju sa serverom. OPC server se sastoji od više objekata: server, grupa i merna tačka. OPC server objekat održava informacije o serveru, služi kao kontejner OPC grupa objekata. OPC grupa objekata omogućava mehanizme za dobavljanje i logičku organizaciju OPC tačaka.

U cilju povećanja fleksibilnosti rešenja opisanog u radu, dodatno je korišćeno programsko okruženje CORBA (*Common Object Request Broker Architecture*), koje omogućava da programske komponente pisane u različitim programskim jezicima i koje se izvršavaju na različitim računarima rade kao jedna celina.

2. ANALIZA PROBLEMA

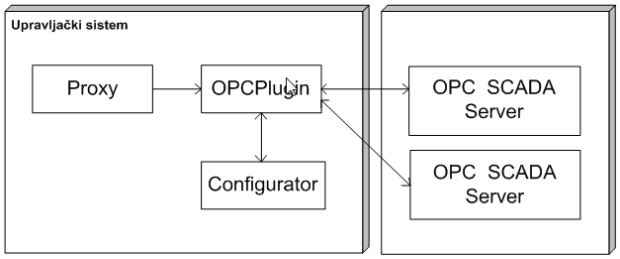
Osnovni zadatak ovog rada je razvoj integracionog modula za komunikaciju OPC protokolom (u daljem tekstu OPCPlugin). OPCPlugin treba da omogući komunikaciju i akviziciju podataka sa SCADA OPC DA servera (u daljem tekstu: SCADA server).

NAPOMENA:
Ovaj rad proistekao je iz diploskog-master rada čiji mentor je dr Branislav Atlagić, red. prof.

Podaci koji se prikupljaju sa SCADA servera mogu predstavljati celobrojne - digitalne vrednosti, ili realne - analogne vrednosti. OPC tačke (*data points*) predstavljaju vrednosti merenja, stanja prekidača (rasklopne opreme) i položaj teretnih menjača unutar elektroenergetskih postrojenja. Svaka od primljenih vrednosti može posedovati dodatnu informaciju vezanu za datu tačku.

Tačke čije je vrednosti potrebno čitati sa SCADA servera se identifikuju imenom tačke. Potrebno je napraviti listu svih tačaka čije se vrednosti čitaju sa SCADA servera. Pri tome, SCADA sistem poseduje internu logiku postavljanja stanja servera. Zbog toga, različiti SCADA serveri imaju različite konstante kojima opisuju stanja digitalnih vrednosti. Na primer, vrednost 0 primljena kao vrednost prekidača se može predstavljati stanje "otvoren" ili "zatvoren" u zavisnosti od SCADA servera. Zato svaka OPC tačka, sem imena, mora posedovati i tip fizičkog uređaja kojem pripada, kako bi mapiranje tačke u model odredišnog sistema bilo uspešno.

Pozicija razvijene komponente je prikazana na slici 1.



Slika 1: Pozicija razvijene komponente

OPC Plugin mora biti u mogućnosti da uspostavi vezu sa SCADA serverom i da izvrši proveru stanja funkcijom definisanom OPC standardom ili na drugi način definisan od strane klijenta (*vendor specific*). Dužnost OPCPlugina je da pronadje aktivan SCADA OPC server i da sa njim uspostavi vezu. Podaci se čitaju isključivo sa aktivnog SCADA servera. U zavisnosti od uspešnosti uspostavljanja veze sa SCADA OPC serverom i trenutnim stanjem uspostavljenje veze, OPCPlugin će započeti akviziciju podataka ili nastaviti da traži aktivan SCADA server.

OPCPlugin treba omogućiti čitanje vrednosti sa SCADA servera sinhrono i asinhrono. U slučaju asinhrone komunikacije potrebno je obezbediti mehanizam provere validnosti komunikacionog kanala (*keep-alive mechanism*).

3. KONCEPT REŠENJA

Zaduženja i funkcije OPCPlugina su sledeća:

- konfiguracija tačaka i komunikacionog kanala,
- uspostavljanje veze sa SCADA serverom i praćenje njegovog stanja,
- čitanje vrednosti tačaka sinhronim ili asinhronim mehanizmima,
- mapiranje pročitanih vrednosti,
- slanje mapiranih vrednosti višim slojevim sistema.

Plugin komponenta sinhronizuje svoj rad prateći automat stanja višeg programskog nivoa. Proxy, a samim tim i plugin, kao deo sistema u realnom vremenu mora zadovoljiti zahteve vezane za mašinu stanja sistema. To podrazumeva pravovremenu i odgovarajuću reakciju na sledeće scenarije:

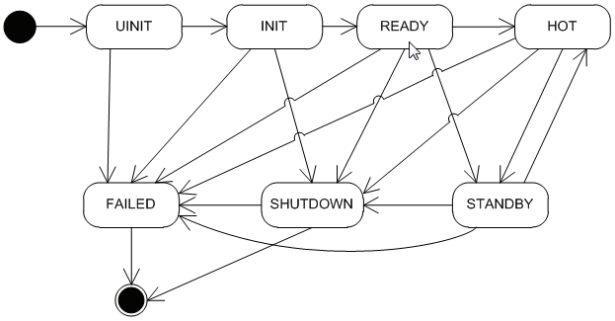
- Pokretanje sistema,
- Preključivanje sistema,
- Aktuelizacija,
- Zaustavljanje rada sistema.

Parametri uspostavljanja komunikacije predstavljaju identifikaciju servera i njegovu mrežnu adresu. U slučaju da nije moguće pronaći aktivan SCADA server, ostatak sistema se o tome mora obavestiti. Obaveštavanje o nemogućnosti pronalaženja aktivnog SCADA servera vrši se postavljanjem kvaliteta tačaka u sistemu na vrednost koja reprezentuje nemogućnost uspostave veze sa eksternim sistemom. Takodje, šalje se informacija koja će biti vidljiva u grafičkom okruženju i zabeležena u listi događaja. Obaveštenje je potrebno poslati i u slučaju gubitka veze sa SCADA OPC serverom, kao i nakon promene stanja SCADA OPC servera sa aktivnog na pasivno.

Životni vek komponente sistema u realnom vremenu podrazumeva prolazak kroz seriju promena stanja sistema. Svaka komponenta sistema vrši određene zadatke u zavisnosti od stanja u kom se sistem nalazi (slika 2). Stanja sistema su data u nastavku:

- UNINIT,
- INIT,
- READY,
- HOT,
- STANDBY,
- FAILED,
- SHUTDOWN.

Svaka komponenta mora biti u istom stanju na jednom čvoru klastera. Za vreme promene, dozvoljeno je da odredjene komponente budu u odgovarajućem stanju različitom od stanja sistema na ograničeni vremenski period.



Slika 2. Stanja automata

Sledeće funkcije su predviđene za komandovanje OPCPlugin-om od strane Proxy-a:

- *StartInit*: Započinjanje procesa inicijalizacije. Plugin-u se prosledjuje konfiguracija iz integratione konfiguracione datoteke.
- *GetConfiguratorNames*: Prosledjivanje imena konfiguratora.
- *SetConfigurator*: Prosledjivanje reference konfiguratora.
- *EndInit*: Završetak inicijalizacije plugin-a,

- *Start*: Promena stanja iz Init u Ready.
- *Resume*: Pokretanja procesa uspostave veze sa SCADA OPC serverom i akvizicije podataka.
- *Standby*: Pauziranje akvizicije podataka i periodične provere stanja eksternog sistema, raskida se veza sa SCADA OPC serverom ako je uspostavljena.
- *Shutdown*: Oslobadjaju se zauzeti resursi. Zaustavlja se rad OPCPlugin-a.
- *GetServerState*: Čitanje informacije o stanju sistema.

Akcije vezane za komunikaciju sa eksternim sistemom u zavisnosti od stanja veze i stanja eksternog sistema su date u nastavku:

- Ako veza sa serverom nije uspostavljena, započinje se proces uspostave veze sa OPC SCADA serverom.
- Ako veza ne može biti uspostavljena, pokušava se uspostava veze sa sekundarnim serverom (odnosno sledećim u listi ako ih ima više).
- Ako je veza uspešno uspostavljena, proverava se stanje servera. Ako je server aktivan, signalizira se uspešna uspostava veze sa aktivnim SCADA serverom OPCPlugin modulu. Ako server nije aktivan pokušava se uspostaviti veza sa sekundarnim serverom.
- Ako ni nakon preključivanja nije moguće uspostaviti vezu sa aktivnim OPC SCADA serverom, vrši se preključivanje sve dok jedan od servera ne postane dostupan i aktivan ili dok se ne primi signal za zaustavljanje rada OPCPlugin-a.
- Ako je veza sa aktivnim SCADA serverom uspostavljena i stanje eksternog sistema predje u pasivno, veza sa serverom se prekida i prekid veze sa eksternim sistemom se signalizira OPCPlugin modulu.
- Ako je veza ka eksternom sistemu izgubljena, takodje se vrši signalizacija OPCPlugin modulu.

Nakon signala za početak akvizicije podataka za svaku grupu tačaka dodaće se OPC grupa na OPC serveru. Nakon toga se za svaku tačku iz grupe tačaka dodaje OPC tačka u odgovarajuću OPC grupu. Ako je mehanizam pribavljanja podataka sinhron, pokrenuće se programska nit koja će periodično vršiti čitanje podataka sa OPC servera. Ako je tip pribavljanja podataka asinhron, OPCPlugin će se pretplatiti na odgovarajuću grupu tačaka, preuzeće inicijalne vrednosti sa servera i nadalje će preuzimati podatke sa servera samo ako se vrednosti tačaka promene. Pročitane vrednosti se šalju DAServer-u nakon mapiranja. Nakon prijema signala za prestanak akvizicije podataka, grupe tačaka sa asinhronim mehanizmom pribavljanja podataka prekidaju pretplatu, dok grupe tačaka sa sinhronim pribavljanjem podataka zaustavljaju čitanje. OPC grupe i OPC tačke koje su bile dodate za svaku grupu tačaka se uklanjaju.

4. REALIZACIJA

OPCPlugin je realizovan kao deljena biblioteka (*dynamic link library*) sa imenom OPCPlugin.dll., koju učitava Proxy komponenta. Biblioteka implementira CORBA objekat. Sve komponente unutar modula su dinamički instancirane kako ne bi dolazilo do preklapanja resursa prilikom učitavanja više instanci biblioteke u memorijski prostor Proxy procesa.

OPCPlugin biblioteka eksportuje samo jednu funkciju CreatePluginReference() koja vrši stvaranje OPCPlugin CORBA instance, a kao povratnu vrednost vraća referencu na plugin objekat.
OPCPlugin implementira sledeće sprege prema ostatku sistema:

- *IPlugin*: definiše spregu prema Proxy komponenti, preko koje Proxy upravlja OPCPlugin komponentom.
- *IDynDataConfigurator*: definiše spregu pomoću kog se OPCPlugin konfiguriše.

Komunikacija unutar sistema (*middle tier*) ostvaruje se CORBA komunikacionim protokolom.

Učitavanje Plugin-a vrši se funkcijom LoadLibrary(), nakon čega se stanje automata OPCPlugin-a postavlja na stanje UNINIT.

OPCPlugin se mora prijaviti DAServer sistemu kako bi mogao započeti upis vrednosti na DAServer, odnosno započeti klijentsku sesiju.

Nakon poziva *Resume* funkcije od strane Proxy servera OPCPlugin započinje akviziciju podaka. OPCPlugin signalizira modulu za komunikaciju sa eksternim sistemom početak rada i zatim čeka na signal uspešno uspostavljene veze sa aktivnim SCADA serverom, ili signal za zaustavljanje akvizicije od strane Proxy servera. Nakon signala za uspešno uspostavljanje veze sa eksternim sistemom OPCPlugin modul startuje modul za upravljanje grupama tačaka kako bi se započela akviziciji podataka.

Od ovog momenta zaustavljanje akvizicije vrši se u sledećim slučajevima:

- Nakon signala za zaustavljanje akvizicije podataka od strane Proxy servera. U tom slučaju OPCPlugin zaustavlja rad modula za upravljanje grupama tačaka i šalje signal za zaustavljanje rada modulu za komunikaciju sa eksternim sistemom.
- U slučaju primanja signala gubitka veze sa eksternim sistemom. U tom slučaju OPCPlugin zaustavlja rad modula za upravljanje grupama tačaka. Nakon toga OPCPlugin čeka na signal uspešno uspostavljene veze sa aktivnim SCADA serverom kako bi po uspostavi ponovo startovao modul za upravljanje grupama tačaka.

Uspostavljanje veze sa SCADA serverom vrši se pozivom funkcije CoCreateInstanceEx. Za uspostavljanje veze funkcija koristi jedinstvenu identifikaciju servera (CLSID) i ime mašine na kojoj se server nalazi. Identifikacija servera se čita iz registra ključeva SCADA OPC server računara korišćenjem ProgID programske identifikacije SCADA OPC servera. Vrednost ProgID programske identifikacije čita se iz konfiguracione datoteke.

Provera stanja SCADA servera vrši se pozivom funkcije GetStatus() definisane IOPCServer spregom OPC specifikacije.

Ako dodje do greške pri upisu pročitanih podataka na DAServer, signalizira se greška i plugin prelazi u FAILED stanje.

U slučaju asinhronog mehanizma primanja podatka, u slučaju da SCADA OPC server podržava verzije 3.0 OPC specifikacije, moguće je pokrenuti mehanizam provere stanja komunikacionog kanala za grupu tačaka. SCADA server se obaveštava kojom periodom treba slati prazne pakete u slučaju da nema promena vrednosti tačaka. Startuje se tajmer koji meri vreme izmedju dva uzastopna primljena paketa odredjene grupe tačaka. Ako vreme timera prekorači unapred definisanu vrednost, veza sa SCADA serverom se smatra prekinutom i započinje se proces ponovne uspostave veze.

Po prijemu vrednosti koje je potrebno obraditi *handle* tačke se koristi kao ključ u mapi *handle*-ova. Koristeći internu konfiguraciju tačke, ustanovljava se kojoj sistemskoj tački odgovaraju primljene vrednosti. Zatim se vrši mapiranje i generiše se *SItemUpdateEx* struktura koja se šalje DAServer-u. Pozitivna logika i merna jedinica se uzimaju u obzir pri mapiranju. SItemUpdateEx modeluje strukturu koju je potrebno poslati DAServer-u nakon prijema i mapiranja tačke. Opis polja *SItemUpdateEx* strukture je dat u nastavku:

- *ItemID*: Identifikacija tačke.
- *Value*: Vrednost tačke
- *Alarm*: Vrednost alarma.
- *Tag*: Vrednost Tag-a.
- *CustomData* : Prenos dodatnih SCADA vrednosti.
- *Quality*: Kvalitet.
- *ClientTimestamp*: Vremenski pečat.
- *ValueMask*: Bit maska načinjenih promena.

Kao što je već navedeno, vrednosti koje se primaju sa SCADA servera predstavljaju:

- Vrednost tačke,
- Vemenski pečat,
- Kvalitet tačke.

Vrednost tačke je tipa *VARIANT* što znači da u sebi može saržiti bilo koji skup podvrednosti različitog tipa podataka čak i nizove podataka. *VARIANT* vrednost će sadržati celobrojnu vrednost za digitalne elemente odnosno realnu vrednost za analogne elemente. *VARIANT* vrednost može sadržati i skup kontrolnih bita vezan za odredjen element. Kontrolni biti mogu nositi informaciju o prekoraćenju vrednosti, alarmu, grafičkom prilagodjavanja simbola ili bilo koju drugu vrednost. Ove vrednosti se mapiraju na *CustomData* polje *SItemUpdateEx* strukture.

5. TESTIRANJE

Unit testiranje vršeno je *CppUnit* programskim paketom. Testiranjem su obuhvaćeni sledeći procesi.

- Konfiguracija
- Mapiranje.

Testovi curenja memorije vršeni su *BoundsChecke, VisualLeakDetector* i *Perfmon* programskim alatima. Praćeni su sledeći parametri:

- Privatna memorija,
- Virtuelna memorija,
- Memorijski set,
- Broj handle-ova, i
- Broj niti.

Testiranje je vršeno sa različitim konfiguracijama sistema u zavisnosti od dostupnosti i stanja SCADA servera kao i mehanizma dobavljanja podataka. Testovi curenja memorije vršeni su u sledećim scenarijima:

- Primarni i sekundarni SCADA serveri nisu dostupni.
- Primarni i sekundarni SCADA serveri su dostupni, ali nisu aktivni.
- Primarni i sekundarni SCADA serveri periodično menjaju dostupnost, i stanje.
- Periodična provera stanja SCADA servera GetState() funkcijom.
- Periodična provera stanja SCADA servera čitanjem namenske tačke za proveru stanja.
- Primanje podaka sa SCADA servera sinhronim mehanizmom akvizicije podataka.
- Primanje podataka sa SCADA servera asinhronim mehanizmom akvizicije podataka.
- Primanje podaka sa SCADA servera korišćenjem sinhronog i asinhronog mehanizma akvizicije podakata.

Testovima je dokazano da se programsko rešenje validno ponaša.

Provera performansi razvijenog rešenja na ispitnom sistemu pokazalo je da programsko rešenje može obraditi približno 32000 tačaka u sekundi.

6. ZAKLJUČAK

Opisano rešenje zadovoljava početne zahteve, pružajući standardan okvir za sve buduće aplikacije koje zahtevaju programsku spregu korišćenjem OPC protokola. Rešenje optimizuje protok podataka unutar sistema i na mrežnom nivou, omogućava asinhrono dobavljanje podaka sa OPC SCADA servera, omogućava organizaciju tačaka po grupama sa različitim prioritetima i dinamikom. Implementacija je uspešno testirana na postojećim sistemima, u realnim aplikacijama.

Dalja unapredjenja bi podrazumevala proširenje skupa testova, kao i implementaciju OPC servera namenjenog za testiranje, kako bi testiranje funkcionalnosti vezane za komunikaciju sa OPC serverom bilo poboljšano.

7. LITERATURA

[1]. V.Kovačević, M.Popović, Sistemska programska podrška u realnom vremenu, FTN Novi Sad, 2002.

[2]. Dale Rogerson, Inside COM, Redmond Washington, 1997.

[3]. OPC specification 3.0.

[4]. Michi Henning, Steve Vinoski, Advanced CORBA® Programming with C++, 1999.

[5]. DMS RealTime System States Functional Requirement Specification.

Kratka biografija:



Dragan Krajnović rođen je u Boru 1985. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva - Računarske tehnike i računarske komunikacije odbranio je 2011. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

OBUČAVANJE NEURONSKIH MREŽA PRIMENOM LEVENBERG-MARQUARDT ALGORITMA NA PRIMERU FUNKCIJE PRENOSA

IMPLEMENTATION OF LEVENBERG-MARQUARDT ALGORITHM FOR TRAINING NEURAL NETWORK ON TRANSFER FUNCTION EXAMPLE

Nataša Majstorović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

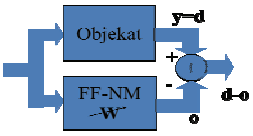
Kratak sadržaj – *U ovom radu je implementiran Levenberg-Marquardt algoritam za obučavanje neuronskih mreža u programskom jeziku C#. Ovaj algoritam je primenjen na obučavanje ponašanja dinamičkog modela opisanog jednostavnom diskretnom funkcijom prenosa drugog reda i rezultati testiranja su upoređeni sa rezultatima Backpropagation algoritma. Takođe, izvršeno je poređenje rada ovih algoritama u programskom paketu MATLAB.*

Abstract – *This paper contains implementation of Levenberg-Marquardt algorithm for neural network training in C# programming language. This algorithm was applied to learn behavior of dynamic model described with a simple second order transfer function and the results were compared with results of Backpropagation algorithm. Results, obtained from these two algorithms were also compared with results that were produced by MATLAB software package.*

Ključne reči: *Neuronska mreža, Levenberg-Marquardt algoritam, Backpropagation algoritam*

1 UVOD

U proteklim godinama, veštačke neuronske mreže su prešle put od istraživačkih laboratorija do primena u realnim situacijama. Veštačke neuronske mreže su u stanju da prepoznaju smisao u komplikovanim ili nepotpunim podacima, da prepoznaju obrasce koji su neprepoznatljivi ljudima zbog svoje složenosti i koji se ne mogu dobiti korišćenjem drugih kompjuterskih tehnika. Sposobnost neuronske mreže da proizvoljno mapira ulaze na izlaze joj omogućava da simulira ponašanje drugog sistema. Obučavanje neuronske mreže ulazno-izlaznim podacima iz objekta predstavlja postupak identifikacije sistema. Mogu se vršiti direktna i inverzna identifikacija. U ovom radu je korišćena direktna identifikacija (slika1). Kako bismo što lakše i na što bolji način rešili zadati problem, postoje različite arhitekture neuronske mreže, kao i različiti algoritmi učenja među neuronskim mrežama. U ovom radu korišćene su *feed-forward* neuronske mreže i u programskom paketu Microsoft Visual C# 2008 je implementiran Levenberg-Marquardt algoritam (LM). Algoritam je primenjen na obučavanje ponašanja dinamičkog modela opisanog jednostavnom funkcijom prenosa. Dobijeni rezultati su upoređeni sa rezultatima dobijenim u već implemetiranom Backpropagation algoritmu (BP) [1]. Takođe, ovi algoritmi su testirani sa istim parametrima u programskom paketu MATLAB R2008b, pomoću ugrađenog alata *nntool*. Na kraju su svi ovi rezultati analizirani.



Slika 1. *Direktna identifikacija sistema*

2 NEURONSKE MREŽE

Neke veštačke neuronske mreže zaista predstavljaju modele bioloških, neke ne, ali istorijski, inspiracija za oblast neuronskog računanja došla je od želje za stvaranjem veštačkog sistema, sposobnog za obavljanje sofisticiranih „inteligentnih“ računanja, sličnih onima koje obavlja ljudski mozak. Topologije mreže mogu biti različite zavisno od problema koji se rešava, vrste ulaznih podataka, vrste izlaznih podataka i drugih faktora. To je razumljivo s obzirom da se neuronske mreže primenjuju u najrazličitijim oblastima ljudske delatnosti.

U odnosu na arhitekturu mreže razlikujemo *feed-forward* i rekurentne neuronske mreže. *Feed-forward* mreže, koje su korišćene u ovom radu, dozvoljavaju kretanje signalu samo u jednom smeru, od ulaza ka izlazu. To znači da jedan neuron može da vrši uticaj samo na neuron koji se nalazi u narednom sloju (sloju koji je bliži izlazu). *Feed-forward* mreže obično brzo daju izlaz za dat ulaz.

Pored strukture, algoritam učenja je najvažniji faktor razlikovanja među neuronskim mrežama. Razlikujemo nadgledano učenje i učenje koje se ne nadgleda. Mreža obučavana nadgledanim učenjem, koje je korišćeno u ovom radu, koristi ulazne podatke za koje su poznati izlazi, pa se razlika željenog i izračunatog koristi za korekciju težina neuronske mreže. Neuronska mreža uči iz iskustva, gde se proces učenja sastoji iz ponavljanja tri zadatka: računanje izlaza, upoređivanje dobijene vrednosti sa željenom, podešavanje težina. Cilj je da se težine postave na vrednosti za koje će greška biti minimalna.

Postoje i dva moda učenja. Jedan je *on-line* učenje, a drugo *batch* učenje. Vrednosti vektora delta određuju za koliko se težine menjaju. Kod *on-line* učenja, težine se ažuriraju čim su sve vrednosti vektora delta poznate, pre predstavljanja sledećeg trening vektora. Za razliku od ovog, *batch* trening skladišti sve vrednosti delte za svaki trening vektor, ali su težine ažurirane tek na kraju trening epohe. *On-line* učenje tokom jednee pohe zaheva više ažuriranja težina [2].

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Aleksandar Erdeljan, docent

2.1 Backpropagation algoritam

BP algoritam spada u nadgledano učenje, koje može koristiti oba moda učenja i predstavlja generalizaciju Delta (Gradient descent) pravila. Ovaj algoritam se može podeliti na dva koraka. Prvi korak se vrši unapred kroz mrežu i izračunava se vrednost svih izlaznih jedinica, u zavisnosti od ulaza. Potom, računa se greška u odnosu na željene izlaze. Sledi drugi korak, koji prenosi grešku do svake jedinice od izlaznog (poslednjeg) sloja mreže, ka ulazima (unazad) i na osnovu grešaka se izračunavaju promene odgovarajućih težina. Brzinu učenja određujemo na osnovu parametra λ, koji učestvuje u ažuriranju težina. BP algoritam se koristi za perceptrone sa više slojeva i oni moraju imati nelinearnu funkciju aktivacije da bi se ovaj algoritam mogao primeniti. Ovaj algoritam spada u iterativne postupke i često mu treba duže od drugih metoda da se izvrši. Konvergencija nije zagarantovana i često može da konvergira ka bilo kom lokalnom minimumu [3].

2.2 Levenberg-Marquardt algoritam

LM algoritam spada u nadgledano, *batch* učenje. On interpolira između Gauss-Newton i Gradient descent algoritma, jednostavan je, spada u iterativne postupke i veoma je robustan algoritam. Robustniji je od Gauss-Newton algoritama, što znači da u većini slučajeva nađe rešenje i kada krene daleko od traženog minimuma. U većini slučajeva LM algoritam je u stanju da za isti broj iteracija proizvede najmanju kvadratnu grešku, u odnosu na druge algoritme. Ali kako raste broj težina u mreži, njegova prednost opada, jer koristi inverznu matricu za ažuriranje težina, pa ako se model jako poveća, ovo računanje postaje komplikovanije i zahteva više memorije. Takođe, ovaj algoritam je veoma osetljiv na inicijalne vrednosti težina i ne uzima u obzir uzlaze, koji odskaču od drugih, što može voditi do pretreniranosti. Ovaj algoritam se ne pokazuje dobro za probleme vezane za prepoznavanje oblika i zahteva dosta memorije. Međutim, ukoliko se radi o problemima aproksimacije funkcija, ovaj algoritam najbrže konvergira za mreže koje imaju do nekoliko stotina težina. Ova osobina je posebno primećena, ako je zahtevan veoma tačan trening. Suština ovog algoritma je rešiti jednačinu:

$$(J^T J + \lambda I)\delta = J^T E \tag{1}$$

gde se pomoću vrednost vektora δ vrši ažuriranje težina neurona. J predstavlja Jakobijanovu matricu za sistem, a J^TJ predstavlja aproksimaciju Hessiana. E je vektor grešaka izlaza u odnosu na svaki ulaz. Najvažniju ulogu u ovom algoritmu igra podešavanje parametara λ, jer na osnovu njega se omogućava da ovaj algoritam koristi prednosti oba algoritma od kojih je iskombinovan. Prednost je, takođe, aproksimacija Hessiana, jer se tako štedi na vremenu u računanju. λ se podešava u svakoj iteraciji. Ako je smanjenje E brzo, koristi se manja vrednost, kako bi bio bliži Gauss-Newton algoritmu, da se ne bi promašio minimum, ili išlo dužom putanjom (što se može desiti u Gradient descent-u). Dok, ako iteracija daje nedovoljno smanjenje reziduala, λ se povećava prilazeći bliže Gradient descent algoritmu da bi se ubrzao algoritam. U interesu je da se greška brzo smanjuje, ali i da se algoritam kreće najkraćom putanjom, tj. da se koristi veći korak ukoliko je algoritam na pravoj putanji da bi se izbeglo komplikovano i memorijski zahtevno računanje Hessiana. Zato su λ i Hessian vrlo značajni za ovaj algoritam. λ se sabira sa svakim članom dijagonale Hessiana pre nego što se rešava gradijent sistema. Onda se rešava LM izraz (1) najčešće primenom LU dekompozicije, ali se sistem može rešavati samo ukoliko aproksimiran Hessian nije singularan. Ukoliko ovo nije slučaj, izraz se može rešiti nekom drugom dekompozicijom [4][5].

3 REALIZACIJA PROGRAMSKOG REŠENJA

Programsko rešenje je realizovano programskim jezikom C# uz oslonac Microsoft Visual Studio 2008 i upoređeno je sa rešenjem koje daje nntool toolbox iz programskog paketa MATLAB R2008b.

3.1 Realizacija Levenberg-Marquardt algoritma

Dati algoritam je realiovan prateći sledeće korake. Prvo je izračunat Jakobijan, na osnovu sledeće formule:

$$\begin{matrix} \frac{de_{11}}{dw_1} & \cdots & \frac{de_{11}}{dw_N} \\ \vdots & \ddots & \vdots \\ \frac{de_{k1}}{dw_1} & \cdots & \frac{de_{k1}}{dw_N} \end{matrix} \tag{2}$$

gde e_{11} predstavlja grešku na prvom izlazu mreže za prvi ulaz, a e_{k1} predstavlja grešku na prvom izlazu mreže za k-ti ulaz. Ovde je realizovan algoritam sa samo jednim izlazom, pa je zato ovakva formula za Jakobijan. Sa w_1 do w_N su predstavljene sve težine i bijasi za datu mrežu. Zatim, izračunat je gradijent greške, pomoću formule:

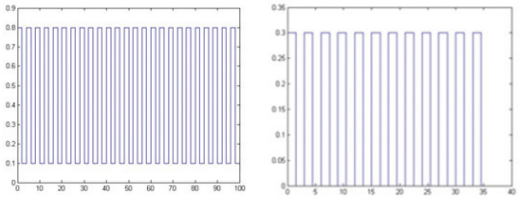
$$g = J^T E. \tag{3}$$

Nakon izračunavanja gradijenta, dobijeni su svi potrebni članovi za izračunavanje δ i ažuriranje težina. δ je dobijena rešavanjem izraza (1), primenjujući LU dekompoziciju. Ažurirane su težine, dodavajući vrednost dobijene δ. Ako suma kvadratne greške, posle ažuriranja težina nije smanjena, odbacuju se nove težine, povećava se vrednost λ, ponovo rešava izraz (1) i nastavlja sa narednim koracima. Ukoliko je suma kvadratne greške smanjena, smanjuje se λ i time završava jedna epoha.

4 REZULTATI TESTIRANJA NA PRIMERU FUNKCIJE PRENOSA

Testirana su dva algoritma: LM i BP algoritam. Prvo su testirani u programskom jeziku C#, a potom i u MATLAB alatu nntool. Analizirana je srednja kvadratna greška ovih algoritama i praćene su epohe u kojima je najmanja greška dostignuta.

Cilj je bio obučiti mrežu da radi kao funkcija prenosa drugog reda. Na ulaz je doveden pravougaoni signal sa maksimalnom vrednosti 0,8 a minimalnom 0,1 i na ovom signalu je vršeno obučavanje. Mreža je testirana u toku obučavanja, izdvajanjem podataka obučavajućeg skupa, kao i posle obučavanja, sa novim pravougaonim ulaznim signalom, sa maksimalnom vrednosti 0,3 a minimalnom 0 (slika 2).



Slika 2. *Ulazni signali*

Obučavajući skup se sastoji od ulaza, koji su dovođeni na mrežu i željenih izlaza (targeta). Pošto se radi o

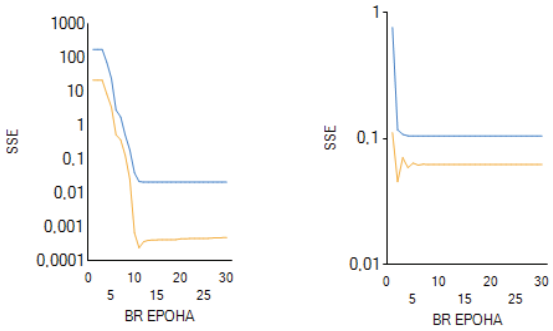
dinamičkom modelu, opisanog diskretnom funkcijom prenosa:

$$G(z) = \frac{0{,}00684\, z^{-1} + 0{,}00543\, z^{-2}}{0{,}821 - 0{,}7788 z^{-1} + 0{,}729\, z^{-2}} \quad (4)$$

a implementirana je *feed-forward* neuronska mreža bez povratne sprege, veštački su zakašnjeni ulazi i izlazi iz modela, tj. na mrežu je dovođeno četiri ulaza: prethodni ulaz u(k-1), ulaz pre njega u(k-2), kao i prethodni izlaz y(k-1) i izlaz pre njega y(k-2), gde je k posmatrani vremenski trenutak.Veličina obučavajućeg skupa je bila 200 trening parametara.

Tokom testiranja neuronske mreže menjan je broj skrivenih neurona, broj epoha, broj trening parametara, kao i vrednosti parametara λ. Program se morao više puta pokretati sa istim vrednostima ovih parametara, jer se na početku zadaju random vrednosti težina, pa da bi se što bolje uočilo koliko ovi parametri utiču na ponašanje testiranih algoritama.

Podaci za testiranja, koja su vršena u toku obučavanja, u C# programskom jeziku su dobijeni tako što je izdvajan svaki osmi podatak iz obučavajućeg skupa, za ulaz pravougaonog signala. Posle svake epohe obučavanja je vršeno testiranje sa ovim podacima i rezultati su poređeni sa rezultatima u MATLAB- u.



Slika 3. *Prikaz greške LM (levo) i BP algoritma (desno) dobijenih u C#*

Na slici 3. je prikazano jedno rešenje u programskom jeziku C#, gde je λ 1, broj neurona u skrivenom sloju 25, broj trening parametara 200 i broj epoha 30. Rezultati obučavanja su predstavljeni plavom bojom, dok su rezultati testiranja, koja su vršena sa izdvojenim podacima iz obučavajućeg skupa, posle svake epohe, predstavljeni narandžastom bojom.

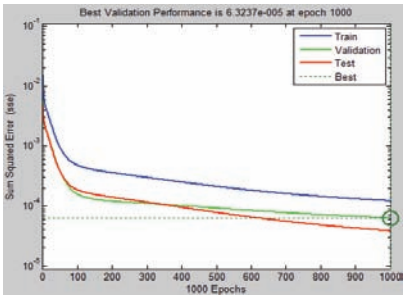
Vidi se da je greška manja za LM algoritam, nego što je to slučaj za BP, tj. da LM algoritam konvergira brže nego što je to slučaj sa BP algoritmom. U MATLAB-u su rezultati LM algoritma, takođe, bolji u odnosu na BP algoritam. Ako se pogleda opcija Regression kod MATLAB-a (pokazuje koliko dobro izračunati izlazi prikazuju željene vrednosti) vidi se da je vrednost za BP 0,925 dok je za LM 0,9984.

Ako se uporedi BP algoritam u C# programskom jeziku sa MATLAB-om sa istim parametrima, u MATLAB-u daje malo bolje rezultate, tj. greška obučavanja iznosi oko 0,01 dok u C# iznosi oko 0,1. Greška testiranja u MATLAB-u je oko 0,0001 dok u C# oko 0,08.

LM algoritam u MATLAB-u daje mnogo manju grešku obučavanja (oko 10^{-5}), u odnosu na 0,1 u C#. MATLAB, takođe, daje manju grešku testiranja (oko 10^{-4}), u odnosu na C# koji daje grešku 0,01.

4.1 Uticaj broja epoha na algoritme

Ukoliko se broj epoha poveća na 1000, onda se u BP algoritmu u MATLAB-u vidi da greška ima veći pad (slika 4) i da je Regression veći, tj. 0,99926. Može se opet desiti da bude veća greška i lošiji Regression, ali je bolja situcija nego sa manjim brojem epoha. U programskom jeziku C# se ne vidi takvo smanjenje greške za ovaj broj epoha.



Slika 4. *Greška BP algoritma u MATLAB-u*

Ukoliko se u MATLAB-u, za povećan broj epoha, posmatra LM algoritam, neće se ništa promeniti u veličini greške, jer će u pvih 20-ak epoha dostići „validation stop“, jer posle toga dolazi do pretreniranosti (greška testa raste, dok se greška treninga povećava).

4.2 Uticaj početnih težina na algoritme

Prilikom testiranja, uočeno je da početne vrednosti težina mnogo više utiču na izvršavanje LM algoritma, nego što je to slučaj sa BP algoritmom. Uticaj početnih vrednosti težina na LM algoritam je veći u programskom jeziku C#, nego što je to u MATLAB-u.

4.3 Uticaj broja skrivenih neurona

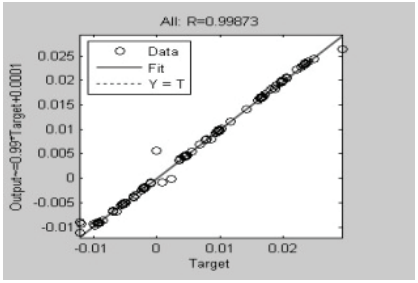
Povećanjem broja skrivenih neurona na 50, greška obučavanja u C# programskom jeziku kod LM algoritma postaje jako velika (iznad 100). Tada greška testiranih podataka dosta odstupa od greške obučenih podataka. Ukoliko se ponovi ispitivanje u MATLAB-u, sa istim parametrima, dobija se mala greška, ali se testiranje mnogo ranije završava, jer testiranje počinje da odstupa mnogo od treninga (tj. nastaje „validation stop“).

Za složenije probleme, BP algoritam zahteva više neurona u skrivenom sloju, kako bi obučavanje bilo moguće. Prilikom obučavanja BP algoritmom događa se da se greška smanjuje, pa se onda opet povećava. Razlog ovome je što ovaj algoritam može upasti u lokalni minimum, iako u blizini postoji drugi lokalni minimum koji je manji. Ovo se može poboljšati povećanjem broja skrivenih neurona, ali postoji i gornja granica ovog broja, posle kojeg se ponovo zapada u lokalni minimum. Ukoliko je manji broj skrivenih neurona od potrebnog, u MATLAB-u se može videti da je Regression loš, a i ne vidi se smanjivanje greške.

4.4 Uticaj vrednosti λ

Ukoliko je parametar λ postavljen na 10, broj skrivenih neurona na 25, broj epoha 100 i broj trening parametara 200, greška LM algoritma ne odstupa mnogo, ni u C# programskom jeziku, ni u MATLAB-u. Tj. povećanjem λ (od početnih 0,01 do 10) se ne povećava greška (greška učenja u C# programskom jeziku je oko 0,01, a greška testa oko 0,001), ali se u MATLAB-u, prilikom dešavanja „validation stop“-a, može videti mnogo veći porast greške testa u odnosu na pad greške, nego što je to bio slučaj sa manjim vrednostima λ. U opciji Regression se povećanje

λ se tek primećuje na trećoj decimali, tj. poklapanje je malo manje, u odnosu na manje vrednosti (slika 5).



Slika 5. *Opcija Reggresion u MATLAB-u*

Za iste vrednosti parametara, BP algoritam u C#-u daje grešku učenja oko 50, a greška testa mu je promenljivai osciluje od 0,01 do 10, u zavisnosti od epohe. U MATLAB-u se isto primećuje veliko oscilovanje za ove vrednosti. Velika greška se ne može smanjiti ni sa povećanjem broja neurona ni epoha. Prilikom smanjivanja vrednosti λ obučavanje BP algoritma je sporije. Ako se poveća, raste i brzina, ali onda i daje lošije rezultate.

4.5 Uticaj trening parametara

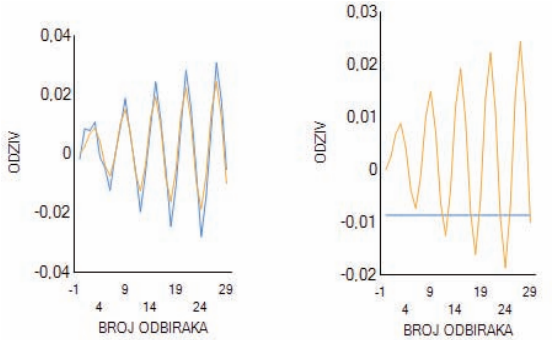
Ukoliko se smanji broj trening parametara iz obučavajućeg skupa na 50, greška treninga i testa se ne menja u MATLAB–u, a ni u C# programskom jeziku za bilo koji algoritam, ali Regression osciluje. Sa manjim brojem trening parametara, mreža je lošije obučena i greške testiranja teže prate greške obučavanja.

4.6 Poređenje nntool i C#

Testiranjem je utvrđeno da MATLAB nntool daje bolje rezultate u odnosu na algoritam realizovan u C# programskom jeziku. nntool koristi mnoge metode kako bi poboljšao treniranje mreže. Ovde se pre početka treniranja izbacuju ulazni podaci koji se mnogo ne menjaju, tj. koji imaju malu varijaciju. Takođe, vrši se normalizacija ulaznih podataka, validacija podataka, optimizacije samih algoritama i omogućuje se uvid u dodatne opcije, kao što je Regression, te zato nntool daje stabilnije rezultate i MATLAB dokazuje da ima moćan ugrađeni aparat za rad sa neuronskim mrežama [6].

4.7 Dovođenje novog ulaza

Da bi se bolje utvrdilo koliko dobro neuronska mreža aproksimira funkciju prenosa, vršeno je poređenje odziva funkcije prenosa sa odzivom obučene neuronske mreže u C# programskom jeziku. Nakon što je neuronska mreža obučena dovođenjem pravougaonog signala na ulaz, vršeno je testiranje sa novim ulazom (slika 2- desno). Dobijeni rezultati su prikazani na slici 6.



Slika 6. *Prikaz odziva funkcije prenosa i neuronske mreže*

Na slici 6 se mogu videti rezultati odziva funkcije prenosa i neuronske mreže za LM (na slici levo) i za BP algoritam (desno). Testiranje je vršeno kada je broj skrivenih neurona 25, broj epoha 40 i broj trening parametara obučavajućeg skupa 30. Može se primetiti da kod LM algoritma izlaz neuronske mreže (predstavljen plavom bojom), mnogo bolje prati stvaran odziv funkcije prenosa (predstavljen narandžastom bojom), nego što je to slučaj sa BP algoritmom.

Mreža može da da dobar odziv i na ulaz koji nije ranije „videla“, tj. za koji nije ranije obučavana, iako je mreža bez povratnog prenosa, a model dinamički.

5 ZAKLJUČAK

U radu je opisan i realizovan LM algoritam za obučavanje neuronskih mreža u C# programskom jeziku. Analizirani su i testirani rezultati na funkciji prenosa drugog reda. Ovde je upotrebljen jednostavan primer funkcije prenosa za testiranje neuronske mreže, jer je akcenat bio na uticaju parametara obuke na realizovan algoritam, kao i ponašanje LM algoritma u odnosu na BP algoritam u C# programskom jeziku i MATLAB-ovom nntool-u.

Nakon analize, utvrđeno je da neuronska mreža bez povratne sprege može da aproksimira dinamički model opisan funkcijom prenosa drugog reda. Međutim, i jednostavna funkcija prenosa se teško obučava ovakvom NM. Pri tome, postupak obuke je veoma osetljiv na parametre obuke poput broja epoha, broja skrivenih neurona, vrednosti λ itd. Koliko je dobra aproksimacija zavisi i od funkcije prenosa, od ulaza za koji se traži odziv, od broja prethodnih ulaza i prethodnih targeta koje se dovode na ulaz (što je veći broj, bolja je aproksimacija), kao i od izbora algoritma za obučavanje neuronskih mreža.

6 LITERATURA

[1] AForge Controls Library, Andrew Kirillov, 2006

[2] Nikola K. Kasabov, “Foundations of neural networks, fuzzy systems, and knowledge engineering”, The MIT Press, 1996.

[3] Halber White, “Some Asymptotic Results for Learning in Single Hidden- Layer Feedforward Network Models”, Journal of the American Statistical Association, Vol. 84, No. 408, 1989

[4] Magnus Magnus Nørgaard, “Neural networks for modelling and control of dynamic systems: a practitioner’s handbook”, Springer-Verlag, London 2000.

[5] Eiji Mizutani and James W. Demmel, „On structure of the residual Jacobian matrix arising in neural – network learning“, KES’ 01, N. Baba et al. (Eds.), IOS Press, 2001.

[6] MathWorks - MATLAB and Simulink for Technical Computing

Kratka biografija:



Nataša Majstorović rođena je u Kikindi, 22. septembra 1986. godine. Upisala je Fakultet tehničkih nauka u Novom Sadu, 2005. Godine, na departmanu za Računarstvo i automatiku i kasnije opredelila za smer Automatika i upravljanje sistemima.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.38:616-7

SISTEM ZA ESTIMACIJU MEHANIČKIH PARAMETARA HODA

SYSTEM FOR ESTIMATION OF MECHANICAL PARAMETERS OF GAIT

Zorica Nikolić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu je dat sistem baziran na akcelerometarskim senzorima, čijom primenom se vršilo određivanje uglova segmenata noge pri hodu i procena sile reakcije podloge.*

Abstract – *Subject of this paper is accelerometer-based system and its application in determing gait parameters during level walking and evaluating ground reaction force .*

Ključne reči: *Akcelerometar, Parametri hoda, Uglovi segmenata noge, Sila reakcije podloge*

1. UVOD

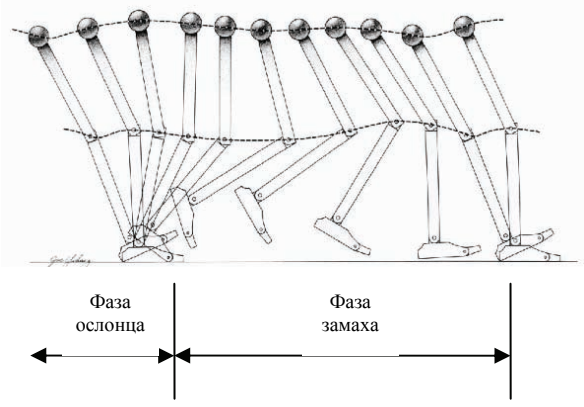
Predmet ovog rada je konstruisanje sistema baziranog na akcelerometarskim senzorima zbog njihove jednostavne primene i niske cene. Akcelerometri vrše merenje ubrzanja segmenata noge i njihovom primenom se mogu dobiti razni parametri hoda, kao što su pozicija, brzina i pre svega ugao segmenata noge. Istraživanje prezentovano u ovom radu je motivisano željom da se isproba metoda, koja će koristiti kao osnov matematičko modelovanje sistema.

2. MODELOVANJE LJUDSKOG KRETANJA

2.1. Hod i parametri hoda

Hod predstavlja složeno pomeranje u kome učestvuje veliki broj mišića i potrebna je izvrsna koordinacija pokreta.

Ljudski hod se može podeliti na dve osnovne faze, kao što se može uočiti na slici 1. Faze predstavljaju vreme kada je stopalo u kontaktu sa podlogom (faza oslonca) što iznosi oko 60% perioda hoda i kada noga nije u kontaktu sa podlogom (faza zamaha), koja predstavlja oko 40% perioda hoda [1].

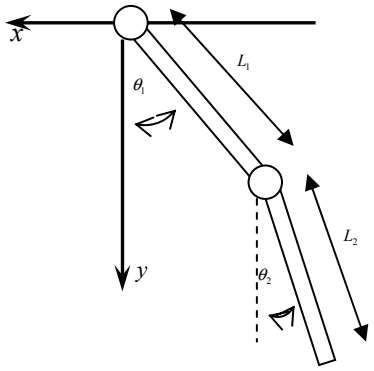


Slika 1. *Prikaz faza hoda*

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Nikola Jorgovanović, van.prof.

2.2. Faza zamaha

Faza zamaha se može modelovati dvostrukim matematičkim klatnom 0. Tokom pokreta unapred, noga koja napušta zemlju zamahnuta je unapred iz kuka, koji predstavlja prvi zglob, dok je drugi zglob koleno - između potkolena i natkolena (slika 2.).



Slika 2. *Matematički model dvostrukog klatna*

Iz matematičkog modela dvostrukog klatna dobijaju se jednačine koordinata prvog i drugog segmenta, a njihovim diferenciranjem se takođe mogu dobiti njihove brzine i ubrzanja.

$$\ddot{x}_1 = -\ddot{\theta}_1 L_1 \cos\theta_1 + \dot{\theta}_1^{\,2} L_1 \sin\theta_1 \quad (1)$$

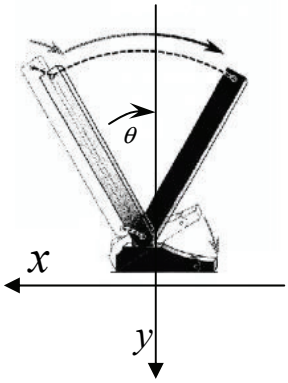
$$\ddot{y}_1 = -\ddot{\theta}_1 L_1 \sin\theta_1 - \dot{\theta}_1^{\,2} L_1 \cos\theta_1 \quad (2)$$

$$\ddot{x}_2 = \ddot{x}_1 - \ddot{\theta}_2 L_2 \cos\theta_2 + \dot{\theta}_2^{\,2} L_2 \sin\theta_2 \quad (3)$$

$$\ddot{y}_2 = \ddot{y}_1 - \ddot{\theta}_2 L_2 \sin\theta_2 - \dot{\theta}_2^{\,2} L_2 \cos\theta_2 \quad (4)$$

2.3. Faza oslonca

Period kada noga udari u podlogu sa petom, masa tela tada ima oslonac i počinje da se kreće ka napred, tako da se sistem može opisati putem inverznog klatna 0.



Slika 3. *Matematički model inverznog klatna*

Slika 3. daje prikaz pokretanja noge u fazi oslonca, na osnovu tog modela su izvedene jednačine pozicije, brzine i ubrzanja noge.

$$\ddot{x} = \ddot{\theta}L\cos\theta - \dot{\theta}^2 L\sin\theta \quad (5)$$

$$\ddot{y} = \ddot{\theta}L\sin\theta + \dot{\theta}^2 L\cos\theta \quad (6)$$

2.4. Centar mase tela i sila reakcije podloge

Centar mase tela COM (*Center of mass*) predstavlja zamišljenu tačku u kojoj je koncentrisana masa celog tela. Sile koje deluju na čoveka pri hodu mogu se podeliti na:

1) Spoljašne sile: gravitaciona, sila rekcije podloge – GRF (*Ground reaction force*),
2) Unutrašnje sile: sile ligamenata i mišića,
3) Inercione sile: posledica ubrzanja.

Ukupna sila koja deluje na centar mase tela može se matematički izraziti jednačinom (7) [4,5]

$$M\vec{a}_{COM} = \vec{F}_{GRF} + M\vec{g} \quad (7)$$

$$a_{COM} = \frac{Mg - F_{GRF}}{M} \quad (8)$$

U izrazima (7) i (8) veličine koje figurišu su a_{COM} koja predstavlja ubrzanje centra mase tela u smeru gravitacionog ubrzanja g, F_{GRF} je sila reakcije podloge, dok je M masa tela. Iz izraza (8) se takođe može videti da se poznavanjem ubrzanja centra mase može proceniti sila reakcije kojom podloga deluje na telo, i obrnuto, merenjem sile reakcije podloge, može se izračunati ubrzanje, pomeranje centra mase tela.

Na osnovu vertikalnog ubrzanja centra mase može se vršiti određivanje faze hoda [6]. Pikovi u signalu ovog ubrzanja odgovaraju trenucima kada noga udari u podlogu, jer time dolazi do pojave sile reakcije podloge. Detekcijom pikova se mogu utvrditi sukcesivni koraci tj. može se utvrditi kada je i koja noga u kontaktu sa podlogom i predstavlja fazu oslonca i kada se noga odvojila od podloge i otpočela fazu zamaha.

3. OPIS PRIMENJENIH ALATA I STRUKTURA

3.1. Akcelerometarski sistemi

Jednoosni akcelerometar se sastoji iz mase koja je oprugom pričvršćena za kućište akcelerometra. Masa se može kretati samo u jednom pravcu koji predstavlja senzitivnu osu akcelerometra. Putem drugog Njutnovog zakona se može izračunati ubrzanjem mase a.

Akcelerometar pored toga što meri linearno kinematičko ubrzanje koje nastaje pokretanjem tela za koje je pričvršćen, meri i statičko ubrzanje zemljine teže, odnosno projekciju tog ubrzanja na senzitivnu osu akcelerometra. Izlazno ubrzanje akcelerometra po horizontalnoj osi se može predstaviti izrazom (10), gde su $a_{\|}$ i $a_{\perp}$ projekcije kinematičkog ubrzanja:

$$\vec{a} = \vec{g} + \vec{a}_{\|} + \vec{a}_{\perp} \quad (9)$$

$$a_x = g\sin\theta - a_{\|}\cos\theta - a_{\perp}\sin\theta \quad (10)$$

Za potrebe ovog rada korišćeni su dvoosni akcelerometri ADXL311 firme *Analog Device*. Ovo integrisano kolo je izrađeno mikromašinskom tehnikom (MEMS *Micro-Electro-Mechanical Systems*) i zbog toga je veoma malih dimenzija 5mm x 5mm x 2mm. Izlazni signal je analogni naponski signal, koji je proporcijalan ubrzanju sa ofsetom jednakim polovini napona napajanja integralnog kola. Pri naponu napajanja od 5V ofset iznosi 2.5V, dok je tipična izlazna osetljivost ovog senzora 312mV/g.

Ovo integrisano kolo karakteriše niska cena, mala potrošnja i minijaturne dimenzije. Ove karakteristike stavljaju ovaj senzor u odličan izbor za komercijalne biomedicinske primene.

3.2. AD kartica NI DAQ 6024E

AD kartica NI DAQ 6024E firme *National Instruments* poseduje 16 naponskih analognih ulaza rezolucije 12 bita sa ulaznim opsegom od -10V do 10V. Maksimalna frekvencija odabiranja po kanalu iznosi 200kHz, što je i više nego dovoljno da posluži za snimanje signala ubrzanja segmenata noge. Ova AD kartica može da snima 16 analognih kanala pri *Single Ended* načinu povezivanja, kojim se digitalizuje napon između svakog analognog ulaza i mase AD kartice i 8 analognih kanala pri diferencijalnom načinu povezivanja gde napon koji se digitalizuje predstavlja razliku napona između dva analogna ulaza.

3.3. Akvizicija signala

Akvizicija signala je vršena tako što su izlazi modula akcelerometara povezivani na PCMCIA AD karticu NI DAQ 6024E pomoću koje je vršena digitalizacija analognog signala akcelerometara. U radu je AD kartica korišćena u *Single Ended* modu, pa je snimala 8 signala, jer je svaki od 4 akceleromatara posedovao po dva analogna izlaza koji su odgovarali senzitivnim osama akcelerometara.

Za komunikaciju sa karticom, snimanje i analizu podataka upotrebljen je programski paket *MATLAB*. Korišćenjem funkcija *MATLAB*-ovog *Data acquisition toolbox*-a izvršeno je podešavanje parametara kao što su dužina snimanja, frekvencije odabiranja, broj kanala, veličina prihvatnog memorijskog bafera.

4. OPIS I UPOTREBA METODE

4.1. Modelovanje faza hoda

Koristeći kao osnov jednačinu za izlaz akcelerometra u izrazu (10), modelovanje faze zamaha vršimo tako što ćemo u navedenoj jednačini na mestu normalnog i paralelnog kinematičkog ubrzanja uvrstiti izraze za ubrzanje po *x*-osi (1),(3) i *y*-osi (2),(4).

$$a_{x1} = g\sin\theta_1 + \ddot{\theta}_1 L_1 \quad (11)$$

$$\ddot{\theta}_1 = \frac{1}{L_1}(a_{x1} - g\sin\theta_1) \quad (12)$$

$$a_{x2} = g\sin\theta_2 + \ddot{\theta}_1 L_1\cos(\theta_1 - \theta_2) + \dot{\theta}_1^2 L_1\sin(\theta_2 - \theta_1) + \ddot{\theta}_2 L_2 \quad (13)$$

$$\ddot{\theta}_2 = \frac{1}{L_2}(a_{x2} - g\sin\theta_2 - \ddot{\theta}_1 L_1\cos(\theta_1 - \theta_2) - \dot{\theta}_1^2 L_1\sin(\theta_2 - \theta_1)) \quad (14)$$

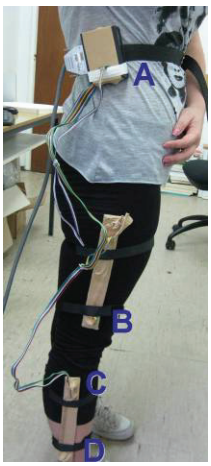
Izlaz akcelerometra za fazu oslonca se dobija upotrebom izraza (5) i (6), odakle se dobija jednačina za izračunavanje ugla noge.

$$a_x = g\sin\theta - \ddot{\theta}L \quad (15)$$

$$\ddot{\theta} = \frac{1}{L}(g\sin\theta - a_x) \quad (16)$$

4.2. Prikaz korišćenog akcelerometrijskog sistema

Snimanje ubrzanja segmenata noge je vršeno postavljanjem četiri dvoosnih akcelerometara u sagitalnoj ravni, tako što je su ose akcelerometra paralelne i normalne na skelet noge, prikazano na slici 4.

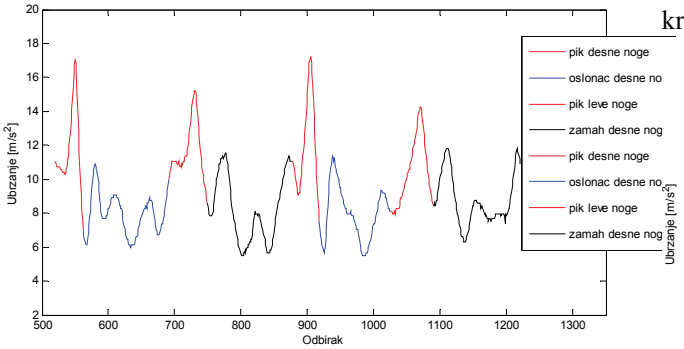


Slika 4. *Izgled akcelerometarskog sistema*

Akcelerometar obeležen slovom A služio je za snimanje ubrzanja pretpostavljenog centra mase, akcelerometar B je korišćen za snimanje ubrzanja natkolena, dok su akcelerometri C i D postavljeni za snimanje ubrzanja potkolena (Slika 4.).

4.3. Izgled dobijenih signala

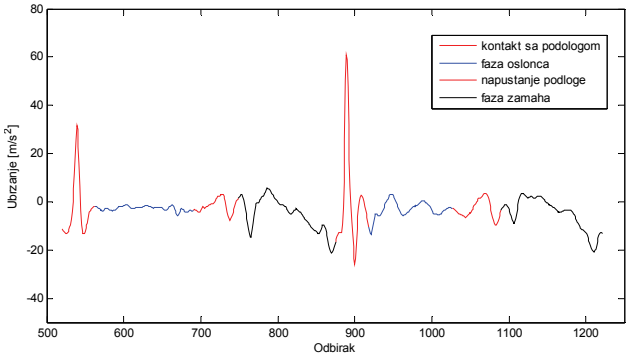
Signali su dati nakon osnovnih konverzija, naime, oduzet im je konstantni ofset od 2.5V koji predstavlja polovinu napona napajanja, zatim je vršeno skaliranje iz dobijenih vrednosti izraženih kao napon u gravitaciono ubrzanje (izlazna osetljivost korišćenih akcelerometra je 0.3V/g) i na kraju je izvršeno prebacivanje vrednosti iz jedinica gravitacionog ubrzanja.



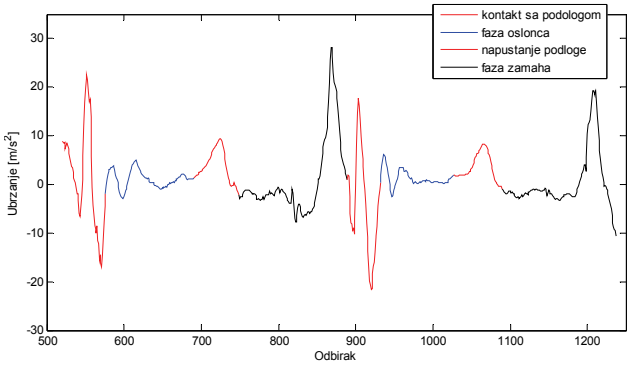
Slika 5. *Vertikalno ubrzanje centra mase*

Na slici 5. je data vertikalna komponenta akcelerometra A. Prema ovom signalu je vršeno uočavanje i podela faza, tako što se mogu videti četiri pika označenih crvenom bojom. Kako je ovaj akcelerometar bio postavljen desno na struku, može se videti razlika u visinama pikova, pri udaru desne noge u podlogu koja je bliža pa se jače detektuje (prvi i treći pik), u odnosu na pikove koje izaziva udar leve noge u podlogu (drugi i četvrti).

Na slikama 6. i 7. su data horizontalna ubrzanja potkolena i natkolena. Dva velika pika figurišu inicijalni kontakt pete desne noge sa podlogom. U fazama oslonca, vrednost signala uzima vrednosti oko nule, jer je malo kinematičko ubrzanje, dok je projekcija gravitacionog ubrzanja isto bliska nuli, jer se osa nalazi nalazi normalno na pravac gravitacionog vektora. U fazi zamaha se može videti jači uticaj kinematičkog ubrzanja, usled pokreta.

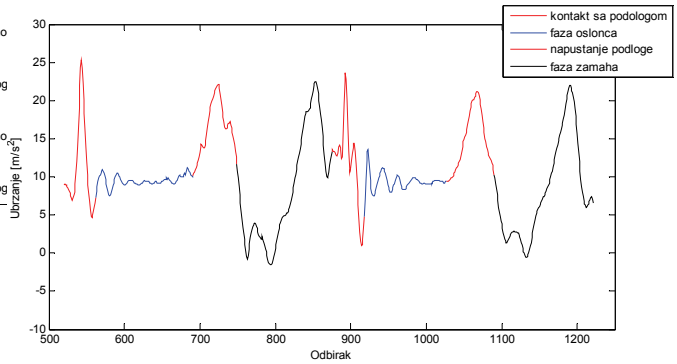


Sika 6. *Horizontalno ubrzanje potkolena*

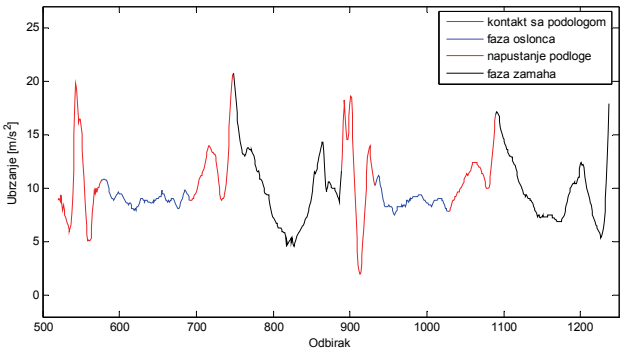


Slika 7. *Horizontalno ubrzanje natkolena*

Na slikama 8. i 9. se mogu videti vrednosti ubrzanja vertikalnih komponenti potkolena i natkolena i takođe se može izvršiti izdvajanje intervala karakterističnih za svaku fazu. Tokom faze oslonca, osa je skoro paralelna sa vektorom gravitacionog ubrzanja i ima malo kinematičkog ubrzanja pa stoga se vrednosti nominalno kreću oko vrednosti gravitacionog ubrzanja.

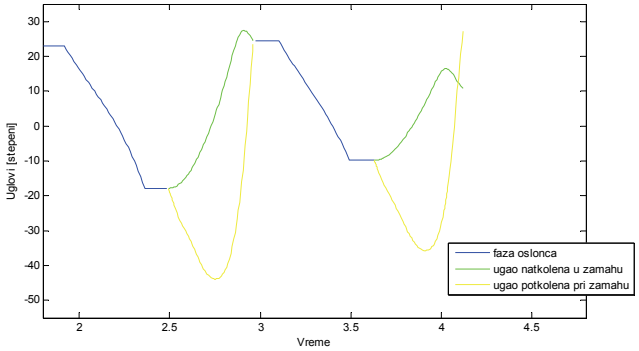


Slika 8. *Vertikalno ubrzanje potkolena*



Slika 9. *Vertikalno ubrzanje potkolena*

5. REZULTATI I RAZMATRANJA



Slika 10. *Inklinacije segmenata noge*

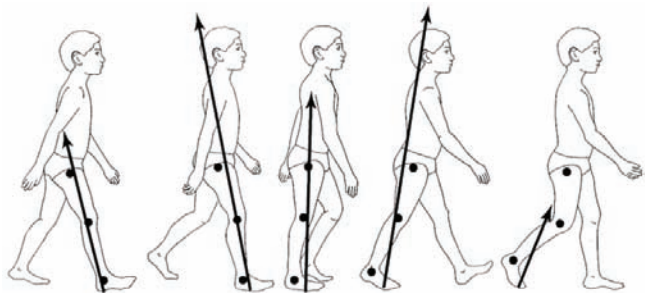
U fazi oslonca na slici 10. pri pokretanju tela unapred dolazi do konstantantnog smanjivanja vrednosti ugla i to od oko $(20^\circ; 30^\circ)$ do $(-10^\circ; -20^\circ)$. Kod pojave pikova, da bi se izbegao njihov uticaj, zadržavaju se poslednje vrednosti rezultata.

U fazi zamaha, predstavljena su dva ugla, ugao inklinacije natkolena – zelenom bojom i potkolena - žutom bojom. Ugao natkolena menja vrednosti od oko -20° do oko 30° što se dešava pomeranjem noge od izdizanja noge od podloge, preko podizanja kolena u najviši položaj, da bi se zatim pred kraj faze malo smanjilo i ispravilo neposredno pre oslanjanja na podlogu. Na početku faze zamaha, potkoleno ide još unazad za pripremanje za veći zamah, to se vidi na grafiku ugla potkolena, čija vrednost ide do -40°, a zatim se u zamahu vrednost kreće ka izjednačavanju sa uglom natkolena pre nego što se osloni noga.

Da bi se dobili dobri rezultati, mora se dosta paziti na početne uslove, jer njihova loša postavka na početku u prvoj fazi, utiče na izgled i tačnost ostatka rezultata.

Veoma je važno uzeti u obzir šum signala što donosi grešku, posebno usred vibracija mišića. Najveće greške ipak nastaju usled nestabilne postavke akcelerometarskog sistema, kod koga je dolazilo do čestih pomeranja, jer je na trake za povezivanje delovalo i pomeranje usled rada mišića. Zbog robusnosti sistema postavljenog na subjekta, signali su mereni na kratkom rastojanju, tačnije mereni su na uzorku od 4-5 koraka, što predstavlja ograničenje u razmatranju.

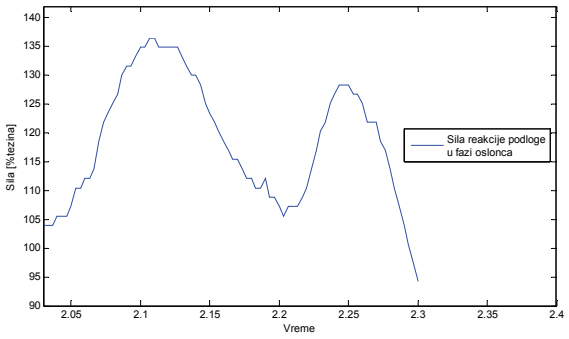
6. SILA REAKCIJE PODLOGE



Slika 11. *Vektor sile reakcije podloge*

Strelice prikazane na slici 11 označavaju magnitudu i smer sile reakcije podloge, koja deluje na desnu nogu.

Korišćenjem izraza (8) procenjena je sila reakcije podloge u fazi oslonca, data u procentima po težini tela slika 12.



Slika 12. *Sila reakcije podloge*

Izgled grafika u obliku slova M je karakterističan za normalan hod [7]. Prvi breg nastaje inicijalnim kontaktom pete sa podlogom, zatim tokom oslonca dolazi do spuštanja vrednosti, jer koleno savijanjem absorbuje silu, a zatim nastajanje drugog brega izaziva odbacivanje noge od podloge pri odvajanju od iste. Sa slike 12 se može videti da je za održavanje oslonca tela i ravoteže pri hodu, potrebno da se primeni sila veća od težine tela.

7. ZAKLJUČAK

Ciljevi budućeg rada odnosiće se na bežično povezivanje senzora uređaja sa računarom i razmatranje mogućnosti automatske dijagnostike bazirane na ovom uređaju. Nastavak istraživanja može ići u smeru da se dobijeni uglovi koriste kao vrednosti povratne sprege na osnovu kojih će se izvršavati upravljanje u električnoj stimulaciji FES (*Functional Electrical Stimulation*), a može se sve proširiti i uvođenjem modelovanja momenata zglobova, kao i delovanja aktiviranja korišćenih mišića.

8. LITERATURA

[1] Jessica Rose, James G. Gamble, "Human walking", 3rd Edition, California, 2006.

[2] Erol Uyar, Özgün Baser, "Investigation of bipedal human gait dynamics and knee motion control"

[3] Koki Kameta, Akinori Sekiguchi, Yuichi Tsumaki, Yoshikazu Kanamiya, "Walking Control around Singularity Using a Spherical Inverted Pendulum with an Underfloor Pivot", Japan 2007.

[4] H. Martin Schepers, Peter H. Veltink, "Ambulatory Estimation of Center of Mass Displacement During Walking", *IEEE* VOL. 56, No. 4, April 2009.

[5] Slawomir Winarski, Alicija Rutkowska-Kucharska, "Estimated ground reaction force in normal and pathological gait", *Acta of Bioengineering and Biomechanics*, Vol. 11, No. 1, 2009.

[6] Choon-Young Lee, "Estimation of Walking Behavior Using Accelerometers in Gait Rehabilitation"

[7] Dr. Robert L. Williams II, Engineering Biomechanics of Human Motion, Supplement for: ME 467 / BME 567, Ohio, 2011.

Kratka biografija:



Zorica Nikolić rođena u Vršcu 1985. god. Matematičku gimnaziju u Beogradu završila 2004. god. Apsolvent je Fakulteta tehničkih nauka, odsek Elektrotehnika i računarstvo, katedra Automatika i upravljanje sistemima.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9

OBUČAVANJE NEURONSKIH MREŽA PRIMENOM BAYESIAN REGULATION ALGORITMA NA PRIMERU FUNKCIJE PRENOSA

IMPLEMENTION OF THE BAYESIAN REGULATION ALGORITHM FOR TRAINING NEURAL NETWORK ON TRANSFER FUNCTION EXAMPLE

Ivana Todosijević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu je opisana implementacija Bayesian regulation backpropagation algoritma za obučavanje neuronskih mreža u programskom jeziku C# uz oslonac na Microsoft Visual Studio 2010. Algoritam je testiran na primeru učenja dinamičkog ponašanja sistema modelovanog funkcijom prenosa i njegovi rezultati su uporedjeni sa Backpropagation algoritmom. Zatim su u programskom paketu Matlab u ugradjenom alatu nntool testirani isti algoritmi sa istim parametrima i na istoj funkciji prenosa. Na kraju su analizirani i upoređeni rezultati.*

Abstract – *This paper describes the implementation of the Bayesian regulation backpropagation algorithm for training neural networks in Microsoft Visual Studio 2010. Algorithm was tested on the example of learning dynamic system behavior modeled by transfer function and its results were compared with the Backpropagation algorithm. Same algorithms, using the same input parameters and transfer function were tested in Matlab programming package, using built in nntool toolbox. Finally, the results are analyzed and compared.*

Ključne reči: *neuronska mreža, Bayesian regulation backpropagation algoritam*

1. UVOD

Oblast koju danas poznajemo kao neuronske mreže nastala je kao rezultat spoja nekoliko veoma različitih pravaca istraživanja: obrade signala, neurobiologije i fizike. Istorijski gledano, inspiracija za razvoj neuronskih mreža proizilazi iz želje da se konstruiše veštački sistem sposoban za prefinjeno, možda „inteligentno", izračunavanje na sličan način kao što to ljudski mozak rutinski izvodi. S jedne strane, to je pokušaj da se razume rad ljudskog mozga, a sa druge da se to stečeno znanje primeni u obradi složenih informacija. [1]

Ključno pitanje u razvoju neuronskih mreža ima pitanje generalizacije tj. koliko će dobro mreža napraviti predikcije za slučajeve koji nisu u obučavajućem skupu. Neuronske mreže, kao i druge fleksibilne nelinearne metode procene, mogu da imaju problema sa "overfitting"-om koji se javlja kod mreža koje su previše složene. *Overfitting* je posebno „opasan" jer može lako dovesti do predviđanja koja su daleko van dometa podataka za obuku, tj. tada se dobijaju pogrešni rezultati predviđanja. Jedan od načina da se izbegne overffiting a da se dobija dobra generalizacija jeste Bayesian regularizacija [2].

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Aleksandar Erdeljan, docent.

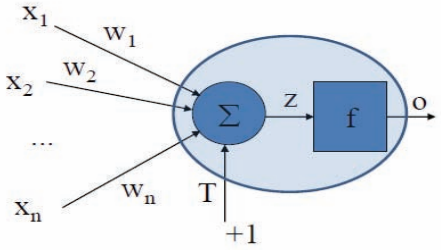
2. TEORIJSKE OSNOVE NEURONSKIH MREŽA

Neuronske mreže (*Neural Networks - NN*) predstavljaju novu generaciju sistema za informaciono procesiranje koje pokazuju osobinu učenja, memorisanja i generalizacije na osnovu podataka kojima se obučavaju.

Postoje dve kategorije neuronskih mreža: veštačke i biološke neuronske mreže. Predstavnik bioloških neuronskih mreža je nervni sistem živih bića. Veštačke neuronske mreže su po strukturi, funkciji i obradi informacija slične biološkim neuronskim mrežama, ali se radi o veštačkim tvorevinama. One su kolekcija matematičkih modela koji simuliraju neke od posmatranih osobina bioloških nervnih sistema i povlače sličnosti sa prilagodljivim biološkim učenjem. Sačinjene su od velikog broja međusobno povezanih neurona (obrađujućih elemenata) koji su povezani svojim vezama koje sadrže propusne (težinske) koeficijente.

2.1. Model veštačkog neurona

Na slici 1 prikazan je model veštačkog neurona.



Slika 1. *Model veštačkog neurona*

Za veštački neuron može se reći da se dizajnira s idejom da oponaša osnovne funkcije biološkog neurona. Svaki neuron poseduje sledeće elemente:

- Ulazi - povezuju neuron sa ostalim neuronima
- Težine sinapsi (w) - koristimo ih da iskažemo koliko je koji ulaz bitan, jak.
- Bias - slobodan ulaz koji nam određuje minimalni nivo signala (*treshold*) potreban za aktivaciju neurona
- Nelinearna funkcija - u odnosu na pobudni signal neurona formira izlazni signal, služi da ograniči izlazni signal.
- Izlaz - ulaz u sledeće neurone ili izlazno rešenje ako je neuron poslednji u nizu.

2.2. Model neuronske mreže

Neuronsku mrežu čine:

- Arhitektura (topologija) mreže: predstavlja specifično povezivanje neurona u jednu celinu. U zavisnosti od uloge u mreži slojevi mogu biti ulazni, skriveni i izlazni, a veze između slojeva jednosmerne veze unapred- *feed-forward* i povratna sprega - *feedback*.
- Prenosna (aktivaciona) funkcija neurona: ograničava amplitudu izlaznog signala neurona. Tipično se uzima normalizacija izlaza na interval [0,1] (Sigmoidna funkcija) ili [-1,1] (Bipolarna Sigmoidna funkcija).
- Zakoni učenja: Proces učenja je proces podešavanja težina sinapsi u cilju postizanja odgovarajućeg (željenog) izlaza za dati pobudni signal. Postoji učenje sa nadgledanjem (*supervised learning*) i učenje bez nadgledanja (*unsupervised learning*). [3]

3. PRIMENJENI ALGORITMI

3.1. Backpropagation algoritam

Algoritam obuke mreže "širenje greške unazad" bio je presudan za široku komercijalnu upotrebu neuronskih mreža, te je neuronske mreže učinio široko upotrebljavanom i popularnom metodom u različitim područjima. U osnovi, tokom obuke mreža propagira ulaz kroz mrežu od ulaznog do izlaznog sloja, a zatim određuje grešku i tu grešku propagira unazad sve do ulaznog sloja ugrađujući je u formulu za učenje. Standardni algoritam mreže "širenje greške unazad" uključuje optimizaciju greške koristeći deterministički algoritam gradijentnog opadanja (*eng. gradient descent*). Glavni nedostatak ovog algoritma je problem čestog pronalaženja lokalnog umesto globalnog minimuma greške. Backpropagation algoritam se sastoji od ponovljenih primena dva prolaza:

- Prolaz unapred: u ovom koraku mreža se aktivira na jednom uzorku i računa se odziv za svaki neuron u mreži sve do izlaznog sloja
- Prolaz unazad: u ovom koraku računa se greška mreže i ona se koristi za ažuriranje težina. Skriveni slojevi su vezani za grešku indirektno posredstvom sinapsi do sledećeg sloja pa se, počev od izlaznog sloja, greška propagira unazad kroz mrežu, sloj po sloj. Ovo se čini rekurzivnim računanjem lokalnog gradijenta svakog neurona. [4]

3.2 Bayesian regulation backpropagation algoritam

Jedan od najboljih načina za poboljšanje generalizacije i izbegavanje *overfitting*-a jeste regularizacija. [2] Upravo je zato MacKay 1992. godine predložio Bayesian regularizaciju da bi se prevazišao problem u interpolaciji nesređenih i heterogenih podataka. Bayesian regularizacija proširuje uobičajenu korišćenu funkciju cilja:

$$F = E_d = \frac{1}{N}\sum_{i=1}^{k}(e_i)^2 \qquad (1)$$

koja predstavlja sumu kvadrata grešaka, dodajući sumu kvadrata težina E_w i dva Bayesian hiperparametra α i β, koji ukazuju u kom pravcu (minimalne greške, ili minimalne težine) će ići proces učenja. Time funkcija cilja postaje:

$$F = \beta E_d + \alpha E_w . \qquad (2)$$

Prema Hagan-u i Foresse (1997), kada se ovako proširena funkcija cilja koristi zajedno sa *Levenberg-Marquardt* algoritmom koji predstavlja još jedan algoritam za obučavanje neuronskih mreža, mreža će imati manje težine, što će dovesti do toga da mreža ima blaži izlaz, tj. izlaz neće imati puno oscilacija i samim tim su manje šanse da dođe do *overfitting*-a. Još jedna važna karakteristika ovog algoritma je da obezbeđuje da se proceni efektivni broj parametara koji je potreban da se obuči model – tj. broj težina i biasa zaista potrebnih da se reši određeni problem. Hiperparametar koji to omogućuje je γ i on se razlikuje od ukupnog broja parametara u mreži (težine + biasi). Nakon dovoljnog broja iteracija ovaj parametar treba da ostaje približno isti bez obzira koliki je ukupan broj parametara u mreži i tada se pretpostavlja da je mreža obučena. Kada efektivni broj parametara prestane da se menja, idealan broj neurona u skrivenom sloju je postignut. Kada se koristi Bayesian regularizaciju, važno je da se mreža trenira dok ne stigne do konvergencije tj. dok suma kvadrata grešaka, suma kvadrata težina i efikasan broj parametara ne dostignu konstantne vrednosti. [5][6]

Koraci rešavanja algoritma:

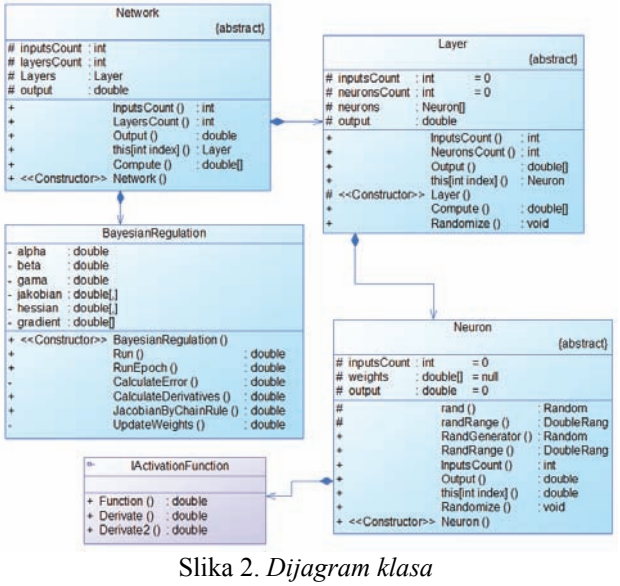
- Slučajno inicijalizovati α, β, γ i težine za sve slojeve
- Ažurirati težine po formuli:

$$w(k+1) = w(k) + [2\beta J^T J + 2\alpha I]^{-1} g \qquad (3)$$

  gde je *I* jedinična matrica, *J* Jakobijan matrica, a *g* je gradijent greške.
- Izračunati α, β i γ
- Ponavljati korake 2 i 3 dok se ne dostigne zadati cilj.

4. REALIZACIJA PROGRAMSKOG REŠENJA ZADATKA

Programsko rešenje zadatka je realizovano programskim jezikom C# i sastoji se u realizaciji Bayesian regulation backpropagation algoritma. Za programsko rešenje su uvedene klase prikazane na slici 2.



Slika 2. *Dijagram klasa*

Algoritam se svodi na rešavanje jednačine:

$$(J^T J + \lambda I)\delta = J^T E \tag{4}$$

gde je J matrica Jakobijana za sistem, λ faktor prigušenja (*learning rate*), δ faktor za ažuriranje težina koji želimo da pronađemo i E je vektor greške koji sadrži greške izlaza za svaki vektor ulaza korišćen za trening mreže. δ nam govori koliko bismo trebali da promenimo težine u mreži kako bismo dobili moguće bolje rešenje. Matrica $J^T J$ je Hesijan. Kada su izračunate greške za svaki neuron, računamo izvode težina (*weights derivates*) u funkciji *CalculateDerivates*. Potom te izvode kopiramo u matricu Jakobijana. Kako je algoritam realizovan sa jednim izlazom, broj redova Jakobijana jednak je broju trening parova. Broj kolona Jakobijana jednak je broju težina (zajedno sa biasima) u mreži. Tako na primer za N trening parova i w-težina u mreži matrica Jakobijana je:

$$J = \begin{bmatrix} \frac{\partial e(x_1, w)}{\partial w_1} & \cdots & \frac{\partial e(x_1, w)}{\partial w_w} \\ \vdots & \ddots & \vdots \\ \frac{\partial e(x_N, w)}{\partial w_1} & \cdots & \frac{\partial e(x_N, w)}{\partial w_w} \end{bmatrix} \tag{5}$$

gde na primer, $\frac{\partial e(x_1, w)}{\partial w_1}$ predstavlja grešku na prvom izlazu mreže za prvi ulaz i prvu težinu. Zatim izračunamo gradijent greške po formuli:

$$g = J^T E \tag{6}$$

i aproksimiramo matricu Hesiana pomoću matrice Jakobijana:

$$H \approx J^T J\,. \tag{7}$$

Definišemo funkciju cilja (2), i pomoću LU dekompozicije rešimo jednačinu (4). Ažuriramo težine po formuli (3), i izračunamo novu sumu kvadratnih grešaka. Ako nova suma kvadratne greške nije smanjena, odbacuju se nove težine, povećava se vrednost λ i ponovo se rešava jednačina (4). Ukoliko je suma kvadratne greške smanjena, smanjuje se λ, ažuriraju se hiperparametri α, β i γ koristeći formule:

$$\gamma = W - (\alpha \cdot tr(H^{-1}))\,, \tag{8}$$

$$\beta = (N - \gamma)/(2 \cdot E_d)\,, \tag{9}$$

$$\alpha = \gamma/(2 \cdot E_d) \tag{10}$$

gde je W ukupan broj težina zajedno sa biasima, a *tr* je funkcija *trace* koja predstavlja trag matrice. Kada se ažuriraju parametri jedna epoha je završena.

5. TESTIRANJE I ANALIZA REZULTATA

Testiranje je vršeno na primeru diskretne funkcije prenosa drugog reda:

$$G(z) = \frac{y}{u} = \frac{0.006594z^{-2} + 0.007166z^{-1}}{0.7788z^{-2} - 1.773z^{-1} + 1}, \tag{11}$$

što znači da trenutni izlaz iz neuronske mreže zavisi od prošlog i predprošlog ulaza, kao i od prošlog i pretprošlog izlaza. Kao ulaz u funkciju prenosa uzet je pravougaoni signal amplitude 0.5. Mreža je testirana na skupovima od 300 i 500 trening parametara (BrTP) s tim da je za test podatke uziman svaki peti par iz trening skupa. Takođe su menjani faktor prigušenja, broj neurona u skrivenom sloju i broj epoha kako bismo dobili što bolje performanse. Analizirani su rezultati srednje kvadratne greške Bayesian regulation backpropagation algoritma (SSE BR) i već realizovanog Backpropagation algoritma (SSE BP). [7] Parametri za koje su testirani algoritmi u programskom jeziku C# i u programskom paketu Matlab u alatu *nntool* su bili sledeći:

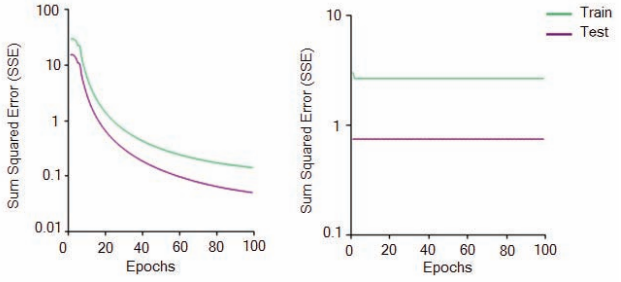
- Broj epoha (Br): 100, 300
- Learning rate (λ): 0.0001, 0.01, 10
- Broj neurona u skrivenom sloju (BrNSS): 20, 30

U tabeli 1 su prikazani dobijeni rezultati za 300 trening parametara i za 100 epoha.

Tabela 1: *Rezultati testiranja u programskom jeziku C#*

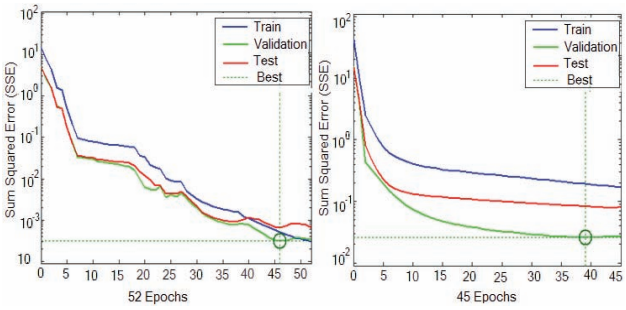
BrTP	BrE	BrNSS	λ	SSE BR	SSE BP
300	100	20	10	0,291	7,689
			0.01	0,413	0.754
			0.0001	0,260	0,754
		30	10	0,063	7,687
			0.01	0,050	0,752
			0.0001	0,090	0,754

Na osnovu rezultata iz tabele 1 vidimo da Bayesian regularizacija daje bolje rezultate, tj. manju grešku. Takođe vidimo da kada smanjimo λ greška će se malo smanjiti kod Bayesian regulation algoritma, dok kod Backpropagation greška se neće puno promeniti (promena je tek na petoj decimali). Sa povećanjem λ, Bayesian regulation backpropagation algoritam će davati malo manju grešku na kraju stote epohe od one kad je λ bio 0.01, ali će grafik imati veći pad. Greška će se posle desete epohe naglo smanjiti, pa će od desete nastaviti postepeno da opada. Backpropagation algoritam će za veliko λ davati jako veliku grešku čak i kada povećamo broj neurona u skrivenom sloju. Kada povećavamo broj neurona u skrivenom sloju, greška kod Bayesian regulation backpropagation algoritma će se smanjiti. Ukoliko povećamo broj epoha, greška će nastaviti da se smanjuje dok ne dostigne zadati cilj. Kada povećamo broj trening parametara dobijemo bolje rezultate, greška je tada manja kod oba algoritma. Iz tabele 1 vidimo da su najbolji rezultati dobijeni kada je λ=0.01 i kada je broj neurona u skrivenom sloju 30 (sl. 3).



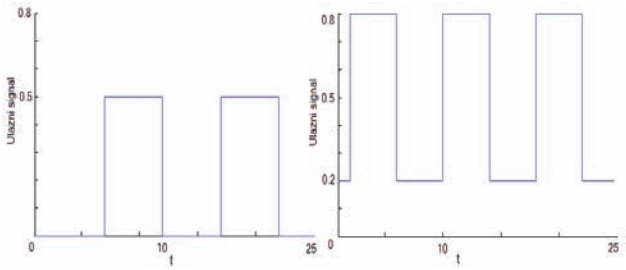
Slika 3. *Poređenje rezultata SSE kod Bayesian regulation algoritma (levo) i Backpropagation algoritma (desno) u programskom jeziku C#*

Na slici 4 vidimo rezultate dobijene u Matlab-u za trening skup od 300 parametara, λ=0,01 i broj neurona u skrivenom sloju 30.

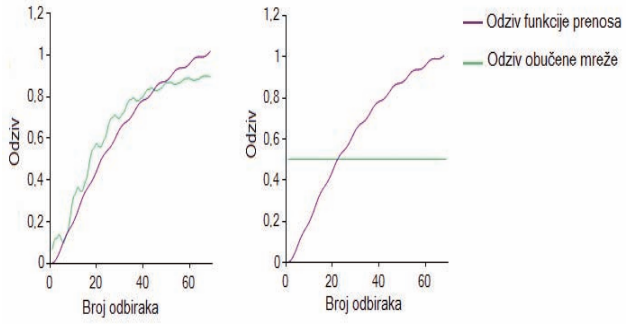


Slika 4. *Poređenje rezultata SSE kod Bayesian regulation algoritma (levo) i Backpropagation algoritma (desno) u programskom paketu Matlab*

Sa slike 4 vidimo da Bayesian regulation backpropagation algoritam daje bolje rezultate u odnosu na Backpropagation algoritam. Kada uporedimo rezultate ovih algoritama u C#-u i Matlab-u zaključujemo da Matlab daje bolje rezultate, tj. manja je greška u Matlabu i pre je dostignuta. Takođe, vidimo da su test rezultati približno isti u C#-u i Matlab-u, dok je samo obučavanje mreže bolje u Matlab-u. To je zato što Matlab koristi razne optimizacione metode u cilju poboljšanja treninga mreže. Menjanjem parametara u *nntoo*l-u greška se smanjuje odnosno povećava kao u tabeli 1.



Slika 5. *Prikaz pravougaonih signala za obučavanje mreže (levo) i za testiranje mreže (desno)*



Slika 6. *Uporedni prikaz odziva funkcije prenosa na signal pomerene amplitude i odziva obučene mreže Bayesian regulation backpropagation algoritmom (levo), odnosno Backpropagation algoritmom (desno).*

Takođe je mreža testirana sa drugim ulaznim signalima sa kojima nije prethodno obučavana kako bismo videli koliko će dobro neuronska mreža aproksimirati funkciju prenosa. Nakon što je neuronska mreža obučena dovođenjem pravougaonog signala amplitude 0,5, mreža je testirana pravougaonim signalom pomerene amplitude (slika 5). Na slici 6 vidimo uporedni prikaz odziva funkcije prenosa na signal pomerene amplitude i odziva obučene mreže Bayesian regulation backpropagation algoritmom (levo), odnosno Backpropagation algoritmom (desno).

Sa slike 6 vidimo da i u ovom primeru Bayesian regulation backpropagation algoritam daje bolje rezultate u odnosu na Backpropagation algoritam i da Backpropagation algoritam nije obučio mrežu. Takođe zaključujemo da statička neuronska mreža može da aproksimira dinamički model opisan funkcijom prenosa. Od broja prethodnih ulaza i prethodnih izlaza koji se dovode na ulaz neuronske mreže kao i od algoritma za obučavanje zavisi koliko će dobra ta aproksimacija biti.

6. ZAKLJUČAK

U radu je opisana implementacija Bayesian regulation backpropagation algoritma za obučavanje neuronskih mreža i njegovo poređenje sa Backpropagation algoritmom. Analizirani su rezultati dobijeni testiranjem na jednostavnoj funkciji prenosa u programskom jeziku C# i programskom paketu Matlab. Nakon analize rezultata vidimo da Bayesian regularizacija daje bolje rezultate od Backpropagation algoritma i da bi greška bila manja, predlaže se da se koristi manji learning rate ili da se poveća broj neurona u skrivenom sloju. Ako uporedimo rezultate dobijene u C#-u i Matlab-u vidimo da Matlab daje bolje rezultate, što pokazuje veću tačnost i pouzdanost ovih algoritama pisanih u Matlab-u, a samim tim pokazuje i da je nntool veoma kvalitetan alat za obučavanje neuronskih mreža.

7. LITERATURA

[1] M. Hagan, H. Demuth, and M. Beale, "*Neural Network Design"*, Boston, MA: PWS Publishing, 1996.

[2] Neal, R. M. (1994) "*Bayesian Learning for Neural Networks"*, Ph.D. Thesis, Dept. of Computer Science, University of Toronto

[3] J. Hertz, A. Krogh, R. G. Palmer, "*Introduction to the Theory of Neural Computation"*, Addison-Wesley, 1991.

[4] Yves Chauvin, David E. Rumelhart *"Backpropagation: theory, architectures, and applications* **",** Lawrence Erlbaum Associates, Inc. (1995)

[5] MathWorks – Matlab and Simulink for Technical Computing

[6] D. Foresee and M. Hagan, "*Gauss-Newton Approximation to Bayesian Learning"* Proceedings of the 1997 International Joint Conference on Neural Networks

[7] AForge Controls Library, Andrew Kirillov, 2006

Kratka biografija:



Ivana Todosijević rođena je u Kikindi 1986. godine. Od oktobra 2010. godine je apsolvent na Fakultetu tehničkih nauka, odsek Računarstvo i automatika, katedra za automatiku i upravljanje sistemima.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

PODELA GRAFA EVOLUTIVNIM PSO ALGORITMOM

EVOLUTIONARY PSO ALGORITHM FOR GRAPH PARTITIONING

Aleksandar Veselinović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj *– U radu je opisana implementacija podele grafa evolutivnim PSO optimizacionim algoritmom. Podela grafa se svodi na rešavanje optimizacionog problema u kome se čvorovi grafa grupišu u particije sa ciljem da se minimizuju veze između particija uz ograničenje da particije budu podjednake veličine. Evolutivni PSO algoritam je primenjen na podelu grafova koji opisuju različite modele distributivne elektroenergetske mreže, i rezultati su analizirani za različite parametre algoritma. Rezultati dobijeni pomoću evolutivnog PSO algoritma upoređeni su sa rezultatima dobijenim pomoću osnovnog PSO algoritma.*

Abstract *– This paper presents implementation of the graph partitioning of real electric power distribution network based on EPSO (Evolutionary Particle Swarm Optimisation) algorithm. Graph partitioning must be done so that there are as fewer connections between partitions as possible with the restriction that sizes of partitions have to be equal. Results are analyzed for various parameters. Also, results obtained by the EPSO algorithm are compared with results obtained using the basic PSO algorithm.*

Ključne reči*: Podela grafa, EPSO algoritam, PSO algoritam, Evolutivni algoritam*

1. UVOD

Često je potrebno predstaviti sisteme i uzajamne odnose koji postoje u sistemu. Grafovi su prirodan model za takvu reprezentaciju. Kako bi se paralelizacijom optimizovao rad sistema, potrebno je izvršiti podelu na particije, tako da u svakoj particiji budu elementi koji su povezani čvrstim vezama.

Jedan od načina podele sistema jeste primena algoritama za podelu grafova. Postupak podele grafa na više particija se postavlja kao optimizacioni problem gde je potrebno naći takvu podelu da veze među particijama budi minimalne, a pri tom da se zadovolji uslov da su sve particije približno jednake veličine.

Za podelu grafa koriste se opšti kao i specijalizovani algoritmi. Za dati problem podele grafova, koji opisuju različite modele distributivne elektroenergetske mreže, primenjen je Evolutivni PSO (EPSO) algoritam. Zanimljiva karakteristika ovog algoritma je to što on dopušta dvostruku interpretaciju, tj. može se posmatrati kao varijanta PSO algoritma ili kao varijanta evolutivnih algoritama [1].

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Aleksandar Erdeljan, docent

2. TEORIJA GRAFOVA

Grafovi su matematički objekti koje veoma često srećemo u svakodnevnom životu. Zbog velikog spektra primena, kao i izuzetno jednostavne veze definicije i osnovnih svojstava, grafovi su našli veliku primenu u naukama kao sto su elektrotehnika, računarstvo, hemija, fizika, biologija itd. Grafovi su posebno interesantni jer se pomoću njih veoma jednostavno mogu modelirati složeni problemi. Teorija grafova nalazi primenu i za opisivanje modela električnih mreža, mreža gasovoda, za predstavljanje saobraćajnica, itd.

2.1. Podela grafa

Modeli distributivne elektroenergetske mreže, koji su predmet razmatranja u ovom radu, su opisani grafom G=(V,E) koji se sastoji od čvorova (V) i grana (E). Problem podele grafa se bazira na podeli neorijentisanog grafa sa čvorovima i granama koji imaju određene težine. Čvorovi grafa predstavljaju elemente mreže koje je potrebno razvrstati i podeliti u određene particije. Svaki čvor je određen svojim identifikacionim brojem, a od informacija sadrži listu susednih čvorova, listu grana preko koje je povezan sa susednim čvorom, kao i *ID* particije kojoj čvor pripada. Grane koje povezuju čvorove definisane su svojim identifikacionim brojem, težinom grane, i informacijom o čvorovima koje povezuje.

3. KRITERIJUM OPTIMALNOSTI

Kako bi se particije optimalno kreirale potrebno je definisati kriterijum optimalnosti. Čvorove je potrebno podeliti na definisani broj particija tako da njihova povezanost unutar particije bude maksimalna, a pri tom da veličina particije ne prelazi maksimalnu dozvoljenu veličinu particije. Maksimalna povezanost unutar particije ujedno znači i minimalnu povezanost između particija. Veličina particije i se definiše kao suma veličina svih čvorova ($V\check{C}$) koji u nju ulaze:

$$VP_i = \sum_{j \in P_i} V\check{C}_j \tag{1}$$

Ograničenje u vidu maksimalne dozvoljene veličine particije (MDV) je:

$$MDV = (1+\varepsilon) \cdot \frac{1}{n_{part}} \sum_i VP_i \tag{2}$$

gde je:

VP_i - veličina particije i

n_{part} - željeni broj particija

ε- tolerancija odstupanja od srednje veličine particije, koja ima vrednost iz intervala:

$$x \in [0, \frac{(n_{part} - 1)}{n_{part}}] \tag{3}$$

Kriterijum optimalnosti je:

$$F = \max \sum_{part} f_{part} \quad (4)$$

gde je:

$$f_{part} = \sum_{p,q \in P_k} CM_{p,q} \quad (5)$$

f_{part}- funkcija definisana za svaku particiju P_k, pri čemu su čvorovi p i q raspoređeni u particiju P_k.

$CM_{p,q}$- kvadratna simetrična matrica povezanosti čvorova čiji elementi predstavljaju broj veza između čvorova p i q.

U cilju balansiranosti particija definisano ograničenje je formulisano na sledeći način (VP_k- veličina particije k):

$$VP_k \leq MDV, \forall k \in \{1,2,...n_{part}\} \quad (6)$$

4. ANALIZA EVOLUTIVNOG PSO ALGORITMA

Evolutivni PSO algoritam poseduje karakteristike standardnog Particle Swarm Optimization algoritma (PSO) i Evolutivnih Algoritama (EA).

4.1. Evolutivni algoritmi

Evolucija je neprekidan proces, u kojem se jedinke prilagođavaju promenljivim uslovima okoline. Evolutivni algoritmi predstavljaju algoritme za pretraživanje zasnovane na mehanizmima prirodne genetske evolucije jedne populacije jedinki pod dejstvom okruženja i genetskih operatora. EA su robusne i adaptivne metode koje se najčešće koriste za pronalaženje optimalnog rešenja problema. Osnovna tri mehanizma na kojima počivaju evolutivni algoritmi su mehanizmi selekcije, rekombinacije i mutacije. EA reprodukcioni ciklus je opisan sledećim pseudo kodom:

```
Napravi inicijalnu populaciju rešenja
Izračunaj stepen pogodnosti jedinke
While uslov prekida nije dozvoljen do
    Odredi jedinke za reprodukciju
    Izvrši rekombinaciju među jedinkama
    Mutiraj jedinke
    Izračunaj st. pogodnosti modifikovanih jedinki
    Generiši novu populaciju rešenja
End while
```

4.2. PSO algoritam

PSO (Particle Swarm Optimization) predstavlja stohastičku optimizacionu tehniku zasnovanu na kolektivnoj inteligenciji populacije. Populacija se sastoji od niza čestica koje se kreću kroz višedimenzionalni prostor i menjaju svoje pozicije u zavisnosti od sopstvenog iskustva i iskustva ostalih jedinki. Iz iteracije u iteraciju, svaka čestica generiše izdanak po definisanom "pravilu kretanja", koje se zasniva na činjenici da svaka čestica pamti koordinate u prostoru problema koje su povezane sa najboljim rešenjem koje je postignuto na nivou čestice (*localBest*), kao i na nivou celog roja čestica (*globalBest*). Ažuriranje pozicije čestice i (x_i), započinje ažuriranjem njene brzine kretanja (v_i) za svaku dimenziju (d) prema formuli:

$$v_{i,d} = W \cdot v_{i,d} + C_1 \cdot rand() \cdot \left(localBest_{i,d} - x_{i,d} \right) + C_2 \cdot rand() \cdot \left(globalBest_{d} - x_{i,d} \right) \quad (7)$$

gde je:

W - parametar inercije

C_1 - parametar individualnosti

C_2 - socijalni parametar

$(localBest_{i,d} - x_{i,d})$ - individualna komponenta

$(globalBest_{d} - x_{i,d})$ - socijalna komponenta

Pozicije čestica se ažuriraju prema formuli:

$$x_i = x_i + v_i \quad (8)$$

Čestice se kreću u prostoru pretrage u blizini čestice najbolje na nivou roja, i ne istražuju ostali prostor. Kvalitet čestice se procenjuje na osnovu kriterijuma optimalnosti kojeg je potrebno zadovoljiti. Uslov za promenu vrednosti lokalne i globalne najbolje čestice je da čestica nakon promene pozicije ima bolju vrednost kriterijuma optimalnosti od njene dotadašnje najbolje pozicije na nivou čestice tj. na nivou populacije. Osnovni koraci PSO algoritma su opisani sledećim pseudo kodom:

```
Napravi inicijalnu populaciju od n rešenja (čestica)
While uslov prekida nije dozvoljen do
    foreach česticu i
        Ažuriranje brzine i pozicije čestice i
        Procena kvaliteta čestice i
        Promena najboljih vrednosti čestica
    end for
    Vrati gBest
End while
```

4.3. EPSO-Evolutionary Particle Swarm Optimization

EPSO algoritam je evolutivni algoritam koji pravilo kretanja čestica definisano u osnovnom PSO algoritmu koristi kao mehanizam rekombinacije. Promenljive veličine u EPSO algoritmu su podeljene na objektne parametre (pozicije X) i strategijske parametre (težine W). Ovaj hibridni algoritam kombinuje dobre osobine evolutivnih algoritama i osnovnog PSO algoritma. Sledi pseudo kod sa osnovnim koracima EPSO algoritma:

```
Napravi inicijalnu populaciju od n rešenja (čestica)
While uslov prekida nije dozvoljen do
    foreach česticu i
        Replikacija čestica
        Mutacija strategijskih parametara
        Reprodukcija
        Procena kvaliteta čestice i
        Promena najboljih vrednosti čestica
        Selekcija n čestica
    end for
    Vrati gBest
End while
```

Tokom replikacije generiše se r kopija čestica (uključujući i originale), sledi mutacija strategijskih parametara svake čestice i generisanje novih r izdanaka sa r različitih pozicija za svaku česticu pomoću definisanog pravila rekombinacije. Selekcija se vrši nad generisanim izdancima čestica, tokom koje se biraju najbolje čestice koje će formirati narednu generaciju. Generacijom izdanaka čestica se upravlja pomoću operatora mutacije i rekombinacije, pod kontrolom selekcije. [2]

Pravilo kretanja čestica u EPSO algoritmu je veoma slično kao kod PSO algoritma, tj. bazira se na tri komponente: brzini, memoriji i kooperaciji. Grafički prikaz kretanja čestica je prikazan na slici 1. Razlika u odnosu na osnovni PSO algoritam je mutacija težinskih parametara. Pravilo kretanja EPSO algoritma je formulisano na sledeći način:

$$X_i^{(k+1)} = X_i^{(k)} + V_i^{(k+1)}$$
$$V_i^{(k+1)} = w_{i1}^* \cdot V_i^{(k)} + w_{i2}^* \cdot (b_i - X_i) + w_{i3}^* \cdot (b_g^* - X_i) \cdot P \quad (9)$$

gde je:

b_i-najbolje rešenje na nivou čestice

b_g-najbolje rešenje na nivou roja čestica

$X_i(k)$-pozicija čestice i u generaciji k

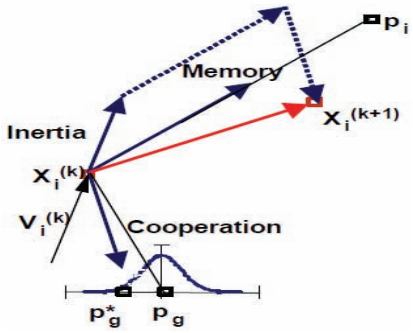
w_{i1}-težinski faktor koji utiče na brzinu kretanja čestice

w_{i2}-težinski faktor koji kontroliše uticaj *localBest* čestice

w_{i3}-težinski faktor koji kontroliše uticaj *globalBest* čestice, odnosno razmenu informacija među česticama

P- faktor komunikacije

Simbolom * označeni su parametri nad kojima se primenjuje mutacija.



Slika 1. *Kretanje čestica u EPSO algoritmu [1]*

Mutacija strategijskih parametara w u w^* se vrši pomoću multiplikativne logoritamske raspodele:

$$w_i^* = w_i [\log N(0,1)]^\tau \quad (10)$$

ili na osnovu aditivne Gausove raspodele:

$$w_i^* = w_i + \sigma \cdot N(0,1) \quad (11)$$

gde su σ i τ parametri koji utiču na stepen mutacije i unapred su definisani.

Još jedna bitna karakteristika EPSO algoritma je to što se mutacija ne odnosi samo na komponente definisane u pravilu kretanja, već i na najbolju česticu na nivou roja. Mutacija *globalBest* čestice se vrši na sledeći način:

$$b_g^* = b_g + w_{i4}^* \cdot N(0,1) \quad (12)$$

gde je:

w_{i4} - četvrti strategijski parametar koji se odnosi na česticu i. On kontroliše veličinu okoline *globalBest* čestice, gde postoji najveća verovatnoća za pronalazak pravog najboljeg globalnog rešenja.

Evolutivni algoritmi su se pokazali kao veoma dobre metode prilikom primene na složenije probleme i sisteme, dok samo-adaptivni modeli utiču na poboljšanje efikasnosti algoritma u većini slučajeva. U evolutivnim algoritmima samo-adaptivna karakteristika se ogleda u manipulisanju operatorom rekombinacije pod kontrolom selekcije. Posmatrajući EPSO algoritam sa stanovišta EA, EPSO predstavlja evolutivni algoritam sa pravilom kretanja pozajmljenim od osnovnog PSO algoritma, reprezentatovanim kao oblik rekombinacije. Pošto EPSO usvaja pravilo kretanja čestica, može da se posmatra i kao poseban slučaj PSO algoritma, gde se kretanjem čestica upravlja pomoću težina koje se menjaju mehanizmom samo-adaptacije, i time omogućuju veće pretraživanje lokalnog prostora. Primenom mutacije težinskih parametara smanjuje se mogućnost čestica da dostignu lokalni minimum.

5. REALIZACIJA PROGRAMSKOG REŠENJA ZADATKA

Programsko rešenje zadatka je realizovano programskim jezikom C# uz oslonac na razvojno okruženje Microsoft Visual Studio 2008.

5.1. Učitavanje i formiranje grafa

Nakon učitavanja podataka,koji su smešteni u fajlovima „čvorovi.txt“ i „grane.txt“, koji opisuju čvorove i grane grafa, kreiraju se objekti tipa *Čvor* i *Grana*, a na osnovu njih se vrši kreiranje početnog grafa.

5.2. Realizacija EPSO algoritma

Adaptacijom EPSO algoritma na problem podele grafova, čestice koje predstavljaju neku podelu, imaju dimenziju jednaku broju čvorova koje je potrebno podeliti po participama. Svaka čestica predstavlja neku podelu po participama. Pozicije čestica se ažuriraju definisani broj iteracija, u skladu sa definisanim kriterijumom optimalnosti. Cilj je pronaći najbolje rešenje, odnosno najbolju poziciju čestice.

Za rešavanje definisanog problema podele grafova, replikacija čestica je rađena sa stepenom replikacije r=3. U svakoj iteraciji roj čestica se utrostručuje, i svakim od rojeva se manipuliše u cilju dobijanja najboljeg rešenja. Nad prvim repliciranim rojem se primenjuje EPSO algoritam, dok se nad preostala dva roja primenjuju osnovni mehanizmi evolutivnih algoritama: ukrštanje i mutacija.

Za roj manipulisan EPSO algoritmom, u svakoj iteraciji za svaku česticu, vrši se mutacija strategijskih parametara na osnovu Gausove aditivne mutacije definisane izrazom (11). Parametar σ, koji utiče na promenu težinskih parametara čestica je fiksno definisan, i njegovu vrednost najbolje je držati na niskom nivou, kako bi obezbedili laganu i postepenu promenu težinskih parametara. U svakoj iteraciji vrši se i mutacija najbolje čestice roja prema izrazu (12), gde W_{i4} predstavlja četvrti strategijski parametar nad kojim je takođe primenjena aditivna Gausova mutacija. Na osnovu testiranja algoritma parametru mutacije težinskih faktora je dodeljena vrednost σ =0.2, dok se kao najbolja vrednost pri mutaciji najbolje čestice roja pokazala vrednost σ=0.1. Sledeći korak je ažuriranje brzine (Vi) i pozicije (Xi), na osnovu pravila kretanja čestica koje je definisano izrazom (11). Neophodno je algoritam prilagoditi datom diskretnom problemu, tako da se vrši modifikacija brzine zaokruživanjem na najbliži ceo broj, kako bi dobili diskretne vrednosti:

$$V_i = round(V_i) \quad (13)$$

Pozicije čestica moraju imati celobrojne vrednosti iz intervala [0, n_{part}-1], tako da na kraju se vrši modifikacija pozicija po modulu n_{part}:

$$X_i = |\mathrm{mod}(X_i, n_{part})| \quad (14)$$

gde je mod(x,y) funkcija koja izračunava ostatak pri deljenju x i y.

Nad preostala dva replicirana roja primenjeni su standardni evolutivni mehanizmi ukrštanja i mutacije. Cilj ukrštanja je obezbediti novi kvalitet, u smislu novih čestica. Nad česticama se vrši ukrštanje u jednoj tački. Ukrštanje se primenjuje na odabrane parove roditelja. Kod prvog repliciranog roja par roditelja koji se ukršta

predstavlja trenutna čestica i lokalna najbolja čestica, dok kod drugog repliciranog roja ukršta se trenutna čestica sa česticom izabranom na slučajan način. Ukrštanje i mutacija vode ka boljim rezultatima u smislu kvaliteta populacije. Stepen mutacije se drži na niskom nivou, da se dobre čestice dobijene ukrštanjem ne bi izgubile.

Na osnovu definisanog kriterijuma optimalnosti procenjuje se kvalitet izdanaka celokupnog roja čestica. Rozicija X_i se pamti kao najbolja na nivou individue (*localBest*) i celog roja (*globalBest*), ukoliko je bolja od dotadašnje najbolje pozicije.

Poslednji korak u realizaciji algoritma je postupak selekcije. Ovim postupkom se među izdancima svih čestica biraju čestice sa najboljim vrednostima kriterijumske funkcije, koje će da "prežive" i formiraju narednu generaciju. Postupak selekcije je zapravo to što čini ovaj algoritam specifičnim, pošto ovaj postupak obezbeđuje neku vrstu "takmičenja" među česticama. Nakon definisanog broja iteracija, EPSO algoritam vraća poziciju *globalBest* čestice kao konačno rešenje problema, koja predstavlja podelu čvorova po participama.

6. REZULTATI TESTIRANJA

Testiranje algoritma je vršeno na računaru sa Intel Core2Duo procesorom na 2 GHz i 2 GB RAM memorije. Kao test modeli korišćeni su realni modeli elektroenergetskih mreža, čije karakteristike su prikazane u tabeli 1.

Tabela 1. *Modeli elektroenergetskih mreža*

Model mreže	Broj čvorova	Broj grana
bg54	54	44
bg63	63	52
pe106	106	52
it206	206	286
bg5x	315	260

Analizirani su rezultati prosečnih vrednosti funkcije cilja. Za manje modele mreže (bg54, bg63, pe106) rađena su testiranja sa 40 ponavljanja, a za preostale modele sa 20 ponavljanja, za svaku od kombinacija parametara prikazanih u tabeli 2.

Tabela.2. *Tabela parametara*

Broj particija	2, 3, 4, 5 i 6
Stepen tolerancije	0.1 (10%)
Broj iteracija	5000
Stepen mutacije težina σ	0.2
Stepen mutacije *globalBest*	0.1
Faktor komunikacije *P*	1
Broj čestica	100

U tabeli 3. prikazani su rezultati podele grafa dobijeni pomoću evolutivnog PSO i osnovnog PSO algoritma. Testiranja oba algoritma su rađena za iste kombinacije parametara. Na osnovu dobijenih rezultata može se zaključiti da evolutivni PSO algoritam vrši dobru podelu grafa za definisane parametre. Poređenjem rezultata dobijenih pomoću ova dva algoritma može se zaključiti da evolutivni PSO algoritam daje značajno bolje rezultate u svim slučajevima podele. Razlike u dobijenim rezultatima su najupečatljivije pri podelama većih grafova, kao i pri podelama na veći broj particija. Analiziranjem rezultata oba algoritma, prednost evolutivnog PSO algoritma se ogleda u proširivanju prostora pretrage rešenja, odnosno uvođenjem mehanizma samo-adaptivne mutacije težinskih parametara kao i laganom promenom najbolje čestice rešenja.

Tabela 3. *Rezultati osnovnog PSO i EPSO algoritma*

Br.part.	Model	PSO	EPSO
2	bg54	392	398
	bg63	475	482
	pe106	149	161
	it206	853	914
	bg5x	2379	2442
3	bg54	385	385
	bg63	455	471
	pe106	146	155
	it206	746	894
	bg5x	2190	2389
4	bg54	363	373
	bg63	444	456
	pe106	138	148
	it206	674	852
	bg5x	2062	2351
5	bg54	336	354
	bg63	419	447
	pe106	129	142
	it206	603	829
	bg5x	1949	2336
6	bg54	310	352
	bg63	360	422
	pe106	120	132
	it206	530	820
	bg5x	1933	2275

7. ZAKLJUČAK

U radu je opisana primena evolutivnog PSO algoritma na problem particionisanja realnih modela elektrodistributivnih mreža predstavljenih grafovima. Opisani su osnovni pojmovi teorije grafova, kao i algoritam korišćen za realizaciju datog problema. Nakon testiranja i analize rezultata dobijenih pomoću evolutivnog PSO algoritma, može se zaključiti da ovaj algoritam daje mnogo bolje rezultate u odnosu na osnovni PSO algoritam.

8. LITERATURA

[1] Vladimiro Miranda, Hrvoje Keko, Alvaro Jaramillo "EPSO: Evolutionary Particle Swarms", Vol.66, Springer, 2007

[2] Vladimiro Miranda, Nino Fonesca "EPSO- Best-of-Two-Worlds Meta-Heuristic Applied to Power System Problems", Conference on Evolutionary Computation, Hawaii, 2002

[3] J.A.Bondy, U.S.R. Murty S.N.Deepa, „Graph theory", „Graduate texts in matematics", Vol. 244., Springer, 2008

Kratka biografija:



Aleksandar Veselinović je rođen 19.01.1987. u Gračacu. Završio je srednju elektrotehničku školu „Mihajlo Pupin" u Novom Sadu 2005. godine i iste godine upisao Fakultet tehničkih nauka u Novom Sadu, odsek Elektrotehnika i računarstvo, smer Računarstvo i automatika, usmerenje Automatika i upravljanje sistemima.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

DINAMIČKA PRERASPODELA MODELA DISTRIBUTIVNE ELEKTROENERGETSKE MREŽE PRIMENOM WF I LMSR ALGORITAMA

DYNAMIC REPARTITIONING OF THE MODEL OF POWER DISTRIBUTION NETWORK USING WF AND LMSR ALGORITHMS

Dejan Čotra, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu je opisana implementacija preraspodele delova već izdeljenog modela distributivne elektroenergetske mreže u kojem je došlo do poremećaja balansiranosti delova mreže usled dinamičke promene statusa pojedinih elemenata (zatvaranje i otvaranje prekidača). Preraspodela se vrši pomoću dva algoritma: 1) WF algoritma i 2) LMSR algoritma. Izvršeno je poređenje ovih algoritama na realnom primeru modela mreže.*

Abstract – *This paper describes implementation of redistribution of the partitions which represent already divided model of distribution power network, in which has occurred a disorder in balance of the size due to dynamic changes in the state of some elements (closing and opening of switches). Redistribution is done using two algorithms: 1) WF algorithm and 2) LMSR algorithm. Algorithms were tested on actual power network model.*

Ključne reči: *Model elektrodistributivne mreže, WF difuzioni algoritam, LMSR algoritam*

1. UVOD

Jedan od ciljeva svakog distribuiranog sistema je ravnomerno opterećenje računara i efikasno korišćenje raspoloživih resursa. Podaci o celokupnom modelu elektroenergetske mreže mogu biti smešteni na jedan računar, ali je za takvo rešenje potreban veliki memorijski prostor i snažan (skup) hardver. Upravo iz tog razloga se često pribegava alternativnom rešenju gde je obrada podataka na više računara, u okviru distribuiranog sistema. Ovakvim pristupom se znatno povećavaju zahtevi softverskog dela sistema, ali se zato smanjuje cena sistema (jeftiniji hardver) i potencijalno povećava brzina rešavanja problema.

Glavni zadatak ovog rada je kreiranje distribuiranog sistema u kome se teži da svi računari koji ga čine budu podjednako opterećeni podacima o mreži nad kojima se vrše proračuni, u cilju što brže obrade podataka. Koreni predstavljaju grupe međusobno povezanih elemenata elektroenergetske mreže koji se mogu istovremeno proračunavati. Više korena čini jedan region. Na svaki od raspoloživih računara u okviru distribuiranog sistema smešta se po jedan region. Proračun elemenata jednog od regiona ne utiče na proračun elemenata drugih regiona.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Aleksandar Erdeljan, docent.

Veličine regiona bi trebalo da budu približno iste, tj. da sadrže približno istu količinu podataka kako bi računari distribuiranog sistema bili približno jednako opterećeni. Inicijalno se koreni raspoređuju tako da svi računari budu približno isto opterećeni, ali u realnom vremenu dolazi do promene stanja provodne opreme (zatvaranje i otvaranje grana) pa se i topološki izgled mreže menja. Ove promene ogledaju se u činjenici da neki elementi moraju da se prebace na neki drugi računar u cilju istovremene obrade, pa može doći do remećenja balansiranosti sitema. Kada je jedan procesor preopterećen, a neki dugi suviše malo opterećen, sistem se nalazi u nebalansiranom stanju. Kao reakcija na ovakvo stanje prebacuje se deo podataka sa preopterećenih računara na malo opterećene kako bi se ponovo uspostavilo balansirano stanje. Pri tome je potrebno voditi računa da na svakoj mašini budu elementi koji su međusobno povezani pa se istovremeno proračunavaju.

Zadatak algoritama za preraspodelu je da u što kraćem vremenskom roku odluči koji se podaci prebacuju i gde kako bi se ponovo uspostavilo balansirano stanje. Prilikom izvršavanja algoritmi za preraspodelu moraju da minimizuju neke parametre, kao što su: ukupna količina podataka koju šalju svi regioni (*TotalV*), maksimalna količina podataka koju šalje/prima jedan region (*MaxV*) [1], broj veza između korena različitih regiona (*EdgeCut*). Minimizacijom ovih parametara smanjuje se vreme preraspodele, tj. vreme od nastanka nebalansiranog stanja u sistemu do uspostavljanja ponovnog balansiranog stanja. Algoritmi za preraspodelu koji su implementirani u ovom radu su:

1) *WF* (*Wavefront Diffusion*) [1] [2]

2) *LMSR* (*Locally Matched Scratch-Remap*) [1] [3]

2. MODELOVANJE MREŽE, TOPOLOŠKA ANALIZA I PROBLEM PRERASPODELE

Teorija grafova nalazi veliku primenu u praksi, a između ostalog se koristi i za opisivanje modela električnih mreža. Električni elementi u elektroenergetskoj mreži (transformatori, prekidači, itd.) mogu se prikazati kao grane grafa, a mesta njihovog spajanja kao čvorovi.

Prilikom modelovanja elektroenergetske mreže korišćene su četiri različite klase podataka. Prva klasa opisuje elemente kao što su prekidači, transformatori, zove se *Grana* i sadrži parametre: *idelementa* (jedinstveni id za svaku granu), *idčvorova* (identifikatore čvorova koje povezuje grana), *tipelementa* (informacija da li je element transformator, prekidač itd.), *status* (informacija da li je element zatvoren ili otvoren). Drugom klasom podataka

opisana su mesta spajanja grana (prekidači, transformatori, itd.), ova klasa podataka zove se *Chvor* i sadrži parametre: *idčvora* (jedinstveni identifikator čvora), listu grana koje ih povezuju sa drugim čvorovima, listu susednih čvorova, *idoblasti* (identifikator oblasti kojoj pripadaju). Treća klasa podataka, *Oblast*, predstavlja skup povezanih elemenata, ima sledeće parametre: *idoblasti* (jedinstveni identifikator oblasti), listu grana koje pripadaju datoj oblasti, listu čvorova koji pripadaju datoj oblasti, *idregiona* (identifikator procesora na kojem se obrađuju podaci koji pripadaju datoj oblasti), *težina* (ukupan broj grana koji pripadaju datoj oblasti). Četvrta klasa podataka, *Region*, predstavlja skup oblasti čiji se podaci proračunavaju na istom računaru, sadrži sledeće parametre: *idregiona* (jedinstveni identifikator regiona), listu oblasti koju sadrži, *težina* (suma težina oblasti koje sadrži), kao i metode koje nam daju informacije o trenutnoj opterećenosti regiona, tj. procesora na kojem se podaci datog regiona obrađuju.

U fazi topološke analize izdvajaju se samo napajane oblasti, one koje sadrže u sebi aktivnu granu tipa *supply line*. Napajane oblasti se nazivaju *koreni*. *Koreni* predstavljaju grupe povezanih elemenata koji moraju da se proračunavaju zajedno a njihove vrednosti nemaju uticaja na elemente drugih *korena.* Na ovaj način se formira od inicijalnog jediničnog grafa, koji se kreira prilikom učitavanja podataka, težinski graf i time se dobija strukturu podataka koju koriste algoritmi za preraspodelu. *WF* i *LMSR* algoritam koriste korene kao težinske čvorove a otvorene veze između njih kao grane, gde je težina korena jednaka broju grana koji se nalazi u njoj, a težina grane je jednaka broju otvorenih veza između dva korena.

Koreni se inicijalno raspoređuju u regione, svaki region se obrađuje na posebnom procesoru. Cilj distribuiranog sistema koji je implementira u ovom radu je da vodi računa o opterećenju procesora, tj. da ne dozvoli situacije u kojima jedan od računar za obradu podataka postaje preopterećen. Optimalna vrednost veličine regiona je:

$$Rs = \sum_{i=1}^{n} \frac{Rt(i)}{n} \qquad (1)$$

gde je $Rt(i)$ težina i-tog regiona, a n ukupan broj regiona. Region je izbalansiran ako njegova težina ne prelazi vrednost $Rs\cdot(1+x)$, gde je x vrednost iz skupa $x\in\{0,(n-1)/n\}$.

Vremenom dolazi do promena stanja opreme (zatvaranje i otvaranje prekidača). Zatvaranjem prekidača između korena različitih regiona može doći do prelaska jednog od korena u susedni region, ako pri tome težina regiona koji prima koren pređe vrednost $Rs\cdot(1+x)$, kaže se da je sistem prešao u nebalansirano stanje.

Relizovan je distribuirani sistem u okviru kojeg se algoritmi izvršavaju centralizovano. Jedan računar u okviru sistema (koordinator) ima informacije gde se svi elementi trenutno nalaze i koje je njihovo stanje (otvoren ili zatvoren). Koordinator prima informacije od klijenata o promeni stanja prekidača i prosleđuje ih računarima na kojima se obrađuju realni podaci o elementima mreže. U slučaju zatvaranja prekidača između dva korena koji se nalaze u različitim regionima dolazi do prelaska manjeg korena (ima manji broj elemenata) u region većeg. Kako koordinator ima informacije gde se koji element nalazi može da proračuna da li je region koji je primio koren postao preopterećen. Kada ustanovi da je neki od regiona postao preopterećen koordinator poziva algoritme za preraspodelu koji proračunavaju koji elementi treba da se prebace i gde kako bi se uspostavilo novo balansirano stanje. Zatim inicira proces preraspodele i šalje informacije o preraspodeli računarima koji obrađuju podatke, oni su tokom tog vremena blokirani. Oni primaju te informacije (koje podatke treba da prebace i gde) od koordinatora i započinju proces razmene koristeći *WCF* protokol za komunikaciju.

3. WF ALGORITAM

WF algoritam [1][2] spada u grupu difuzionih algoritama, koreni se pri balansiranju kreću u vidu „talasa“ od preopterećenih regiona ka malo opterećenim regionima odatle i ime algoritma *WaveFront* (*wave* engl. talas). Izborom regiona donora počinje izvršavanje *WF* algoritma. Region donor je onaj koji u tekućoj iteraciji šalje korene kako bi se uspostavilo njegovo balansirano stanje. Kada se izbalansira trenutni region donor, algoritam bira novi region koji će slati korene. Algoritam nastavlja svoj rad sve dok se ne postigne balansiranost svih regiona. Iterativnim postupkom (za svaku iteraciju se bira novi donor i samo on može da šalje korene) postiže se da region donor prilikom slanja korena ima na raspolaganju najveći mogući izbor korena. Prilikom odlučivanja koji se koren šalje prednost imaju koreni koji se ne nalaze u svom početnom regionu (region u kojem su se nalazili pre početka rada algoritma), čime se smanjuje ukupan broj podataka koji menjaju region (minimizuje se parametar *TotalV*).

Prva faza algoritma je izbor regiona donora. Odabir se vrši tako što se za region donor bira region čiji je odnos veličina *outflow* i *inflow* najveća vrednost. Ove dve veličine se dobijaju na sledeći način: Rešavanjem sistema linearnih jednačina $L\cdot\lambda = b$ [2], gde je:

$$L=\begin{bmatrix} \deg(1) & -1/0 & \dots & -1/0 \\ -1/0 & \deg(2) & \dots & -1/0 \\ -1/0 & -1/0 & \ddots & -1/0 \\ -1/0 & -1/0 & \dots & \deg(n) \end{bmatrix}_{n\times n} \quad \lambda=\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix}_n \quad b=\begin{bmatrix} Rt(1)-Rs \\ Rt(2)-Rs \\ \vdots \\ Rt(n)-Rs \end{bmatrix}_n \qquad (2)$$

L je matrica veličine nxn, n predstavlja ukupan broj regiona. Formira se tako što za $L_{i,j}$, kad je $i = j$ (elementi na dijagonali) vrednost jednaka $deg(i)$ ukupan broj veza koji i-ti region ima sa korenima koji se nalaze u drugim regionim, kad je $i \neq j$ popunjava se sa -1 ako su i-ti i j-ti region susedi ili 0 ako nisu susedi (ako ne postoji ni jedna potencijalna veza koja ih povezuje). Vektor rešenja λ, koji se još naziva difuziono rešenje (*diffusion solution*), vektor b_i se izračunava kao trenutna težina i-tog regiona umanjena za optimalnu vrednost regiona Rs.

Vektor λ se dobija metodom *Gauss-Seidel* za numeričko rešavanje sistema linearnih jednačina. Zatim se na osnovu tog rešenja za svaki region i ($i = 1\dots n$), računa vektor Rl, gde je $Rl_{i,j}$ ($j = 1\dots n$), jednako $\lambda_i - \lambda_j$, ako je $i \neq j$ tj. 0 ako je $i = j$. Pozitivne vrednosti se sabiraju i one daju *outflow*, zbir apsolutnih vrednosti negativnih veličina čini vrednost *inflow*. Kada se ove dve vrednosti izračunaju za sve

regione, iteracija počinje za region čiji je odnos *outflow/inflow* najveći.
Sledeća faza u radu *WF* algoritma je izbor korena koji šalje region donor. Koreni (težinski čvorovi) koji imaju otvorene veze sa čvorovima drugih regiona nazivaju se granični. Prva faza pri izboru čvora koji se šalje je sortiranje.
Sortiranje se vrši po sledećem kriterijumu: prvi u listu se smeštaju čvorovi koji imaju bar jednu otvorenu vezu sa čvorom koji je u nekom malo opterećenom regionu (*underweight*), zatim idu čvorovi koji imaju bar jednu otvorenu vezu sa čvorom koji je u nekom preopterećenom regionu (*overweight*), na kraju idu čvorovi koji imaju otvorene veze samo sa čvorovima koji se nalaze u balansiranim regionima (da se ne bi narušila balansiranost regiona ostvarena u prethodnim iteracijama). Prilikom slanja čvorova prednost uvek imaju čvorovi koji se ne nalaze u svom *home* regionu (region u kojem su se nalazili pre početka rada algoritma), oni se još nazivaju i prljavi (*dirty*) čvorovi. Ovakvom strategijom slanja se postiže da čvorovi prave više „skokova“ iz regiona u region a da pri tome ne povećavaju ukupnu količinu podataka koja se prebacuju sa računara na računar u cilju uspostavljanja balansiranog stanja.
U slučaju da je čvor kandidat za slanje sused sa više *underweight* regiona šalje se u onaj region sa čijim čvorovima ima više otvorenih veza čime se minimizuje ukupan *EdgeCut* (ukupan broj otvorenih veza između čvorova različitih regiona).

4. LMSR ALGORITAM

LMSR algoritam [1][3], za razliku od difuzionih algoritama koji imaju informaciju o tome gde se koji podaci trenutno nalaze, ne poseduje informacije tog tipa nego novo balansirano stanje proračunava iz početka (*from scratch*). Trenutni raspored čvorova po regionima, tj. informacije o trenutnom stanju u *LMSR* algoritam se integrišu pomoću matrice sličnosti. Metrike koje može da minimizuje ovaj algoritam su: ukupna količina podataka koja se prenosi (*TotalV*) i maksimalna količina podataka koju šalje/prima jedan region (*MaxV*). U ovom radu korišćena je *TotalV* metrika. Minimizacija *TotalV* parametra postiže se pretragom matrice i preslikavanjem *id*-jeva regiona, u tu svrhu implementirana su dva algoritma *Greedy* i *Brute Force*.
Nakon proračunavanja novog balansiranog stanja konstruiše se matrica sličnosti na sledeći način [3]:

1) Formira se matrica sličnosti veličine $n \times n$, gde je n broj regiona. Redovi matrice predstavljaju stare regione, a kolone predstavljaju novoproračunate regione. Svaki element matrice $S_{i,j}$ predstavlja količinu podataka starog regiona i koja se nalazi u novoizračunatom regionu j.

2) Odabira se tačno n elemenata tako da svaki red i kolona sadrže tačno jedan selektovani element, tako da se neka metrika, u našem slučaju *TotalV*, optimizuje. Biraju se elementi tako da njihova suma bude maksimalna, time se dobija najveće moguće preklapanje između starih i novih regiona, a samim tim minimalni *TotalV*.

3) Za svaki selektovani element $S_{i,j}$ vrši se preslikavanje tj. region i postaje region j.

Jedan od algoritama implementiranih u ovom radu je *Greedy* algoritam pretrage matrice sličnosti. Optimalno rešenje nalazi u jednom prolazu. Sortira elemente matrice u opadajući poredak. Zatim kreće sa preslikavanjem, svaki novi region mora da bude preslikan. Kada se preslikaju svi regioni algoritam menja *id*-jeve u skladu sa odlukom o preslikavanju. Ne garantuje nalaženje minimalnog rešenja, ali ima veoma malo vreme izvršavanja.
Pored *Greedy* algoritma pretrage matrice implementiran je i *Brute Force* algoritam. Za razliku od *Greedy* algoritma garantovano nalazi minimalno rešenje. *Brute Force* algoritam za svaku moguću kombinaciju preslikavanja ispituje koliki bi bio *TotalV* u slučaju da se baš takvo preslikavanje izvrši. Kada se radi o raspodeli na četiri regiona potrebno mu je 24 iteracije da nađe rešenje, prilikom podele na šest regiona potrebno mu je 720 iteracija, povećanjem broja regiona i broj iteracija se znatno povećava a samim tim i vreme izvršavanja algoritma.
Primećeno je prilikom testiranja da algoritmi daju identične rezultate i da je pri tome vreme izvršavanja *Greedy* algoritma drastično manje. Takođe, treba naglasiti da sa povećanjem broja regiona vreme izvršavanja *Greedy* algoritma ne varira toliko kao vreme *Brute Force* algoritma. Upravo iz ovih razloga u nastavku rada dati su samo rezultati *LMSR* algoritma implementiranog uz pomoć *Greedy* pretrage matrice sličnosti.

5. REZULTATI TESTIRANJA

Algoritmi su implementirani u programskom jeziku C# u *Microsoft .NET* okruženju. Za komunikaciju je korišćen *WCF* protokol. Testiranje je izvršeno na računarima sa *Intel Core2Duo* procesorom na 2 *GHz* i 2 *GB RAM* memorije, sa *Windows* operativnim sistemom.
WF i *LMSR* algoritmi testirani su na tri različite elektroenergetske mreže:

1) Mreža 1 - umanjeni model elektroenergetske mereže Beograda, broj elemenata 164597, 54 težinska čvora

2) Mreža 2 - prošireni model elektroenergetske mereže Beograda, broj elemenata 166391, 63 težinska čvora

3) Mreža 3 - model elektroenergetske mereže grada Milana, broj elemenata 425613, 206 težinskih čvorova

Za sve tri mreže testovi su rađeni za podele na četiri, šest i osam regiona. Rađeno je po 100 iteracija za svaku od podela. Tolerancija je za sve tri mreže ista, $x = 0{,}1$. Kvalitet algoritama je ocenjivan na osnovu sledećih parametara: *Greška* (E) maksimalno dozvoljeno odstupanje od optimalne veličine regiona. *EdgeCut* (Ec) ukupan broj otvorenih veza između čvorova koji se nalaze u različitim regioma. *TotalV* (Tv) ukupna količina podataka koja napušta svoj region i prebacuje se u neki drugi u cilju uspostavljanja balansiranosti. *MaxV* (Mv) maksimalna količina podataka koju šalje/prima jedan region prilikom preraspodele. *Vreme Algoritma* (T) vreme za koje algoritam odluči koji čvorovi napuštaju svoj region i u koji se prebacuju, kako bi se postigla balansiranost. Prosečne vrednosti parametara za sve tri mreže date su u tabeli 1.

Tabela 1. *Rezultati testiranja za sve tri mreže*

x = 0.1		Mreža 1		Mreža 2		Mreža 3	
Reg	Alg	WF	LMSR	WF	LMSR	WF	LMSR
4	E[%]	7.3	7.49	5.45	9.57	7.38	7.89
	Ec	146.38	177	98.83	199	630.88	719
	Tv	19969.6	31791.4	14105.3	24639.3	5806.6	41150.8
	Mv	15188.8	16567.2	11242.5	14008.4	5799.7	19882.2
	T[ms]	9.04	0.28	6.7	0.27	5.58	0.41
6	E[%]	8.06	9.36	6.9	7.3	7.3	9.84
	Ec	149.1	251	91.56	230	623.56	688
	Tv	30631.6	49491.4	18327.4	45063.4	6407.6	47005.8
	Mv	16561.6	26427.8	10708.9	18455.6	5904.4	18203.5
	T[ms]	30.82	0.49	24.54	0.3	13.15	0.59
8	E[%]	7.61	9.93	9.43	9.8	8.55	9.32
	Ec	207.61	237	107.96	219	705.59	773
	Tv	40682.2	46251.9	20653.9	51412.3	9508.9	34937.3
	Mv	23760.9	27311.5	13100.8	21154.5	7924.0	12199.9
	T[ms]	41.6	0.8	27.56	0.51	16.18	0.76

Kad su u pitanju parametri *Greška*, *EdgeCut*, *TotalV*, *MaxV* algoritam *WF* daje bolje rezultate za sve tri mreže, bilo da se radi o podeli na četiri, šest ili osam regiona. Kako daje manje prosečne vrednosti za *Grešku* može se konstatovati da *WF* daje kvalitnije particionisanje, tj. ujednačenije opterećuje procesore koji obrađuju podatke. *EdgeCut* predstavlja broj otvorenih veza između čvorova različitih regiona i kada se neka veza zatvori dolazi do međuprocesorske komunikacije i eventualnog nastanka nebalansiranog stanja, te je upravo zato dobro da ovaj parametar bude što manji. *TotalV* i *MaxV* su parametri koji direktno utiču na vreme preraspodele, vreme od nastanka nebalansiranog stanja do ponovnog uspostavljanja balansiranosti. U činjenici da algoritam *WF* daje mnogo bolje rezultate za ova dva parametra ogleda se njegova glavna prednost u odnosu na *LMSR* algoritam. Jedini parametar u kojem je *LMSR* pokazao apsolutnu dominaciju je *Vreme Algoritma*, za sve tri mreže bilo da se radi o podeli na četiri, šesti ili osam regiona.

Vreme preraspodele je vreme koje protekne od trenutka pojave nebalansiranog stanja do momenta kad se ponovo uspostavi balansirano stanje, tj. kad se prebace svi podaci koji menjaju svoj region.

Kako je testiranjem ustanovljeno da algoritam *WF* daje bolje vrednosti za sve parametre koji su direktno vezani za vreme preraspodele, za sve tri mreže, pri odlučivanju o preraspodeli korišćeni su rezultati ovog algoritma i dobijeni su rezultati prikazani u tebeli 2 (sve vrednosti su date u milisekundama).

Tabela 2. *Prosečno vreme preraspodele WF algoritmom za sve tri mreže*

Reg	Mreža 1	Mreža 2	Mreža 3
4	255.85	223.23	119.58
6	419.55	257.24	151.33
8	494.74	305.58	216.61

Povećanjem broja regiona povećava se i vreme preraspodele, trend koji se javlja za sve tri mreže, kao što se može videti u tabeli 2. Razlog ovome je što se sa povećanjem broja regiona povećava i količina podataka koja se prebacuje sa računara na računar (Tabela 2, parametar Tv), ali takođe značajan faktor jeste i povećanje broja mašina koje učestvuju u komunikaciji.

Napomena: Prilikom računanja vremena preraspodele nije uzeto u obzir vreme koje je potrebno da se završe trenutni proračuni i da računari budu spremni za razmenu, nego samo vreme od trenutka kada počne razmena do njenog završetka.

6. ZAKLJUČAK

U radu je opisana implementacija preraspodele već izdeljenog modela distributivne elektroenergetske mreže kao i implementacija samih algoritama za preraspodelu (*WF* i *LMSR*). Algoritmi su testirani na tri mreže, koje predstavljaju realne modele elektroenergetskih mreža gradova Beograda i Milana. Relizovan je distribuirani sistem u okviru kojeg se algoritmi izvršavaju centralizovano. Jedan računar u okviru sistema (koordinator) ima informacije gde se svi elementi trenutno nalaze, a ostali računari služe za obradu realnih podataka.

Na osnovu svih dobijenih rezultata i uporednih analiza predlaže se korišćenje *WF* algoritma. Uprkos nešto slabijem *Vremenu Algoritma*, koje predstavlja samo mali deo ukupnog vremena preraspodele, ovaj algoritam daje mnogo kvalitetnije rezultate kada su u pitanju svi ostali parametri *Greška*, *EdgeCut*, kao i *TotalV*, *MaxV* koji imaju najveći uticaj na vreme preraspodele, i samim tim najviše utiču na vreme međuprocesorske komunikacije i minimizaciju iste.

LITERATURA

[1] Kirk Schloegel, George Karypis, Vipin Kumar. „Wavefront Diffusion and LMSR: Algorithms for Dynamic Repartitioning of Adaptive Meshes“. IEEE Transactions on Parallel and Distributet Systems, Vol. 12, No. 5, May 2001.

[2] Kirk Schloegel, George Karypis, Vipin Kumar. „Multilevel diffusion schemes for repartitioning of adaptive meshes“. Journal of Parallel and Distributed Computing 47, 109-124(1997) Article No. PC971410

[3] Kirk Schloegel, George Karypis, Vipin Kumar, Rupak Biswas, Leonid Oliker. „A Performance Study of Diffusive vs. Remapped Load-Balancing Schemes“. University of Minnesota, Department of Computer Science & Engineering/Army HPC Center, Minneapolis, MN.

Kratka biografija:



Dejan Čotra rođen je 03.01.1985. u Sremskoj Mitrovici. Završio je gimnaziju „Stevan Puzić“, Ruma, 2004. god. Iste godine upisao je Fakultet Tehničkih Nauka u Novom Sadu, odsek elektrotehnika i računarstvo, smer računarstvo i automatika, usmerenje automatika i upravljanje sistemima.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

ANALIZA I PRIMENA VIŠEFAZNOG ALGORITMA ZA PODELU MODELA PODATAKA ZASNOVANOG NA GRAFOVIMA

ANALYSIS OF MULTILEVEL GRAPH PARTITIONING ALGORITHM

Suzana Kuburić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu je opisana implementacija podele grafa na particije sa ciljem da između particija bude što manja povezanost. Podela se vrši pomoću višefaznog algoritma za podelu grafa. Algoritam je testiran na grafovima koji predstavljaju model podataka realnih elektroenergetskih mreža i izvršeno je poređenje dobijenih rezultata.*

Abstract – *This paper describes an implementation of the graph partitioning where as small as possible number of connections between partitions is obtained. The partitioning is performed using Multilevel graph partitioning algorithm. The algorithm was tested on graphs which represent data models of real power networks and obtained results were compared.*

Ključne reči: *Podela grafa, Višefazna podela grafa, Kernighan-Lin (KL) algoritam*

1. UVOD

Podela grafa predstavlja važan problem koji ima široku primenu u mnogim oblastima kao što su računarske nauke i VLSI kola [1]. Grafom se mogu predstaviti sistemi kod kojih postoje međusobno spregnuti elementi koje je potrebno podeliti u grupe tako da u svakoj grupi budu elementi koji su povezani čvrstim vezama.

Za podelu grafa koriste se razni algoritmi kao što su PSO algoritam, genetski algoritmi, multilevel algoritmi i mnogi drugi. U ovom radu za problem podele primenjena je višefazna podela grafa. Pomoću ove podele, graf se deli u tri faze, gde se u prvoj fazi ukrupnjava, zatim se vrši inicijalna podela i nakon toga se usitnjava graf i poboljšavaju rezultati. Višefazna podela grafa se odlikuje dosta velikom brzinom rada u odnosu na slične algoritme.

2. TEORIJA GRAFA

Pri optimizaciji sistema pojavljuje se problem podele neke grupe elemenata na više manjih podgrupa, odnosno particija. Elementi koju su međusobno bolje povezani bi trebalo da se nalaze u istim particijama, u odnosu na ostale za koje nije bitno u kojoj particiji se nalaze. Objekti koje želimo da podelimo se nalaze u grafu i predstavljeni su čvorovima, a veze između njih su predstavljene granama.

Dva čvora mogu biti povezana sa više grana. U tom slučaju ove grane se prikazuju kao jedna veza, a broj grana koje ona predstavlja se definiše kao njena težina.

NAPOMENA:

Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Aleksandar Erdeljan, docent.

Isto tako više čvorova se može prikazati jednim čvorom čija težina označava broj čvorova koji se nalaze u tom čvoru.

2.1. Podela grafa

Problem podele grafa možemo definisati na sledeći način: Zadat je graf $G = (V,E)$, gde V predstavlja sve čvorove grafa G. Potrebno je V podeliti na particije $V_1, V_2 ... , V_k$ tako da je $V_i \cap V_j = 0$ za $i \neq j$, a $\bigcup_i V_i = V$. Broj grane E čiji čvorovi pripadaju različitim particijama treba takođe svesti na minimalan. [2]

Problem podele grafa se može proširiti na grafove čiji čvorovi imaju određene težine i povezani su granama između grafova. U ovom slučaju cilj je da se čvorovi podele u disjunktne podgrupe tako da zbir čvorova, odnosno težina čvorova u svakoj podgrupi bude isti, a suma težina grana između podgrupa što manja. U ovom radu predstavljena je višefazna podela grafa.

3. KRITERIJUM OPTIMALNOSTI

Kako bi se čvorovi optimalno rasporedili po particijama neophodno je podeliti ih tako da povezanost unutar particije bude najveća moguća, a da veličina particije ne prelazi maksimalnu dozvoljenu (MDV)

$$MDV = (1 + x) \cdot \frac{\sum_i VG_i}{n_{part}} \tag{1}$$

gde je:

VG_i - veličina grafa,

n_{part} - željeni broj particija,

x - tolerancija odstupanja od srednje dozvoljene veličine regiona, koja ima vrednost iz intervala:

$$x \in [0, \frac{(n_{part} - 1)}{n_{part}}] . \tag{2}$$

Kriterijum optimalnosti je:

$$F = \max \sum_p f_p \tag{3}$$

gde je:

$$f_p = \sum_{r,q \in R_p} CM_{r,q} \tag{4}$$

f_p - funkcija definisana za svaku particiju p (R_p), a

r i q – čvorvi raspoređeni u particiji p, gde je $p = 0,..n_{part}$,

$CM_{p,q}$ – kvadratna simetrična matrica povezanosti čvorova čiji elementi predstavljaju broj veza između čvorova p i q.

Particije se formiraju uz ograničenje:

$$PS_p = \sum_{r \in R_r} GS_r \leq MDV \quad (5)$$

gde je:
PS_p - veličina particije p u kome su koreni r veličine GS_r.

4. VIŠEFAZNI ALGORITAM PRERASPODELE GRAFA

Višefazna podela grafa se sastoji iz tri faze. U prvoj fazi se vrši ukrupnjavanje grafa, jer se time dobija manji graf pa i manji prostor dopustivih rešenja. Druga faza se zasniva na tome da se dobijeni ukrupnjeni graf deli na određeni broj particija koje imaju približno istu težinu čvorova. Nastoji se da težina grana između tih particija bude što manja. U trećoj fazi dobijene particije se usitnjavaju do početnog grafa sa tim da se pri svakom usitnjavanju vrši poboljšavanje odnosno smanjuje težina grana između particija.
U svakoj fazi mogu se primeniti različiti algoritmi.

4.1. Ukrupnjavanje grafa

Početni graf $G(V,E)$ koji se sastoji od čvorova V i grana E, prikazuje se nizom manjih grafova $G_1, G_2, \ldots . G_m$ za koje važi da $|V|>|V_1|>|V_2|>\ldots>|V_m|$ [2]. Dakle, ukrupnjavanje grafa G_m bi se moglo zapisati kao preslikavanje $g_m : V_m \rightarrow V_m+1$ tako da je $|V_m|=|V_m+1| = r,\ r > 1$.
Za ukrupnjavanje su korišćena četiri algoritma i to:

- nasumičnog povezivanja,
- povezivanje grana sa većom težinom,
- povezivanje grana sa manjom težinom i
- povezivanje velikih grupa.

Algoritam nasumičnog povezivanja odabira na slučajan način jedan čvor iz grafa. Ako ovaj čvor još nije uparen (povezan sa nekim čvorm), onda se takođe na slučajan način odabira jedan od njegovih susednih neuparenih čvorova, ako postoje, i on se uparuje sa tim susednim čvorom. Težine čvorova se sabiraju. Ako zbir težina prelazi maksimalno dozvoljenu težinu (MDV) čvorovi neće biti upareni. Ako izabrani čvor nema susednih čvorova on će ostati neuparen.
Drugi korišćeni algoritam za ukrupnjavanje je algoritam za *povezivanje grana sa većom težinom* (HEM - *Heavy Edge Matching*). Cilj višefazne podele grafa je da odredi particije koje će između sebe imati što manju težinu grana. Ako se upare čvorovi između kojih su grane sa većim težinama, ostaće grane sa manjim težinama. Ovako se smanjuje mogućnost da se dobije velika težina grana između particija.
HEM algoritam uzima na slučajan način jedan čvor, nalazi njegove susedne čvorove i povezuje ga sa susednim čvorom sa kojim se povezuje preko grane sa najvećom težinom u odnosu na sve ostale njemu incidentne grane. U slučaju nemogućnosti uparivanja sa tim čvorom, uparuje ga sa sledećim koji ima najveću tezinu grana.
Korišćen je i algoritam za *povezivanje grana sa manjom težinom* (LEM - *Light Edge Matching*) koji umesto da što više smanji ukupnu težinu grana, nastoji da dobije što veću težinu grana na kraju. To se postiže algoritmom koji je sličan kao HEM sa tim što se povezuju čvorovi sa najmanjim težinama grana između njih. Ovaj algoritam u kratkom vremenskom periodu određuje particije i uparuje veliki broj čvorova.
Algoritam za povezivanje velikih grupa (HCM - *Heavy Clique Matching*) je takođe korišćen za ukrupnjavanje grafa. Grupa predstavlja potpuno povezani podgraf u grafu *G=(V,E)*. Ako imamo podgrupu definisanu sa $G_u=(U,E_u)$ gde E_u predstavlja sve grane, a U sve čvorove iz grafa G_u, na osnovu odnosa U i E_u može se odrediti koliko je podgrupa G_u blizu grupi. Konkretno $2|E_u|/(|U|(|U|-1))$ teži jedan ako je U grupa. Ovaj odnos U i E_u naziva se gustina grana**.** [2]
HCM povezuje čvorove koji imaju najveću gustinu grana. Kao i kod prethodnih algoritama na slučajan način se izabere jedan neukrupnjeni čvor. Nakon toga se računa gustina grana za svaki njemu susedni čvor i povezuje se čvor sa najvećom gustinom grana.
Gustina grana se računa pomoću formule

$$\frac{2(ce(u)+ce(v)+e\omega(u,v))}{(v\omega(u)+v\omega(v))(v\omega(u)+v\omega(v)+1)}. \quad (6)$$

U ovoj formuli u predstavlja čvor koji je na početku slučajno izabran i koji želimo da spojimo, dok v predstavlja njegov susedni čvor. Ostale oznake su:
$e\omega(u,v)$ - suma težina grana između čvorova u i v,
$ce(u)$ - suma težina grana koje sadrže čvor u,
$ce(v)$ - suma težina grana koje sadrže čvor v,
$v\omega(u)$ - težina čvora u,
$v\omega(v)$ - težina čvora v.

4.2. Inicijalna podela grafa

U ovoj fazi neophodno je podeliti graf na određeni broj particija koje će imati što sličniji zbir težina čvorova i što manju težinu grana između njih. Algoritmi korišćeni za ovu podelu su sledeći: Kernighan-Lin (KL) algoritam [2], algoritam za podelu narastajućih grafova, greedy algoritam za podelu narastajućih grafova.
Prvi od korišćenih algoritama je *Kernighan-Lin* (KL) algoritam. Ukrupnjeni graf se slučajnim odabirom čvorova deli na dati broj particija gde svaka ima približno istu težinu. KL algoritam se zasniva na tome da se iz tako dobijenih particija uzima po jedan čvor iz dve particije i njihovom zamenom se smanjuje težina grana između patricija [2]. Da bi se postigli bolji rezultati koristi se Fiduccia-Mattheyses (FM) algoritam [3] koji predstavlja modifikovani KL algoritam. Razlika između ova dva algoritma je u tome što FM algoritam uzima po jedan čvor i njega prebacuje u drugu particiju, a ne vrši zamenu čvorova kao kod KL algoritma.
Parametri na osnovu kojih se odabira koji će čvor preći u drugu particiju su dobiti (*gain*) i balansiranosti. Dobit čvora predstavlja broj za koliko bi se smanjio zbir težina presečnih grana ukoliko bi se taj čvor prebacio iz particije u kojoj se nalazi u drugu particiju. On se može predstaviti jednačinom:

$$g_v = \sum_{(v,u)\in E \wedge P[v]\neq P[u]} \omega(v,u) - \sum_{(v,u)\in E \wedge P[v]=P[u]} \omega(v,u) \quad (7)$$

gde P predstavlja particiju grafa $G=(V,E)$, g_v dobit čvora v, a $\omega(u,v)$ težinu grane (u,v) [2].

Balans predstavlja uslov da težina svake particije zadovolji ograničenje.

Kod *algoritma za podelu narastajućih grafova* (GGP - *Graph growing partitioning algorithm*) se izabere na slučajan način jedan čvor i uzimaju se svi čvorovi oko njega dok se ne dostigne željena veličina particije i tako za sve particije. Ovaj algoritam je veoma osetljiv na izbor prvog čvora, jer sa različitim početnim čvorom dobijaju se različite težine grana između particija tako da od početnog čvora najviše zavisi kolika će biti težina grana između particija.

Greedy algoritam za podelu narastajućih grafova (GGGP - *Greedy graph growing partitioning algorithm*) je u velikoj meri sličan sa prethodnim GGP algoritmom. Međutim, za razliku od GGP algoritma, kada izabere jedan čvor on ne uzima sve čvorove koji su povezani sa njim, već računa dobit za te čvorove i čvor koji ima najveću dobit povezuje sa tim čvorom. Na ovaj način dobijanje težine grana između particija u mnogome manje zavisi od izbora početnog čvor.

4.3. Finija preraspodela grafa

Pošto se posle ukrupnjavanja grafa broj čvorova smanjio, finijom raspodelom se usitnjava graf pa se vraćaju čvorovi koji su tokom ukrupnjavanja nestali. Graf je i dalje podeljen na particije. Čvor koji je nestao prilikom ukrupnjavanja se pripaja particiji u kojoj se nalazi čvor sa kojim je bio uparen. Posle vraćanja svakog čvora proverava da li se zamenom čvorova može doći do boljeg rešenja. To se vrši pomoću KL algoritma. Ovim postupkom se može dosta poboljšati podela jer se pri usitnjavanju dobijaju čvorovi čijom se zamenom poboljšava rešenje.

5. REALIZACIJA PROGRAMSKOG REŠENJA ZADATKA

Implementacija softverskog rešenja realizovana je programskim jezikom C# uz oslonac na razvojno okruženje Microsoft Visual Studio 2008.

5.1. Učitavanje i formiranje grafa

Graf se formira učitavanjem dva fajla. Fajl „*nodes.txt*“ opisuje čvorove grafa, dok fajl „*edge.txt*“ opisuje grane grafa. U fajlu sa čvorovima svaki red predstavlja ID čvora i u tom redu se nalazi težina tog čvora. Kod grana takođe svaki red predstavlja ID grane, ali se u njemu nalaze tri broja koja su odvojena separatorom. Prvi broj predstavlja težinu grane, a druga dva ID čvorova koje ta grana povezuje. Pri učitavanju fajla kreiraju se objekti tipa *Node*, *Edge* i *Graph*.

5.2 Rezulati tesiranja

Testiranje algoritma je vršeno na računaru sa Intel Core2Duo procesorom na 2 GHz i 2 GB RAM memorije. Kao test model korišćen je graf sa 54 čvora i 44 grane.

Testovi su vršeni za različite kombinacije algoritama uzimajući iz svake grupe po jedan algoritam (12 različitih kombinacija).

Analizirani su rezultati prosečnih vrednosti funkcije cilja, vremena izvršavanja i relativno odstupanje od idealne veličine particije. Kako bi se dobile prosečne vrednosti, vršeno je po 50 testiranja. Graf je deljen na 2, 4, 6 i 8 particija.

Table 1. *Dobijene prosečne težine grana između particija*

		2	4	6	8
KL	RM	19	56	111	152
	HEM	30	73	111	112
	LEM	29	99	143	160
	HCM	16	60	80	117
GGP	RM	13	56	86	110
	HEM	12	51	86	105
	LEM	15	65	107	135
	HCM	12	39	61	76
GGGP	RM	17	32	58	73
	HEM	13	25	43	44
	LEM	16	44	58	74
	HCM	12	25	37	43

Na osnovu rezultata prikazanih u tabeli 1 vidi se da najbolje rezultate daje GGGP algoritam u kombinaciji sa HCM algoritmom pri svim podelama. Približno dobre rezultate daje i kombinacija GGGP i HEM algoritma. Takođe može se primetiti da generalno GGGP algoritam daje dosta bolje rezultate u odnosu na GGP i KL algoritme, kao i da HCM algoritam daje skoro uvek najbolje rezultate u poređenju sa RM, HEM i LEM algoritmima.

Table 2. *Prosečno vreme izvršavanja (u sekundama)*

		2	4	6	8
KL	RM	0.6	1.8	3.6	6.6
	HEM	0.6	2.5	5.4	9
	LEM	0.6	2.7	5	10.3
	HCM	0.6	2.2	3.6	6.5
GGP	RM	3.2	5.1	5.9	6.5
	HEM	4.4	5.2	7.4	9.1
	LEM	4.1	6.2	7.4	8
	HCM	3.5	5.1	6.1	7.2
GGGP	RM	1.5	1.6	1.5	1.4
	HEM	1.8	2.2	2	2
	LEM	1.8	2.1	1.9	1.9
	HCM	1.6	1.6	1.6	1.5

Pri podeli na dve particije GGGP ne prikazuje najbolje vreme izvršavanja, dok najbrže radi KL i to sa svim algoritmima. GGGP ipak ima i pri podeli na 4, 6 i 8 particija slično vreme izvršavanja, dok KL za svaku narednu veću podelu skoro udvostručuje vreme izvršavanja. Takođe HEM i LEM algoritmi su uvek malo sporiji u odnosu na RM i HCM koji su približno iste brzine. GGP je dosta sporiji od KL i GGGP i takođe mu se sa porastom broja particija udvostručuje vreme izvršavanja.

Najmanja odstupanja od idelane particije daje KL algoritam i to u većini slučajeva u kombinaciji sa LEM algoritmom. Sa RM, HEM, LEM i HCM se dobijaju slična odstupanja. GGGP je za nijansu lošiji od KL.

Table 3. *Dobijena prosečna odstupanja od idealne veličine particije*

		2	4	6	8
KL	RM	2.9	8.1	17.1	28.5
	HEM	2	8.9	16.2	28.9
	LEM	2.7	7.5	16	28
	HCM	2.1	8.8	19.4	34.6
GGP	RM	3.4	12.1	22	35.4
	HEM	4.9	15.1	25.1	34.1
	LEM	4	12.5	22.9	36.9
	HCM	3.5	14	21.8	29.4
GGGP	RM	3.1	11.8	23.5	35.8
	HEM	3.7	14.5	28.2	39.2
	LEM	3.4	11.9	24.4	38.5
	HCM	2.4	13.8	27.1	38.6

6. ZAKLJUČAK

U radu je opisana podela grafa pomoću višefaznog algoritma za podelu grafa. Analizirani su rezultati dobijeni testiranjem na test grafu.

Nakon analize rezultata utvrđeno je da najbolje rezultate daje kombinacija HCM i GGGP algoritma. Najveća razlika pri dobijenim rezultatima se vidi kod podele na 8 particija, gde ovi algoritmi daju dosta bolje rezultate. Kod ove kombinacije algoritama vreme izvršavanja je konstantno za sve podele, a balansiranost je slična kao i kod ostalih kombinacija algoritama.

7. LITERATURA

[1] Henderickson Bruce, Leland Robert „A multilevel algorithm for partitioning graphs“, Proceedings of ACM/IEEE conference on Supercomputing, San Diego, 1995.

[2] Karypsis George, Kumar Vipin „A fast and high quality multilevel scheme for partitioning irregular graphs“ SIAM Journal of Sciantic Computing, vol. 20, no.1, 1998.

[3] C.M.Fiduccia i R. M. Mattheyses, „A linear-time heuristic for improving network partitions“, „Design automation conference, 19th ACM/IEEE“, str. 175-181, 1982.

[4] J.A.Bondy, U.S.R. Murty S.N.Deepa, „Graph theory“, „Graduate texts in matematics“, Vol. 244., Springer, 2008.

Kratka boigrafija



Suzana Kuburić je rođena 07.09.1986. u Zrenjaninu. Završila je opšti smer „Zrenjaninske gimnazije“ 2005. godine i iste godine upisala Fakultet Tehničkih Nauka u Novom Sadu, odsek elektrotehnika i računarstvo, smer računarstvo i automatika, usmerenje automatika i upravljanje sistemima.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.39

SSH PROTOKOL

SSH PROTOCOL

Jovana Lugonja, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu su opisane osnovne karakteristike SSH protokola, njegova arhitektura i područje primene, zahtevi koji se postavljaju po pitanju korišćenja različitih verzija i mogućnosti koje se pružaju korisnicima. Akcenat rada je na programskoj implementaciji SSH protokola u C# programskom jeziku korišćenjem VS 2008 razvojnog okruženja, sa ciljem obezbeđenja pouzdanog prenosa podataka u okviru korisničke aplikacije. U praktičnom delu rada predstavljen je način realizovanja SSH klijenta uz pomoć SecureBlackbox biblioteke klasa.*

Abstract – *This work describes the basic characteristics of Secure Shell (SSH), its architecture and areas of application, the requirements that consider different versions and features offered to the users. The emphasis is on the program implementation of the SSH protocol in C# programming language using VS 2008 development environment, in order to provide reliable data delivery within custom applications. Implementing the SSH client by using SecureBlackbox class library is presented in the practical part of the paper .*

Ključne reči: *SSH-1, SSH-2, arhitektura protokola, Zlib, Diffie-Hellman, SecureBlackbox.*

1. UVOD

SSH je mrežni protokol namenjen za sigurno povezivanje na udaljeni računar preko nepouzdanog protokola kakav je Internet. Reč je o protokolu koji pripada aplikacijskom sloju TCP/IP modela, ali može funkcionisati i preko drugih Transportnih protokola. On osigurava tajnost svih prenesenih podataka tako što kriptuje sav saobraćaj između klijenta i servera i osigurava integritet prenesenih podataka računanjem MAC (*message authentication code*) kodova za svaki paket.

U postupku spajanja, SSH omogućava klijentu da autentifikuje servera, tj. da utvrdi da je server baš taj za kog se predstavlja, i tako se osigura od mogućeg čovek-u-sredini (*man-in-the-middle*) napada. Za razliku od *telnet* protokola ovde se lozinke prenose sigurnim kriptovanim kanalom, i tako onemogućavaju prisluškivanje (*eavesdropping*). Nakon uspostavljanja veze međusobne autentifikacije, sloj konekcije SSH protokola omogućava razne usluge.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bila dr Dragana Bajić, red.prof.

2. RAZVOJ SSH

SSH protokol je u svojoj verziji SSH-1 nastao 1995. godine, a razvio ga je Finac Tatu Ylönen. Motivacija za izradu tog protokola bio je slučaj napada na računarsku mrežu njegovog univerziteta u kojem su napadači saznavali lozinke korisnika prisluškivanjem saobraćaja na mreži. Prva verzija (SSH1), predložena kao IETF Internet Draft, imala je puno sigurnosnih propusta. Zbog toga SSH Communications Security 1996. godine predstavlja novu verziju protokola koja je nekompatibilna sa starom. Međutim, zbog toga što je program bio komercijalan, tj. nije bio besplatan kao prva verzija, on nije bio dobro prihvaćen i SSH-1 verzija protokola je još nekoliko godina ostala popularnija i šire korišćena od SSH-2 verzije. To se promenilo tek pojavom OpenSSH implementacije koja je razvijena pod OpenBSD licencom što znači da je potpuno besplatna i *open source*.

Standard je definisan u nizu RFC dokumenata.

- RFC4250 - The Secure Shell (SSH) Protocol,
- RFC4251 - The Secure Shell (SSH) Protocol Architecture,
- RFC4252 - The Secure Shell (SSH) Authentication Protocol,
- RFC4253 - The Secure Shell (SSH) Transport Layer Protocol,
- RFC4254 - The Secure Shell (SSH) Connection Protocol i
- RFC4335 - The Secure Shell (SSH) Session Channel.

2.1. Kompatibilnost verzija

Danas postoje programi koji i dalje koriste staru, SSH-1, verziju protokola, zbog čega je potrebno obezbediti kompatibilnost. Ako imamo slučaj da klijent koristi stari protokol (SSH-1), a server koristi novi (SSH-2), tada bi server trebalo da prijavi svoju verziju protokola kao 1.99 (tj. identifikacijski niz počinje sa „SSH-1.99“). U tom slučaju novi klijenti treba to da prepoznaju kao ekvivalent protokola 2.0 i da započnu razmenu ključeva prema SSH-2 verziji protokola. Stare verzije klijenta niz „1.99“ protumače kao verziju SSH-1 protokol i počinju razmenu koristeći stariji protokol. Naravno, server treba prijaviti verziju kao 1.99 samo ako podržava i staru verziju protokola i ako je konfigurisan tako da prihvata i SSH-1 konekcije. Ako imamo slučaj da klijent koristi novi protokol, a server koristi stari protokol, tada je moguće da klijent odmah nakon slanja svog identifikacijskog niza (a pre primanja serverovog) pošalje i podatke za razmenu ključeva. S obzirom da server koristi stariju verziju

protokola, neće razumeti podatke i prekinuće konekciju. Rešenje je da klijent primi serverov identifikacijski niz u kojem server javlja da koristi verziju SSH-1. Klijent će u tom slučaju nakon prekidanja veze ponovo uspostaviti vezu (ali sada se prijavljujući kao klijent koji koristi staru verziju) i dalje postupati po starom protokolu.

Izgled inicijalnog uspostavljanja SSH veze u praksi dat je na primeru preuzetom iz log datoteke programa Putty. Dakle, na klijentskoj strani je program Putty, dok je na serverskoj strani OpenSSH implementacija (Slika 1):

```
Event Log: Writing new session log (SSH raw data mode) to file: putty.log
Event Log: Looking up host "192.168.0.5"
Event Log: Connecting to 192.168.0.5 port 22
Event Log: Server version: SSH-1.99-OpenSSH_3.9p1
Event Log: We claim version: SSH-2.0-PuTTY_Release_0.60
Event Log: Using SSH protocol version 2
```

Slika 1. Uspostavljanje SSH komunikacije

3. Arhitektura SSH komunikacije

SSH se temelji na modelu klijent/server. Server s jedne strane osluškuje zahteve na unapred zadatom mrežnom portu, a klijent ih po potrebi šalje serveru. SSH server osluškuje zahteve klijenata na TCP portu 22.
Sastoji se od tri glavne komponente/sloja:

Transportni protokol (*The Transport Layer Protocol*), pruža autentifikaciju servera, zaštitu prenosa, i integritet prenesenih podataka. Takođe, opcionalno pruža i uslugu kompresije podataka. Obično radi kao aplikacijski protokol iznad TCP/IP protokolnog steka, ali ne zavisi od njega, već može raditi iznad svakog protokola koji pruža pouzdan prenos podataka.

Autentifikacijski protokol (*The User Authentication Protocol*) zadužen je za autentifikaciju korisnika serveru na koji se on pokušava spojiti, kojem pokušava pristupiti. Za njegov rad je potrebno da Transportni sloj protokola već bude u funkciji.

Protokol konekcije (*The Connection Protocol*) je zadužen za multipleksiranje više logičkih kanala u jedan kriptovani tunel. Takođe upravlja zahtevima korisnika za uslugama kao što su: zahtev za pokretanjem pseudoterminala (*pty*) i ljuske operativnog sistema (*shell*), izvršavanje naredbe na serveru, pokretanje podsastava (*scp, sftp*) ili pak prosleđivanje mrežnih portova (*port forwarding*). Radi iznad transportnog i autentifikacijskog sloja.

Nakon uspostave sigurnog transportnog kanala klijent obično pošalje zahtev za pokretanjem autentifikacijskog servisa (*ssh-userauth*). Takođe, nakon uspešne autentifikacije klijent će poslati zahtev za pokretanjem protokola konekcije (*ssh-connection*). To omogućava da novi protokoli budu definisani i da koegzistiraju sa postojećim protokolima.



Slika 2. Arhitektura SSH protokola

3.1 Algoritmi kompresije

Ako je u postupku pregovaranja algoritama dogovoren i algoritam kompresije, tada će polje sa korisnim podacima (samo to polje) biti komprimovano.
Kompresija je nezavisna za svaki smer, tj. možemo koristiti jedan algoritam za smer klijent-server, a drugi algoritam za kompresiju u smeru server-klijent. Međutim, preporučeno ja da metoda kompresije bude ista u oba smera.

Ime algoritma	Zahtev	Opis algoritma
None	obavezno	Bez kompresije
Zlib	opcionalno	ZLIB (LZ77) kompresija

Tabela 1. Prikaz definisanih algoritma kompresije

U gornjoj tabeli navedeni su algoritmi kompresije koje SSH protokol definiše.

Prva je 'none' metoda. Ona označava da nema kompresije, i prema SSH protokolu ta je metoda obavezna u svim SSH implementacijama. Druga metoda, označena kao „zlib", jeste zlib metoda kompresije opisana u dokumentima RFC1950 (*ZLIB Compressed Data Format Specification version 3.3*) i RFC1951 (*DEFLATE Compressed Data Format Specification version 1.3*).

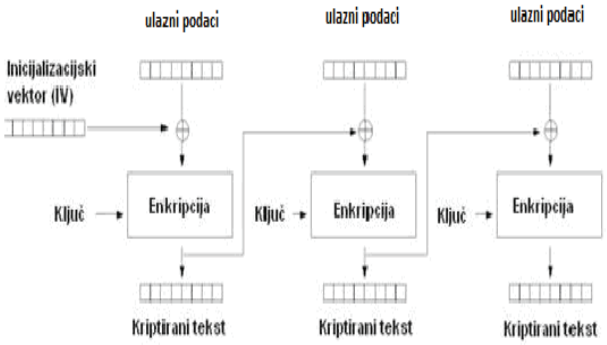
3.2 Algoritmi enkripcije

Algoritam za enkripciju se bira u postupku pregovaranja algoritama, dok se ključ enkripcije dobija Diffie-Hellman razmenom ključeva. Nakon što je završena razmena ključeva, svaki paket se kriptuje. Enkripcija se vrši nezavisno u oba smera, tj. postoji poseban ključ i inicijalni vektor za smer enkripcije klijent-server, dok se za smer server-klijent koristi drugi ključ i inicijalni vektor.

Osim toga, moguće je čak da se koristi jedan algoritam enkripcije za jedan smer, a drugi algoritam za drugi smer, međutim to nije preporučljivo. Dužina ključeva enkripcije je minimalno 128 bita.

Svi algoritmi rade u CBC (*cipher block chaining*) modu (osim arcfour algoritma), što znači da se pre kriptovanja vrši XOR operacija između ulaznog teksta i prethodnog kriptovanog bloka (ili inicijalnog vektora, ako se radi o prvom paketu).

Princip rada u CBC modu prikazan je na slici 3.



Slika 3. CBC način enkripcije

3.3 Integritet podataka

Integritet podataka zaštićen je tako što se uz svaki paket šalje i MAC (*message authentication code*) koji se računa iz tajnog ključa, rednog broja paketa i sadržaja samog paketa.

3.4 Algoritmi javnog ključa

SSH protokol je dizajniran tako da može raditi s sa bilo kojim algoritmom javnog ključa. Ti algoritmi se takođe dogovaraju nakon inicijalizovanja konekcije. Ipak, daleko su najviše korišćeni DSS (*Digital Signature Standard*) i RSA algoritmi. Format zapisa njihovih ključeva za primenu u SSH protokolu označen kao "ssh-dss" i "ssh-rsa". DSS algoritam obavezno moraju podržavati sve implementacije dok je RSA opcionalan. Oni se koriste za generisanje digitalnog potpisa, što omogućava autentifikaciju servera.

3.5 Razmena ključeva

Nakon što su utvrđeni svi potrebni algoritmi koji će se koristiti, kreće postupak razmene ključeva i autentifikacije servera.
Sledi opis kompletnog postupka korak po korak.

V_C označava klijentov identifikacijski niz, V_S je serverov identifikacijski niz, p i g su unapred poznati parametri Diffie-Hellman razmene, p je veliki prosti broj, a g je generator za grupu GF(p), K_S je serverov javni ključ, I_C je klijentova SSH_MSG_KEXINIT poruka, a I_S je serverova SSH_MSG_KEXINIT poruka.

- klijent generiše veliki slučajni broj x i računa broj e, e = g^x mod p. Klijent šalje broj e serveru.
- Server generiše slučajni broj y i računa broj f, f = g^y mod p. Server prima broj e. Tada računa tajni ključ K, K = e^y mod p. takođe računa H, H = hash(V_C || V_S || I_C || I_S || K_S || e || f || K). Server potpisuje H svojim privatnim ključem i dobija potpis s. Server šalje (K_S || f || s) klijentu.
- Klijent proverava da javni ključ K_S stvarno pripada serveru (bilo pomoću certifikata ili pomoću lokalne baze ključeva). Klijent može prihvatiti serverov ključ i bez provere, ali to čini protokol ranjivim na man-in-the-middle napade. Klijent takođe računa tajni ključ K, K = f^x mod p, računa H = hash(V_C || V_S || I_C || I_S || K_S || e || f || K) i proverava da li je s ispravan potpis od H.

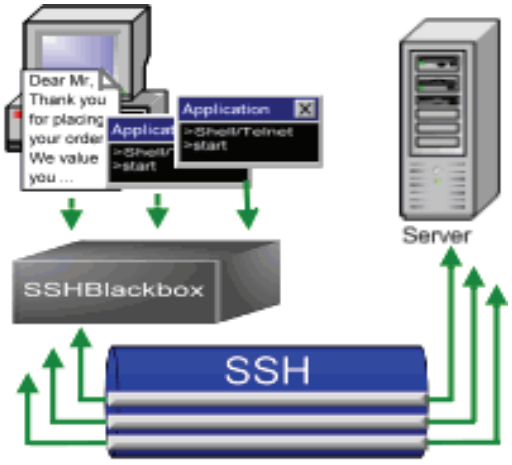
Kao rezultat razmene dobijamo K, što je tajni ključ poznat samo klijentu i serveru, a ne i potencijalnim napadačima koji su prisluškivali saobraćaj.

4. IMPLEMENTACIJA SSH PROTOKOLA

Praktična implementacija SSH protokola predstavlja deo projekta namenjenog administriranju mreže kablovskog operatora. Ideja je bila da se povezivanje, kao i izvršavanje komandi između klijentskih računara tehničke podrške i udaljenih servera obavi upravo pomoću mehanizama koji garantuju poverljivost i integritet podataka i verodostojnost subjekata komunikacije.

4.1 Programska implemetacija SSH sigurnosti

Program je implementiran u programskom jeziku C#, u Microsoft Visual Studio 2008 razvojnom okruženju. Za realizaciju SSH protokola iskorišćena je biblioteka klasa SSHBlackbox, idejno rešenje kompanije ELDOS. SSHBlackbox omogućava povezivanje sa udaljenim serverima preko SSH protokola ili kreiranje sopstvenih SSH servera (serverske komponente su dostupne u VCL i NET izdanju). Pruža niz korisnih klasa koje podržavaju pristup udaljenim računarima, udaljeno izvršavanje komandi, različite šeme autentifikacije što dozvoljava klijentu da se autentifikuje pomoću nekoliko šema u isto vreme, preusmeravanje saobraćaja kao i druge SSH funkcionalnosti. Na slici koja sledi dat je grafički prikaz komunikacije uz pomoć SSHBlackbox rešenja.



Slika 4 : Grafički prikaz SSHBlackbox rešenja

Povezivanje SSHBlackbox biblioteke sa razvojnim okruženjem je jednostavno i odvija se automatski pokretanjem instalacionog programa.

Za upotrebu određene klase projektu je potrebno dodati referencu. Može se dodati iskaz *Using* kodu, koji stavke prostora imena datih klasa dovodi u područje važenja. Aktiviranje SecureBlackbox komponenti zahteva licencni ključ, bez koga se one ne mogu koristiti. SSH komponente imaju unapred definisane vrednosti za uspostavljanje veze.

Međutim, u određenim slučajevima može biti neophodno da se prilagode određene opcije konekcije.

SSHBlackbok uključuje sledeće komponente i funkcije vezane za SSH protokol čiji je tabelarni prikaz dat u nastavku.

Imena komponenti	*Kratak opis*
ElSSHClient	Obezbeđuje klijentsku stranu funkcionalnosti za SSH veze.
ElSimpleSSHClient	Pojednostavljena verzija ElSSHClient.
ElSSHServer	Obezbeđuje serversku stranu funkcionalnosti za SSH veze.
ElSSHPublicKeyClient	Obezbeđuje klijentsku stranu funkcionalnost za daljinsko upravljanje SSH ključevima na serveru..
ElSSHPublicKeyServer	Implementira SSH javni ključ-podsistem servera (koristi se za daljinsko upravljanje ključevima).

Tabela 2. Prikaz osnovnih klasa SecureBlackbox biblioteke

Za programsko rešenje komunikacije između računara tehničke podrške sa jedne, i menadžment dela mreže sa druge strane uzeta je klasa ElSSHClient, budući da je bilo potrebno implementirati samo SSH klijenta.
ElSSHClient se koristi za dodavanje podrške SSH klijent strani TCP konekcije (strani koja inicira vezu). Da bi se veza započela potrebno je pozvati metodu Open i na taj način kreirati instancu klijenta. Poziv ove metode inicira SSH rukovanje (handshake) i treba da se izvrši kada je Transportni sloj spreman, odnosno kada je uspostavljena TCP konekcija.
Nakon pozivanja ove metode SSH klijent će pokušati da uspostavi bezbednu vezu. Konekciju čini sekvenca OnSend i OnReceive događaja. Kada su kriptovani podaci spremni za slanje, ElSSHClient će inicirati OnSend događaj. Kada ElSSHClient ima potrebu za prijemom podataka inicira se OnReceive događaj. Ako se koristi rad u neblokiranom režimu, može se iskoristiti metoda DataAvailable koja bi klijentu ukazivala da postoje neki podaci spremni za čitnaje. ElSSHClient podržava rad sa SSH1 i SSH2 verzijom i dozvoljava upotrebu svih algoritama enkripcije, razmenu ključeva i MAC algoritma.
Cilj implementacije programa bio je da se u WINDOWS okruženju automatizuje upravljanje mrežnim resursima na jedan centralizovan i siguran način. Ovim programskim paketom omogućen je lakši, brži i jednostavniji rad uz poseban akcenat na zaštiti jednog takvog sistema.

5. ZAKLJUČAK

Bezbednost računarske mreže podrazumeva visoku zaštitu mrežne opreme i komunikacije na svim nivoima implementacijom bezbednosnih protokola i mehanizama. Budući da je mnogo jednostavnije uvesti sigurnost na višim nego na nižim slojevima arhitekture, SSH će se i dalje nametati kao privlačno i brzo rešenje. Treba imati na umu da SSH pruža određen nivo zaštite prilikom korišćenja Interneta, no ne i potpunu zaštitu. SSH ne proverava podatke koji se šalju i pretpostavlja pouzdanost krajnjih tačaka komunikacije. Ukoliko se ne koristi primerena zaštita antivirusnih programa, te krajnje tačke mogu biti kompromitovane.
Osim programske sigurnosne podrške, izuzetno je važno naglasiti i korisničko ponašanje jer je upravo neodgovornost korisnika najčešći uzrok narušavanja bezbednosti. Osim primene programske sigurnosne zaštite potrebno je i dobro zaštititi korisnička imena, lozinke, osetljive podatke na računaru, obazirati se na automatska sigurnosna upozorenja, ne preuzimati datoteke sa nepouzdanih izvora i slično.
Dakle, nužno je da korisnici budu dovoljno upućeni u rad protokola i posebno oprezni. Uprkos nabrojenim manama, zaključak je da SSH protokol praktičan i koristan za upotrebu, zavisno o čovekovim potrebama, te da puno toga može pružiti, a još više ako se znaju ceniti njegove prednosti.

6. LITERATURA

[1] Ylönen, T., C. Lonvick, Ed., "*The Secure Shell (SSH) Protocol Architecture*", RFC 4251, 2006.
[2] Ylönen, T., C. Lonvick, Ed., "*The Secure Shell (SSH) Transport Layer Protocol*", RFC 4253, 2006.
[3] Ylönen, T., C. Lonvick, Ed., "*The Secure Shell (SSH) Authentication Protocol*", RFC 4252, 2006.
[4] Ylönen, T. and C. Lonvick, Ed., "*The Secure Shell (SSH) Connection Protocol*", RFC 4254, 2006.
[5] Lehtinen, S., C. Lonvick, Ed., "*The Secure Shell (SSH) Protocol Assigned Numbers*", RFC 4250, 2006.
[6] Cusack, F., Forssen M., "*Generic Message Exchange Authentication for the Secure Shell Protocol (SSH)*", RFC 4256,2006.
[7] Friedl M., Provos N., Simpson W. "*Diffie-Hellman Group Exchangefor the Secure Shell (SSH) Transport Layer Protocol*", RFC4419,2006.
[8] Bellare, M., Kohno T., Namprempre C. "*The Secure Shell (SSH)*
Transport Layer Encryption Modes", RFC4344, 2006.
[9] Song, X.D., Wagner, D., and X. Tian, "*Timing Analysis of Keystrokes and SSH Timing Attacks*" , Paper given at 10th USENIX Security Symposium, 2001.

Kratka biografija:



Jovana Lugonja rođena je u Bugojnu (BIH) 1986. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Komunikacione tehnologije i obrada signala odbranila je 2011.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

УДК: 621.317

PET/CT У НУКЛЕАРНОЈ МЕДИЦИНИ

PET/CT IN NUCLEAR MEDICINE

Петар Зобеница, *Факултет техничких наука, Нови Сад*

Област –ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *У раду су приказане основе PET/CT уређаја, његова примена у медицини, заштита од зрачења као и специфичност рада са пацијентима.*

Abstract – *This paper presents the basics of PET/CT devices, its application in medicine, radiation protection and the specificity of work with patients.*

Кључне речи: *Позитронски емитери, PET/CT, квалитет слике, заштита од зрачења.*

УВОД

Спој позитронске емисионе томографије (PET) и компјутерске томографије (CT) назива се PET/CT уређај. То један хибридни уређај који обједињује PET и CT уређаје у једну целину.

PET/CT је неинвазивна дијагностичка процедура и представља један од највећих напредака у савременој имиџинг дијагностици (такозвани молекуларни имиџинг) у откривању малигних обољења.

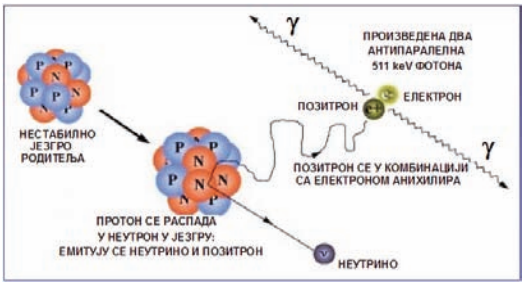


Слика 1. *Изглед PET/CT уређаја*

1.ФИЗИКА PET/CT

1.1. Позитронски емитери

Позитронски емитери су неутронско дефицијентни изотопи који постижу стабилност преласком протона (p^+) у неутрон (n^0). Тада се емитује позитрон (e^+), који се анихилира са електроном (e^-). Нискоенергетски електрон и позитрон анихилацијом могу да пређу само у два фотона (гама зрака) од 551 keV који су антипаралелни у оквиру позитрона. Два анихилацио на гама зрака се емитују у супротним смеровима и коинцидентно детектују помоћу детектора PET скенера.



Слика 2. *Изглед анихилације*

1.2. Радиофармацеутици

У производњи радиофармацеутика најчешће се примењују радиоактивни изотопи елемената који су највише заступљени у организму, као што су угљеник (^{11}C), азот (^{13}N) и кисеоник (^{15}O), док се уместо атома водоника (H), захваљујући истој величини користи флуор (^{18}F). Предност ^{18}F у односу на остале радиоактивне изотопе је дуже време полураспада, што омогућава дистрибуцију и ван места производње, као и дуготрајнију синтезу сложенијих радиофармацеутика. Остали наведени радионуклиди и од њих направљени радиофармацеутици, због веома кратког времена полураспада, морају да се производе у близини PET/CT уређаја и непосредно пре употребе. Радиоактивни изотопи (радионуклиди), производе се у циклотрону. Због релативно кратког времена полураспада, велики број инсталација садржи циклотрон у непосредној близини, како би се радиофармацеутик за што краће време администрирао пацијенту.



Слика 3. *Изглед модерног акцелератора*

Најчешће коришћени трејсер у онкологији је ^{18}F (флуоро-деокси-глукоза, тј. FDG) који прати сличан

НАПОМЕНА:
Овај рад проистекао је из дипломског-мастер рада чији ментор је др Весна Спасић-Јокић, ред. проф.

метаболички пут до глукозе *ин виво*, осим што се не метаболише са CO_2 и водом, него остаје заробљен унутар ткива. То га чини врло погодним за употребу као трејсер глукозе која се упија. То је од интереса у онкологији јер ћелије рака имају више од просечне стопе метаболизма глукозе, стога се лако примете на снимцима.

Изотоп	Време полураспада (минута)	Максимална позитронска енергија (MeV)	Позитронски домет у води (FWHM у mm)	Начин производње
^{11}C	20,3	0,96	1,1	циклотрон
^{13}N	9,97	1,19	1,4	циклотрон
^{15}O	2,03	1,70	1,5	циклотрон
^{18}F	109,8	0,64	1,0	циклотрон
^{68}Ga	67,8	3,89	1,7	генератор
^{82}Rb	1,26	3,15	1,7	генератор

Слика 4. *Примери изотопа који се користе у PET*

2. ПРИМЕНА ПОЗИТРОНСКИХ ЕМИТЕРА У PET

2.1. Медицинске импликације

Позитронски емитери имају највећу примену у онкологији (85%), неурологији (10%) и кардиологији (5%). Једна од најважнијих примена PET-а у онкологији је детектовање карцинома: плућа, дојке, дебелог црева, бубрега, мозга, простате, тестиса, панкреаса, итд. у стадијумима где их остале имиџинг технике не могу детектовати.

У неурологији PET се примењује за дијагностиковање дегеративних можданих обољења (Алцхајмерова, Хантингтонова и Паркинсонова болест), као и за откривање епилепсије, поремећаје кретања, итд.

Примена PET-а у кардиологији је за дијагностиковање коронарне артеријске болести, проверу стања леве коморе, а такође помаже идентификовању пацијената који би имали користи од реваскуларизационих процедура као што су стентовање и бајпас операције.

2.2. Предности

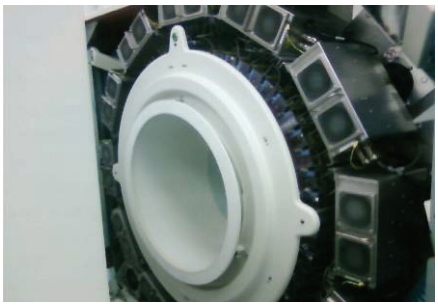
Предност PET-а је веома краткотрајна радиоактивност, а излагање радијацији је екстремно ниско. Количина супстанце је толико мала да не утиче на нормалне процесе у телу. Осетљивост уређаја за детекцију радиоактивног зрачења је врло велика, а квантитативна испитивања су прецизна, јер гама фотони имају високу енергију и не апсорбују се у ткивима организма.

2.3. Недостаци

Недостатак је употреба јонизујућег зрачења, а поред тога неопходно је водити бригу о радиоактивном отпаду који се ствара након третмана PET-ом. Због тога што се неки од радиофармацеутика распадају брзо и што делују веома кратко, морају се припремити у лабораторији близу PET скенера, а уз то је потребно да се циклотрон налази у непосредној близини.

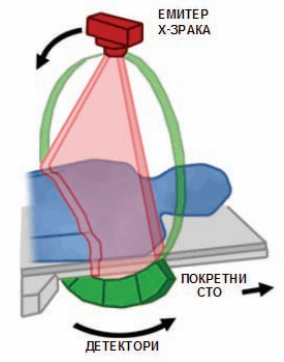
3. PET/CT УРЕЂАЈ

PET/CT уређај се састоји од два модула, а то су PET модул и CT модул. Први део PET/CT уређаја је CT модул, а други део је PET модул. Електрон и позитрон након анихилације, прелазе у два гама зрака. Оба зрака se детектују брзом коинциденцијом (5 ns) помоћу детектора који се налазе у PET модулу и дају линију одзива (Line Of Response - LOR). Након детектовања, гама зраци се претварају у електричне сигнале које веома брзо региструје електроника уређаја. Касније се ти сигнали обрађују у компјутеру и формира се коначна слика.



Слика 5. *Изглед детекторског прстена*

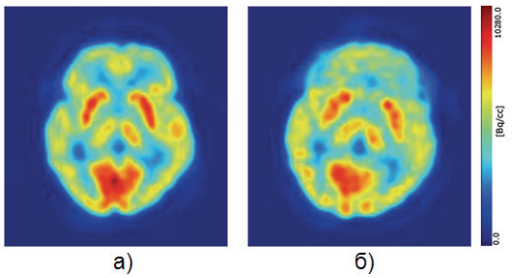
Компјутерска томографија (CT) мери атенуацију (апсорпцију и расејање) фотона X-зрака кроз пацијента, слично рентген уређају, и пружа морфолошке и анатомске податке о стању органа. У CT систему, X-зрак је генерисан у емитеру X-зрака који се ротира око пацијента пружајући информације о атенуацији фотона кроз пацијента из различитих углова.



Слика 6. *Принцип рада CT-а*

4. ФОРМИРАЊЕ И КВАЛИТЕТ СЛИКЕ

Након детекције гама зрака и њиховог претварања у електричне сигнале, следи формирање коначне слике у зависности од модула којим је вршено снимање (PET, CT или PET/CT). Веома је важно да слика буде добро реконструисана и да буде задовољавајућег квалитета, јер у неким случајевима пацијентов живот може бити доведен у питање. Као један од проблема приликом формирања слике је померање пацијента приликом снимања, што може изазвати нежељене промене слике.

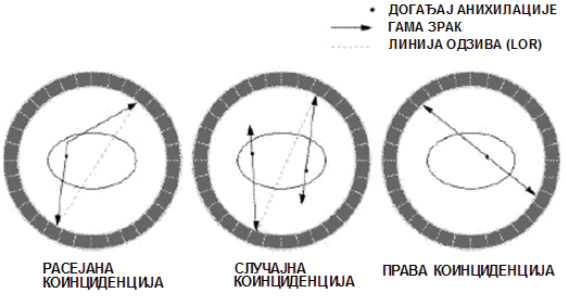


Слика 7. *Снимак без покрета (а)и са покретом (б)*

Покрети пацијента ограничавају остварив просторну резолуцију у PET-у и могу да буду унутар фрејма и између фрејмова. Покрети пацијента имају потенцијално велики утицај на квалитет слике и на

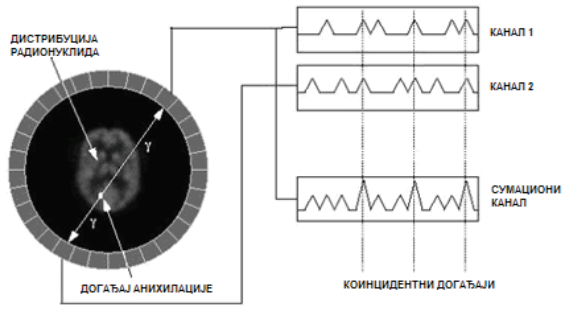
тачност анализе кинетике трејсера. Квантитативна компензација покрета је могућа, може бити интегрисана у клиничку рутину и помаже смањивању или чак елиминисању ефеката повезаних са покретима пацијента.

Постоје четири категорије коинцидентних догађаја: права коинциденција, расејана коинциденција, случајна коинциденција и вишеструка коинциденција.



Слика 8. *Типови коинцидентних догађаја*

Коинциденција повећава осетљивост у смислу побољшања резолуције (5-10) mm, у односу на SPECT (15-20) mm. Софтвер за реконструкцију слике користи као улазне податке коинцидентне догађаје, мерене по свим угаоним и линеарним позицијама, да реконструише слику која приказује локализацију и концентрацију радиоизотопа унутар равни органа који је сниман.



Слика 9. *Коинцидентна детекција код PET*

Гама фотони су детектовани коинциденцијом у детекторском прстену тако да два детектована гама фотона представљају праву линију дуж које се догађај десио. Скуп таквих линија је прикупљен и обрађен уз помоћ софтверског алата за процесирање слике да би се створила слика активности, а тиме и функционалности.

Један од постојећих програма за управљање целом апликацијом је Singo. То је Сименсов програм који управља целокупним процесом (рад уређаја, снимање, аквизиција, формирање и фузија слике, приказ резултата).

Стандардна вредност везивања радиофармацеутика (SUV, *Standard Uptake Value*) се рачуна по формули:

$$SUV = \frac{\text{активност у ROI (MBq)/запремина (ml)}}{\text{инјектована активност (MBq)/маса пацијента (g)}}$$

Што је већи SUV, већи је и ризик од болести. SUV се прати током терапије, код зона са везивањем SUV>1, а SUV > 2,5 је индикатор канцера.

Ограничења SUV-а: полу-квантитативна мера метаболизма глукозе, дефинисање ROI, хетерогеност тумора (некроза, променљив састав), промена волумена тумора током времена, а мањи тумори се теже снимају.



Слика 10. *Вредности SUV*

Дефиниција мете може бити квалитативна (стручни клинички преглед) и квантитативна (вредност интезитета воксела).

Физика имиџинга – боље слике се постижу помоћу:

- Дизајна детектора
  - Просторна резолуција
  - Осетљивост
- Обраде слике:
  - Корекције за физичке ефекте
  - Алгоритми за реконструкцију слике
- Анализа података и биолошко моделовање – боља интерпретација слика

5. ЗАШТИТА ОД ЗРАЧЕЊА

Приликом снимања PET/CT скенером, пацијенти и особље се излажу јонизујућем зрачењу. Зрачење може да изазове генетске поремећаје, неплодност, катаракту, опекотине, канцер и смрт. Зато је потребно посветити посебну пажњу заштити особља и пацијената од јонизујућег зрачења и потребно је предузети све неопходне заштитне мере како би се минимализовала њихова изложеност зрачењу и спречили могући инциденти.

Пожељно је да особље има што мање физичког контакта са пацијентима како би се избегла могућа контаминација зрачењем. То се постиже специјалним пројектовањем зграде у којој се врше PET/CT снимања. Најефикаснија заштита од гама зрака је употреба материјала високе густине (бетон 2350 kg/m^3, олово 11340 kg/m^3) и велике дебљине. За заштиту од зрачења донете су одређене регулативе од стране Међународне агенције за атомску енергију (IAEA), такође, постоје и законске регулативе и у Србији које се односе на заштиту од зрачења.

6. СМЕШТАЈ ПАЦИЈЕНАТА-МАНИПУЛАЦИЈА

Због употребе јонизујућег зрачења, рад особља са пацијентима представља својеврстан изазов. Пацијенти којима се администрира радиофармацеутик ради прегледа PET-ом имају посебан третман и користе посебне просторије како не би контаминирали особље, или евентуално друге пацијенте. Постоји неколико просторија у којима бораве пацијенти током прегледа PET-ом и те просторије морају бити физички одвојене од осталих просторија у којима бораве пацијенти или особље. Такође, пацијенти имају и посебан тоалет који се приликом екскреције (након прегледа) контаминира, те је и он одвојен од осталих просторија.

7. МАНИПУЛАЦИЈА РАО

Радиоактивни отпад (РАО) је свака врста материјала која је контаминирана одређеном дозом

радиоактивности, стога се мора посветити велика пажња правилној манипулацији. Приликом и након третмана радиоактивни отпад (стаклене посуде, шприцеви, игле, рукавице, папир, ампуле са нискоактивним радионуклидима, краткоживећи радионуклиди) се привремено смешта у канту предвиђену за то, која је оклопљена дебелим оловним блоковима како би се минимизовала контаминација. Касније се тај отпад односи на другу локацију предвиђену за одлагање РАО у којој се држи одређено време.

8. ЛИТЕРАТУРА

[1] J. Langner, „Motion Compensation in Positron Emission Tomography: Performance of a Clinical Integration at the PET centre Dresden-Rossendorf“, Дрезден, 2009.

[2] http://www.fda.gov/Radiation-EmittingProducts/, (2011).

[3] http://www.wikipedia.com

[4] В. Спасић-Јокић, „POSITRON EMISSION TOMOGRAPHY (PET) IN MEDICAL IMAGING“, Београд, (2008).

[5] N. S. Rehfeld, „Monte Carlo Simulations in Positron Emission Tomography Reconstruction“, Берлин, 2007.

Кратка биографија:



Петар Зобеница рођен је у Инђији 1986. Средњу техничку школу у Инђији завршио је 2004. године. Од школске 2004/05 студент је Факултета техничких наука у Новом Саду.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004(043.3)

RAZVOJ SISTEMA ZA AUTOMATIZOVANO TESTIRANJE SOFTVERA ZA MULTIMEDIJALNU VIZUALIZACIJU

SYSTEM FOR AUTOMATED TESTING OF MULTIMEDIA VISUALISATION SOFTWARE

Ivan Perić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Zadatak ovog rada jeste realizacija sistema koji omogućuje automatizovano testiranje aplikacije za multimedijalnu vizualizaciju. Sistem je realizovan na primeru TDMS Web aplikacije uz pomoć JUnit i Selenium 2 alata. Analizirani su i drugi alati, ali je zaključeno da je ovaj set alata najbolji izbor, što je dokazano kroz realizaciju ovog sistema.*

Abstract – *Task of this study is developing a system for automated testing of multimedia visualisation software. System was implemented using TDMS Web application for testing, and JUnit and Selenium as tools. Other tools were also considered, but this selection was made after carefull consideration, and the quality of the choice was proven through implementation of this system.*

Ključne reči: *testiranje, automatizovano testiranje, Unit testiranje, JavaScript, Selenium, veb programiranje, multimedijalni softver.*

1. UVOD

Glavne prednosti automatizovanog softverskog testiranja su mogućnosti da ponovi prethodno zabeležene i definisane akcije, uporedi rezultate sa očekivanim ponašanjem i pruži izveštaje i obaveštenja zainteresovanim stranama o uspešnosti ili neuspešnosti izvršenja testova. Kad se testovi jednom napišu, moguće ih je upotrebiti neograničen broj puta i lako ih je proširiti da izvrše zadatke koje je mnogo teže ili nemoguće izvršiti ručnim testiranjem. Zbog ovoga se automatizovano testiranje smatra esencijalnom komponentom uspešnog razvoja softvera.

Kao što automatizovano testiranje preuzima primat od ručnog, tako i razvoj veb aplikacija teži ka tome da učini desktop aplikacije zastarelim. Naime, sa razvojem tehnologije dolazi i do značajnih unapređenja u oblasti komunikacione infrastrukture, a to čini udaljene resurse sve pristupačnijim korisnicima. Mreže ili oblaci servera visokih performansi omogućuju pristup mnogobrojnim servisima i podacima koji su pouzdani i sigurni zahvaljujući visokoj redundantnosti. Sa korisničke, tj klijentske, strane sve češće je dovoljno samo imati veb *browser* koji uz pomoć *HTML*, *CSS* i *JS* tehnologija omogućuje potpun pristup servisima.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Dragan Ivetić, red.prof.

Ovaj rad predstavlja predlog razvoja sistema za automatizovano testiranje softvera, a zbog prethodno spomenutih tendencija, kao primer je uzeto testiranje veb aplikacije za multimedijalnu vizualizaciju. Pomenuta aplikacija je veb klijent kompanije **Telvent DMS** koji koristi servise njihovog *DMS Softvera*.

2. TESTIRANJE SOFTVERA

Testiranje softvera je istraživanje koje se sprovodi u cilju pružanja informacija, kreatorima subjekta testiranja i ostalim zainteresovanim stranama, o kvalitetu proizvoda ili servisa koji se testira [1]. Ono takođe omogućava objektivni i nezavisni pogled na softver da bi poslovni aspekt razvoja mogao da shvati rizike i beneficije razvoja tog softvera. Tehnike testiranja uključuju, ali nisu ograničene na, proces pokretanja programa ili aplikacije sa namerom pronalaženja grešaka (bagova).

Ipak, testiranje ne može identifikovati sve greške u softveru. Umesto toga, ono daje poređenje između stanja i ponašanja proizvoda naspram načela, principa i mehanizama pomoću kojih neko može prepoznati problem. Ova načela mogu sadržati specifikacije, ugovore [2], slične proizvode, starije verzije istog proizvoda, korisnička očekivanja, relevantne standarde, primenjive zakone, i druge kriterijume.

3. UNIT TESTIRANJE

U programiranju, unit testiranje je metoda pomoću koje su individualne jedinice izvornog koda testirane da bi se videlo da li su spremne za upotrebu. Jedinica koda je najmanji deo aplikacije koji može biti testiran. U proceduralnom testiranju, ova jedinica može biti individualna funkcija ili procedura. U *OOP*, jedinica je obično metoda. Ovi testovi su kreirani od strane programera ili ponekad u procesu razvoja od strane testera koji koriste metodu bele kutije.

Idealno, svaki slučaj testiranja je nezavistan od drugih. Zamene kao što su *stub* metode, lažni objekti ili test pojas mogu pomoći kod testiranja modula u izolaciji. Implementacija unit testa može varirati od veoma manuelne (olovka i papir), do formalizovane kao deo *build* automatizacije.

Cilj unit testiranja je da izoluje svaki deo programa i da pokaže da je svaki od njih individualno korektan [3]. Ova vrsta testiranja pronalazi greške još u ranim fazama razvoja softvera.

4. TESTIRANI SOFTVER

Pošto se ovaj rad bavi razvojem sistema za automatizovano testiranje, neophodno je isprobati takav sistem na postojećem proizvodu. Za potrebe ovog rada je korištena *DMS Web klijent* aplikacija koja je zasnovana na *Java* tehnologijama i *AJAX* rešenjima, i koji omogućuje interakciju korisnika sa *DMS* sistemom pomoću multimedijalne vizualizacije.

4.1 Multimedijalna vizualizacija

Multimedijalni sadržaj je onaj koji koristi kombinaciju različitih formi sadržaja. Stoga, on obuhvata sve kombinacije: tekstualnog, audio i video sadržaja, kao i sadržaja sa nepokretnim slikama, interakcijom i animacijom.

Većina modernog softvera poseduje *GUI* koji pruža korisnicima mogućnost interakcije sa implementiranim sistemom koji se nalazi u pozadini. Ovakav interfejs predstavlja prezentaciju dostupnih podataka i funkcionalnosti, pomoću multimedijalne vizualizacije. Uglavnom je to kombinacija tekstualnog, interaktivnog i sadržaja sa nepokretnim slikama, ali su interfejsi uglavnom obogaćeni i ostalim formama sadržaja.

4.2 DMS Web klijent

DMS veb klijent aplikacija je rešenje firme **TelventDMS** za omogućavanje pristupa podacima i ograničenoj funkcionalnosti *DMS Softvera* klijentima na veb-u, upotrebom standardnih veb *browser*-a (*IE 7* ili noviji, *FF 3* ili noviji, *Chrome*).

DMS softver predstavlja tehničko-informacioni sistem koji omogućava kompletan uvid u tehničke podatke, analizu rada elektro-distributivne mreže, primenu ekspertskih metoda za unapređenje rada sistema, kao i upravljanje i planiranje u elektro-distributivnom sistemu [4].

WebDMS aplikacija je veb aplikacija koja pruža većinu funkcionalnosti pomenutog *DMD* alata. Ova aplikacija je nastala iz potrebe za pristupom određenim podacima i ograničenom broju funkcija iz *DMS Software*-a preko Interneta, a pomoću standardnih Internet pretraživača. Osnovni zadatak aplikacije je naravno bio prikaz elektro-distributivnih mreža, a kasnije su dodate i druge funkcionalnosti, poput upravljanja mrežom. Zbog korišćenja mrežne komunikacije bezbednost ove aplikacije je bila kljucan faktor u njenom razvoju.

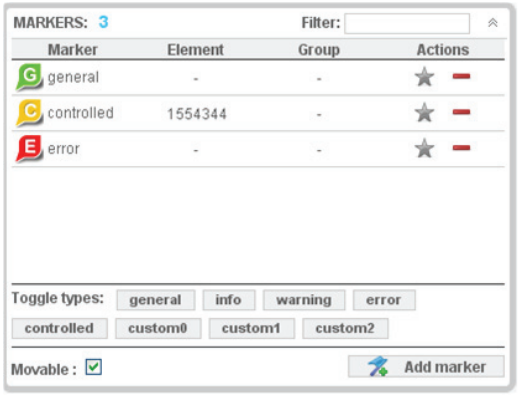
Komponente *WebDMS* aplikacije mogu se podeliti u četiri grupe: veb komponente, servisne komponente, kontrolne komponente, *domain* objekti. Od značaja za ovaj rad je samo *DMS Map* veb komponenta.

4.3 Opis testirane komponente

Komponenta koja je korištena za testiranje jeste *DMS Map* (*Graphic Controlware - GCW*) koja je zadužena za prikaz, monitoring i manipulaciju mrežom distributivnog sistema. GCW može biti programski podešen da prikazuje različite prikaze mreže, kao sto su šematski ili geografski prikazi. On može biti integrisan u bilo koje veb rešenje razvijano od treće strane. Takođe, enkapsulira prikaz mreže sa multifunkcionalnim *toolbar*-ovima i različitim navigacionim alatima koji omogućavaju intuitivnu interakciju koja je prilagođena korisnicima. Primarno je testiran njen modul koji radi sa markerima. Dodatno je testiran i *GPS Path* modul.

4.4 Markeri

Markeri se koriste da označe lokacije ili elemente mreže koji su od posebnog interesa u nekom trenutku ili periodu vremena. Markeri mogu biti ručno pozicionirani na mapu ili automatski posle nekih operacija. Prozor "*Markers*" pokazuje listu markera.



Slika 1. *Prikaz instanciranih markera*

Postoje 4 kolone: *Marker* (prikazuje ikonicu i vrstu markera), *Element* (prikazuje ID elementa, ako je marker povezan sa nekim elementom na mapi), *Group* (prikazuje ID grupe kojoj marker pripada) i *Actions* (zvezda menja vidljivost markera, dok znak minus briše marker).

Ispod te liste se nalaze dugmad za pokazivanje i sakrivanje određenih markera prema njihovoj vrsti. U donjem desnom ćošku se nalazi dugme "*Add marker*" koje sluzi kreiranju novih markera. Markeri se kreiraju u dva koraka. Prvi je odabir tipa markera iz liste unapred definisanih i dodatnih tipova markera. Zatim se klikne na "*Next*". Drugi korak je da se marker dodeli elementu ili da se specificiraju koordinate na mapi gde bi marker trebao da bude postavljen. Klikom na dugme "*Set marker*" se završava proces kreiranja novog markera.

4.5 GPS putanje

Namena ovog modula je da se koristi prilikom praćenja ekipa za intervencije na mrežama koje se nalaze na terenu (*GPS tracking*). Ova funkcionalnost se koristi u koordinaciji sa *Comtop* serverom. *PDA* uređaj šalje svoje koordinate *Comtop* serveru u *WGS 84* formatu. Na istom serveru se čuva i log položaja za svaku ekipu.

Scenario korišćenja ove funkcionalnosti je sledeći: Korisnik *GCW* otvara *Comtop* radni tiket za jednu od ekipa. Posle toga, pozicija i putanja te ekipe ce biti vidljiva na *GCW*. Korišćenjem pomenutih *API* metoda, *Comtop* aplikacija je u mogućnosti da prikaže poziciju i putanju određene terenske ekipe, kao i pozicije i putanje svih terenskih ekipa ako je u pitanju korisnik koji je nadležan za upravljanje terenskim ekipama (*super user*). *API* omogućuje postavljanje specifičnog simbola kao markera, i podešavanje osobina prikaza putanje – boja i debljina linije.

Podaci se osvežavaju na *Comtop* serveru, i propagiraju *DMS Map* komponenti pozivanjem odgovarajućih metoda, u zavisnosti od prirode promena koje su nastale.

5. SISTEM ZA AUTOMATIZOVANO TESTIRANJE

Za razliku od testiranja koda veb aplikacija, gde je za pravljenje automatizovanih testova dovoljno koristiti *framework* za unit testiranje za željeni programski jezik (recimo *JUnit* za *Javu*, *NUnit* za *.NET*, itd), kod testiranja *GUI*-a veb aplikacija potrebno je koristiti dodatne alate koji će ili simulirati rad *browser*-a, ili upravljati željenim *browser*-om.

Pošto su za razvijanje veb aplikacije koja će služiti kao primer testiranja u ovom radu, korištene *Java* tehnologije, *Java* će biti korištena i za pisanje automatizovanih testova. Stoga je i *JUnit* izabran kao *framework* za pisanje unit testova.

5.1 JUnit

JUnit je *open source framework* za testiranje, napisan za *Java* programski jezik. Međutim, postao je toliko popularan da je preveden i prilagođen za korišćenje u drugim programskim jezicima: *C# (NUnit), Python (PyUnit), Fortran (fUnit), C++ (CPPUnit)*. Pomoću ovih alata, moguće je automatizovano izvršavati jedinične testove iz *IDE* razvojnog okruženja.

JUnit je integrisan u mnoga razvojna okruženja (*Eclipse, NetBeans, JBuilder, JDeveloper, Sun Java Enterprise Studio*). Prednost integrisanog *JUnit*-a je u tome što je olakšan rad pri testiranju koda jer se vrši preko bogatog grafičkog interfejsa, a mnoge operacije, kao što je na primer kompajliranje testova i izvršavanje testova, su automatizovane. *JUnit* je nastao na osnovu originalnog okruženja za testiranje u *SmallTalk*-u, koje je napisao **Kent Beck** (*SUnit*). Nakon toga su **Kent Beck** i **Erich Gamma** napravili *JUnit* za *J2SE* i *J2EE platformu*.

TestCase klasa je osnovna klasa za testiranje. Sadrži pomoćne metode za kreiranje i pokretanje testova. Kada se izvrši *TestCase*, automatski se pokreću sve metode klase koja počinje sa test. *JUnit* koristi refleksiju da odredi koje testove da pokrene, olakšavajući na taj način dodavanje novih metoda za testiranje u postojeći slučaj testiranja. Refleksija je proces kojim program može da uoči i modifikuje svoju strukturu i ponašanje.

5.2 Selenium 2

Kao mogući kandidati za dodatni alat su bili razmotreni *WebUI Test Studio, HtmlUnit, Watij* i *Selenium*, zbog svoje rasprostranjenosti i kvaliteta. Svaki od razmatranih alata ima svoje prednosti i mane. Ipak, *Selenium* se izdvojio kao najbolja opcija u trenutku pisanja ovog rada.

Iako je *Selenium* 1 imao dosta mana i ograničenja, *Selenium 2* koji predstavlja integraciju *Selenium*-a 1 i *WebDriver*-a ima ih veoma malo, a vremenom se i one gube kako napreduje njegov razvoj. Naime, dostupan je na svim platformama, moguće je pokretati testove na pravim instancama *Firefox*-a, *Internet Explorer*-a, *Safari*-a, *Opera*-e, *Chrome*-a (drajveri za *Safari* i *Operu* su u još u fazi testiranja), ali je moguće i pokrenuti test na *HtmlUnit* instanci. Podržava pisanje testova u *Java*-i, *C#*-u, *Ruby*-u, *PHP*-u, *Perl*-u i *Python*-u. Dodatno, podržava rad i sa mobilnim platformama kao što su *Android* i *iPhone*. Od svih dodatnih alata koje pruža *Selenium*, najpoznatiji je verovatno *Selenium IDE* (*plug-in* za *Firefox*) koji omogućuje snimanje akcija korisnika i potom generiše test u željenom programskom jeziku. *Selenium* dolazi pod open source licencom.

Selenium 2 je alat za automatizaciju testiranja koji poseduje mnoštvo korisnih osobina, uključujući kohezivni i *OO API*, kao i odgovor na probleme starijih implementacija *Selenium*-a. Developeri *Selenium* 1 i *WebDriver*-a su se složili da oba alata imaju svoje prednosti i da bi njihovo ujedinjenje dovelo do robusnijeg automatizacionog alata [5].

WebDriver je brz *framework* za automatizovano testiranje veb aplikacija. On ima drugačiji pristup od *Selenium* 1. Umesto da bude *JS* aplikacija koja se pokreće unutar *browser*-a, on koristi mehanizam koji najviše odgovara za datom *browser*-u. Za *Firefox*, on je implementiran kao ekstenzija, dok za *Internet Explorer* koristi *IE*-ove kontrole za automatizaciju. Promenom mehanizma kontrole *browser*-a, moguće je zaobići ograničenja koja su dodeljena *browser*-u od strane *JS* bezbednosnog modela. U takvim slučajevima, automatizacija kroz *browser* ne bi bila dovoljna, stoga *WebDriver* može da koristi mogućnosti koje nudi *OS*. Npr, na *Windows*-u je moguće simulirati kucanje teksta na nivou *OS*-a i time bliže modelovati kako korisnik vrši interakciju sa *browser*-om [6].

Kada se ova dva *framework*-a uporede, primetno je da su slabosti jednog zapravo prednosti drugog. Npr, dok *WebDriver* pristup za podržane *browser*-e zahteva mnogo rada od strane *framework* developera, *Selenium* lako može biti proširen. Obratno, *Selenium* uvek zahteva postojeći *browser*, dok *WebDriver* pruža mogućnost lake i veoma brze emulacije kroz implementaciju zasnovanu na *HTML Unit*-u. *Selenium* ima odličnu podršku za mnoge česte situacije koje treba testirati, dok *WebDriver* pruža mogućnost da se izađe iz *JS sandbox*-a i to otvara neke zanimljive mogućnosti.

6. ORGANIZACIJA TESTOVA

Posle sastavljanja skupa alata za testiranje u cilju izgradnje sistema za automatizovano testiranje, potrebno je napisati testove za željenu aplikaciju koji bi pokazali validnost i funkcionalnost testirane aplikacije, ali koji bi i ukazali na kvalitet razvijenog sistema za testiranje.

Kao što je prethodno navedeno, objekti testiranja su bili *Marker* i *GPS Path* moduli *GCW*-a, tj *DMS Mape*. Posto je cilj testiranja verifikacija funkcionalnosti *GUI*-a, bilo je potrebno testiranje i na nivou *API*-a (*TDMS JS API*) i na nivou prikaza u browser-ima (*Internet Explorer* i *Mozilla Firefox*) za sta je korišten *Selenium* 2 *WebDriver API* i *WebDriver Java* klijentska biblioteka.

Klase koje su korištene za testiranje su podeljene na tri nivoa: prolazi, testovi prolaza i upravljačke klase.

Prolazi su zapravo testovi koji pozivaju neophodne *API* metode *(TMDS JS API* ili *WebDriver API*) u određenim metodama od kojih je svaka određena celina testiranja

(inicijalizacija, nalaženje elemenata, markiranje elemenata, testiranje instanci markera,...). Pored toga svaka od tih metoda je podeljena na vise sekcija tako da su svi krucijalni pozivi odvojeni u zasebne sekcije. Svaka sekcija je okružena *try-catch* blokom koji hvata sve izuzetke (generički *Exception*). Pored hvatanja izuzetaka, postoji i programska provera rezultata izvršenja svih poziva API metoda. Na osnovu ta dva mehanizma, svaka metoda pruža informaciju o uspešnosti njenog izvršavanja i o detaljima poziva koji eventualno nije uspešno izvršen.

Testovi prolaza su klasične *Java Unit* test klase koje pokreću dodeljeni prolaz kroz kod za specifični *browser* i specifične konfiguracione parametre. Svaka metoda ovih klasa je zaseban Unit test (*@Test*, osim metoda *setUpBeforeClass()* i *tearDownAfterClass()*, koje služe, respektivno, inicijalizaciji i deinicijalizaciji objekata i parametara neophodnih za pokretanje prolaza kroz kod) i svaka od njih poziva *assertTrue()* za odgovarajuću metodu u dodeljenoj klasi prolaza kroz kod.

Test Runner i *Test Logger* su **upravljačke klase** koje, respektivno, omogućavaju pozivanje testova spolja (komandna linija, druga aplikacija,...) i generisanje *HTML* izveštaja o rezultatu pokrenutih testova.

7. ZAKLJUČAK

Zadatak ovog rada bila je realizacija sistema za automatizovano testiranje veb aplikacija za multimedijalnu vizualizaciju. Sistem je realizovan korišćenjem *Java* programskog jezika i *Selenium* 2.0 *Java* klijentske biblioteke. Aplikacija koja je sluzila za testiranje je *TDMS Web aplikacija* koja predstavlja veb klijenta (pokreće iz veb *browser*-a) za *DMS Softver*. U ovom radu je opisan postupak razvoja ovakvog sistema, kao i opis svih njegovih komponenata.

Pošto su za razvijanje veb aplikacije koja će služiti kao primer testiranja u ovom radu, korištene *Java* tehnologije, *Java* će biti korištena i za pisanje automatizovanih testova. Stoga je i *JUnit* izabran kao framework za pisanje unit testova. Kao mogući kandidati za dodatni alat su bili razmotreni *WebUI Test Studio, HtmlUnit, Watij i Selenium*, zbog svoje rasprostranjenosti i kvaliteta. Svaki od razmatranih alata ima svoje prednosti i mane. Osobine kao sto su: lakoća izrade testova, širok spektar mogućnosti koje pruža, fleksibilnost, realizacija drajvera za najviše korištene veb *browser*-e kao i drajver za generički *browser*, brzina realizacije testova, i još mnoge druge, su dovele do zaključka da je za realizaciju sistema za automatizovano testiranje veb aplikacije za multimedijalni prikaz, *Selenium 2.0 framework* trenutno najbolji izbor.

Izvršenje testova koji su pisani za potrebe ovog rada traje nekoliko minuta. Za to vreme oni su uspeli da nađu dve greške u kodu. Kasnije se, kroz višestruke iteracije egzekucije tih testova utvrdilo da su ovo i jedine dve greške u testiranim modulima.

Ipak, bilo je potrebno nekoliko dana da se takvi testovi napišu. Sa druge strane, primenom metode ručnog testiranja, programer bi gotovo izvesno našao obe greške u roku od jednog dana, ali bi potrošio nekoliko dana da locira i ispravi greške, i da utvrdi ispravnost novog koda. Iako bi, naizgled, ručno testiranje oduzelo istu količinu vremena kao i proces pisanja i izvršavanja automatskih testova, to važi samo za prvu iteraciju testiranja. U kasnijim iteracijama, metoda ručnog testiranja zahteva minimalno istu količinu vremena, dok automatizovano sada zahteva samo vreme izvršenja testova.

Zaključak je da se kombinacija *JUnit*-a i *Selenium*-a 2 može prilično efikasno iskoristiti za testiranje veb aplikacije koja obavlja multimedijalnu prezentaciju većeg broja podataka, čak i kada se koriste AJAX asinhroni pozivi radi efikasnije prezentacije i interakcije sa prezentovanim sadržajem.

8. LITERATURA

[1] C. Kaner, Exploratory Testing, Orlando FL: Quality Assurance Institute Worldwide Annual Software Testing Conference, 2006.

[2] A. Leitner, I. Ciupa, B. Meyer, A. Fiva, „Contract Driven Development = Test Driven Development - Writing Test Cases“, Proc. of ESEC/FSE'07: Eur. Soft. Eng. Conf. and the ACM SIGSOFT Symp. on the Found. Of Soft. Eng. 2007, Septembar 2007.

[3] A. Kolawa, D. Huizinga, Automated Defect Prevention: Best Practices in Software Management, Wiley-IEEE Computer Society Press, 2007.

[4] Telvent DMS, Uvod u DMS Software, Telvent DMS, 2009.

[5] ThoughtWorks, „Selenium History“, http://seleniumhq.org/about/history.html, istorijat Selenium-a.

[6] S. Stewart, „Introducing WebDriver“, http://google-opensource.blogspot.com/2009/05/introducing-webdriver.html, o WebDriver-u.

Kratka biografija:



Ivan Perić rođen je u Kikindi 1986. god. Završio je gimnaziju u Kikindi 2005. godine. Fakultet tehničih nauka u Novom Sadu je upisao 2005. godine, dok je master studije upisao 2009. godine. Ispunio je sve obaveze i položio je sve ispite predviđene studijskim programom sa prosečnom ocenom 8.56.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

KOMPRESIJA VREMENSKE SERIJE LINEARNIM INTERPOLACIONIM METODOM

TIME SERIES COMPRESSION USING LINEAR INTERPOLATION METHOD

Petar Vasilić, *Fakultet Tehničkih Nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu je opisano korišćenje SLIM (Simple linear interpolation method) algoritma za kompresiju vremenskih serija (time series). Komprimovane vremenske serije se čuvaju u bazi podataka i kasnije po potrebi mogu biti rekonstruisane. Algoritam je realizovan kao mrežni servis koristeći Microsoft WCF (Windows Communication Foundation) tehnologiju.*

Abstract – *This paper presents SLIM (Simple linear interpolation method) algorithm which is used for time series compression. Compressed time series are saved into a data base, and later, if there is need to do so, they can be reconstructed. Algorithm is realized in the form of service using Microsoft WCF (Windows Communication Foundation) technologies.*

Ključne reči –*Vremenske serije (time series), linearni interpolacioni metod, rekonstrukcija vremenske serije*

1. UVOD

Vremenske serije se koriste u mnogim oblastima života, a posebno u naučnim oblastima kao što su inženjerstvo, ekonomija, sociologija itd. Vremenska serija (*time series*) predstavlja niz obzervacija (u automatici to su najčešće podaci dobijeni kao merenja prikupljena sa akvizicionih uređaja, mada to mogu biti i razni drugi signali) koje se beleže u ravnomerno raspoređenim, diskretnim vremenskim trenutcima [1]. Osnovni podaci svake vremenske serije su vrednosti merene veličine i vremenski trenuci u kojima su merenja izvršena, a mogu imati i dodatne informacije kao što su kvalitet vrednosti ili merna jedinica.

Smanjenje prostora potrebnog za skladištenje podataka ili smanjenje opterećenja telekomunikacionih linkova kod prenosa vremenskih serija su najčešći razlozi za korišćenje algoritama koji vrše njihovu kompresiju. Cilj je obezbediti maksimalno smanjenje količine podataka sa minimalnim gubicima u kvalitetu podataka. Takva, proređena vremenska serija se snima u bazu podataka. Ako se vremenska serija želi pročitati iz baze tom prilikom ona mora biti rekonstruisana zbog toga što podaci snimljeni u bazi ne mogu biti interpretirani kao originalni signal. Rekonstrukcija obuhvata proces aproksimiranja originalnog signala na osnovu podataka o komprimovanom signalu koji se nalaze snimljeni u bazi.

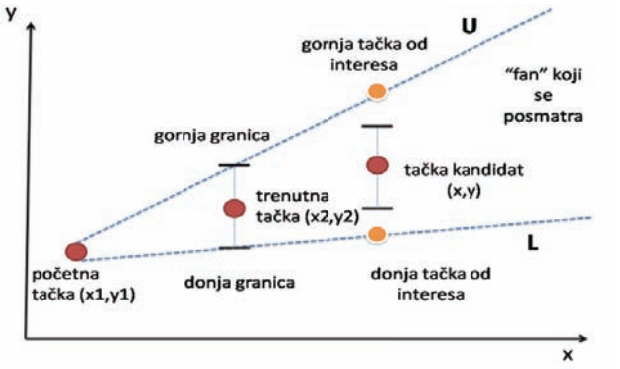
Kvalitetna rekonstrukcija će rezultirati signalom koji je blizak polaznom signalu koji je komprimovan.Neki od algoritama kompresije su *SLIM_1, SLIM_2*, *box_car, backslope, swinging door* i tako dalje. Njihovim daljim proučavanjem ustanovljeno je da *SLIM_1* i *SLIM_2* za definisane zahteve (veliki procenat kompresije uz solidan kvalitet rekonstruisanog signala) daju najbolje rezultate.

NAPOMENA:
Ovaj rad je proistekao iz diplomskog – master rada čiji je mentor bio dr Aleksandar Erdeljan, docent.

2. LINEARNI INTERPOLACIONI METOD – OBJAŠNJENJE ALGORITMA

Lepezasti interpolatori (*fan interpolators*) su se vremenom pokazali kao veoma korisni pa su zbog toga postali veoma popularni. Nazivaju se još i SLIM algoritmi [2].

Algoritam će najbolje biti objašnjen kroz mali primer.



Slika 1. *Početak primene algoritma, tačke od značaja*

Algoritam počinje od prve tačke u seriji, ona se proglašava za *početnu tačku*, ili oslonac "lepeze" koju prestavljaju poluprave L i U. Poluprava U se formira preko formule za jednačinu prave kroz dve tačke koja glasi:

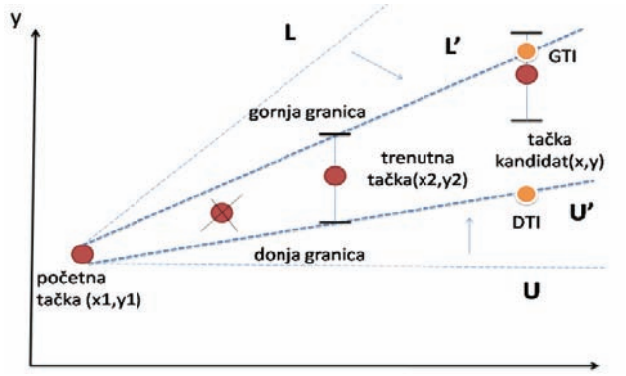
$$(Y - Y1) = \frac{(Y2-Y1)}{(X2-X1)} * (X - X1)$$

gde su $(X_1,Y_1) = (x_1,y_1)$ koordinate*početne tačke* a $(X_2,Y_2) = (x_2,y_2$ + gornja granica)koordinate *trenutne tačke*.Na sličan način dobijamo i polupravu L, sa razlikom da se koordinate (X_2,Y_2) računaju kao: $(X_2,Y_2) = (x_2,y_2$ - donja granica). Na ovakav način se dobijaju opisipoluprave L (*lower fan boundary*) i poluprave U (*upper fan boundary*).

Gornja i donja granica (*max_dev*) predstavljaju maksimalno dozvoljeno odstupanje odizmerene vrednosti. Te vrednosti su unikatne za svaki signal [2].

Sledeći korak je određivanje koordinata *gornje i donje tačke od interesa*. To se uradi tako što se u jednačine polupravih L i U uvrsti X koordinata *tačke kandidata.*Sve tri tačke, *gornja tačka od interesa* (GTI), *donja tačka od interesa* (DTI) i *tačka kandidat*(TK) imaju istu X koordinatu, tj odnose se na isti vrenenski trenutak. Zatim se porede Y koordinata TK sa Y koordinatom GTI i DTI. Za trenutnu TK važi da je Y(TK) < Y(GTI) i Y(TK) > Y

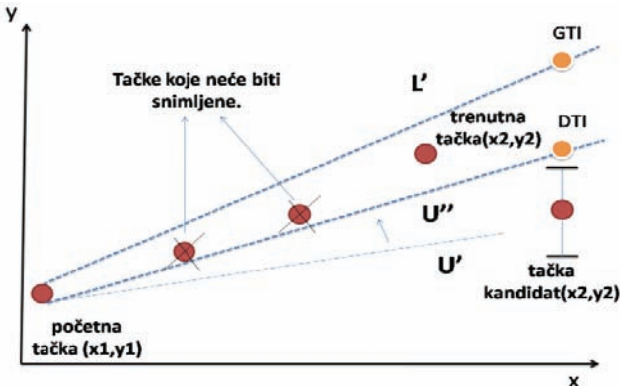
(DTI), pa iz toga sledi da se *tačka kandidat*(TK) nalazi unutar ograničavajuće "lepeze".
Sledeći korak je provera da li je potrebno korigovati ograničavajuće prave L i U. Prava U se proverava tako što poredimo da li je (Y(TK) + gornja granica) < Y(GTI), i ukoliko jeste tada je potrebno sračunati novu jednačinu ograničavajuće prave U. Prava L se proverava tako što proverimo da li važi da je (Y(TK) - gornja granica) > Y(DTI), i ukoliko važi onda je potrebno odrediti novu jednačinu prave L. U primeru su oba uslova ispunjena pa se *tačka kandidat* proglasi za *trenutnu tačku,* i onda se na već opisan način kroz *početnu tačku* i novu *trenutnu tačku* formiraju prave U' i L'. Na kraju opisanog postupka dobijena jesituacija prikazana na slici 2.



Slika 2. *Formirane L' i U', nova trenutna i tačka kandidat*

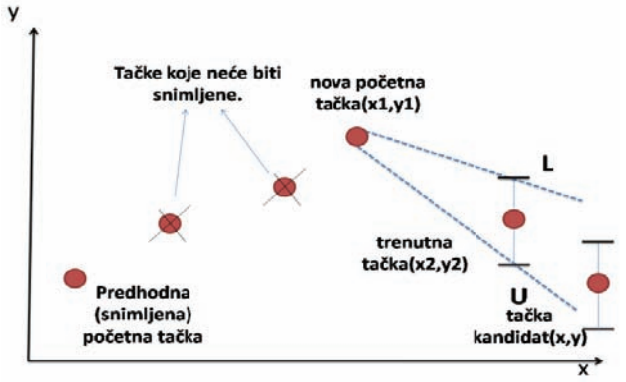
Za novu *tačku kandidata* uzima se sledeći član vremenske serije.Prethodna *trenutna tačka* je preskočena, i ona sigurno neće biti sačuvana kao član komprimovanog (proređenog) signala.
Vidljivo je, iz položaja prave L' u odnosu na L i U' u odnosu na U da sa svakim korakom algoritma ograničavajuća "lepeza" postaje sve uža, što znači da algoritam sa svakim narednim korakom postaje sve osetljiviji na "oscilacije" vrednosti u vremenskoj seriji.
Sada ponovo,preko jednačina za U' i L' izračunamo vrednosti Y koordinate za *gornju tačku od interesa* (GTI) i *donju tačku od interesa* (DTI), i poredimo ih sa Y koordinatom *tačke kandidata* (TK). Vidljivo je da se trenutna TK nalazi unutar "lepeze", pa se zbog toga algoritam nastavlja. Unutar "lepeze" je i tačka (Y(TK) - donja granica), pa se zbog toga mora sračunati novo U''. Međutim, pošto je Y(GTI) < (Y(TK) + gornja granica)poluprava L' mora da se sačuva kao gornje ograničenje "lepeze".
Ako bi se sračunalo novo gornje ograničenje u ovoj TK to bi dovelo do širenja "lepeze", a to nije cilj pa se zbog toga zadržava staro gornje ograničenje. Isto tako, ako bi bilo (Y(TK) - donja granica) < DTI tada se nebi računalo novo donje ograničenje. Kada se sračuna nova gornja granica dobijemo situaciju prikazanu na slici 3.
U situaciji prikazanoj na slici 3 *početna tačka*je i dalje ista kao na početku algoritma, dok je prethodna *tačka kandidat* postala *trenutna tačka*. Sada postoje dve tačke koje neće biti sačuvane kao članovi komprimovanog (proređenog) signala.

Formirana je nova donja granična prava "lepeze" U'', dok je L' ostala nepromenjena. *Za novu tačku kandidata* uzima se sledeći član vremenske serije.



Slika 3. *Formitana U'' ,nova trenutna i tačka kandidat*

Ponovo se na isti način izračunaju Y koordinate *gornje i donje tačke od interesa* i porede se sa Y koordinatom *tačke kandidata*.Vidi se da je Y(TK) < Y(DTI), pa se iz toga može zaključiti da se tačka kandidat nalazi ispod donje granice ograničavajuće "lepeze". Ovo znači da algoritam sada mora da definiše novu *početnu tačku* i da ponovo započne da proverava tačke vremenske serije.
Početna tačka se snima u bazu podataka, a za novu *početnu tačku* se uzima poslednja snimljena*trenutna tačka*na taj način sto se preuzme njena X koordinata, a za njenu (Y) koordinatu nove *početne tačke* se uzima vrednost (Y – donje ograničenje) od poslednje snimljene *trenutne tačke.*Da se kojim slučajem tačka koja ispada izvan ograničenja nalazila iznad ograničavajuće "lepeze" koordinate nove *početne tačke* bi se uzele na sličan način,samo bi se za Y koordinatu nove *početne tačke* uzela koordinata (Y + gornje ograničenje) poslednje snimljene *trenutne tačke.*
Tačka kandidat koja je ispala iz ograničavajuće lepeze postaje *trenutna tačka,* i kroz njenu gornju i donju granicu se konstruišu nove ograničavajuće prave L i U i na taj način se definiše nova "lepeza".



Slika 4. *Formirana nova ograničavajuća "lepeza"*

Nova ograničavajuća "lepeza" je formirana i algoritam može da nastavi sa proređivanjem tačaka. Kada se ponovo desi da *tačka kandidat* ispadne iz ograničenja trenutna *početna tačka* će ponovo biti snimljena u bazu podataka, formiraće se nove prave L i U i algoritam će nastaviti sa izvršavanjem.
Kada se pogleda slika 4 vidljivo je da dve tačke koje su eliminisane imaju jako malo odstupanje od prave provučene kroz dve *početne tačke* (snimljenu i novu) koje se nalaze na slici. Gubljenje ta dva podatka neće se u

velikoj meri odraziti na kvalitet signala na taj način što će se degradacija signala držati na podnošljivom nivou.

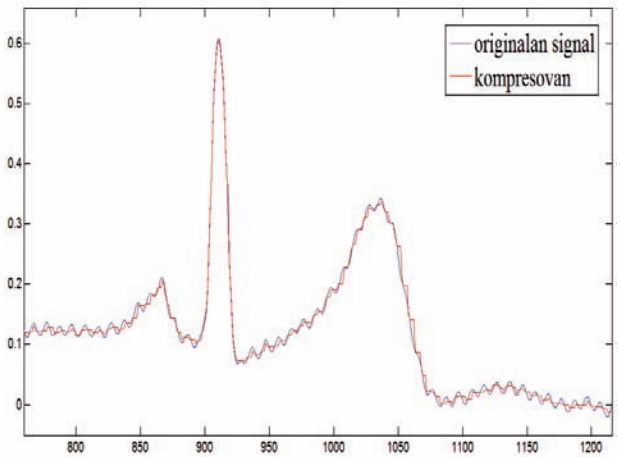
3. PREGLED REZULTATA RADA ALGORITMA

Za testiranje realizovanog algoritma korišćena su tri različita signala. To su EKG signal, godišnje merenje potrošnje električne energije i merenje napona.Izabrani signali su dosta različite prirode da bi se istakle razlike u primeni algoritma na različitim tipovima signala. Imamo signal koji ima jako veliki raspon amplituda i koji se dosta brzo i naglo menja (godišnja potrošnja), signal koji uglavnom osciluje u oblasti oko jedne vrednosti i ne menja se previše (merenje napona 230V), i signal koji se po brzini promene nalazi između prethodna dva (EKG).
EKG signal je snimljen u 4170 tačaka, i njegov raspon je od 0-0,65 V.

Tabela 1. *Rezultati SILM-a primenjenog na EKG signal*

MAX_DEV	**0.001**	**0.002**	**0.005**	**0.01**	**0.03**
BROJ TAČAKA	3093	2405	1217	676	249
% KOMPRESIJE	23%	40%	71%	83%	94%

Vidljivo je da sa povećanjem dozvoljenog odstupanja (*max_dev*) raste i procenat kompresije. Teoretski je moguće postaviti *max_dev* na veliku vrednost i na taj način dobiti veoma velik procenat kompresije, ali u tom slučaju će rekonstruisani signal mnogo odstupati od originalnog signala. Isto tako mala vrednost *max_dev* će rezultovati veoma malim procentom kompresije i rekonstruisanim signalom koji se gotovo poklapa sa originalnim. Zbog toga za *max_dev*treba odabrati vrednost koja će obezbediti dobar odnos veličine kompresije i kvaliteta rekonstruisanog signala.



Slika 5. *Originalan i rekonstruisan EKG signal*

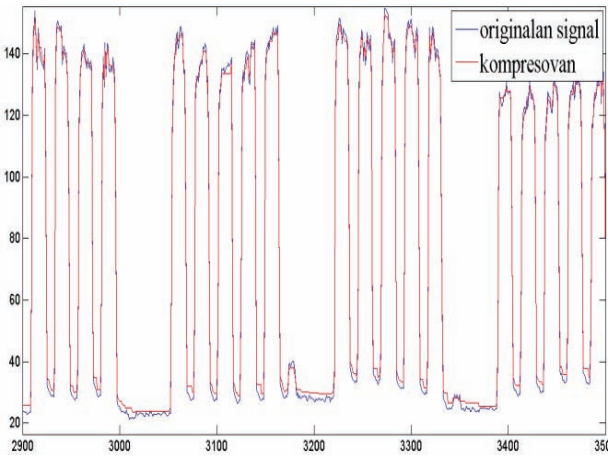
Kvalitetan rekonstruisan EKG signal se dobija za *max_dev* =0,005. Dobija se 71% kompresije signala.
Na slici 5 je prikazan jedan ciklus EKG signala. Vidljivo je da rekonstruisan signal (crvena boja) veoma dobro prati originalan signal (plava boja). Zbog velike kompresije i dobre rekonstrukcije da se zaključiti da je EKG signal veoma podesan za komprimovanje ovim algoritmom.

Sledeći signal predstavlja godišnju potrošnju, vremensku seriju koja se često pojavljuje u sistemima koji nadgledaju potrošnju u elektrodistributivnim mrežama.
Signal je snimljen u 8760 tačaka, a amplituda signalaje u rasponu od 0 – 160.Iz broja tačaka je vidljivo da se snimanje vrednosti vrši svakog sata, u periodu od godinu dana.

Tabela 2. *Rezultati SLIM primenjenog na signal potrošnje*

MAX_DEV	**0.5**	**1**	**3**	**5**	**10**
BROJ TAČAKA	5845	4587	3094	2511	1954
% KOMPRESIJE	33%	47%	64%	71%	78%

Ovaj signal nije toliko pogodan za primenu SLIM algoritma kao EKG signal, i to zbog njegove osobine da su oscilacije u signalu veoma brze, a i velike po amplitudi. Brze promene u signalu dovode do toga da je potrebno snimiti više tačaka da bi komprimovani signal pratio dinamiku originalnog signala.



Slika 6. *Originalan i rekonstruisan signal potrošnje*

Na slici je prikazan deo signala (oko 600 uzastopnih tačaka). Vrednost *max_dev* je 3, a procenat kompresije je 64%. Procenat kompresije je dosta dobar, ali vidljivo je da signal nije dobro rekonstruisan u nekim intervalima (donji delovi signala).
To se dešava zbog toga što su oscilacije amplitude signala u tim intervalima unutar *max_dev*.Ako su zahtevi takvi da je potrebno da se rekonstruisani i originalni signal preklope u sitnijim detaljima vrednost *max_dev* mora da se smanji, a samim tim bi se smanjio i procenat kompresije.
Poslednji signal predstavlja napon meren uređajem koji se koristi za merenje važnih vrednosti vezanih za potrošnju električne energije (napon, struja, snaga). Signal je snimljen u 3610 tačaka, i nema velike oscilacije u amplitudi zbog toga je merena vrednost napona, ako nema nekih preteranih smetnji i prekida napajanja, stalno u neposrednoj okolini vrednosti 230V.
Zbog brzih promena u okolini vrednosti 230V potrebno je jako smanjiti *max_dev* da bi se detalji originalnog i rekonstruisanog signala preklopili.

Tabela 3. *Rezultati SLIM primenjenog na signal napona*

MAX_DEV	**0.1**	**0.5**	**0.75**	**1**	**1.5**
BROJ TAČAKA	2837	1228	803	560	313
% KOMPRESIJE	21%	66%	77%	84%	91%

Ovakvi signali se najčešće analiziraju da bi se utvrdilo da li je bilo nekih većih smetnji ili gubitka napajanja. Sa ovim algoritmom takve informacije mogu se očuvati sa dosta velikom vrednošću *max_dev* i sa jako velikom kompresijom signala.

4. REALIZACIJA SERVISA KORIŠĆENJEM WCF

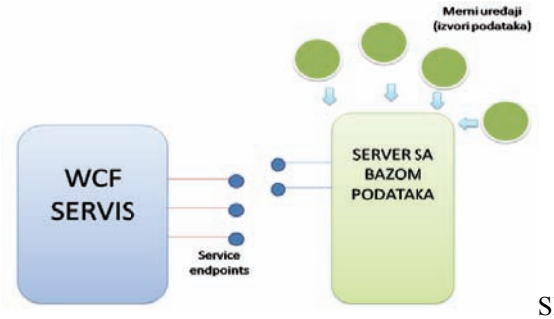
WCF (*Windows Communication Foundation*) je veoma koristan deo .NET okruženja koji omogućava kreiranje servisno orijentisanih aplikacija, neformalno poznatih kao *web servisi* [4]. Servisi mogu biti dostupni preko Interneta ili preko lokalne mreže u nekom preduzeću.

Korišćenjem WCF tehnologije i programskog jezika C# realizovan je servis koji pruža usluge kompresije i rekonstrukcije vremenske serije.

Korisnik servisa je najčešće neki server koji sakuplja podatke sa mernih uređaja i skladišti ih u fajl sistem ili u memoriju. Da bi se formirala vremenska serija podaci se moraju snimati neko određeno vreme (na primer 5 minuta). Kada je signal formiran tada se on prosleđuje servisu koji vrši njegovu kompresiju.

Servis se poziva tako što mu se pošalje niz podataka koji on prihvata. Svaka tačka u tom nizu mora da sadrži dva podatka. Jedan je *time stamp*, vremenski trenutak u kome je izvršeno snimanje vrednosti merene veličine, a drugi je vrednost merene veličine u tom trenutku.Servis na date podatke primenjuje SLIM algoritam i klijentu vraća niz u kome se nalazi komprimovana vremenska serija.

Kada neki korisnik od servera zatrazi da pogleda signal tada se iz baze pročita komprimovana vremeska serija, i pošalje se servisu sa zahtevom za rekonstrukciju signala. Servis rekonstruiše signal i vraća ga nazad serveru koji ga prosleđuje klijentu. Šematski prikaz sistema prikazan je na slici 7.



Slika 7.

Šematski prikaz sistema

Interfejs koji se pruža korisnicima se sastoji od dve funkcije. Prva funkcija, *proredíPodatke,* kao ulazni parametar dobija *dozvoljeno odstupanje* tipa double i niz čiji su elementi instance klase *Data*. Klasa *Data* predstavlja jedan član vremenske serije, i njeni atributi su *vrednost* tipa float i *timeStamp* tipa DateTime. Druga funkcija, *interpolirajPodatke* kao parametar prima niz instanci klase *Data* koji predstavlja proređenu vremensku seriju i vrstu interpolacije koja će se koristiti (*zeroHold* ili *linearna*). Obe funkcije kao povratnu vrednost vraćaju nit instanci klase *Data* i poruku o uspešno ili neuspešno obavljenom zadatku.

5. ZAKLJUČAK

Rad se bavi proučavanjem linearnih interpolacionih metoda, i njihove primene na kompresiju vremenskih serija.Nakon procesa simulacije u softverskom paketu MATLAB i analize različitih algoritama utvrđeno je da SLIM_1 algoritam najviše odgovara postavljenim zahtevima.

Primenom algoritma na različite vrste signala utvrđeno je da algoritam daje bolje rezultate na manje promenljivim signalima (EKG ima bolje rezultate nego signal potrošnje električne energije). Zbog ovoga se uviđa velika mogućnost primene ovakvih rešenja za signale u procesnoj industriji. Velika većina tih signala je dosta statična (nivo u nekom rezervoaru ne može naglo promeniti). Korišćenjem ovakvog softverskog rešenja koje će komprimovati signal pre čuvanja mogu se ostvariti velike uštede u vremenu pisanja i čitanja u bazu podataka, kao i uštede u memorijskom prostoru potrebnom za skladištenje podataka.

Programski jezik C# i Microsoft .NET okruženje su iskorišćeni da bi se realizovalo softversko rešenje u vidu WCF servisa koji klijentu pruža usluge kompresije i rekonstrukcije vremenske serije. Predviđeno je da klijent ovog servisa bude server koji u sebi ima bazu podataka koja se koristi za skladištenje vremenskih serija.

6. LITERATURA

[1] Hua LU, Xia HU, Yanjie Deng,"*An Intuitive Time Series Compression Method*",University of Electronic Science and Technology of China,Journal of Computational Information Systems (2010) 189-195

[2] Peter A. James, "*Data Compression For Process Historians*" ,1995, Chevron Research and Technology Company,Richmond, CA 94802-0627

[3] Matthew J. Watson, Antonios Liakopou, "*A Practical Assessment of Process Data Compression Techniques*",Chemical Process Modeling and Control Research Center,Lehigh University, Bethlehem, Pennsylvania 1998

[4] Adam Freeman, "*Introducing Visual C# 2010*", 2010 , "Springer Science & Business Media", ISBN : 978-1-4302-3171-4

7. BIOGRAFIJA



Petar Vasilić rođen je u Vrbasu 21.02.1987 god. 2006 god završio je gimnaziju „Svetozar Miletić“ u Srbobranu. Apsolvent je na Fakultetu tehničkih nauka u Novom Sadu, odsek Računarstvo i automatika, katedra za automatiku i upravljanje sistemima.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

PRISTUP VREMENSKIM SERIJAMA UPOTREBOM OPC UA INTERFEJSA

ACCESSING TIME SERIES USING OPC UA INTERFACES

Vladimir Ber, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj –*U radu je opisan način čitanja i pisanja vremenskih serija preko interfejsa OPC UA (OLE for Process Control Unified Architecture) servera. Podaci vremenske serije su skladišteni u Microsoft SQL Server 2008 bazi podataka, i u radu je velika pažnja posvećenapostizanjušto boljih performansi prilikom rada sa bazom podataka. Komunikacija između servera i klijenata je realizovana preko Microsoft WCF (Windows Communication Foundation) tehnologije.*

Abstract –*This paper describesthe way of reading and writing time series using interfaces of OPC UA (OLE for Process Control Unified Architecture) server. Time series are being stored in Microsoft SQL Server 2008 database, and great attention is dedicated obtaining the best possible performance when working with the database. Communication between the server and the client is realized using Microsoft WCF (Windows Communication Foundation) technology.*

Ključne reči – *Vremenske serije, nadzorno-upravljački sistemi*

1.UVOD

U dvadeset prvom veku sa stalnim inovacijama u svim oblastima tehnike, rastu i zahtevi savremenih kompanija. Njihovi zahtevi se odnose pre svega na rad sa velikim brojem informacija koje moraju biti pribavljene u što kraćim vremenskim trenucima. U ovu sliku uklapa se i sama OPC UA arhitektura koja standardizuje i diktira načine razmena informacija u savremenoj procesnoj automatici. Vremenske serije se nameću kao jedan od oblika podataka koji se danas možda i najčešće razmenjuju, najviše zbog njihovog samog oblika (vremenske značke – vrednosti) koji je ljudima veoma blizak i pogodan za analizu. Jedan od problema koji se ovde javlja je vreme pristupa vremenskim serijama, koje korisnici zahtevaju da bude što manje, a sa druge strane ono je ograničeno, i teško se smanjuje bez napredka računarskih tehnologija.Vremenska serija predstavlja skup posmatranja x_t, od kojih je svaki zabeležen u specifičnom trenutku t. Diskretne vremenske serije su one vremenske serije u kojima set vremenskih trenutaka T_o, odgovara setovima podataka u kojima su obavljena posmatranja u ekvidistantnim trenucima. Kontinualne vremenske serije se pribavljaju kada se podaci beleže tokom određenog kontinualnog vremenskog perioda, npr. T_o=[0,1] [1].

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Aleksandar Erdeljan, docent.

U ovom radu opisan je način čitanja i skladištenja diskretnih vremenskih serija, koje mogu, a ne moraju biti ekvidistantno raspoređene u vremenu. Skladištenje vremenskih serija predstavlja veliki problem pre svega zbog velike količine podataka koja se neprestano povećava. Kao predmet ovog rada konstruisan je OPC UA server u programskom jeziku C# koji komunicira sa klijentima, a podatke skladišti u SQL 2008 bazu podataka. OPC UA (OLE for Process Control Unified Architecture) predstavlja najnoviju OPC specifikaciju, predstavljenu od OPC fondacije i značajno se razlikuje od svojih prethodnika. OPC specifikacija definiše set standardnih objekata, metoda i interfejsa koji se koriste za razmenu podataka u procesnoj automatici. OPC fondacije je objavila setove standarda koji su široko zatupljeni u industriji, a neki od njih su OPC DA (Data Access) OPC A&E (Alarm and Events) i OPC HDA (Historical Data Access) koji je najzastupljeniji u okviru ovog rada.

2. OPC HDA

OPC HDAje specifikacija definisana od strane OPC fondacije, deo je OPC UA specifikacijei koristi se za pribavljanje i analizu istorijskih podataka sa ciljem prikaza, analize, optimizacije itd. [2]. OPC serveri najčešće pribavljaju istorijske podatke koji su skladišteni u relacionim bazama podataka. Prema specifikaciji aplikacije mogu da koriste OPC HDA i za upis istorijskih podataka, koji se skladište u bazu podataka.

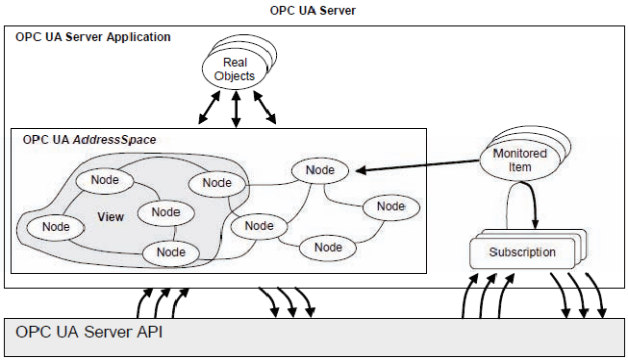
Ovakvi podaci se beleže u obliku vrednost – vremenska značka (*time stamp*) u bazu podataka, a njima mogu biti pridodate još neke vrednosti, npr. kvalitet vrednosti koji označava razlog nepostojanja vrednosti, ili zastavica koja nam govori da li su podaci modifikovani itd. (u okviru ovog projekta podaci u bazi podataka su modelovani tako da sadrže sve ove atribute).

3.OPC UA ARHITEKTURA

OPC UA sistemska arhitektura modelovana je po principu klijent - server koji se ponašaju kao partneri. Arhitektura u ovom radu zasnovana je na principu jednog servera i više klijenata koji mu šalju zahteve.

OPC UA Part 1 – Concepts [3] definiše i način na koji je potrebno modelovati serversku stranu. Na slici 1 prikazani su svi glavni elementi OPC UA Servera, i načini na koji su oni međusobno povezani.

Real objekti su fizički ili softverski objekti kojima može da pristupa samo serverska aplikacija.Oni omogućavaju lakši rad serveru i nisu vidljivi od strane spoljašnjih aplikacija. OPC UA Server API je deo servera koji implementira osnovnu funkcionalnost servera, ovaj deo prihvata zahteve od klijenata, obrađuje ih i šalje rezultate.



Slika 1. *Arhitektura OPC UA Servera*

Adresni prostor (*address space*) je modelovan kao set čvorova (*node*) kojima mogu da pristupaju klijenti putem interfejsa i metoda koji su definisani u specifikaciji OPC-a. Pogled (*view*) je podskup adresnog prostora. Koristi se za ograničavanje pogleda na adresni prostor klijentima (svaki klijent ima svoj pogled).Najvažniji deo servera predstavljaju čvorovi, koji su reprezentacije stvarnih objekata, njihove definicije i reference. Čvor u sebi može sadržati atribute, reference na druge čvorove, i osobine koje se mogu menjati. Svaka Node klasa nasleđuje osnovnu BaseNode klasu koja u sebi sadrži osnovne podatke koje treba da sadrži svaka node klasa.

BaseNode klasu nije moguće instancirati, već je po specifikaciji omogućeno samo njeno nasleđivanje. Postoji osam vrsta node klasa (Variable, Variable Type, Object, Object Type, Reference Type, Data Type, Method, View), i one zajedno modeliraju objekte u adresnom prostoru.

Variableklasa se koristi za prezentovanje podataka koji mogu biti prosti ili složeni. Instance ove klase nikad nisu definisane same, već su delovi nekih drugih čvorova (u većini slučajeva instanci Object klase) i opisani su preko VariableType klase. Objekti se koriste za definisanje sistema, sistemskih komponenti, objekata iz stvarnog sveta i softverskih komponenti, instance su Objectklase a opisani su sa ObjectType klasom. Ponekad su sistemi preobimni, i u tim slučajevima klijenti nemaju potrebu za posmatranjem celog adresnog prostora. U drugim slučajevima server ne želi da dozvoli svim klijentima jednak pristup čvorovima u adresnom prostoru. Zbog ovih slučajeva potrebno je uvesti pojam pogleda (*View*), gde svaka instanca klase View definiše jedan podskup pogleda na adresni prostor, koji se odnosi na jednog klijenta. DataType klasa se koristi za opis prostih i složeni tipova podataka. Reference su definisane kao instance ReferenceType klase, i vidljive su od strane klijenta. Methodklasa definiše funkcije koje je klijent u stanju da pozove koristeći neke od definisanih servisa OPC specifikacije.

Pristupanje podacima na serveru omogućeni su preko servisa koji su definisani u delu *OPC UA Part 4 – Service Drafts* [4]. Servisi su podeljeni u više grupa, od kojih za pristup vremenskim serijama najzanimljivije dve, grupa servisa za upravljanje čvorovima i servisi za upravljanje atributima.

Servis za upravljanje čvorovima služi za dodavanje i brisanje čvorova i referenci između njih. Svi dodati čvorovi nastavljaju da postoje u adresnom prostoru i kada nestane klijent koji ih je napravio. Prilikom dodavanja čvorova moguće je odmah postaviti reference na njegove roditelje, ako su oni već definisani, a prilikom brisanja moguće je u okviru jedne instrukcije obrisati i roditelja i sve njegove potomke. Svaki od ovih servisa kao rezultat vraća rezultat koji je tipa StatusCode (niz od 32 bita) koji predstavlja uspešnost operacije.

Preko servisa za upravljanje atributima korisnici mogu pristupati atributima koji su delovi čvorova. U ovoj grupi nalaze se servisi Read, HistoryRead, Write i HistoryUpdate.

Read servis sluzi za čitanje jednog ili više atributa, jednog ili više čvorova. On kao rezultat vraća najsvežiju vrednost čvora, koja se nalazi unutar maksimalnog dozvoljenog vremena (*maxAge*) koju je zadao klijent.

Servis HistoryRead se koristi u okviru standarda za pribavljanje istorijskih događaja i podataka.

Za HistoryRead servis se može reći da je složen servis, pre svega zato što zahteva veliki broj informacija poslatih od strane klijenta koje server treba da procesira. Među ovim vrednostima nalazi se identifikator čvora čije podatke klijent želi da pročita, vremenski interval iz kojeg klijent želi da pročita podatke i da li je u taj interval potrebno uključiti granice ili ne, da li je potrebno vratiti serverski, klijentski ili oba time stamp-a i možda najvažniji od svih je tip podataka koje klijent želi da mu budu vraćeni. Tipovi podataka koji mogu biti zatraženi od servera su sirovipodaci (*Raw*) i oni predstavljaju skladištene podatke koji mogu biti kompresovani, zatim vrednosti promenjene nakon skladištenja (*Modified*), interpolirane vrednosti (*Interpolated*) - veličine dobijene nekom od interpolacijai procesirane vrednosti (*Processed*) kao što su minimalna, maksimalna ili srednja vrednost.

Writeservis se koristiza upis vrednosti jednog ili više čvorova. Server podatke skladišti u SQL Server 2008 bazu podataka.

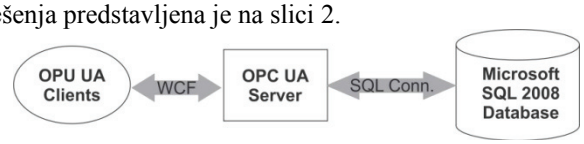
Podaci koji se razmenjuju između klijenata i servera predstavljeni su kao struktura podataka, koja u sebi sadrži: vrednost merenja, vremenski trenutak u kojem je klijentposlao serveru podatak, vremenski trenutak kada je server zabeležio podatak u bazu podataka, kvalitet podatka koji je opisan binarnim nizom. U SQL bazi podataka u tabeli Values koja čuva ove podatke primljene od strane servera, nalazi se i vrednost modified, tipa Boolean, koja daje informaciju o tome da li su podaci skladišteni u tom redu sirovi podaci ili modifikovani.

HistoryUpdateservis, predstavlja jedan od servisa za upravljanje atributima i služi za „promenu“ jedne ili više vrednosti čvorova. Pod promenom podrazumevaju se više različitih operacija. Operacije ubacivanja (*Insert*) koja ubacuje podatke u željenom trenutku (*timestamp*) ukoliko podatak sa tom vremenskom značkom ne postoji, a u suprotnom vraća statusni kod o neuspeloj operaciji.

Zamena (*Replace*) je operacija koja zamenjuje podatak iz baze podataka koji ima zadatu vremensku značku, sa novom zadatom vrednošću, ukoliko podatak koji je potrebno zameniti ne postoji u bazi, server će vratiti kod o neuspeloj operaciji i neće ubaciti novu vrednost. Osveženje (*Update*) predstavlja uniju prethodne dve operacije, ako podatak već postoji u bazi podataka on će ga samo zameniti sa novom vrednošću, u suprotnom će kreirati novi slog u bazi podataka. Brisanja (*Delete*) služi za uklanjanje podataka iz Values tabele, ukoliko podatak sa zadatom vremenskom značkom ne postoji server klijentu vraća kod o neuspeloj operaciji.

4. IMPLEMENTACIJA SERVERA I KLIJENATA

Osnovna ideja softverskog rešenja je da se formira OPC server koji pruža sa jedne strane usluge OPC klijentima (definisane po standardu za OPC UA), a sa druge strane vrši čitanje i upis podataka u Microsoft SQL bazupodataka.Gruba predstava arhitekture programskog rešenja predstavljena je na slici 2.



Slika 2. *Arhitektura rešenja*

Softversko rešenje realizovano je programskim jezikom C# u Microsoft.NET okruženju. Pomenuta komunikacija između OPC klijenata i servera je ostvarena preko Microsoft-ovog WCF servisa [5]. WCF (Windows Communication Foundation) je okruženje za konstruisanje servisno orijentisanih aplikacija i stoga je bilo veoma pogodno za realizaciju ovog programskog rešenja. Zbog podrške ovog okruženja za rad sa nitima servis je organizovan tako da omogućava istovremen rad sa više klijenata. Pri tome je bilo potrebno obratiti pažnju da se ne naruši konzistentnost podataka, koja je očuvana korišćenjem ugrađene C# klase ReaderWriterLock, koja nudi mogućnost posebno, zabrane čitanja, odnosno pisanja resursu od strane niti.

Podaci primljeni od strane OPC klijenata se skladište u SQL Server 2008 bazu podataka. U bazi podataka, podaci su organizovani u više tabela (ObjectTable, VariableTable i ValueTable). ObjectTable u sebi sadrži sve potrebne informacije o instancama objekata, a VariableTable o njenim poljima. ValueTable čuva vrednosti merenja primljene od strane OPC servera, u toku testiranja ova tabela je najopterećenija jer se najviše upita vrši upravo nad njom.Realizacija upisa podataka u bazu urađena je na tri načina, sa ciljem da bi se na završetku rada mogla uporediti sva tri načina i izabralo rešenje koje ima najbolje performanse prilikom upisa.

Prvi način unosa podataka realizovan je direktnim pozivanjem INSERT operacija upotrebom ADO.NET tehnologije, dok se drugi i treći način oslanja na upotrebu SQLBulkCopy funkcije (koja se nalazi u System.Data.SQLClient namespace-u), sa različitim načinima skladištenja podataka pre unosa u bazu podataka. Čitanje iz baze podataka je takođe ostvareno preko ADO.NET tehnologija. U okviru ovog programskog rešenja, čitanje nije realizovano na više načina, jer to ne bi donelo značajne promene u poboljšanju rada celokupnog softverskog rešenja. Dobijanje boljih performansi kod čitanja iz SQL baze podataka moguće je ostvariti prilikom organizovanja same baze podataka i njenih tabela.

ADO.NET se nekad posmatra kao evolucija ADO tehnologija (*ActiveX Data Objects*), mada je bio značajno promenjen tako da se može smatrati kao sasvim novi proizvod. On predstavlja set komponenti koji je dostupan programerima za pristup podacima i servisima za podatke. Često se koristi za pristup i modifikovanje relacionih baza podataka, mada se može koristiti i za pristup ne relacionim bazama podataka. U okviru ovog softverskog rešenja ADO.NET je korišten za rad sa SQL bazom podataka, odnosno u samom kodu su formirani upiti i naredbe u identičnom obliku kao u SQL-u, i na taj način prosleđivani bazi na obrađivanje. Prednost ovakvog načina rada sa bazom je u tome što je većina stvari automatizovana i zbog toga programer ne mora da vodi mnogo računa o detaljima prilikom komunikacije sa bazom. Microsoft SQL Server u sebi sadrži popularnu terminalsku naredbu pod nazivom *bcp,* koja služi za premeštanje grupa podataka iz jedne tabele u drugu. Klasa *SqlBulkCopy* omogućava pisanje koda koji ima sličnu funkcionalnost. Postoje i drugi načini unosa podataka u SQL tabele (npr. INSERT naredba, koja je ranije pomenuta), ali *SQLBulkCopy* nudi mnogo bolje performanse u odnosu na njih. Mana ove klase je to što omogućuje rad samo sa Microsoft SQL bazom podataka. Rad ove klase se razlikuje u odnosu na ostale u tome što ona ne prosleđuje trenutno svaki podataka bazi podataka na skladištenje (i na taj način manje opterećuje samu bazu) već interno kod sebe skladišti podatke na određeni vremenski period, a zatim ih u grupama prosleđuje na skladištenje. Privremeno skladištenje podataka pre unosa u bazu izvedeno je na dva načina i predstavlja dva različita rešenja koja su kasnije testirana.

Rešenje koje kao privremeno skladište koristi dva tekstualna fajla, prilikom pisanja podataka koristi se jadan fajl, a prilikom čitanja podataka koje server želi da pošalje u bazu koriste se oba. Prilikom implementacije servera bilo je bitno razmišljati šta se dešava u trenucima dok server, odnosno jedna njegova nit, čita podatke iz tekstualnog fajla (txt) koje želi da prosledi u bazu na skladištenje, i tom vremenskom periodu, tekstualni fajl je zaključan i ni jedna druga nit nemože da mu pristupi. Tada je potrebno sve druge niti preključiti na drugi tekstualni fajl i dozvoliti im upis novih vrednosti u njega, jer u suprotnom one bi bile blokirane dok nit za čitanje ne oslobodi fajl. Veoma je bitno napomenuti da je dobra odlika ovakvog server to što podatke koje još nije prosledio bazi podataka na skladištenje, zapisuje interno u tekstualni, a ne u neku od programskih kolekcija, jer u slučaju otkaza servera nemože doći do gubitka podataka.

Za razliku od prethodnog rešenja, treća izvedba servera skladišti podatke interno u C# DataTable kolekciju, pre upisa u SQL Server. Unapred znajući da će ovakav server dati najbolje rezultate, on je namerno konstruisan na ovaj način, jer je kasnije bio dobar reper u testiranju druga dva servera. Loša strana ovog servera je to što može doći do gubitka podataka koji se nalaze u DataTable kolekciji prilikom *failover*-a. Vremenski period na koji se vrši upis u bazu je podesiv u kodu (u procesu testiranja je bio 10s), a može se još više smanjiti tako da gubitak podataka bude zanemarljiv.

5. TESTIRANJE PERFORMANSI SERVERA I REZULTATI

Sam proces testiranja programskog rešenja bilo je potrebno osmisliti što bliže stvarnim situacijama, kako bi on što realnije dočarao performanse rada servera. Test je osmišljen tako da bi što više opteretio server i na taj način prikazao kako on uspeva da se izbori sa slivanjem velikog broja informacija u određenom vremenskom intervalu. Grubo se može reći da svaki OPC server komunicira sa dve vrste klijenata, pa su tako i u ovom radu serveri bili testirani istovremeno sa istim brojem klijenata iz ove dve grupe. Prvu vrstu klijenata predstavljaju merno akvizicio-

ni uređaji koji mere analogne i digitalne veličine u određenim, uglavnom ekvidistantnim vremenskim trenucima, i šalju ih serveru na obradu i skladištenje. Druga vrsta klijenata predstavlja delove sistema koje od OPC servera potražuju informacije o signalima u određenim trenucima ili vremenskim intervalima, a bitno je da zahtevi za podacima iz bliže prošlosti budu češći nego potražnja za podacima iz daleke prošlosti.
Testiranje je vršeno na računaru sa Intel Core2Duo procesorom na 2.2GHz i 2MB keš memorije, 4GB DDR2 ram memorije na 800Mz. U toku testiranja bio je pokrenut jedan OPC server koji je komunicirao sa korisnicima i Microsoft SQL bazom podataka. Sa serverom je komuniciralo 12 klijenata, po 6 iz svake grupe, koji su slali zahteve za čitanje, odnosno pisanje, na svakih pola sekunde. Performanse rada servera snimane su pomoću *Windows*-ovog *Performance Monitor*-a, a uključeni su sledeći brojači koji su odabrani iz široke lepeze monitora, preporučenih od strane Brenta Ozara u njegovoj knjizi [6]:

-Processor:% Processor time, koji daje izveštaj o zauzeću procesora u procentima;
-PhysicalDisk:Avg.Disk Queue Length, koji predstavlja prosečan broj niti koje čekaju procesor na izvršenje;
-PhysicalDisk:Avg. Disk msec/Read, predstavlja prosečno vreme trajanja jednog upisa u milisekundama.
U narednim tabelama 1, 2 i 3 prikazani su rezultati testiranja OPC UA servera. Testiranja su obavljena istim metodama.

Tabela 1.*Performanse OPC UA servera implementiranog pozivanjem INSERT operacija*

	Avg. Disk msec/Read	Avg.Disk Queue Length	% Processor time
Minimalna vrednost	0,47	0	26,09
Maksimalna vrednost	3,07	7	50,31
Prosečnavrednost	0,97	1	33,04

Tabela 2.*Performanse OPC UA server implementiranog pozivanjem SQLBulkInsert metode sa podacima iz tekstualnog fajla*

	Avg. Disk msec/Read	Avg.Disk Queue Length	% Processor time
Minimalna vrednost	0,49	0	35,46
Maksimalna vrednost	1,60	8	93,90
Prosečna vrednost	0,65	2	61,07

Tabela 3.*Performanse OPC UA server implementiranog pozivanjem SQLBulkInsert metode sa podacima iz DataTable-a*

	Avg. Disk msec/Read	Avg.Disk Queue Length	% Processor time
Minimalna vrednost	0,47	0	21,33
Maksimalna vrednost	0,97	5	82,24
Prosečna vrednost	0,60	1	45,35

Na osnovu prikazanih tabela može se zapaziti da OPC UA server koji vrši upis pozivanjem INSERT operacija najmanje opterećuje procesor, ali je najsporiji što se tiče upisa. OPC Server koji koristi SQLBulkInsert metodu i radi sa tekstualnim fajlom, daje bolje rezultate prilikom upisa podataka, ali više zauzima procesor zbog rada sa tekstualnim fajlom. Za razliku od njega, server koji podatke belezi u DataTable kolekciju pre upisa, dao je najbolje rezultata, ali on ne mora biti najbolje rešenje jer prilikom njegovog korišćenja može doći do gubitka podataka.

6. ZAKLJUČAK

U okviru ovog rada predstavljen je način pristupa i beleženja vremenskih serija u Microsoft SQL Server bazu podataka, koristeći interfejse konstruisanog OPC UA servera.
Predstavljena su tri načina rada sa upisom podataka u bazu podataka. Prvi način je lakši za implementaciju, manje opterećuje procesor, ali znatnije opterećuje rad baze podataka, koja može biti usko grlo. Drugi primer je znatno teži za implementaciju, zahteva poznavanje konkurentnog programiranja kako se ne bi narušila konzistentnost podataka, češće zauzima procesor, ali je pouzdan i znatno manje opterećuje bazu podataka, što u velikom broju slučajeva može biti razlog dizajnera servera da se odluče za ovakvo rešenje.
I treći primer, koji je sličan po implementaciji sa drugim, ali za razliku od njega podatke nesnimljene u SQL bazu podataka beleži u DataTable klasu (a ne u tekstualni fajl) što daje bolje performanse, ali je u određenim situacijama nepouzdan.

7. LITERATURA

[1]James D. Hamilton. „*Time series analysis"*,Princeton University Press, Oxford, 1994.
[2] OPC Foundation. „*OPC UA Part 11 – Historical Access"*Version 1.0 July 2006.
[3] OPC Foundation. „*OPC UA Part 1 – Concepts"*Version 1.0 July 2006.
[4] OPC Foundation. „*OPC UA Part 4 – Service Drafts"*Version 1.0 July 2006.
[5] Jay Hilyard, Stephen Teilhet. „*C# 3.0 Cookbook"*Third edition,O'Reilly Media, Sebastopol CA,December 2007.
[6] Brent Ozar. „*Professional SQL Server 2008 Internals and Troubleshooting"*, Wrox press,January 2010.

8. KRATKA BIOGRAFIJA



Vladimir Ber rođen je u Zrenjaninu 19.08.1987. godine. Završio je osnovnu školu „Vuk Karadžić" u Zrenjaninu, a zatim upisao srednju elektro tehničku školu „Nikola Tesla" u Zrenjaninu. Fakultet tehničkih nauka upisao je 2006. godine na smeru za računarstvo i automatiku.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9

EKSTERNA AUTENTIFIKACIJA PO MODELU IDENTITETA ZASNOVANOM NA TVRDNJAMA

CLAIM-BASED IDENTITY MODEL EXTERNAL AUTHENTICATION

Igor Javni, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu je prikazano jedno rešenje eksterne autentifikacije po modelu identiteta zasnovanog na tvrdnjama. Rad sadrži motive za razvoj specijalizovanog* ***web*** *servisa za izdavanje sigurnosnih tokena i opisuje njegovu ulogu i načine korišćenja u realizaciji eksterne autentifikacije. Rad sadrži ključne implementacione detalje softverskog rešenja realizovanog* ***C#*** *programskim jezikom u* ***.NET*** *4.0 okruženju.*

Abstract – *This paper introduce one solution of claim-based identity model external authentication. The paper contains main motives for development of specialized security token service and describes its role and usage in the implementation of external authentication. The paper contains the most important implementation details of software solution implemented with C# programming language in .NET 4.0 framework.*

Ključne reči: *Model identiteta zasnovan na tvrdnjama, eksterna autentifikacija, servis za izdavanje sigurnosnih tokena*

1. UVOD

U ovom radu je prikazana primena eksterne autentifikacije korišćenjem modela identiteta zasnovanog na tvrdnjama (*claims-based identity model*), kao i primena kontrole pristupa zasnovane na tvrdnjama (*claims-based access control - CBAC*). Tradicionalni pristup autentifikacije korisnika podrazumeva da svaka aplikacija sadrži programsku logiku koja vodi računa o autentifikaciji. Zadatak rada je da prikaže moguće rešenje u kome aplikacije ne sadrže programsku logiku koja vodi računa o autentifikaciji korisnika, već se ta odgovornost delegira tako što se autentifikacija izmešta u specijalizovane servise. Specijalizovani *web* servis koji vrši autentifikaciju korisnika se naziva *identity provider* (*IdP*) i sastoji se od više komponenti među kojima je svakako najvažnija komponenta servis za izdavanje sigurnosnih tokena (*security token service - STS*). U okruženju u kome postoji *IdP*, korisnik se autentifikuje tako što *IdP*-u pošalje zahtev za autentifikaciju koji sadrži i odgovarajući korisnički akreditiv.

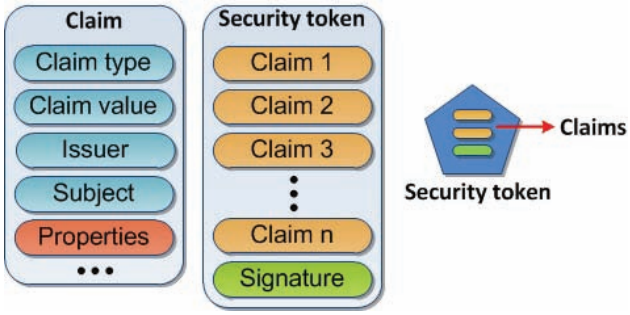
Ukoliko je autentifikacija uspešna, *IdP* izdaje korisniku sigurnosni token koji sadrži identitet korisnika po modelu identiteta zasnovanog na tvrdnjama. Aplikacije koje su izvršile eksternalizaciju autentifikacije se nazivaju *relying parties* (*RP*). One su to učinile tako što su uspostavile odnos poverenja sa odgovarajućim *IdP*-om. Korisnik koristi ovaj sigurnosni token kako bi pristupio *RP* aplikacijama. Korisničke informacije *RP* aplikacijama uvek pristižu u istom, jasno definisanom i standardizovanom formatu, upravo zbog upotrebe sigurnosnih tokena. Ove informacije iz sigurnosnog tokena, koje su predstavljene odgovarajućim tvrdnjama, *RP* aplikacije koriste kako bi vršile kontrolu pristupa korisnika po *CBAC* modelu.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Hajduković, red. prof.

2. OPIS REŠAVANOG PROBLEMA

2.1 Model identiteta zasnovan na tvrdnjama

Kod modela identiteta zasnovanog na tvrdnjama korisnik prezentuje svoj identitet kao konačan skup tvrdnji (*claims*) [1,2]. Tvrdnja predstavlja određenu informaciju o korisniku koja ga na odgovarajući način opisuje. Svaka tvrdnja se kreira od strane izdavaoca koji se najčešće naziva *IdP*. Zadatak *IdP*-a je da kreira identitet korisnika koji sadrži potrebne tvrdnje naspram informacija koje *IdP* poseduje o datom korisniku. Identitet korisnika opisan tvrdnjama mora na odgovarajući način da se prenese između svih zainteresovanih strana i iz tog razloga potrebno je serijalizovati identitet korisnika. Za serijalizaciju identiteta se koristi koncept sigurnosnog tokena. Na slici 1. prikazana je struktura sigurnosnog tokena i struktura tvrdnje.



Slika 1. *Sigurnosni token i tvrdnja*

Sigurnosni token predstavlja serijalizovani skup tvrdnji o identitetu, koji je digitalno potpisan od strane *IdP*-a. Digitalni potpis omogućava da se potvrdi validnost sigurnosnog tokena, pa samim tim i tvrdnji koje su u njemu sadržane. Svaka tvrdnja se sastoji pre svega iz tipa tvrdnje i vrednosti tvrdnje, ali može da poseduje i niz drugih osobina koje je opisuju. Sigurnosni tokeni se prenose u okviru zahteva.

2.2 Kontrola pristupa zasnovana na tvrdnjama

CBAC se primenjuje u okruženju u kome je kao model identiteta izabran model identiteta zasnovan na tvrdnjama. *CBAC* je veoma sličan kontroli pristupa zasnovanoj na

ulogama (*role-based access control - RBAC*). Po *RBAC* modelu autorizacione odluke se definišu upotrebom uloga. Ako korisnik poseduje ulogu koja je potrebna za izvršavanja akcije, pristup mu je dozvoljen. U suprotnom, pristup mu neće biti dozvoljen. *CBAC* može da podrži *RBAC* pristup zato što tvrdnje mogu da se iskoriste za prenos različitih vrsta informacija uključujući i uloge. Po *CBAC* modelu autorizacione odluke se definišu upotrebom proizvoljne logike koja koristi podatke dostupne u tvrdnjama kako bi donela odluku o pravima pristupa [3].

2.3 *Claims-based* arhitekture

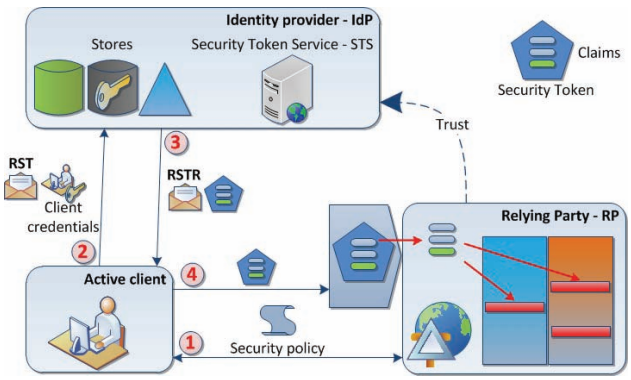
Postoje dve vrste okruženja u kojima je moguće implementirati sisteme koji koriste model identiteta zasnovan na tvrdnjama. One se razlikuju naspram tipa klijenta koji se koristi za pristup *RP* aplikacijama [1,2]. To su :

- Aktivni klijenti – grupa klijenata koji predstavljaju *rich client*-e odnosno *smart client*-e
- Pasivni klijenti – grupa klijenata koji predstavljaju *web browser*-e.

Dizajn sistema na visokom nivou je isti bez obzira na tip korišćenog klijenta. Međutim u procesu implementacije izbor klijenta uslovljava i izbor korišćenih tehnologija i protokola. Pasivni klijenti koriste jednostavne protokole za rukovanje sigurnosnim tokenima poput *HTTP* protokola sa *GET* i *POST* metodama, a sigurnosni token prenose u okviru *HTTP POST* zahteva. Aktivni klijenti koriste mnogo kompleksnije biblioteke za rukovanje sigurnosnim tokena (recimo *WCF* i *WIF* biblioteke u slučaju *.NET* okruženja), pa se samim tim oslanjaju na upotrebu kompleksnijih protokola kao što je *WS-** familija protokola za *web* servise, a sigurnosni token prenose u okviru sigurnosnog zaglavlja *SOAP* poruka. Rad će se pre svega baviti okruženjem aktivnih klijenata, pa samim tim i konkretna implementacija rešenja će pokrivati slučaj aktivnih klijenata.

2.4 Arhitektura sistema aktivnih klijenata

Arhitektura sistema u kome se koriste aktivni klijenti prikazana je na slici 2.



Slika 2. *Arhitektura sistema aktivnih klijenata*

Tok komunikacije i razmena poruka između učesnika na slici 2. su specificirani *WS-Trust* protokolom [4]. *RP* aplikacija je izvršila eksternalizaciju autentifikacije tako što je uspostavila *trust relationship* sa *IdP*-om. U koraku jedan, klijent se obraća *RP*-u kako bi dobio odgovarajuću sigurnosnu politiku koja podrazumeva da za pristup *RP*-u mora predhodno da se autentifikuje kod nadležnog *IdP*-a. U koraku dva, kada klijent sazna kojem *IdP*-u treba da se obrati za autentifikaciju, on kreira odgovarajući zahtev po *WS-Trust* specifikaciji koji se naziva *Request for Security Token* (*RST*) i prilaže svoj klijentski akreditiv. Uloga *IdP*-a je da izvrši autentifikaciju korisnika i, ukoliko je korisnik uspešno autentifikovan, izda sigurnosni token. Token se izdaje u odgovarajućem formatu i sadrži identitet klijenta po modelu identiteta zasnovanog na tvrdnjama. Za konkretan format tokena u radu je izabran *SAML* format tokena, pošto je to najzastupljeniji format tokena u sistemima baziranim na na *WS-** familiji protokola [1]. Sigurnosni token je digitalno potpisan od strane *IdP*-a. *IdP*, pored obrade zahteva za izdavanje tokena, mora da poseduje podršku za skladištenje podataka, među kojima se pre svega misli na korisničke naloge i na korisničke tvrdnje. U koraku tri, *IdP* kreira odgovor na zahtev po *WS-Trust* specifikaciji, koji se naziva *Request for Security Token Response* (*RSTR*), i u okviru njega šalje klijentu sigurnosni token. U koraku četiri, klijent koristi sigurnosni token kako bi uputio poziv ka *RP*-u. *RP* vrši validaciju sigurnosnog tokena tako što proverava digitalni potpis. Ukoliko je token validan, *RP* može pristupiti identitetu sadržanom u okviru sigurnosnog tokena i iskoristiti tvrdnje kako bi po *CBAC* modelu kontrole pristupa donosio autorizacione odluke.

2.5 Bezbednost *web* servisa

Pošto *web* servisi u većini slučajeva koriste *HTTP* i *TCP/IP*, potrebno je postojanje sigurnosnog sloja koji će da omogući bezbednost prilikom korišćenja *web* servisa. Ovim slojem obezbeđujemo zaštitu i bezbednost podataka i poruka. Okruženja sa aktivnim klijentima se oslanjaju na *WS-Security* specifikaciju koja definiše mehanizme digitalnog potpisivanja i enkripcije, čime možemo da garantujemo poverljivost, integritet i neporecivost poruka i sigurnosnih tokena. Enkripcija nam obezbeđuje poverljivost time što omogućava da poruke između učesnika mogu da putuju između više tačaka i da istovremeno budu zaštićene od napada i zloupotreba (*end-to-end security*). Digitalno potpisivanje poruka nam pruža mogućnost neporecivosti i integriteta poruka. Neporecivost nam obezbeđuje da učesnik u komunikaciji nije u mogućnosti kasnije da tvrdi kako on nije kreirao datu poruku i digitalno je potpisao. Integritet nam obezbeđuje da sačuvamo sadržaj poruka od zlonamernih ili slučajnih izmena. Pored poruka, enkripciju i digitalno potpisivanje primenjujemo i na sigurnosne tokene koje izdaje *IdP*.

2.6 Bezbednost komunikacije u okruženju aktivnih klienata

Svaki od učesnika u komunikaciji (klijent, *IdP* i *RP*) poseduje svoj digitalni sertifikat. Učesnik svoj privatni ključ drži u tajnosti, dok javni ključ učesnika je dostupan ostalim učesnicima okruženja.

Klient uz *RST* poruku može da priloži odgovarajući akreditiv ili da iskoristi digitalno potpisivanje svojim privatnim ključem kao mehanizam autentifikacije. *RST* zahtev se dodatno enkriptuje javnim ključem *IdP*-a. Na taj način samo *IdP* može da dekriptuje zahtev.

IdP po prijemu zahteva vrši njegovu dekripciju i potom vrši autentifikaciju korisnika. Korisnik može da pošalje različite akreditive, pa samim tim i mehanizam autentifikacije zavisi od tipa klijentskog akreditiva. Ukoliko je autentifikacija uspešna, *IdP* mora da identifikuje i *RP*

aplikaciju za koju je stigao zahtev. Ukoliko je *IdP* zadužen za izdavanje sigurnosnog tokena za datu *RP* aplikaciju, *IdP* kreira odgovarajući token sa potrebnim tvrdnjama i dodatnim *proof-key* ključem koji se takođe postavlja u okviru sigurnosnog tokena. Ovaj ključ je potreban za *holder-of-key* način potvrde na strani *RP*-a. Token se digitalno potpisuje sa privatnim ključem *IdP*-a i enkriptuje javnim ključem *RP*-a. Time čuvamo integritet i neporecivost tokena (jer je digitalno potpisan od strane *IdP*-a), kao i njegovu poverljivost (jer je enkriptovan sa *RP*-ovim javnim ključem i samo on može da ga dekriptuje). *IdP* kreira još jedan token koji se naziva *proof token* i on sadrži samo *proof-key* i enkriptovan je javnim ključem klijenta. *IdP* kreira *RSTR* koji sadrži odgovarajuće tokene i šalje ga klijentu.

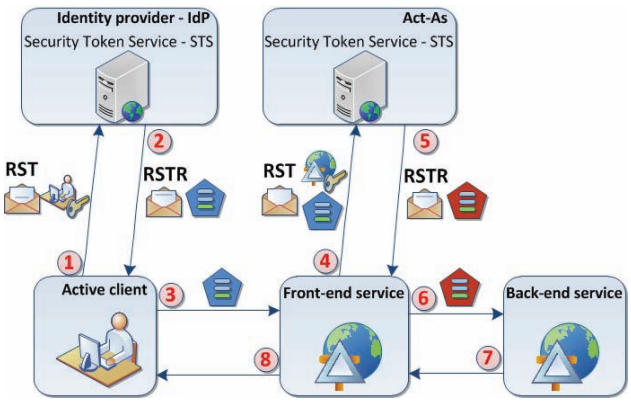
Od klijenta se očekuje ne samo da priključi odgovarajući sigurnosni token prilikom poziva *RP*-a, već i da dokaže da je on vlasnik tog sigurnosnog tokena kako bi se isključila mogućnost krađe sigurnosnog tokena. Ovaj način potvrde se naziva *holder-of-key* i posebno je efektan protiv *man-in-the-middle* napada.

Po prijemu *RSTR* klijent dekriptuje *proof token* kako bi mogao da iskoristi *proof-key*. Prilikom poziva *RP*-a, klijent dodaje sigurnosni token koji je namenjen *RP* aplikaciji i potom koristi *proof-key* kako bi izvršio digitalno potpisivanje ili enkripciju poruke (odnosno određenih delova poruke). *RP* po prijemu zahteva vrši proveru digitalnog potpisa tokena i vrši njegovu dekripciju. Na taj način dolazi i do *proof-key* iz tokena i vrši *holder-of-key* proveru da li je klijent iskoristio odgovarajući *proof-key*. Ukoliko je token bio ukraden prilikom transporta, *holder-of-key* verifikacija će biti neuspešna i *RP* bi trebalo da odbije dolazni zahtev.

Po završetku celokupnog verifikacionog procesa, *RP* aplikacija može da koristi sigurnosni token kako bi pristupila odgovarajućem identitetu, odnosno potrebnim tvrdnjama sadržanim u identitetu.

2.7 Delegacija identiteta

Delegacija identiteta se koristi u situacijama kada identitet klijenta, koji je započeo komunikaciju, propagiramo kroz više nivoa poziva između *web* servisa. Verzija 1.4 *WS-Trust* protokola definiše način na koji možemo da izvršimo delegaciju identiteta. Na slici 3. je prikazan tipičan primer delegacije identiteta.



Slika 3. *Delegacija identiteta*

Prva tri koraka su ista kao i kod scenarija aktivnih klijenata sa slike 2. Klijent vrši autentifikaciju nad *IdP*-om i dobavlja sigurnosni token sa svojim identitetom. Ideja je da *front-end web* servis (*FE*) ne pristupa *back-end web* servisu (*BE*) sa upotrebom svog identiteta, nego će izvršiti *act-as* poziv ka *BE* koristeći identitet aktivnog klijenta. Kako bi to ostvario, u koraku četiri *FE* kreira *act-as RST* zahtev koji sadrži *bootstrap* token klijenta, a takođe sadrži i token sa *FE* autentifikacionim akreditivom [1]. *Act-As STS* vrši obradu zahteva tako što prvo radi autentifikaciju naspram *FE* akreditiva. Ako je autentifikacija uspešna, izdaje se sigurnosni token koji će da sadrži delegacioni lanac identiteta. On će kao prvi identitet sadržati identitet aktivnog klijenta (jer se za njegov identitet i vrši delegacija identiteta), dok će ostali identiteti biti postavljeni kao sledeći članovi delegacionog lanca. U ovom slučaju identitet *FE* biće postavljen kao drugi identitet u nizu delegacionog lanca, a u slučaju da je postojalo još par nivoa poziva između *web* servisa, tada bi i identitet svakog servisa koji bi pravio *act-as RST* zahtev bio postavljen u delegacioni lanac. Sigurnosni token se *FE*-u šalje u okviru *RSTR* odgovora. U koraku broj šest *FE* poziva *BE* tako što u okviru poziva šalje *act-as* sigurnosni token. *BE* ne mora ni biti svestan činjenice da je izvršen *act-as* poziv. On vrši validaciju sigurnosnog tokena, i ako je validacija uspešna koristi prvi identitet iz delegacionog lanca (identitet klijenta u ovom primeru) kao da ga je taj prvi identitet direktno pozvao. *BE* na osnovu tvrdnji iz tog identiteta vrši autorizacione odluke po *CBAC* modelu kontrole pristupa. *BE* ima mogućnost pristupa i ostalim identitetima iz delegacionog lanca, ako mu je to iz nekog razloga potrebno.

Na slici 3. *IdP STS* i *Act-As STS* su prikazani kao odvojeni servisi. Međutim moguće je da jedan *STS* vrši obradu oba tipa zahteva što je i najčešći slučaj i tako je realizovan *STS* u okviru aplikativnog rešenja ovog rada.

3. OPIS REŠENJA PROBLEMA

Aplikativno rešenje rada je implementirano na *C#* programskom jeziku u *.NET* 4.0 *framework*-u. Korišćene su *Windows Communication Foundation* (*WCF*) i *Windows Identity Foundation* (*WIF*) biblioteke iz ovog *framework*-a. Programski alati korišćeni za implementaciju aplikativnog rešenja su *Microsoft Visual Studio 2010*, *Microsoft Internet Information Services 7.5*, *Microsoft SQL Server 2008 R2* i *Active Directory Lightweight Directory Services* (*AD LDS*).

Konkretna implementacija aplikativnog rešenja obuhvata razvoj projekta koji predstavlja specijalizovani *web* servis za izdavanje sigurnosnih tokena kao i odgovarajućih test klijenata i test *RP* aplikacija.

3.1 *WIF* model identiteta zasnovan na tvrdnjama

Za konkretnu implementaciju modela identiteta zasnovanog na tvrdnjama je izabrana implementacija modela u okviru *WIF* biblioteke. *WIF* definiše interfejs ***IClaimsPrincipal*** koji sadrži kolekciju svih ***IClaimsIdentity*** objekata koji su u vezi sa trenutnim sigurnosnim kontekstom. Takođe treba napomenuti da ovaj interfejs poseduje *property* ***Actor*** tipa ***IClaimsIdentity.*** Upravo preko ovog člana se omogućava definisanje delegacionog lanca. Interfejs ***IClaimsIdentity*** pre svega sadrži kolekciju ***Claim*** objekata koji pripadaju identitetu. Klasa ***Claim*** predstavlja tvrdnju sa svojim atributima poput ***ClaimType***, ***Value***, ***ValueType***, ***Issuer*** i ***Subject***.

3.2 Projekat *STS*

Najvažniji projekat rešenja je upravo projekat koji predstavlja implementaciju *STS*-a. Projekat se sastoji iz par logičkih celina, tako da postoji deo sistema koji se bavi primanjem zahteva za izdavanje sigurnosnih tokena i obradom tih zahteva, deo sistema koji se bavi problemima autentifikacije korisnika, i deo sistema koji služi za dobavljanje korisničkih tvrdnji kao i rukovanju odgovarajućim skladištima podataka koja se nalaze u okviru *STS*-a.

ClaimsIdentitySTS predstavlja klasu koja nasleđuje apstraktnu *WIF* klasu ***SecurityTokenService*** i *override*-uje potrebne metode. Ova klasa implementira konkretnu logiku za izdavanje sigurnosnih tokena. U okviru ***SecurityTokenService*** klase nalazi se metoda ***Issue***. Ova metoda kao jedan od parametara prima objekat tipa ***RequestSecurityToken***, a kao rezultat vraća objekat tipa ***RequestSecurityTokenResponse***. ***Issue*** metoda ustvari implementira *Issue binding* definisan *WS-Trust* protokolom. Podrazumevana implementacija procesira dolazni *RST* zahtev kroz *pipeline* za izdavanje tokena (*token issuance pipeline*). Podrazumevana implementacija *pipeline*-a se sastoji iz ukupno deset segmenata obrade zahteva, gde se svaki segment sastoji iz poziva odgovarajućih metoda za obradu *RST* zahteva. Samim tim u okviru implementacije rešenja izvršeno je redefinisanje metode ***GetOutputClaimsIdentity*** i metode ***GetScope***. ***GetScope*** metoda kreira odgovarajući objekat ***ClaimsBasedScope*** klase, dok ***GetOutputClaimsIdentity*** metoda kreira objekat ***IClaimsIdentity*** klase sa potrebnim tvrdnjama za identitet klijenta za koji se po *RST* zahtevu i kreira sigurnosni token. Na osnovu ovog identiteta u krajnjem segmentu *pipeline*-a biće kreiran odgovarajući format sigurnosnog tokena koji će biti poslat klientu u okviru *RSTR* poruke. ***GetOutputClaimsIdentity*** metoda je takođe zadužena za kreiranje odgovarajućeg delegacionog lanca identiteta, odnosno kreiranja odgovarajućeg *act-as* sigurnosnog tokena, ako je u pitanju *act-as RST* zahtev.

ClaimsManager predstavlja pomoćnu *manager* klasu koja omogućava dobavljanje klijentskih tvrdnji iz odgovarajućih skladišta podataka. Njena metoda ***GetClaimsForPrincipal*** se upravo poziva u okviru ***GetOutputClaimsIdentity*** kako bi se kreirao odgovarajući identitet sa potrebnim tvrdnjama. *STS* pruža podršku za korišćenje različitih skladišta podataka za skladištenje korisničkih tvrdnji i iz tog razloga svaka realizacija ***IClaimsProvider*** interfejsa dobavlja tvrdnje preko metode ***GetClaims*** iz odgovarajućeg skladišta. Konkretno rešenje rada poseduje dve implementacije :

- ***ActiveDirectoryClaimsProvider*** – dobavlja tvrdnje iz direktorijumskog servisa *AD LDS*
- ***SqlDatabaseClaimsProvider*** – dobavlja korisničke tvrdnje iz relacione baze podataka *SQL Server 2008*.

Svaki dolazni *RST* zahtev sadrži odgovarajuće klijentske akreditive naspram kojih mora da se izvrši autentifikacija. Za svaki tip klijentskih akreditiva potrebno je kreirati odgovarajući *handler*. *WIF* implementacija sadrži veći broj podrazumevanih *handler*-a za različite tipove sigurnosnih tokena. Međutim u većini slučajeva preporučljivo je kreirati sopstveni *handler* kako bi se omogućila proizvoljna logika. Recimo za *username/ password* tip autentifikacije kreiran je ***CustomUserNameSecurityTokenHandler*** *handler* koji obavlja proizvoljnu autentifikacionu logiku u okviru ***ValidateToken*** metode. U okviru ove metode biće pozvana ***Authenticate*** metoda pomoćne *manager* klase ***ClaimsIdentityAuthenticationManager***. Ova metoda vrši autentifikaciju klijenata time što koristi registrovane *provider*-e koji realizuju ***IAuthenticationProvider*** interfejs. Svaka ***IAuthenticationProvider*** implementacija omogućava autentifikaciju odgovarajućeg tipa sigurnosnog tokena naspram odgovarajućeg skladišta korisničkih naloga. Konkretno rešenje rada poseduje dve implementacije :

- ***UsernameSqlAuthenticationProvider*** – autentifikuje klijente za *username/password* tip autentifikacije naspram relacione baze podataka
- ***UsernameActiveDirectoryAuthenticationProvider*** – autentifikuje klijente za *username/password* tip autentifikacije naspram direktorijumskog servisa *AD LDS*.

4. ZAKLJUČAK

Zadatak rada je bio implementacija specijalizovanog *web* servisa za izdavanje sigurnosnih tokena. Cilj je bio da se kreiranjem ovog servisa izvrši eksternalizacija autentifikacije korisnika. Time smo dobili da *RP* aplikacije više ne moraju da vode računa o konkretnoj implementaciji logike za autentifikaciju korisnika i ne moraju da vode računa o rukovanju skladištima podataka za korisničke naloge i korisničke tvrdnje. Za model identiteta je izabran model identiteta zasnovan na tvrdnjama i time smo dobili da identitet korisnika *RP* aplikacijama dolazi uvek u istom, standardizovanom i jasno definisanom formatu, upotrebom standardizovanih formata sigurnosnih tokena. Posebna pažnja je bila posvećena realizaciji bezbednosti u komunikaciji između odgovarajućih učesnika. U okviru komunikacije između učesnika su obezbeđeni poverljivost, integritet i neporecivost poruka i sigurnosnih tokena.

5. LITERATURA

[1] *Vittorio Bertocci. Programming Windows Identity Foundation, 1st edition. Microsoft Press, 2010.*

[2] *Dominick Baier, Vittorio Bertocci, Keith Brown, Matias Woloski, Eugenio Pace. A Guide to Claims-Based Identity and Access Control, 1st edition. Microsoft Press, 2010.*

[3] *MSDN. Authorization in Claims-Aware Web Applications and Services. Microsoft Corporation. http://msdn.microsoft.com/en-us/library/gg185915.aspx*

[4] *OASIS. WS-Trust 1.4. OASIS Standard Specification, 2009. http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.pdf*

Kratka biografija:



Igor Javni rođen je u Novom Sadu 1986. godine. Fakultet tehničkih nauka upisao je 2005. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Primenjene računarske nauke i informatika odbranio je 2011. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

OPTIMIZACIJA POTROŠNJE ELEKTRIČNE ENERGIJE U STAMBENO - POSLOVNIM OBJEKTIMA

OPTIMIZATION OF ELECTRICAL ENERGY CONSUMPTION IN RESIDENTIAL - BUSINESS BUILDINGS

Goran Knežević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

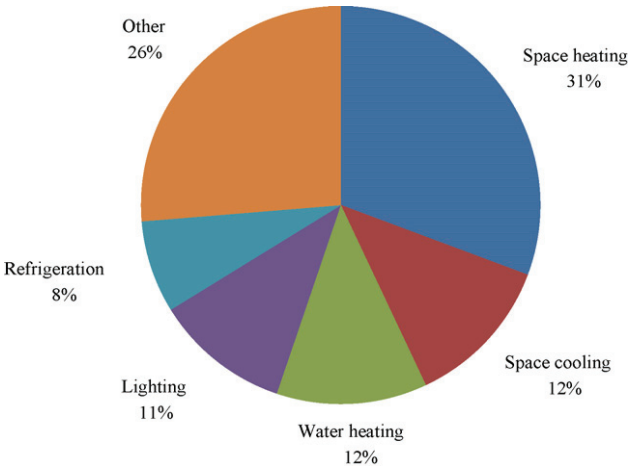
Kratak sadržaj –*Cilj rada je identifikacija uključenih potrošača u stambeno – poslovnim objektima. Identifikacija se vrši na osnovu prikupljenih podataka sa mernih uređaja (napon, struja, snaga i frekvencija). Na osnovu toga korisnik ima uvid u potrošenu električnu energiju pojedinih uređaja u određenom periodu. Za potrebe poslovnih objekata se svakom uređaju dodeljuje nivo prioriteta, na osnovu koga se vrši isključivanje uređaja niskog prioriteta u slučaju kada je potrošnja blizu određene granične potrošnje. Ograničavajući faktor za predstavljeni algoritam je postojanje više uređaja koji se neznatno razlikuju po potrošnji što može dovesti do pogrešne identifikacije.*

Abstract–*This paper presents an advanced algorithm for the identification of turned on electrical consumers in residential - office buildings. Identification is done based on data collected with a measuring device (voltage, current, power and frequency), which measures the total consumption of all observed electrical consumers. According to this, a user has access to the electricity consumption of individual devices in a given period based on a comprehensive measure. For the purposes of office buildings, a priority level is assigned to all devices, based on which the device of low priority is turned off in the case when the total consumption of the building is around a certain limit consumption prescribed by maximal power meter. The limiting factor for the presented algorithm is the existence of multiple devices that are slightly different in consumption which can lead to false identification.*

Ključne reči:*optimizacija potrošnje električne energije, detekcija električnih uređaja, prikaz potrošnje uređaja, genetski algoritam*

1. UVOD

Potrošnja električne energije u Srbiji je u 2010.godini iznosila 35.855 GWh, od proizvedenih 40.980 GWh, od čega se skoro 70% električne energije dobija eksploatacijom fosilnih goriva [1], [2].
Potrošnja električne energije u domaćinstvima čini više od 52% od ukupne potrošnje u Srbiji (52.54%). U okviru strategije EU 20/20/20 do 2020. godine (20% povećanje energetske efikasnosti, 20% učešće obnovljivih energija u energetskoj strukturi i smanjenje emisije štetnih gasova za 20%) jasno je stavljeno do znanja da članice EU i zemlje koje to žele da postanu moraju da se ozbiljno pozabave pitanjem energetske efikasnosti.
Postoji mnogo radova u kojima je obrađena tema energetske efikasnosti. S obzirom na to da se u domaćinstvima najviše potroši električne energije za grejanje i hlađenje prostorija (do 43%) najviše radova se bazira upravo na optimizaciji HVAC (*Heating, Ventilationand Air Conditioning* - HVAC)sistema [3,4] (slika 1).



Slika 1. *Prikaz potrošnje električne energije u stambenim objektima [5]*

U ovom radu je predstavljen jedan od načina smanjivanja potrošnje električne energije koji se zasniva na identifikovanju uključenih električnih potrošača u stambenim i poslovnim objektima i upravljanjem datim uređajima u cilju uštede električne energije.

2. OBRAČUNSKA SNAGA

Obračunska snaga ili vršna snaga je maksimalana izmerena srednja vrednost aktivne snage u vremenu od 15 minuta u toku meseca i meri se maksimetrom [6]. U praksi troškovi vršne snage su u proseku između 30% i 50% ukupnog računa za utrošenu električnu energiju, a može se desiti da cena koju platimo za obračunsku snagu premašuje iznos za utrošenu aktivnu snagu.Cena kilovata vršne snage je 620,52 din. za svaki kilovat (mereno na niskom naponu) do vrednosti ugovorene snage; a ako je potrošnja veća, iznos razlike od ugovorene do ostvarene snage naplaćuje se po 1241,04 din./kW. Opravdanje za ovako visoke cene, ED preduzeća vide iz razloga što mora da se ostvari stabilnost sistema i optimalni režim rada

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Velimir Čongradac, docent.

elektroenergetskog sistema a glavni način da se to ostvari je da kupci električne energije budu u okvirima ugovorene potrošnje. Iz napred navedenog jasno je da se mora analizirati mogućnost smanjenja troškova koje dobijamo na osnovu obračunske snage. Upravljanje vršnim opterećenjem se ostvaruje kontrolisanjem radnih režima najvećih potrošača. Cilj je izbeći istovremeni rad, tj.rasporediti periode punog opterećenja različitih potrošača tako da se ne poklapaju. U slučaju da ukupna potrošnja pređe određenu granicu, bilo bi pogodno isključiti potrošače niskog prioriteta da bi ostali u dozvoljenim granicama potrošnje.

3. KREIRANJE BAZE UREĐAJA

Za identifikaciju uključenih uređaja korišten je VMP-20 instrument za merenje napona, struje i snage koji se preko serijskog porta povezuje sa računarom. Instrument je razvijen na katedri za električna merenja na Fakultetu tehničkih nauka (slika 2).



Slika 2. *VMP-20 instrument*

Sve izmerene veličine se upisuju u bazu podataka (HSQLDB) i to upotrebom algoritma arhivske kompresije (*Archive Compression*). Arhivska kompresija nam omogućava da u bazu podataka upisujemo samo vrednost koja se u određenoj meri razlikuje od poslednje upisane vrednosti.Na taj način je drastično smanjen broj upisanih merenja u bazu podataka. Inače bi za jedan uređaj bilo potrebno 43 200 upisa dnevno, a ovom metodom je broj upisa smanjen do 80%. Princip rada algoritma se zasniva na linearnoj interpolaciji. Poslednja upisana tačka se pamti (*Archived point*). Sledeća izmerena vrednost se zove"zadržana tačka"(*Held point*) i na nju se dodaju i oduzimaju tačke ofseta. Nakon toga se kroz upisanu tačku povuku dve poluprave koje prolaze kroz ofset zadržane tačke. Ako se sledeća izmerena vrednost nalazi u prostoru između dve prave, ta nova tačka se proglašava za "zadržanu tačku" i zatim se računa ofset za nju. Ako se nova izmerena vrednost ne nalazi između dve tačke, ta tačka se upisuje u bazu podataka (slika 3).



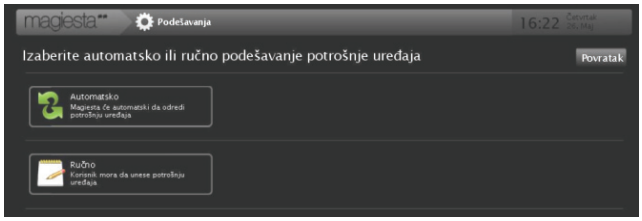
Slika 3. *Prikaz algoritma arhivske kompresije*

Kontrola uređaja, kao što su klima uređaji, bojleri, rasveta i sl., vrši se nekim od protokola za kućnu automatiku i Building management system - BMS (KNX, X10, LON - *Local Operating Network*). Program za komunikaciju sa VMP-om i uređajima kao i kompletna logika je urađen u Java programskom jeziku uz oslonac na Spring, Hibernate i Quartz framework.

Pre svega je potrebno odrediti skup uređaja čiju potrošnju želimo da kontrolišemo.Svakom uređaju se dodeljuje aktivna i reaktivna snaga koju koristi u režimu rada. To se može obaviti na jedan od dva načina:

1. ručnim unosom vrednosti aktivne snage u vatima [W] i reaktivne snage u volt-amperima reaktivnim [var]
2. automatskim određivanjem potrošnje uređaja primenom mernog instrumenta na osnovu razlike potrošnje električne energije u trenutku t0, kada uređaj nije bio uključen i potrošnje u trenutku t1 kada je uređaj uključen (pretpostavka je da se između t0 i t1 ne uključuje niti isključuje neki drugi uređaj)

Korisnik sam bira jedan od dva načina putem menija (slika 4).



Slika 4. *Biranje automatskog ili ručnog unosa potrošnje uređaja*

Svakom elementu iz skupa uređaja čiju potrošnju želimo da kontrolišemo dodeljujemo prioritet iz opsega od 0 do 10, gde 0 označava elemenat (uređaj) sa najmanjim prioritetom, a 10 uređaj sa najvećim prioritetom. Ideja je da se pri prelasku ukupne potrošnje objekta iznad unapred određene vrednosti isključe uređaji najmanjeg prioriteta (npr. klima uređaji, mašina za pranje suđa, bojleri i sl.), a sve u cilju smanjenja maksimalne srednje vrednosti koju meri maksigraf. Skup uređaja zajedno sa odgovarajućim potrošnjama i prioritetima uređaja se smeštaju u bazu podataka (HSQLDB, MySQL i sl.).

4. IDENTIFIKACIJA UKLJUČENIH UREĐAJA (GENETSKI ALGORITAM)

Za identifikaciju uređaja koji su uključeni primenjuje se genetski algoritam [7]. Problem je sličan problemu ranca (Knapsack) u kom je potrebno staviti što više predmeta određene vrednosti i dimenzije, ali tako da se ne prekorači kapacitet ranca [8]. U našem problemu umesto predmeta postoje uređaji koji imaju određenu potrošnju. Prvo trebamo učitati sve uređaje iz skupa uređaja koji se nalaze u bazi podataka i smestiti ih u niz objekata klase Uređaj.

Za prvu generaciju se na slučajan način formiraju hromozomi jedinki kao binarni niz (0 i 1), gde je dužina hromozoma, tj. binarnog niza jednaka broju uređaja u skupu uređaja. Vrednosti gena u hromozomu, odnosno vrednosti elemenata niza mogu biti 0, u slučaju da uređaj nije uključen ili 1 u slučaju da je uređaj uključen.

Fitnes funkcija ili funkcija pripadnosti se izračunava po

sledećoj formuli:

$$fitness_i = H - abs(C_i - M)$$

gde je

$fitness_i$– fitnes i-tog hromozoma

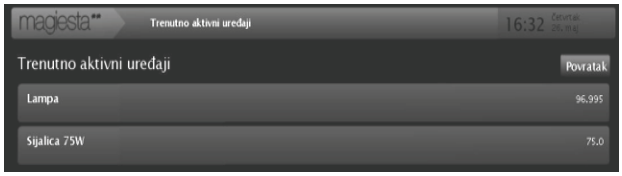
H – maksimalna moguća potrošnja svih uređaja iz skupa uređaja

C_i – potrošnja uključenih uređaja po i-tom hromozomu

M – trenutna izmerena potrošnja električne energije

Što je fitnes veći, to smo bliži našem rešenju (odnosno razlika između merene potrošnje i potrošnje po i-tom hromozomu je manja).

Selekcija jedinki se vrši rulet metodom, dok se ukrštanje vrši u jednoj tački koja se bira na slučajan način. Nakon toga postoji verovatnoća od 0.02 (2%) da će se desiti mutacija jednog od gena u hromozomu novonastalih jedinki, gde se mutacija definiše kao negacija bita. 4% najboljih jedinki se prenosi u sledeću generaciju i na taj način se čuvaju najbolji rezultati iz svake generacije (elitizam).
Genetski algoritam se izvršava svaki put kada se detektuje promena potrošnje veća od 5 W. Na taj način možemo da detektujemo uključenje ili isključenje svih uređaja čija je potrošnja veća od 5W, a sa druge strane smo otporni na šum merenja koji može izazvati lažnu dojavu o novoj aktivnosti uređaja (slika 5).

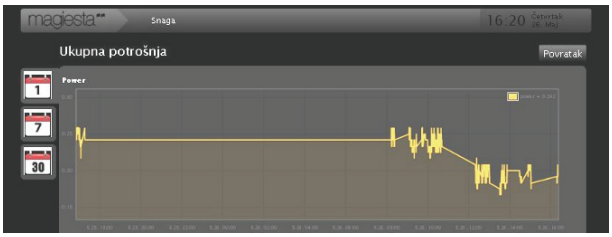


Slika 5. *Prikaz uključenih uređaja*

S obzirom na to da sada imamo informacije o tome koji su uređaji uključeni, koliko troše električne energije i koji prioritet imaju, sa lakoćom možemo isključivati uređaje nižeg prioriteta u slučaju da trenutna potrošnja pređe ugovorenu vrednost snage. Uređaji najvišeg prioriteta (8,9 i 10) se nikada ne isključuju. To su sigurnosni i protivpožarni uređaji, liftovi, rasveta i sl. Isključivanje i uključivanje uređaja je moguće izvršiti nekim od protokola za kućnu automatiku i BMS (X10, KNX, LON). Problem koji X10 protokol ima je taj da nakon što im pošaljemo komandu za uključenje ili isključenje ne možemo sa sigurnošću da kažemo da li je uređaj zaista uključen ili isključen. Navedeni problem je moguće rešiti na taj način što se nakon slanja signala za uključivanje ili isključivanje uređaja očekuje rast ili pad potrošnje u zavisnosti od toga koliko sam uređaj zahteva električne energije.

U domaćinstvima u kojima nemamo maksimetre, možemo zahteve modifikovati tako da potrošnja u skupoj tarifi ne prelazi određenu granicu koju je korisnik zadao. Uređaj koji je u višoj tarifi isključen radi uštede potrošnje bi se mogao ponovo uključiti u periodu niže tarife.

Takođe je napravljen prikaz ukupne potrošnje u objektu u toku dana, nedelje i meseca (slika 6), kao i mogućnost prikaza potrošnje električne energije po uređaju. Pored toga postoji i prikaz trenutno uključenih uređaja.



Slika 6. *Prikaz ukupne potrošnje objekta*

5. ZAKLJUČAK

Primenom prikazanih tehnologija u ovom radu mogu se postići velike uštede kako u poslovnim objektima neprekoračenjem ugovorene snage tako i u stambenim objektima isključivanjem pojedinih uređaja (veš mašine, mašine za pranje suđa i sl.) u višoj tarifi i ponovnim uključivanjem istih uređaja u nižoj tarifi. Glavni nedostatak prikazane metode je u tome što ne možemo sa sigurnošću odrediti koji je od dva ili više uređaja uključen u slučaju kada imaju istu potrošnju (npr. sijalica od 40W i lemilica koja troši isto toliko) ili slučaj kada je suma potrošnje više uređaja jednaka potrošnji nekog uređaja (sijalice od 40 i 60 vati zbirno imaju jednaku potrošnju kao sijalica od 100W). Još jedan faktor o kom se mora voditi računa je komfor korisnika objekta. Naime, malo ko će pristati na uštedu energije, ako mu se komfor boravka drastično smanji. Teško da će gost u hotelu biti zadovoljan ako mu se radi uštede isključi klima ili bojler za toplu vodu.

Dva glavna razloga zbog kojih mali broj ljudi vodi računa o energetskoj efikasnosti su nedostatak finansijskih sredstava u domaćinstvu za ulaganje u energetski efikasne tehnologije i nedostatak znanja o pojedinačnoj potrošnji uređaja i na koji način mogu tu potrošnju smanjiti ili raspodeliti između perioda niže i više tarife [9]. Zbog toga je napravljen prikaz ukupne potrošnje električne energije objekta kao i pojedinačan prikaz potrošnje po uređaju. Na taj način korisnici stambenog objekta imaju uvid u potrošnju

- električne energije u višoj i nižoj tarifi,
- pojedinačnih uređaja
- svih uređaja

na osnovu koje mogu da zaključe gde su mogli da uštede, koji uređaji im nepotrebno troše električnu energiju i na taj način bi im se polako podizala svest o tome kako da smanje račun za električnu energiju, a samim tim bi doprineli i očuvanju životne okoline.

6. LITERATURA

[1] – IEA Energy Statistics, 2010 - http://www.iea.org/stats/index.asp

[2] – EPS tehnički godišnjak 2010 - http://www.eps.rs/g_izvestaj.htm

[3] – Velimir Čongradac, Filip Kulić, "HVACsystem

optimization with CO2 concentration control using geneticalgorithms", Energy and Buildings, Volume 41, 2009 Pages 571–577

[4] – K.F. Fong, V.I. Hanby, T.T. Chow, "System optimization for HVAC energy management using the robust evolutionary algorithm", Applied Thermal Engineering, Volume 29, Issues 11-12, 2009, Pages 2327-2334

[5] – "Energy Efficiency and Renewable Energy, Energy Efficiency Trends in Residentialand Commercial Buildings", U.S. Department of Energy, Editor, 2008

[6] – http://www.energija.co.rs

[7] – Randy L. Haupt, Sue Ellen Haupt, "Practical Genetic Algorithms", Second Edition, 2004

[8] – Maya Hristakeva , Dipti Shrestha , "*Solving the 0-1 Knapsack Problem with Genetic Algorithms* ", 2003

[9] – Bruce Tonn, Jean H. Peretz, "State – level benefits of energy efficiency",Energy Policy, Volume 35, Issue 7, July 2007, Pages 3665-3674

Kratka biografija



Goran Knežević rođen je u Zadru 1987.god. Diplomski-bachelor rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2010. god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

KLASTEROVANJE KOMANDI X10 PROTOKOLA KUĆNE AUTOMATIKE PRIMENOM SUBTRACTIVE CLUSTERING ALGORITMA I NEUROFAZI STRUKTURE

DOMOTICS X10 PROTOCOL BASED CLUSTERING BY APPLYING SUBTRACTIVE CLUSTERING ALGORITHM AND NEUROFUZZY STRUCTURE

Daniel Tertei, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu objašnjeni su principi subtractive clustering algoritma, neurofazi strukture i X10 protokol. Razvijena je testna aplikacija koja klasteruje snimljene X10 komande po vremenskom kriterijumu i njima modeluje scenarije ponašanja ukućana obučavajući neurofazi strukturu. Pored modelovanja scenarija, neurofazi struktura koristi se i za proveru aktuelnosti klaster centara. Aplikacija je dala zadovoljavajuće rezultate.*

Abstract – *This paper presents the principles of the subtractive clustering algorithm, neurofuzzy structure and X10 protocol. On this basis a test application is built which clusters recorded X10 commands with the accent of their time dimension. Afterwards it uses them to model behavioral scenarios of householders by training a neurofuzzy structure. Besides modelling scenarios, the neurofuzzy structure is utilised to test whether cluster centers of relevant scenarios are up-to-date. The testing application has given satisfactory results.*

Ključne reči: *X10 protokol, subtractive clustering, neurofazi struktura*

1. UVOD

Izraz "domotika" odnosi se na principe automatskog upravljanja primenjene u stambenim objektima pomoću računarskih sistema: ventilacija, grejanje, klimatizacija, upravljački sistemi osvetljenja, prenos audio i video signala po kući, bezbednosni sistemi, teleasistencija, Interkom, kontrola kućnih robota, integracija namenskog hardvera idr. Kod nas se češće upotrebljavaju termini "inteligentne zgrade", "pametne kuće", ili "kućna automatika".

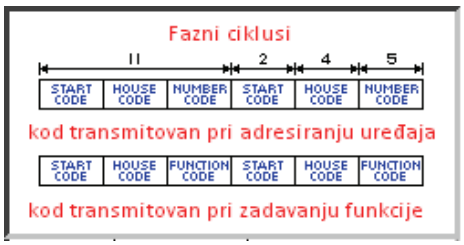
Takođe, domotika se posmatra kao integralni deo „Pametne mreže" [1] u smislu strateškog upravljanja potrošnjom električne energije u domaćinstvu. Cilj ovakvog energetskog menadžmenta je optimizacija potrošnje energije koja dolazi iz elektroprivrede. Totalno integrisani sistem kućne automatike podrazumeva energetski efikasan sistem. Na drugom mestu nalaze se bezbednost i komfor korisnika.

Komponente ovakvih sistema mogu se podeliti u tri kategorije: kontroleri, senzori i aktuatori. Međusobno se povezuju kablovima (bakarne parice, optički, koaksijalni), a moguća je i komunikacija bežičnim putem infracrvenim i radiotalasima. Sa stanovišta standarda i protokola, najzastupljeniji na tržištu i najstariji u svetu je X10 protokol, a zatim sledi složeniji KNX protokol. Ima i brojnih drugih, uglavnom nastalih kao pokušaj da se unapredi spori X10 protokol (20 bit/s) koji kao medijum za prenos signala koristi postojeću kućnu mrežu. Interoperabilnost među protokolima postiže se korišćenjem protokol konvertora, poput X10 u KNX, s tim da sami protokoli moraju biti kompatibilni.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Velimir Čongradac, doc.

2. X10 PROTOKOL

Digitalni podaci kodiraju se na nosilac frekvencije 120kHz koji se transmituje u kratkom vremenskom intervalu u momentu faznih prelaza naizmenične struje u kućnoj instalaciji. Prisustvo signala na frekvenciji 120kHz u trajanju 1msec označava logičko „1", dok njegovo odsustvo logičko „0". Startni kod, kod kuće (housecode) i kod uređaja (unitcode) predstavljaju adresiranje, a kada je poslednji u ovoj grupi kod komande (function code) – izdavanje komande uređaju kućne automatike. Ove tri grupe kodova predstavljaju X10 okvir komande (slika 1) i čine minimum komponenti validnog paketa X10 podataka [2]. Transmituju se, po pravilu, uvek dva puta.



Slika 1. *Adresiranje i izdavanje komande*

Kodovi kuće predstavljeni su engleskim slovima abecede od A do P, a uređaji brojevima od 1 do 16. Stoga postoji 256 mogućih načina adresiranja pojedinačnog uređaja.

Prenos podataka preko X10 protokola je spor zbog čega se ova tehnologija koristi samo za uključivanje i isključivanje uređaja, ili neke druge veoma proste operacije kao što je menjanje intenziteta osvetljenja. Komande protokola koje se koriste u ovom radu biće detaljno predstavljene u nastavku teksta.

Takođe, X10 standard podržava kreiranje tzv. makroa koji omogućavaju korisnicima da postave ceo spektar događaja koji bi se izvršavali uzastopno u određenom predefinisanom terminu. Makro je jedna komanda koja je isprogramirana da izvrši niz događaja (drugih komandi) kao što su scenariji.

3. SUBTRACTIVE CLUSTERING ALGORITAM

Posmatrajmo skup od *K* tačaka datih *m*-dimenzionalnim vektorom u_k , k = 1,2, ... , K. Lakše je posmatrati ovu matricu dimenzija *K* x *m* kao skup skaliranih komandi X10 protokola gde je *K* broj komandi, a *m* odgovara podacima koji opisuju datu komandu.
Algoritam subtractive clustering bazira se na nalaženju mere gustine tačaka skupa u oblasti od interesa – radijusa [3]. Mera gustine date tačke definiše se kao (1):

$$D_k = \sum_{j=1}^{K} \exp\left(- \frac{\left\| u_k - u_j \right\|}{\left(\frac{r_a}{2} \right)^2} \right) \tag{1}$$

gde je r_a pozitivna konstanta. Iz izraza se vidi da će tačka u_k imati veliku gustinu samo ako i u svojoj okolini ima mnogo tačaka. Okolina je određena konstantom r_a. Po računanju mere gustine svake tačke, izabiraju se prvi klaster centri. Neka je tačka u_{C1} sa svojom gustinom D_k jedan od klaster centara. Sledeći korak je osvežavanje mere gustine svih tačaka skupa prema izrazu (2) :

$$D_k' = D_k - D_{C_1} \exp\left(- \frac{\left\| u_k - u_{C_1} \right\|}{\left(\frac{r_b}{2} \right)^2} \right) \tag{2}$$

gde je r_b pozitivna konstanta. Izraz (2) smanjuje meru gustine svake tačke u blizini klaster centra, tako umanjujući šansu da okolne tačke budu izabrane kao klaster centri pri narednim iteracijama. Konstanta r_b definiše meru smanjenja gustine tačaka. Obično se uzima da bude veća od konstante r_a kako bi se izbegli bliski klaster centri, shodno (3):

$$r_b = 1.5 * r_a \tag{3}$$

Posle osvežavanja vrednosti mere gustine svake tačke, nalazi se sledeći klaster centar u_{c2} i ponovi se postupak osvežavanja mera gustina. Procedurom se analiziraju sve tačke skupa u_k i ona se ponavlja sve dok se ne nađe odgovarajući broj klaster centara ili ponestane novih kandidata. Algoritam tretira svaku tačku skupa kao potencijalni klaster-centar, što je velika pogodnost u svrsi izrade ovog master rada, kada se uzme u obzir da su tačke skupa u stvari skalirane komande X10 protokola. Time je onemogućeno da se za rezultate klasterovanja uzimaju nepostojeće komande.

4. NEUROFAZI STRUKTURA

Originalan naziv ove strukture je Adaptivni Neurofazi Sistem Zaključivanja – ANFIS[1]. Njena funkcionalnost zasnovana je na sposobnosti klasifikacije neuronskih mreža i rasplinutom[2] zaključivanju Sugeno tipa [4]. Grubo rečeno, ANFIS predstavlja metodu podešavanja inicijalnog skupa pravila pomoću hibridnog algoritma obuke. Inicijalni skup pravila modeluje se na osnovu podataka obučavajućeg skupa. ANFIS podržava isključivo sisteme Sugeno tipa, koji imaju sledeće karakteristike:

- Fazi funkcija je polinom nultog ili prvog reda;
- Da poseduje samo jedan izlaz koji je dobijen korišćenjem defazifikacione metode otežanog proseka;
- Sve izlazne funkcije pripadnosti su im istog tipa;
- Izlazne funkcije pripadnosti mogu biti ili linearne ili konstantne ;
- Težine svih fazi pravila su jednake i iznose 1 i nema deljenja pravila (različita pravila ne mogu deliti istu izlaznu funkciju pripadnosti, odnosno broj izlaznih funkcija pripadnosti mora biti jednak broju pravila).

4.1 Arhitektura ANFIS – a

Radi jednostavnosti prikaza, uzeto je da se ANFIS sastoji od samo dva fazi pravila određena Sugeno modelom prvog reda [5]:

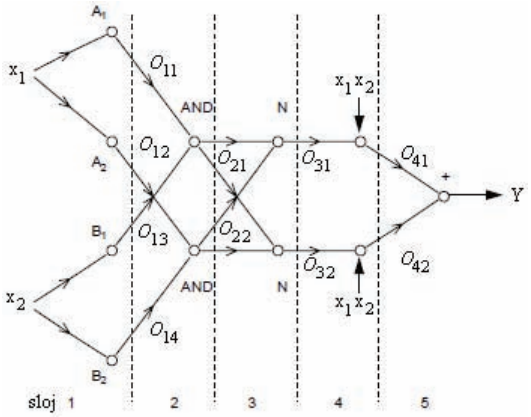
if x_1 *is* A_1 *and* X_2 *is* B_1 *then* $y_1 = c_{11}u_1 + c_{12}u_2 + c_{10}$ (4)

if x_1 *is* A_2 *and* x_2 *is* B_2 *then* $y_2 = c_{21}u_1 + c_{22}u_2 + c_{20}$ (5)

Neka su date jačine pravila α_1 i α_2 za prvo i drugo pravilo, respektivno. Tada je izlaz jednak (6):

$$y = \frac{\alpha_1 y_1 + \alpha_2 y_2}{\alpha_1 + \alpha_2} = \overline{\alpha}_1 y_1 + \overline{\alpha}_2 y_2 \tag{6}$$

ANFIS koji odgovara ovim vrednostima prikazan je na slici 2.



Slika 2. *Struktura ANFIS – a*

Svaki neuron u prvom sloju je adaptivni neuron sa parametarskom aktivacionom funkcijom. Njegov izlaz O_{1n}, $n=1,2,3,4$ predstavlja meru pripadnosti funkciji pripadnosti odgovarajućeg ulaza (x_1,x_2): $\mu_{A1}(x_1)$, $\mu_{B1}(x_2)$, $\mu_{A2}(x_1)$ ili $\mu_{B2}(x_2)$. Primer funkcije pripadnosti je Matlabova funkcija gaussmf.m određena izrazom (7):

$$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{7}$$

U master radu uzeta je funkcija gauss2mf.m kao funkcija pripadnosti ulaznih neurona. Parametri σ i c nazivaju se premisnim parametrima.
Čvorišta drugog sloja ne menjaju vrednost tokom obuke. Izlaz je proizvod ulaznih signala. Za računanje izlaza uopšteno se koristi logička funkcija minimuma – AND

[1] eng. Adaptive Neurofuzzy Inference System
[2] eng. fuzzy

funkcija. Svaki *i*-ti izlaz predstavlja jačinu pravila α_i koje pripada rasplinutom pravilu *i*.
Svaki čvor trećeg sloja predstavlja fiksni čvor čiji je izlaz normalizovana jačina *i* –tog pravila (8):

$$\overline{\alpha}_i = \frac{\alpha_i}{\alpha_1 + \alpha_2}, i = 1,2 \quad (8)$$

Svaki čvor (neuron) u četvrtom sloju je adaptivan sa izlazom (9):

$$\overline{\alpha}_i y_i = \overline{\alpha}_i (c_{i1} x_1 + c_{i2} x_2 + c_{i0}), i = 1,2 \quad (9)$$

gde je $\overline{\alpha}_i$ normalizovana jačina pravila iz trećeg sloja, a parametri $\{c_{i1}, c_{i2}, c_{i0}\}$ predstavljaju parametarski skup ovog čvorai nazivaju se posledičnim parametrima.

4.1 Obuka neurofazi strukture

Kada su premisni parametri poznati, izlaz *Y* računa se kao linearna kombinacija posledičnih parametara (10,11,12):

$$y = \frac{\alpha_1}{\alpha_1 + \alpha_2} y_1 + \frac{\alpha_2}{\alpha_1 + \alpha_2} y_2 \quad (10)$$

$$= \overline{\alpha}_1 (c_{11} x_1 + c_{12} x_2 + c_{10}) + \overline{\alpha}_2 (c_{21} x_1 + c_{22} x_2 + c_{20}) \quad (11)$$

$$= (\overline{\alpha}_1 x_1) c_{11} + (\overline{\alpha}_1 x_2) c_{12} + \overline{\alpha}_1 c_{10} + (\overline{\alpha}_2 x_2) c_{21} + (\overline{\alpha}_2 x_2) c_{22} + \overline{\alpha}_2 c_{20} \quad (12)$$

Hibridni algoritam obuke koriguje posledične parametre c_{ij}, i=1,2; j=0,1,2; pri prolazu unapred (u smeru strelica na slici 2), a premisne parametre σ_i i c_i ,i=1,2,3,4 pri prolasku unazad (u smeru suprotnom od smera strelica na slici 2). U prolasku unapred signal se propagira sve do četvrtog sloja gde se posledični parametri identifikuju po metodi najmanjih kvadrata. Pri prolasku unazad propagira se signal greške i osvežavaju se vrednosti premisnih parametara gradijentnom metodom. To se ponavlja dok se ne postigne kriterijum zaustavljanja određen dozvoljenom greškom obuke.

5. PROGRAMSKO REŠENJE

Komande se izdaju RF transmiterom Marmitek Easytouch 35, a kontroler Marmitek CM15Pro ih prima i prevodu u X10 komande koje potom šalje kroz mrežu. Personalni računar je povezan USB kablom sa kontrolerom i osluškuje poslate komande:

Tabela 1. *Marmitek X10 komande*

Bajt kod	Funkcija	Opis
5a 21 60 xxxxxxxx 5a 21 60 xxxxxxxx	All units off	Isključi sve uređaje sa naznačenim kućnim kodom
5a 21 61 xxxxxxxx 5a 21 61 xxxxxxxx	All units off	Upali sva svetla (sa sposobnošću podešavanja nivoa osvetljenja)
5a 2 0 6d xxxxxxxx 5a 2 1 62 xxxxxxxx 5a 2 0 6d xxxxxxxx 5a 2 1 62 xxxxxxxx	On	Uključi uređaj
5a 2 0 6d xxxxxxxx 5a 2 1 62 xxxxxxxx 5a 2 0 6d xxxxxxxx	Off	Isključi uređaj
5a 2 1 62 xxxxxxx		
5a 3 2 1f 64 65 xxxx ili 5a 2 0 62 xx f7 xxxx 5a 2 1 62 xx f7 xxxx 5a 2 0 62 xx 65 xxxx 5a 2 1 62 xx 65 xxxx 5a 3 2 b 64 65 xxxx	Dim	Smanji intenzitet svetla
5a 3 2 24 65 75 xxxx ili 5a 2 0 6a xx e7 xxxx 5a 2 1 62 xx e7 xxxx 5a 2 0 6a xx 75 xxxx 5a 2 1 62 xx 75 xxxx 5a 3 2 b 65 75 xxxx	Bright	Povećaj intenzitet svetla

5.1 Formiranje obučavajućeg skupa

Termin „Obučavajući skup" vezuje se za ANFIS strukturu. Elementi obučavajućeg skupa biće komande X10 protokola. Format elementa obučavajućeg skupa dat je tabelom:

Tabela 2. *Format X10 komande*

	Opseg
Sat*	0 – 24
Minut*	0 - 60
Sekund*	0 - 60
Kod kuće	1 - 16
Kod uređaja	1 - 16
Funkcija	1 – 8.99
Scenario	1 - N

*Sat, minut i sekund predstavljaju aktuelno vreme pristigle komande u toku dana. Kolona *Funkcija* iz tabele 2 dobija vrednosti prema tabeli 3.

Tabela 3. *Mapiranje kolone Funkcija X10 komande*

Funkcija	**Mapiranje**
All lights on	1
All units off	2
On	3
Off	4
Bright	5.xx*
Dim	8.xx*

Kod komandi *Bright* i *Dim* vrednost *.xx* predstavlja procenat pojačanja/zagušenja. Koloni *Scenario* iz tabele 2 programski se zadaju sledeće vrednosti:

Tabela 4. *Mapiranje kolone Scenario X10 komande*

Scenario	**Mapiranje**
scenario_1	1
scenario_2	2
scenario_3	3
...	...
scenario_n	n

Mapirani broj scenarija zavisi od broja postojećih (sačuvanih) scenarija. Aktuelni scenario uvek dobija vrednost $n = m + 1$, ako ima m postojećih.

5.2 Skaliranje obučavajućeg skupa

Vremenska dimenzija: sat, minut i sekund skalira se na jedan realan broj. To se radi nalaženjem ukupnog broja

proteklih sekundi od ponoći (00:00:00) i njegovim skaliranjem na opseg (0 2400). Takav opseg uzet je kompromisno u cilju što kvalitetnije kasnije obuke ANFIS – a. Kod kuće, kod uređaja, funkcije i scenarija skaliraju se množenjem njihovih vrednosti sa faktorom 100.

Tabela 5. *Format skalirane X10 komande*

	Skalirani opseg
Vreme	0 - 2400
Kod kuće	100 - 1600
Kod uređaja	100 - 1600
Funkcija	100 - 899
Scenario	100 – n*100

5.4 Primena subtractive clustering algoritma

Klaster centre predstavljaju periodične komande koje su se ponovile u vremenskom opsegu +-15min bar četiri puta u ukupnom vremenu snimanja. Perioda iznosi 24h.
Po završetku snimanja komandi, iz pristiglih podataka nalazi se opseg na osnovu koga se definiše radijus vektor za klasterovanje. Dobijeni klaster centri se dalje filtriraju po pravilu da se svi prvi klasteri prvo provere da stižu sa iste adrese (kod kuće i kod uređaja) i drugo da se periodične komande oko njih bar četiri puta ponove (zajedno sa njima). Klasteri koji zadovoljavaju ova dva uslova uzimaju se za konačne klaster centre.
Uzeto je da iz izraza (1) i (2) r_a iznosi 0.5, a r_b 0.15.

5.5 Osvežavanje pomoću neurofazi strukture

Neurofazi struktura u ovom radu ima trostruku primenu: obuka zarad modeliranja novog scenarija, poređenje novih klastera sa postojećim scenarijom i doučavanje postojećeg scenarija novim klasterima. Prvi omogućava konfiguraciju scenarija laičkim korisnicima aplikacije, drugi proveru aktuelnosti starijih scenarija, a treći osvežavanje novim, snimljenim klaster centrima postojeći scenario. Kriterijum aktuelnosti je koren srednjeg kvadrata greške odstupanja izlaza ANFIS – a.
Vektor struktura ANFIS – a je fiksiran: [10 3 10 3] i označava broj funkcija pripadnosti za svaki ulazni podatak obučavajućeg skupa – 10 za vreme, 3 za kod kuće, 10 za kod uređaja i 3 za funkciju. Ovaj vektor uzet je empirijski, iz razloga što su najkompleksniji elementi obučavajućeg skupa – kolone vremena i funkcije, pa je i broj funkcija pripadnosti veći -10, u odnosu na ostale gde je taj broj manji -3. Što je broj funkcija pripadnosti veći, kvalitetnija je i aprokimacija nelinearnih funkcija. Takođe, struktura sa ovakvog kapaciteta uspeva da se obuči i za skup od preko 200 klaster komandi. Tip funkcija pripadnosti ulaza je gauss2mf.m. Tip funkcija pripadnosti izlaza uzet je da bude konstantan ('constant'). Ostali podaci su isti kao u teorijskom poglavlju 4.1.
Potom genfis1.m inicijalizuje strukturu sa parametrima funkcija pripadnosti koji su relevantni na obučavajući skup podataka. Broj pravila se automatski generiše na osnovu zadanog vektora strukture i iznosi: 10 x 3 x 10 x 3 = 900 pravila. Svakom pravilu odgovara jedan izlaz, te ima 900 različitih funkcija pripadnosti izlaza.
Funkciji anfis.m se, pored obučavajućeg skupa i inicijalne ANFIS strukture, zadaje i broj epoha -7. Zbog pogodno izabrane strukture ANFIS – a za slučajeve obuke scenarijima i hibridnog algoritma, greška obuke veoma brzo konvergira ka nuli te je i broj od sedam epoha empirijski uzet, pošto je ponekad dovoljno za obuku i tri epohe.

6. TESTIRANJE APLIKACIJE

Snimljeno je 149 komandi u simuliranom vremenu u trajanju četiri dana sa opremom iz poglavlja 5. Posle klasterovanja nađeno je 34 komandi – klaster centara. Potom je kreirana neurofazi struktura koja obuhvata sve klaster centre pod nazivom "Nedelja" sa greškom obuke manjom od 0.001. Aktuelnost ovog scenarija testirana je poređenjem sa dve grupe novih klaster centara. Jedni su bili pomereni za 20min a drugi za 2h u odnosu na iste klaster centre "Nedelje". Pošto se postojeći klasteri smatraju aktuelnim ako nisu vremenski udaljeni za više od 30min, za prve nije bilo potrebno osvežavanje, a za druge jeste. Posle osvežavanja scenarija "Nedelja" novim klasterima iz druge grupe crtan je uporedni grafik odziva doučene ANFIS strukture za date komande pomoću Matlabove funkcije plot.m. Na njemu se jasno vide pomereni pikovi na relativnoj udaljenosti 2h koji odgovaraju pomeranju starih klaster centara u osveženoj neurofazi strukturi.

7. ZAKLJUČAK

Aproksimacija neprekidne, nelinearne funkcije specificirane preko ulazno/izlaznog skupa podataka je široko rasprostranjen problem [6]. Zbog brze konvergencije pri hibridnoj obuci i specifičnoj arhitekturi ovakav aparat veštačke inteligencije postiže veoma dobre rezultate, posebno kada se uzme u obzir stohastička priroda X10 komandi – bilo kada može da se adresira bilo koji uređaj. Prednost ovog softverskog rešenja leži i u nezavisnosti od sistema automatike koji se implementira: bezbednost, alarmi, osvetljenje, simulacija prisutnosti, komfor, energetska efikasnost i drugi, a bazira se na šest komandi X10 protokola.

8. LITERATURA

[1] http://en.wikipedia.org/wiki/Smart_grid
[2] Rose, B. (2001). Home networks: A standards perspective. *IEEE Communications Magazine,* No. 12, pages: 78-85.
[3] Chiu in Jang et al., 1997.
[4] Jang, Sun & Mizutani (1997); Jang & Sun, (1995).
[5] Fuller, R. (2000). *Introduction to Neuro-Fuzzy Systems*. Springer-Verlag, Berlin.
[6] Tettamanzi, A., Tomassini, M. (2001). *Soft computing*. Springer-Verlag, Berlin.

Kratka biografija:



Daniel Tertei rođen je u Subotici 1986. god. Diplomski-master rad na Fakul¬tetu tehničkih nauka iz oblasti Elektro¬teh¬nike i računarstva – Automatsko upravljanje odbranio je 2011.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.38

REALIZACIJA PID ALGORITMA U LabVIEW PROGRAMSKOM PAKETU

REALIZATION OF PID ALGORITH USING LabVIEW SOFTWARE

Aleksandra Vuković – Rukavina, Josif Tomić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu izložena je realizacija jednostavnog programa za PID regulaciju upotrebom programskog paketa LabVIEW. Program je baziran na primeni gotovih funkcija koje paket već sadrži.*

Abstract – *In this paper, realization of program for PID regulation using LabVIEW software is presented. Program is based on applying functions that software already contains.*

Ključne reči: *LabVIEW, PID algorithm, VI*

1. UVOD

Brzo prihvatanje PC računara u poslednjih 20 godina bilo je uslov za revoluciju u instrumentaciji za testiranje, merenje i automatizaciju. Virtualni instrument se sastoji od softvera i hardvera dodatih PC računaru opšte namene, koji zatim svi zajedno omogućavaju korisniku da kreira merne instrumente koji tačno odgovaraju njegovim potrebama, umesto da se ograniči tradicionalnim instrumentima sa fiksnim funkcijama [1].

Programski paket **LabVIEW**, proizvod kompanije **National Instruments**, koristi se za razvijanje sofisticiranih mernih, test i kontrolnih sistema. Baziran je na korišćenju PC računara i različitih vrsta merno-akvizicionih komponenti. **LabVIEW** (*Laboratory Virtual Instrumentation Engineering Workbench*) je grafički programski jezik koji za formiranje aplikacija koristi ikone umesto tekstualnog koda. Programi pravljeni u **LabVIEW-u**, nazivaju se virtualnim instrumentima (*VI*) [1], jer njihova pojava i operacije imitiraju fizičke instrumente kao što su, npr. osciloskopi i multimetri.

U današnje vreme, Proporcionalno - Integralno - Diferencijalni (PID) algoritam je najčešće korišćeni algoritam u industruji. Najčešće, **PID** se koristi za kontrolu procesa u koje spadaju sistemi za grejanje i hlađenje, praćenje nivoa tečnosti, kontrolu protoka i kontrolu pritiska. Pri upotrebi **PID** kontrole moraju se postaviti promenljive procesa koje se kontrolišu (temperatura, pritisak ili protok) i "asimtota" (vrednost koja se nastoji postići u sistemu). **PID** kontroler određuje izlaznu vrednost sistema koja nastoji da procesnu promenljivu sistema približi "asimptoti".

PID VI-i se mogu koristiti sa opremom **National Instruments-a** da bi se razvile **LabVIEW** kontrolne aplikacije.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je dr Josif Tomić, docent.

I/O oprema, kao **DAQ** uređaji ili **GPIB** kartica mogu se koristiti bi se PC povezao sa sistemom koji se nadgleda. Moguće je koristiti **I/O VI-e** koji već postoje u **LabVIEW** date sa **PID Toolkit-om** za razvoj kontrolnih aplikacija ili se mogu prepraviti primeri dati u **Toolkit-u**. Postoji mogućnost kombinacije ovih **PID VI-a** sa **LabVIEW** matematičkim i logičkim funkcijama da bi se kreirali blok dijagrami za realne kontrolne strategije.

2. PID ALGORITMI

PID kontroler upoređuje asimtotsku vrednost (*setpiont - SP*) sa procesnom promenljivom (*process variable - PV*) da bi se izračunala greška (*error - e*).

$$e = SP - PV \quad (2.1)$$

Izlazni signal *u(t)*, gde je $\boldsymbol{K_c}$ proporcionalno pojačanje kotrolera, se izračunava kao:

$$u(t) = K_c\left(e + \frac{1}{T_i}\int_0^t edt + T_d\frac{de}{dt}\right) \quad (2.2)$$

$\boldsymbol{T_i}$ je vreme integraljenja u minutima (*reset time),* a $\boldsymbol{T_d}$ je vreme diferenciranja u minutima (*rate time*).

2.1 Način rada PID VI-a

Formula (2.3) predstavlja trenutnu grešku:

$$e(k) = (SP - PV_f) \quad (2.3)$$

Proporcionalno dejstvo predstavlja proizvod pojačanja i greške, kao što je prikazano u formuli (2.4):

$$u_p(k) = (K_c * e(k)) \quad (2.4)$$

Trapezoidno integraljenje se koristi da bi se izbegla nagla promena u toku integraljenja ako ***PV*** ili ***SP*** naglo promene vrednost. Koristi se nelinarno podešavanje integraljenja da bi se predupredio premašaj, kao što je prikazano u formuli (2.5).

$$u_i(k) = \frac{K_c}{T_i}\sum_{i=1}^{k}\left[\frac{e(i)+e(i-1)}{2}\right]\Delta t \quad (2.5)$$

Zbog naglih promena ***SP***, diferenciranje se sprovodi samo nad ***PV***, umesto nad ***e***, da bi se izbegao prebačaj. Formla (2.6) predstavlja parcijalno diferenciranje:

$$u_d(t) = -K_c\frac{T_d}{\Delta t}\left(PV_f(k) - PV_f(k-1)\right) \quad (2.6)$$

Izlazni signal je zbir proporcionalnog, integralnog i diferencijalnog dejstva, kao što je dato u formuli (2.7):

$$u(k) = u_p(k) + u_i(k) + u_d(k) \quad (2.7)$$

Realni izlazni signal je ograničen u opsegu navedenom za kontrolni izlaz.

ako je $u(k) >= u_{max}$ onda $u(k)=u_{max}$

i

ako je $u(k) <= u_{min}$ onda $u(k)=u_{min}$

Formula (2.8) predstavlja najviše korišćeni model PID

kontrole:

$$u(k) = K_c\left[(SP - PV) + \frac{1}{T_i}\int_0^t (SP - PV)dt - T_d\frac{dPV_f}{dt}\right] \quad (2.8)$$

PID VI-i koriste korektivni algoritam integralne sume koji olakšava borbu protiv zasićenja i efikasano prebacivanje između ručne i automatske kontrole.

Opsezi za parametre ***SP***, ***PV*** i ***izlaz*** odgovaraju vrednostima u procentima, mada je moguće koristiti i prave vrednosti. Parametri T_i i T_d se zadaju u minutima. U ručnom modu, moguće je promeniti ulaz ručno da bi se promenio izlaz.

Pozivi ovih **PID VI-a** se mogu vršiti iz **While** petlji sa fiksnim vremenom ciklusa. Svi **PID VI-i** su rekurzivni. Višestruki pozivi iz **VI-a** višeg nivoa koriste odvojene i različite podatke.

Planiranje pojačanja se odnosi na sistem u kojem se menjaju parametri zasnovani na uslovima merenja. Unapred planirano pojačanje kontroliše sistem čija se dinamika menja sa uslovima rada.

Sa **PID VI-ima**, moguće je definisati neograničen skup **PID** parametara za raspored pojačanja. Za svaku situaciju, moguće je izvršiti samopodešavanje da bi se osvežili **PID** parametri

3. PID SOFTWARE

Pri izradi sistema kontrole, potrebno je nacrtati dijagram koji u sebe uključuje fizički proces i kontrolne elemente kao što su ventili i merna oprema. Potrebno je dodati povratnu spregu iz procesa i potrebna izračunavanja. Potom se mogu iskoristiti **VI-i** iz **PID Toolkit-a**, u kombinaciji sa matematičkim i logičkim funkcijama iz **LabVIEW-a**, da bi se planirani dijagram preveo u program.

Prema teoriji upravljanja, kontrolni sistem mora uzorkovati fizički proces brzinom 10 puta većom od najmanje vremenske konstante u fizičkom procesu. Brže uzorkovanje ne donosi poboljšanje performansi; prebrz rad svih **VI-a** degradira vremeski odziv **LabVIEW** aplikacije.

Svi **VI-i** unutar petlje izvršavaju se jednom po svakoj iteraciji sa istim vremenom ciklusa. Ako se želi rad nekoliko **VI-a** sa različitim vremenskim ciklusima ali da i dalje međusobno razmenjuju podatke, **VI-i** se moraju razdvojiti u vremenski nezavisne petlje.

Ako front panel ne sadrži grafike koje **LabVIEW** mora osvežavati, **PID Control VI-i** se mogu izvršavati na brzinama reda kHz. Takođe, akcije kao što su pomeranje miša i pomeranje ekrana uticaće na ove brzine.

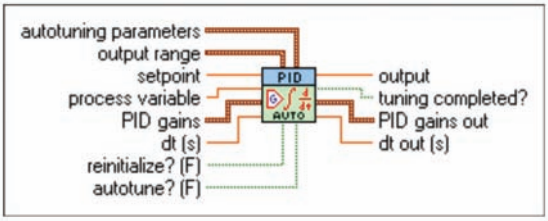
3.1 Funkcije PID Toolkit-a

Mogu se koristiti **PID VI-i** za razvoj sledećih aplikacija na osnovu **PID** kontrole:

- Proporcionalni (P); proporcionalno-integralni (PI), proporcionalno-diferencijalni (PD), proporcionalno-integralno-diferencijalni (PID) algoritmi
- PID sa planiranim pojačanjem
- Samopodešavajući PID
- Error-squared PID
- Lead-Lag kompenzacija
- Generisanje asimptote
- Kaskadna kontrola kroz više petlji
- Feedforward kontrola
- Kontrola premašaja
- Kontrola ratio/bias

3.2 Korišćenje PID Autotuning VI funkcije i Autotuning Wizard

PID Autotuning VI ima ulaz i izlaz koji je konzistentan sa drugim **PID VI-ima**, tako da je moguće zameniti bilo koji **PID VI** sa njim. Poseduje ulaz za **setpoint** (*asimtota*), **process variable** (*procesna promenljiva*), **PID gains** (*PID pojačanja*), **dt**, **output range** (*izlazni opseg*), i **reinitialize?** (*fuknicija reinicijalizacije*). Ulaz **PID gains** je klaster tri vrednosti – porporcionalno pojačanje, vreme integraljenja i vreme diferenciranja.



PID Autotuning VI funkcija

Output range definiše opseg izlaza kontrolera. Početni opseg izlaza je –100 do 100, što odgovara vrednostima datim u terminima procenata pune skale. Međutim moguće je podesiti da se pojačanja kontrolera izražavaju u inženjerskim jedinicama umesto procentima.

Moguće je iskoristiti **dt** za zadavanje vremena ciklusa kontrolne petlje. Početna vrednost je –1, tako da **PID Autotuning VI** koristi takt operativnog sistema za izračunavanja koja se tiču ciklusa petlje. Za determinističko vreme ciklusa, moguće je staviti ovaj ulaz u **PID Autotuning VI**. Takt operativnog sitema ima rezoluciju 1 ms, tako da je potrebno dati **dt** vrednost eksplicitno ako je ciklus kraći od 1 ms. Ako je **dt** pozitivna vrednost u sekundama, **VI** koristi tu vrednost u računanjima bez obzira na proteklo vreme. **National Instuments** preporučuje korišćenje ove metode za brze petlje, kao pri korišćenju hardvera za akviziciju za vremensko podešavanje kontrolera ili za real-time aplikacije

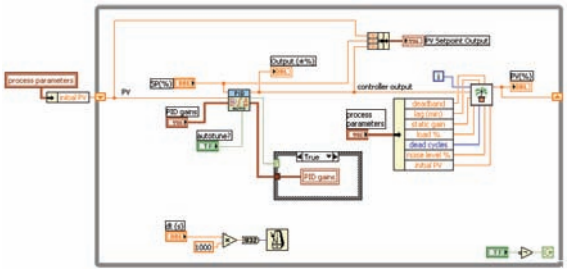
PID Autouning VI inicijalizuje interna stanja pri prvom pozivu **VI-a**. Svi dalji pozivi koriste stanja iz prethodnih poziva. TRUE na **reinitialize?** ulazu inicijalizuje početno stanje, u aplikacijama koje moraju stati i restartovati kontrolnu petlju bez restartovanja cele aplikacije.

Ulaz **autotuning parameters** (*parametri podešavanja*) je klaster parametara koje **VI** koristi za samopodešavanje procesa. **Autotuning Wizard** dozvoljava zadavanje ovih parametara ručno, tako da je moguće ostaviti ulaz **autotuning parameters** neožičen. Ulaz **autotune?** (*samopodešavanje*) uzima **Boolean** vrednost zadatu od strane korisnika. Ožičavanjem **Boolean** kontrole na front panelu aplikacije na ovaj izlaz omogućava da kada korisnik pritisne **Boolean** kontrolu, **Autotuning Wizard** se otvara automatski. **Autotuning Wizard** vodi korisnika kroz proces samopodešavanja.

Način praktičnog korišćenja biće objašnjen na primeru simulacije regulisanja temperature u stakleniku.

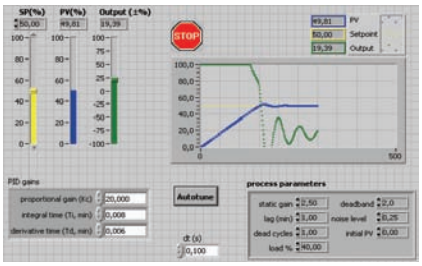
U **LabVIEW** kreirati novi prazan VI (*blank VI*), i na blok dijagram ubaciti **PID Autotuning VI-a** sa **PID** palete

funkcija. Kreirati kontrolu za vrednost asimptote temperature (*SP(%)*) i za vrednosti **PID** parametara (*PID gains* K_c, T_i, T_d). Kreirati indikator za upravljanje (*Output(%)*). Ožičiti odgovarajuće ulaze **PID VI-a**.



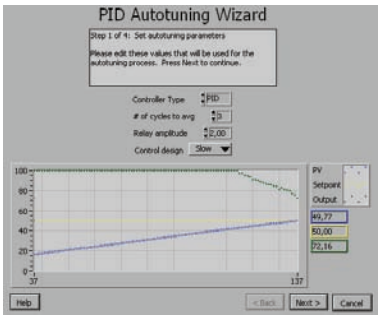
Slika 3.1. *Blok dijagram sistema*

Ubaciti **Plant Simulator VI**, koji simulira ponašanje biljke u stakleniku. Kreirati kontrole za unos promenljivih (*process parameters*) koje modeluju ponašanje biljke. Ožičiti odgovarajuće ulaze. Ožičiti izlaz **Plant Simulator VI-a** sa indikatorom za procesnu promenljivu (*PV(%)*). Izlaz **Output** iz **PID Autotuning VI-a** ožičiti sa ulazom **Manipulated Variable** na **Plant Simulator VI-u**.



Slika 3.2. *Front panel sistema*

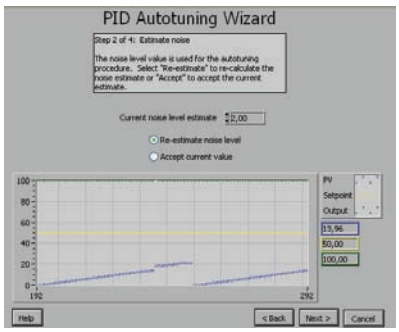
Ubaciti **Wait Until Next ms Multiple** i kreirati kontrolu za unos vremena **dt**, za kontrolu intervala između dva pozivanja petlje, odnosno brzina izvrašavanja programa. Kreirati **Waveform Chart** i ožičiti ga sa procesnom promenljivom, asimptotom i upravljanjem.
Okružiti dati kod sa **While** petljom i kreirati kontrolu za sopiranje petlje. Kreirati **Shift Register** na okviru petlje za procesnu promenljivu (*PV(%)*). Kreirati kontrolu za ulaz **autotune?** da bi se omogućilo pozivanje **Autotunog Wizard-a** i podešavanje. Kreirati **case** strukturu i njen **case selector** ožičiti sa **tuning completed?** izlazom sa **PID Autotuning VI-a**.



Slika 3.3 *Korak I za PID Autotuning Wizard*

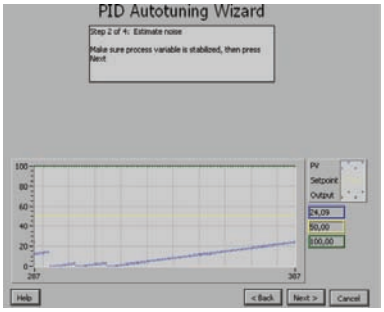
Unutar **True selector label-a** postaviti lokalnu promenljivu **PID gains** i ožičiti je sa izlazom **PID gains out** izlazom **PID Autotuning VI-a**. Klikom na **Autotune** poziva se **Autotuning Wizard** i prikazuje se dijalog za komunikaciju sa korisnikom kao na slici 3.3.

U polju **Controler Type** podesiti tip kontrolera. U polju **# cycles to average** podesiti broj ciklusa petlje za procenu. U polju **Relay amplitude** podesiti amplitudu za eksperiment podešavanja. U polju **Controle design** potrebno je odabrati brzinu dizajna**.** Kliknuti na polje **Next >**.



Slika 3.4 *Korak II: nivo šuma*

Odabrati ponovnu procenu šuma klikom na **re-estimate noise level** ili prihvatanje trenutne vrednosti **accept curent value**. Trenutnu vrednost moguće je upisati u polju **Current noise level estimate**. Kliknuti na polje **Next >** (slika 3.4).



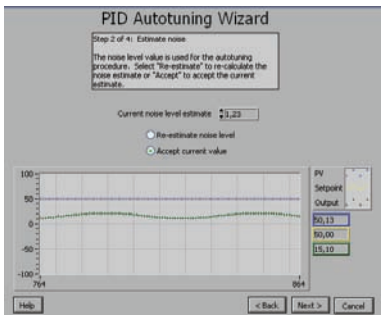
Slika 3.5 *Korak III: stabilizovanje*

Kada korisnik proceni da je sistem stabilizovan kliknuti na polje **Next >** (slika 3.5).



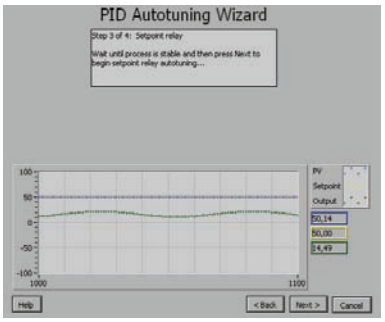
Slika 3.6 *Korak IV: procena šuma*

Autotuning Wizard vrši procenu šuma i korisnik čeka deset ciklusa petlje. Kliknuti na polje **Next >** (slika 3.6).
Kada je sistem izvršio procenu šuma i dao njenu novu vrednost. Korisnik je u mogućnosti da prihvati tu vrednost, da je sam izmeni ili da uputi sistem na ponovnu procenu. Takođe korisnik je u mogućnosti da se vrati na prethodne korake klikom na polje **< Back**, ili da odustane od procesa klikom na polje **Cancel**. Kliknuti na **Next >** (slika 3.7).
Korisnik se upućuje da sačeka da se proces stabilizuje i klikne na polje **Next >** (slika 3.8).



Slika 3.7 *Korak V: priprema podešavanja*

Korisnik se upućuje da sačeka da se proces stabilizuje i klikne na polje **Next >** da bi se počelo sa podešavanjem. Otvara se prozor **PID Autotuning Wizard** dijaloga kao na slici 3.9.



Slika 3.8 *Korak VI: početak podešavanja*

Korisnik se upućuje da sačeka da se procesna promenljiva stabilizuje i klikne na polje **OK**. Otvara se prozor kao na slici 3.10.

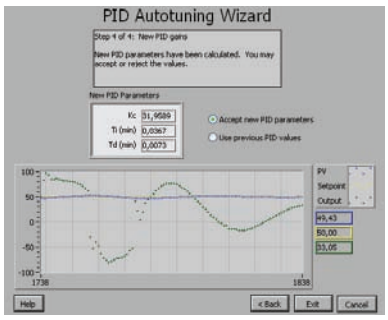


Slika 3.9 *Korak VII: podešavanje u toku*

Sačekati tri ciklusa petlje. Kliknuti na polje **Next >**. Otvara se prozor kao na slici 3.11. Nove **PID** parametre korisnik može prihvatiti sa **accept new PID parameters** ili odbaciti sa **use previous PID values**.

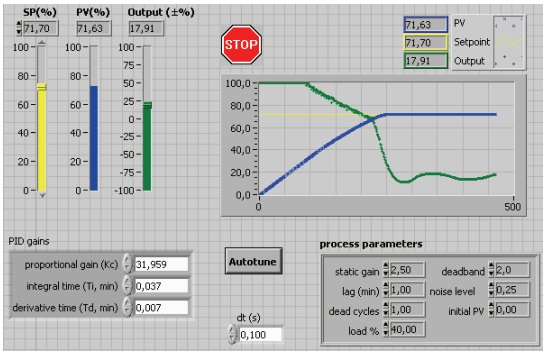


Slika 3.10 *Korak VIII: stabilizacija*



Slika 3.11 *Završni korak*

Klikom na **Exit** korisnik se vraća na front panel gde se proces na dalje izvršava.



Slika 3.11 *Front panel procesa sa novim parametrima*

Vidi se da je proces samopodešavanja pronašao automatski parametre koji obezbeđuju dovoljno brz odziv sa minimumom premašaja i malim oscilacijama u mirnom stanju.

4. ZAKLJUČAK

U ovom radu prikazana je simulacija sistema za **PID** regulaciju sa automatskim podešavanjem **PID** parametara, korišćenjem funkcija **PID Autotuning** i **PID Autotuning Wizard** koji su deo programskog paketa **LabVIEW**. Pokazano je kako na jednostavan način i uz minimalne izmene default podešavanja korišćenog softvera, može se kreirati jedan algoritam za efikasnu regulaciju.

5. LITERATURA

[1] Josif Tomić, Milan Milovanović, "Virtualna instrumentacija primenom LabVIEW programa", FTN Grafički centar GRID, Novi Sad , 2010.
[2] "PID User Manual", National Instruments Corporation, Austin, Texas, 2009.

Kratka biografija:

Aleksandra Vuković – Rukavina rođena 1987. god, Osnovne akademske - bečelor studije završila je na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Mikroračunarska elektronika na usmerenju za Primenjenu elektroniku u julu 2010. god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9

SOFTVER ZA EVIDENCIJU STIPENDISTA

SCHOLARS RECORD SOFTWARE

Nikola Potrebić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu predstavljena je implementacija softvera za evidenciju stipendista. Aplikacija je realizovana kao Web aplikacija korišćenjem Microsoft WCF i Silverlight tehnologije, uz oslonac na .NET 3.5 Framework. Kontrola pristupa u aplikaciji realizovana je korišćenjem ADAM direktorijumskog servisa. Za smeštaj podataka aplikacije korišćena je Oracle baza podataka.*

Abstract – *This paper describes implementation of scholars record software. Application is implemented as a Web application using Microsoft WCF and Silverlight technology, with reliance on .NET 3.5 Framework. Access control in application is implemented using ADAM directory service. Oracle Database is used for storing application data.*

Ključne reči: *Stipendisti, WCF, Silverlight, Web aplikacija, ADAM*

1. UVOD

Aplikacija za evidenciju stipendista (*STDB*) predstavlja softversku podršku za evidentiranje stipendista. Osnovna funkcionalnost softvera je skladištenje podataka o stipendistima, kao i prikaz i modifikacija unetih podataka. Iako je skladištenje stipendista osnovna funkcionalnost ovog softvera, ostavljena je i mogućnost skladištenja drugih tipova aplikanata (srednjoškolaca, diplomaca i ostalih).

Zadatak je realizovan kao *Web* aplikacija zbog portabilnosti, odnosno izbegavanja potrebe za instalacijom aplikacije kod korisnika, što je svojstvo *Web* tehnologije. Takođe je definisan mehanizam autentifikacije i autorizacije korisnika, radi zaštite pristupa aplikaciji. Podaci aplikacije se skladište u relacionoj bazi podataka kao najboljem obliku skladišta za podršku transakcionih zadataka.

Funkcionisanje aplikacije je realizovano na sledeći način:

- Server komponenta obrađuje zahteve korisnika i dostupna je korisnicima 24 sata dnevno.
- Korisnik pristupa aplikaciji iz *Web browsera*. Pošto se kominikacija između klijenta i servera obavlja preko mreže, mrežni saobraćaj je optimizovan , odnosno sveden na najmanji mogući nivo, bez narušavanja funkcionalnost aplikacije.
- Za autentifikaciju korisnika koristi se domenski mehanizam autentifikacije.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Miroslav Hajduković, red.prof.

2. TEHNOLOGIJE

Web aplikacijama se pristupa preko mreže, kao što su Internet i intranet. Veoma su popularne zbog široke prisutnosti *Web browsera* i njihove mogućnosti korišćenja kao klijenata. Ključ njihove popularnosti je mogućnost izmena i održavanja aplikacije bez potrebe za distribucijom i instaliranjem softvera na hiljade klijentskih računara, kao i multiplatformska kompatibilnost. Postoji veliki broj tehnologija za razvoj *Web* aplikacija, a neke od najznačajnijih su *ASP***,** *ASP***.***NET***,** *PHP*, *DHTML*, *Java*, *Flash* i *Silverlight*. *Microsoft .NET framework* predstavlja moćan okvir za razvoj *Web* aplikacija, jer sadrži veliku kolekciju resursa za razvoj softvera i podržava nekoliko programskih jezika.

2.1. *Oracle Database*

Oracle Database (*RDBMS*) je objektno-relacioni sistem za upravljanje bazom podataka koga je razvila kompanija *Oracle*. Sistem logički skladišti podatke u obliku tabela (*tablespace*), a fizički u obliku fajlova sa podacima [1].

Objekti baze podataka se koriste za skladištenje i organizaciju podataka kojima korisnička aplikacija barata. Ti objekti uključuju tabele, indekse, poglede, sekvence i sinonime. Nakon kreiranja nekog objekta, potrebno je specificirati i tip tog objekta. Na primer, kada se kreira tabela, potrebno je definisati tip podataka svake njene kolone. Najčešće korišćeni tipovi podataka su tekstualni znakovi, brojevi i datumi.

U opisanoj aplikaciji korišćen je *Oracle Database* 10 XE (*Express Edition*), koji predstavlja besplatnu distribuciju *Oracle* sistema na *Windows* i *Linux* platformama.

Oracle Database XE poseduje *browser*-orijentisan korisnički alat za administraciju objekata baze podataka, izvršavanje *SQL* naredbi, *PL/SQL* koda i izgradnju *Web* orijentisanih aplikacija. Ono što je mnogo pogodnije koristiti za ovu svrhu, i što je ujedno i korišćeno, je *Oracle SQL Developer* alat, koji predstavlja grafički orijentisan alat za kreiranje i izmenu šeme baze podataka, kao i za manipulaciju podacima u bazi.

2.2. *NHibernate*

NHiberante obezbeđuje objektno-relaciono mapiranje za *Microsoft .NET* platformu [2]. Njegov zadatak je da oslobodi osobe, koje rade na razvoju softvera, značajnog dela zadataka vezanih za rad sa relacionim podacima. *NHibernate* je besplatan alat i nastao je prilagođavanjem *Hibernate* alata **.***NET* okruženju.

Primarna funkcija *NHibernate-a* je mapiranje *.NET* klasa na tabele baze podataka. Pored toga, on takođe obezbeđuje i postavljanje upita nad podacima i preuzimanje podataka.

NHiberante maping fajl je *XML* fajl koji sadrži metapodatke potrebne za objektno-relaciono mapiranje. Metapodaci uključuju deklaraciju klasa za perzistiranje i mapiranja *property*-ja. Naziv *maping* fajla je u formatu naziv.*hbm.xml*. Svaka klasa za perzistiranje mora da sadrži identifikator. Ovaj *property* se koristi za razlikovanje perzistiranih objekata.

2.3. *Windows Communication Foundation*

Windows Communication Foundation (*WCF*) je *Microsoft*-ovo objedinjeno okruženje za izgradnju pouzdanih, sigurnih, transakcionih i interoperabilnih distribuiranih aplikacija [3]. Korišćenjem ovog alata, servis postaje dostupan spoljnim pozivima različitih tehnologija.

WCF distribuirani sistem se gradi kreiranjem tri povezana sklopa:

· *WCF Service assembly* – sadrži klase i interfejse koji predstavljaju funkcionalnosti koje se otkrivaju spoljnim pozivima. Sadrži *WCF* ugovore (*contracts*) i njihovu implementaciju.

· *WCF Servise host* – celina koja uslužuje *Service assembly*.

· *WCF Client* – aplikacija koja pristupa funkcionalnostima servisa.

Osnovni blokovi *WCF* servisa su *Address* – lokacija servisa, *Binding* – način povezivanja klijenta i servisa i *Contract* – opis svake metode koja se otkriva od strane servisa.

2.4. *Silverlight*

Microsoft Silverlight je aplikativno okruženje za pisanje i pokretanje sadržajno bogatih Internet aplikacija, sa naglaskom na multimediju, animaciju i grafiku [4]. Po karakteristikama i nameni sličan je *Adobe Flash* tehnologiji. Izvršno okruženje za *Silverlight* je dostupno većini *Web browsera* kao *plugin*. Trenutna verzija *Silverlight*-a je 4 koja je objavljena u aprilu 2010. Kompatibilan je sa većinom *Web browsera* koji rade na *Microsoft Windows* i *Mac OS* operativnim sistemima.

U *Silverlight* aplikacijama, korisnički interfejs se definiše korišćenjem *Extensible Application Markup Language* (*XAML*) jezika i programira korišćenjem podseta *.NET* razvojnog okruženja. *XAML* se može koristiti za kreiranje vektorske grafike i animacija.

Komunikacija između *Silverlight-a* i *WCF* servisa obavlja se na asinhroni način. Na ovaj način je postignuto da poziv jedne metode ne blokira dalje izvršavanje klijentske aplikacije, već aplikacija nastavlja da obavlja koristan posao dok je poziv metode u toku.

2.5. *Active Directory Application Mode*

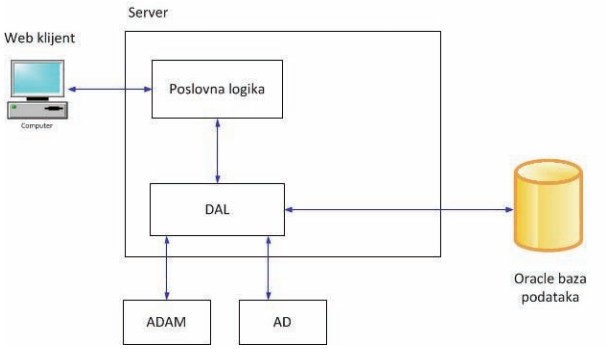
Active Directory Application Mode (*ADAM*) se može opisati kao jedna nova vrsta *Active Directory (AD)* servisa koji je specijalno dizajniran za direktorijumski orijentisane aplikacije, skraćeno *DEA (Directory-enabled Application)* [5].

ADAM je servis koji je zasnovan na *LDAP* protokolu i koristi se kao korisnički servis, za razliku od *AD* servisa koji se koristi kao sistemski servis. *ADAM* servis se može pokrenuti na serverskim računarima i domenskim kontrolerima koji koriste operativne sisteme iz *Windows Server* familije, kao i na klijentskim računarima sa *Windows XP* operativnim sistemima.

3. IMPLEMENTACIJA

3.1. Arhitektura aplikacije

Na slici 1. prikazana je arhitektura *STDB* aplikacije.



Slika 1. *Arhitektura STDB aplikacije*

Korisnički podaci se smeštaju u *Oracle* bazu podataka. Za potrebe aplikacije kreiran je *Oracle user* i u njega učitana šema baze podataka.

U okviru servera realizovana je određena poslovna logika i data mu je dozvola da pristupa podacima smeštenim u bazi podataka, kao i da u nju upisuje korisničke podatke. Takođe, serverski sloj je zadužen za kontrolu pristupa podacima od strane korisnika. Autentifikacija i autorizacija korisnika odvija se u posebnom sloju unutar servera koji se naziva *DAL* (*Data Access Layer*).

ADAM direktorijumski servis koristi se kao skladište sigurnosnih podataka za potrebe autorizacije. On komunicira sa *DAL* komponentom, odnosno *DAL* komponenta od njega zahteva podatke o pravima pristupa korisnika zadatoj koloni baze podataka.

AD je domenski direktorijumski servis koji skladišti podatke o korisnicima sistema u okviru poslovne organizacije. *DAL* komponenta mu pristupa prilikom autentifikacije korisnika koji pristupa sistemu.

Klijentski sloj se sastoji od *Web* aplikacije izrađene u *Silverlight* tehnologiji, koja korisnicima pruža interfejs za korišćenje softvera.

3.2. Upravljanje podacima

Oracle DBMS predstavlja sistem za rad sa relacionom bazom podataka. Međutim, relacioni koncept podataka nije toliko blizak programerima i projektantima softvera, koji rade sa isključivo objektnom reprezentacijom podataka. Iz ovog razloga potrebno je kreirati mehanizam za upis objektnih podataka softvera u relacionu bazu podataka, kao i dohvatanje podataka iz nje. *NHibernate* predstavlja zgodan alat koji omogućava objektno-relacionu konverziju i koji je i korišćen u ovom radu.

Rad u *NHiberante*-u podrazumeva definisanje entiteta i pisanje *xml mapping* fajlova za mapiranje između entiteta i kolona baze podataka. U ovom projektu entiteti i *XML* mapiranja nalaze se u paketu *DataTransfer*.

Kominikacija sa bazom podataka odvija se u sesijama (*Session*). Sesije se kreiraju pomoću fabrike sesija (*SessionFactory*) koja predstavlja reprezentaciju objektno-relacionog mapiranja u memoriji. Pošto je kreiranje fabrike sesija procesorski i vremenski dosta zahtevno, fabrika sesija se kreira samo jednom na početku rada i kasnije se koristi samo njena instanca. To je implementirano pravljenjem jedne *singleton* klase

FactoryClass koja sadrži referencu na fabriku sesija, a koja se poziva na početku izvršavanja aplikacije.
Kada zahtevamo neki podatak iz baze podataka, kreiramo pomoću fabrike sesija novi sesiju i od nje zahtevamo podatke.
Ukoliko radimo upis ili izmenu podataka u bazi, moramo da radimo u transakcionom režimu. To se radi tako što pre operacije upisa ili izmene iniciramo započinjanje transakcije, a po završetku operacije završavamo i transakciju. Po zatvaranju sesije, *NHiberante* će automatski ažurirati bazu podataka.

3.3. Server

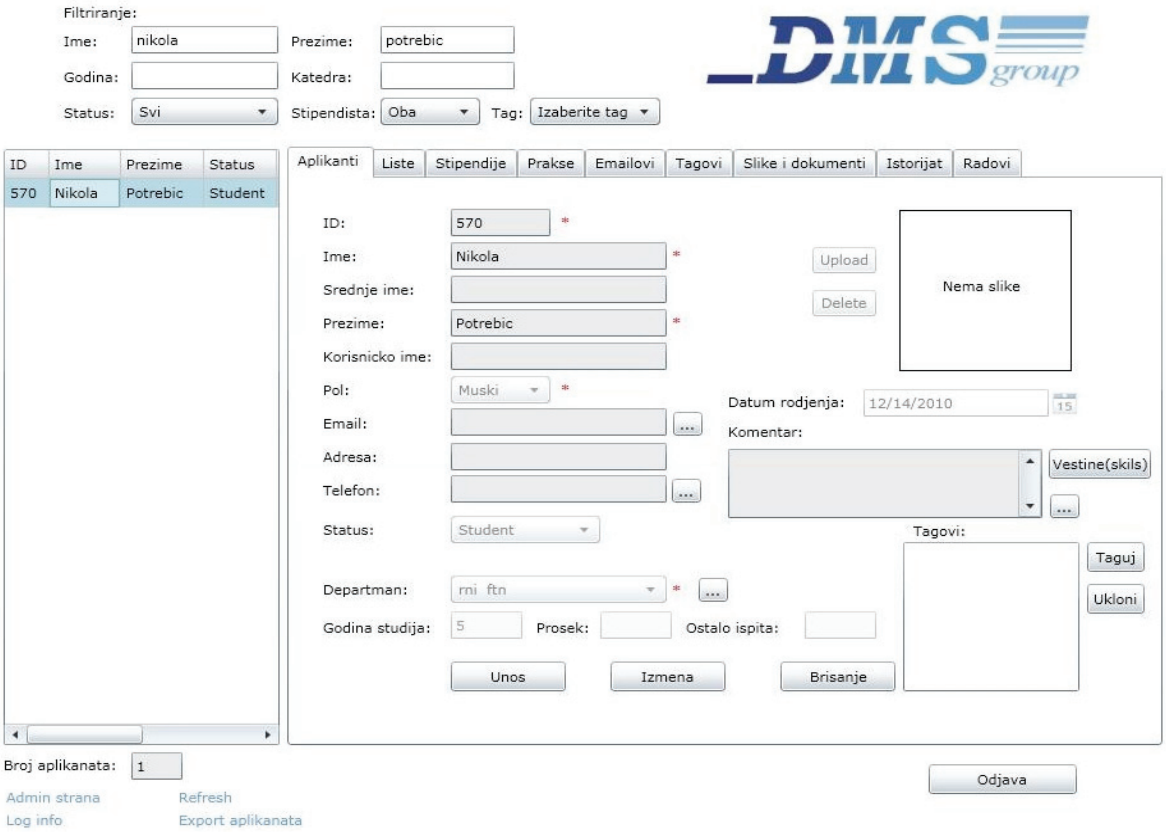
Serverska aplikacija je kreirana kao *WCF* servis. Osnovni zadatak joj je pružanje usluga klijentima pozivom metoda servisa.
Da bi metodama servisa mogli da pristupamo preko mreže, interfejs je označen atributom *ServiceContract*, a svaka metoda interfejsa atributom *OperationContract*. Takođe, sve klase čiji objekti se prenose preko mreže, označeni su atributom *DataContract*, a *property*-ji atributom *DataMember*, što omogućava njihovu uspešnu serijalizaciju.
Metode servera mogu se podeliti u nekoliko osnovnih grupa:
* Metode za pristup podacima – to su setovi metoda za čitanje, upis, modifikaciju i brisanje podataka iz baze podataka. U ovu grupu spadaju i metode za učitavanje slika i dokumenata sa fajl sistema korisnika, kao i njihov *download* sa servera na fajl sistem korisnika.
* Metode za eksport podataka i generisanje izveštaja – podaci se mogu eksportovatu u običan HTML ili *Microsoft Excel* format. Izveštaji se generišu za liste stipendista. Za svaki izveštaj postoji i mogućnost formiranje sumarnog izveštaja.
* Administratorske metode – implementiraju postupak autentifikacije i autorizacije i omogućavaju administriranje liste korisnika aplikacije i dodelu prava pristupa grupama korisnika.

3.3.1. Prava pristupa

Postupak autorizacije korisnika aplikacije implementiran je u okviru klase *SecurityDll*. Samo autorizovani korisnici imaju pravo pristupa aplikaciji.
Postupak autorizacije obavlja se odmah nakon postupka autentifikacije, prilikom prijave korisnika na sistem. Postupak autorizacije korisnika se obavlja korišćenjem *ADAM* direktorijumskog servisa.
Prava pristupa aplikaciji definisana su na nivou kolona baze podataka, što predstavlja najveći mogući nivo granularnosti.
Da bi zaštitili podatke u *NHiberante* entitetima od neovlaščenog pristupa, za svaki entitet je kreirana *wrapper* klasa. *Wrapper* klasa sadrži sve atribute koje ima i *NHibernate* entitet i pri svakom pokušaju pristupa atributu poziva metodu *isGrant SecurityDll* klase koja proverava da li je pristup dozvoljen.
Ukoliko je dozvoljen, vraća korisniku traženu vrednost, a ukoliko nije vraća *null* ili neku drugu predefinisanu vrednost za *ne-null* tipove podataka.
Na ovaj način smo postigli da vrednost atributa mogu da očitaju samo oni koji su za to autorizovani.

3.4. Klijent

Klijentska aplikacije predstavlja *Web* klijenta izrađenog u *Silverlight* tehnologiji. Osnovni zadatak klijenta je da omogući korisnicima svu funkcionalnost koja je definisana u specifikaciji zahteva, a implementirana na strani servisa. Da bi mogla da pristupa *WCF* servisu, odnosno njegovim metodama, klijentska aplikacija sadrži referencu na *WCF* servis. Sva komunikacija sa servisom se obavlja asinhrono.
Web aplikacija se sastoji od stranica. Svaka stranica se sastoji od *.xaml* i *.xaml.cs* fajla. *.xaml* fajl služi za definisanje i raspored komponenti interfejsa na stranici, a *.xaml.cs* fajl sadrži pozadinski kod stranice pomoću kog se upravlja radom aplikacije.
Login.xaml je prva stranica koju korisnik vidi kada adresira ovu aplikaciju, dok je *MainWorker.xaml* glavna stranica aplikacije.
Na slici 2. prikazan je izgled *MainWorker.xaml* stranice.
Korisnik preko l*ogin* stranice unosi svoje kredencijale i šalje ih na proveru. Server kredencijale prosleđuje *AD* servisu koji je zadužen za proces autentifikacije. Ako je autentifikacija uspešna, korisnik se preusmerava na stranicu *MainWorker.xaml,* koja je osnovna stranica aplikacije. U suprotnom, korisniku se ispisuje poruka o neuspešnom logovanju i zadržava se na *login* stranici.
Na glavnoj stranici se jasno mogu uočiti nekoliko regiona:
· Gornji deo stranice rezervisan je za filtriranje stipendista po raznim kriterijumima.
· Levi region je namenjen za prikaz liste stipendista, odnosno njihovih osnovnih podataka.
· Srednji i desni region su spojeni i namenjeni za obavljanje glavnih funkcija aplikacije.
· Donji deo je ostavljen za neke dodatne funkcije.
Funkcionalne celine aplikacije su raspoređene u devet tabova koji se na stranici nalaze u srednjem i desnom regionu:
1. Aplikanti - rad sa stipendistima, prikaz i unos njihovih ličnih podataka, podataka o mestu stanovanja, fotografija, statusa i veština koje poseduju. Stipendista može da ima jedan od četiri predefinisana statusa: student, diplomac, srednjoškolac i ostali. U zavisnosti od statusa, prikazuju se i unose i dodatni podaci. Takođe je u ovom tabu omogućeno i tagovanje stipendiste, kao i uklanjanje taga.
2. Liste - rad sa listama stipendista. Omogućava korisnicima da kreiraju listu stipendista po nekom kriterijumu i da u određenim kontekstima posmatraju samo odabrane stipendiste.
3. Stipendije - prikaz, unošenje, izmena i brisanje stipendija.
4. Prakse - prikaz praksi koje je stipendista obavio u firmi.
5. *Email*-ovi - spisak *email* poruka koje je poslao stipendista.
6. Tagovi - rad sa tag-listama stipendista. Pošto svaki tag može da pripada nekog grupi tagova, tagovi i grupe tagova se prikazuju pomoću hijerarhijske strukture u obliku drveta.
7. Slike i dokumenti - rad sa slikama i dokumentima stipendista. Svaki stipendista može da ima proizvoljan broj svojih slika i dokumenata. Omogućen je *download* slika i dokumenata sa servera na fajl sistem klijenta.



Slika 2. *Izgled stranice MainWorker.xaml*

8. Istorijat - obezbeđuje evidenciju o toku školovanja stipendiste. Evidencija obuhvata dosadašnji prosek u studiranju i broj preostalih ispita.
9. Radovi - evidencija završnih radova stipendiste. Pod radovima se smatraju diplomski, master, doktorski i eventualno neki drugi tip radova.

Tabela za prikaz stipendista se popunjava prilikom inicijalizacije stranice, dok se svi ostali podaci dobavljaju na zahtev i zatim se popunjavaju polja za prikaz. To je realizovano na ovaj način da bi se postigle dobre performanse aplikacije, sa obzirom da se podaci prenose preko mreže.

4. ZAKLJUČAK

U radu je opisan softver za evidenciju stipendista realizovan kao *Web* aplikacija. To je urađeno zbog prednosti kod održavanja i modifikacije *Web* aplikacija. Bilo kakva izmena funkcionalnosti ne zahteva bilo kakvu intervenciju na korisničkoj strani. Međutim, to nosi sa sobom i određene nedostatke. Aplikaciji se pristupa preko javne mreže, što može da dovede do narušavanja bezbednosti aplikacije. Zlonamerni korisnici mogli bi raznim tehnikama da pokušaju da dobiju pristup aplikaciji ili da dođu do podataka koji se čuvaju u bazi podataka, a koji su svakako poverljivi. Takođe, imamo povećan mrežni saobraćaj, što znači da će performanse aplikacije zavisiti i od propusnosti mreže. Realizacija rešenja kao klasične *Widnows* aplikacije prevazišla bi ove mane, ali bi sa sobom donela druge nedostatke, vezane za kasnije održavanje i proširivanje funkcionalnosti aplikacije.

5. LITERATURA

[1] *Oracle Database Express Edition 2 Day DBA* - http://download.oracle.com/docs/cd/B25329_01/doc/admin.102/b25107/toc.htm

[2] *NHibernate - Relational Persistence for Idiomatic .NET* - http://www.nhforge.org/doc/nh/en/index.html

[3] *What Is Windows Communication Foundation* - http://msdn.microsoft.com/en-us/library/ms731082%28v=vs.90%29.aspx

[4] *Christian Wenz, Essential Silverlight, 2007, O'Reilly Media Inc.*

[5] *What Is Active Directory Application Mode* - http://technet.microsoft.com/en-us/library/cc738377%28WS.10%29.aspx

Kratka biografija:



Nikola Potrebić rođen je u Novom Sadu 1985. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektro-tehnike i računarstva – Računarske nauke i informatika odbranio je 2011.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9

СОФТВЕРСКО ОКРУЖЕЊЕ ЗА ВИЗУЕЛНО ПРАЋЕЊЕ SCADA СИГНАЛА У SINAUT SPECTRUM СИСТЕМУ

A SOFTWARE ENVIRONMENT FOR VISUAL MONITORING OF SCADA SIGNALS IN THE SINAUT SPECTRUM SYSTEM

Горан Шмит, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *У раду је презентована реализација софтверског окружења за визуелно праћење SCADA сигнала у SINAUT Spectrum систему уз помоћ графичких компоненти – widget-а.*

Abstract – *This paper presents an implementation of the software environment for visual monitoring of SCADA signals in the SINAUT Spectrum system with grapfical components – widgets.*

Кључне ријечи: *Dashboard, Widget, SCADA, SINAUT Spectrum*

1. УВОД

Развој нових технологија омогућио нам је много једноставније и транспарентније контролисање великих и често дистрибуираних система, веома компликованих за управљање. Компјутерски софтвер је данас присутан у скоро свакој области људског дјеловања и омогућио је разне врсте симулација реалних система као и интеракцију са њима што је инжењерима олакшало пројектовање а корисницима употребу оваквих система. Један такав систем је и електро-енергетски систем чије се управљање данас врши уз помоћ софтверског система познатог као *SCADA (Supervisory Control and Data Acquisition*). *SCADA* омогућава прикупљање вриједности из реалног система, њихово складиштење и управљање овим подацима.

Циљ рада био је имплементирати алат који ће омогућити визуелно једноставан, јасан и конфигурабилан приказ стања елемената електричне мреже као проширење *SINAUT Spectrum* система, комерцијалне *SCADA*-е развијене од стране компаније *Siemens*. Овај алат настао је под именом *Dashboard* за потребе компаније ДМС. Идеја је била да се диспечеру омогући једноставан, а ипак довољно детаљан приказ динамике енергетског система као и да се обезбиједи његова брза и недвосмислена реакција на нежељено понашање цијелог система. У пракси се показало да човјек обраћа више пажње на објекте у околини који се крећу, свијетле, трепере или мијењају боју што је искоришћено у реализацији интерфејса алата представљеног у овом раду. Велику улогу у реализацији ових идеја имале су графичке компоненте – *widget*-и које су омогућиле приказ мјерења. У практичном раду *widget*-и ће оператеру обезбиједити потпуну конфигурабилност својим особинама промјене величине и позиције на *dashbord*-у (контролној табли) и омогућити му да самостално креира и мијења комплетан интерфејс по жељи.

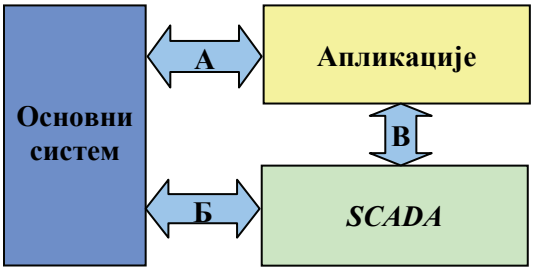
НАПОМЕНА:
Овај рад је проистекао из мастер рада чији ментор је био др Иван Луковић, ред.проф.

2. ОПИС РЈЕШАВАНОГ ПРОБЛЕМА

2.1. *SINAUT Spectrum* систем

SINAUT Spectrum систем је апликација развијена од стране *Siemens*-а која представља отворен, модуларан и дистрибуиран систем за управљање електричним мрежама. Извршава са на *Solaris* оперативном систему и пружа могућност лаког проширивања функционалности и параметризације. Задатак *Spectrum*-а је да прикупља информације из енергетске мреже и да у реалном времену врши обраду, приказ обавјештења, алармирање и командовање.

Архитектура *SINAUT Spectrum* система се састоји из три цјелине и приказана је на слици 1.



Слика 1. *Архитектура SINAUT Spectrum система*

SCADA укључује добављање дигиталних, аналогних и акумулираних вриједности, као и контролу надзирања, складиштење и активирање аларма. Подаци се прикупљају са различитих сензора који су лоцирани на великим удаљеностима. Затим се подаци просљеђују централном рачунару који врши њихову обраду и спроводи адекватну управљачку функцију.

Основни систем испуњава два основна циља. Енкапсулира хардвер и софтвер ситема и пружа конзистентан скуп дистрибутивних сервиса рачунарског система потребан апликацијама.

Апликације представљају широки опсег функционалности које пружа *SINAUT Spectrum* систем. Архитектура је направљена да омогућује врло лако додавање нових алата и функција, као и модификацију постојећих.

Дистрибуирана комуникација је један од разлога зашто је *SINAUT Spectrum* веома флексибилан и конфигурабилан и она се остварује преко посебне магистрале која се зове *Softbus*. Она омогућује сервис

слања порука од тачке до тачке (*point-to-point*) између сервера. Ови сервиси омогућавају успостављање и одржавање комуникације и пренос порука. *Softbus* такође подржава ред чекања порука, што дозвољава да програми комуницирају без упуштања у детаље комуникације и слања порука.

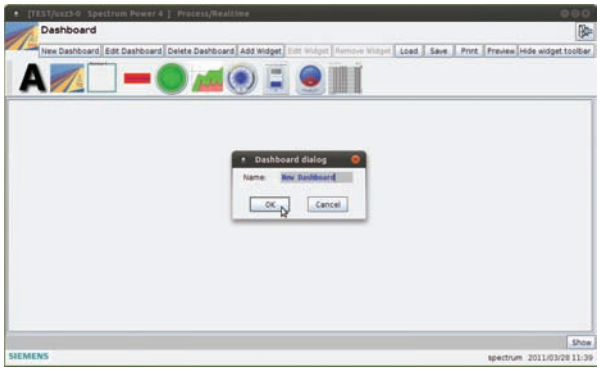
2.2. Технолошка адреса

Да би се омогућило повезивање стварних механичких елемената на терену и оних који се приказују на мапи, потребно је извршити њихово мапирање. Резултат ове операције је технолошка адреса (ТА). Она представља јединствено додељивање имена из физичког у логички свет. Састоји се из пет дијелова означених са B1, B2, B3, Elem и Info. B1 представља ознаку разводног постројења, B2 ознаку напонског нивоа, а B3 означава конкретан уређај у електричној мрежи. Преостала два дијела Elem и Info одређују које могуће вриједности могу да се читају са *SCADA*-е за одабрани уређај. Уколико је потребно очитавати која је тренутна вриједност тренутне (моменталне) снаге трансформатора који ради на напонској мрежи од 220 kV и налази се у Паризу, онда та адреса може да гласи: *B1*:Paris *B2*:220 *B3*:TR01 *Elem*:P *Info*:MvMoment.

3. ОПИС АПЛИКАТИВНОГ РЈЕШЕЊА

Свака компонента *SINAUT Spectrum* система имплементирана је у засебном пакету са својим потпакетима. *Dashboard* као компонента овог система је смјештен у пакет *dashboard* са подпакетом *widgets* у којем се налази имплеметација свих *widget*-а.

Главни прозор *Dashboard*-а визуелно је подијељен у три дијела. У горњем дијелу налази се име алата, дугме за излазак и трака са основним командама за управљање алатом. У доњем дијелу постављено је поље за испис порука битних кориснику, лого *Siemens* компаније и системско вријеме. Ова два дијела представљају стандардан дио интерфејса свих апликација *SINAUT Spectrum* система. Централни дио прозора заузимају трака са *widget*-има и канвас, тј. простор на који се постављају *widget*-и. Изглед главног прозора приказан је на слици 2.

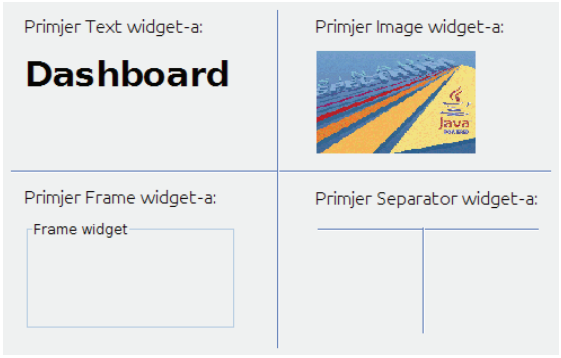


Слика 2. *Главни прозор Dashboard алата*

3.1. *Widget*-и

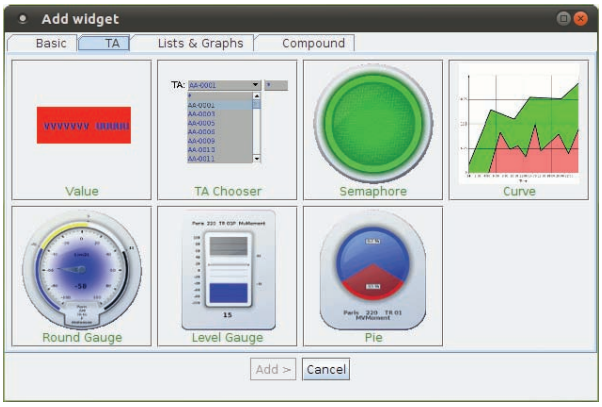
Ријеч *widget* је у енглеском језику настала као скраћеницу за *window gadget*. *Widget* представља самосталну компоненту графичког интерфејса која има неку карактеристичну функционалност. Основне функционалности *widget*-а описаних у овом поглављу обезбјеђују различите начине визуелног приказа мјерења примљених са *SCADA*-е, могућност помијерања *widget*-а по канвасу као и промјену њихове величине. *Widget*-и су по својим особинама подијељени у четири цјелине: *Basic*, *TA*, *Lists & Graphs* и *Compound*.

***Basic* група** *widget*-а служи за визуелно груписање и уређивање осталих *widget*-а на канвасу. Ови *widget*-и немају могућност приказа мјерења или стања елемената мреже па тако и не постоји потреба за подешавањем технолошке адресе. У ову групу спадају: *Text*, *Image*, *Frame* и *Separator widget*. Примјери ових *widget*-а приказани су на слици 3. Као лабела за сваки примјер употребљен је по један *Text widget*, а примјери су раздвојени једним хоризонталним и једним вертикалним *Separator widget*-ом.



Слика 3. *Примјери widget-а из Basic групе*

***TA* група** садржи *widget*-е који омогућавају приказ мјерења као и граница (лимита) из којих мјерене вриједности не би требало да излазе. На слици 4 приказан је таб дијалога на којем корисник може одабрати *widget* из ове групе који жели поставити на канвас.



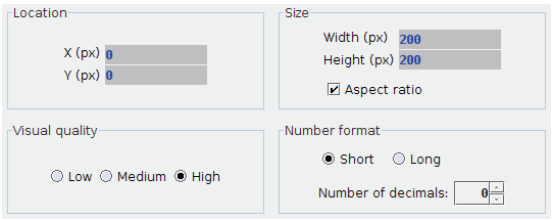
Слика 4. *Примјери widget-а из TA групе*

У ову групу спадају: *Value*, *TAChooser*, *Semaphore*, *Curve*, *RoundGauge*, *LevelGauge* и *Pie widget*. Њихове графичке презентације приказане су на слици 4.

***Lists & Graphs* група** садржи *List widget* и *Loadgraph widget* који су настали од двије постојеће апликације из *SINAUT Spectrum* система које се зову *UIList* и *Loadgraph*. Оваква интеграција других алата у *Dashboard* у облику *widget*-а омогућава кориснику коришћење предности тих алата унутар *Dashboard*-а.

***Compound* група** садржи *widget*-е који представљају предефинисане конфигурације настале комбинацијом до сада наведених *widget*-а. Ове конфигурације су креиране по угледу на постојеће форме у *SINAUT Spectrum* систему које се најчешће користе.

3.2. Конфигурација основних параметара



Слика 5. *Подешавање сновних параметара*

Сви дијалози за конфигурацију *widget*-а садрже један заједнички панел који кориснику омогућава да подеси основне параметре: локације, величине, квалитета приказа и формата исписа бројева.

Како би било омогућено очитавање вредности са технолошке адресе потребно је омогућити постављање адресе *widget*-има. То омогућава компонента *TAChooser*, приказана на слици 6, која се састоји од пет *combobox*-ова у којима се редом, с лијева на десно, бирају сегменти технолошке адресе. Редом, јер се приликом одабира једног ТА дијела аутоматски врши филтрирање вриједности у следећем *combobox*-у.



Слика 6. *TA chooser компонента са опцијама скалирања*

Панел за скалирање, приказана на слици 6 омогућава подешавање распона бројева на скали код *Round gauge* и *Level gauge widget*-а. Постоји аутоматско и ручно скалирање. Код ручног скалирања корисник има могућност да постави максималну и минималну вриједност на скали *widget*-а док се код аутоматског скалирања, ове вриједности израчунавају на основу примљене вриједности која се приказује.

Сваки *widget* који приказује неко мјерење има и могућност примања нове тенолошке адресе једноставним превлачењем адресе са неке друге компоненте у *SINAUT Spectrum* систему.

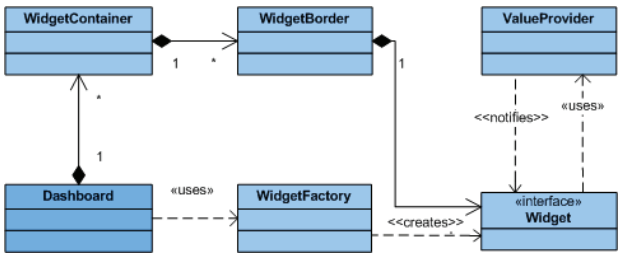
3.3. Функције за штампу приказа

На главној форми постоје дугмад *Print* и *Preview* који омогућавају штампање конфигурације приказане на селектованом табу као и приказ странице пред штампање са могућношћу детаљних подешавања параметара штампе. Оператеру је на овај начин омогућено да одштампа тренутно стање мјерења приказано на *Dashboard*-у или да га преко виртуелног штампача сачува у фајл, нпр. *PDF* фајл. Овако сачувано тренутно стање могуће је поредити са другим стањима направљеним у различитим временским тренуцима и на тај начин пратити понашање система на недељном или мјесечном нивоу.

4. СПЕЦИФИКАЦИЈА РЈЕШЕЊА

Dashboard апликација је имплементирана као компонента *Spectrum* система и мора да насљеђује *AbstractUIClientApplication* која представља апстрактну клијентску апликацију *Spectrum* корисничког интерфејса.

Dashboard се састоји од 4 основна типа компоненти приказаних на слици 7:



Слика 7. *Основне компоненте Dashboard алата*

Value provider компонента обезбјеђује повезивање на једну технолошку адресу и праћење промјене вриједности са ње. Када се вриједност промијени, провајдер генерише догађај и обавјештава оне који га слушају.

Widget омогућава приказ вриједности праћених технолошких адреса и чине га компоненте:

- *Visual* – визуелна репрезентација која се приказује на канвасу *Dashboard*-а
- *Dialog* – компонента за подешавање параметара одабраног *widget*-а
- *DTO* – компонента за пренос параметара (*data transfer object*) који описују неки *widget*. Омогућава и *save*, *load*, *import* и *export* одређеног *widget*-а.

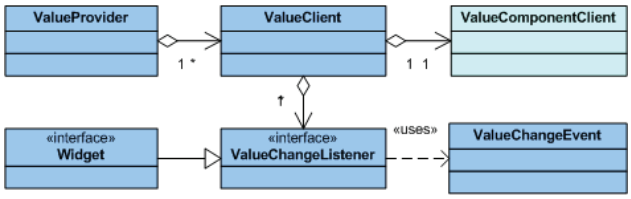
Widget container – компонента која представља канвас на који се смијештају *widget* компоненте.

Widget factory – компонента која је задужена за инстанцирање widget компоненти чије се иницијалне конфигурације читају из конфигурационог фајла.

4.1. *Value provider* компонента

ValueProvider је *singletone* класа која чува све клијенте (*ValueClient*) у хешмапи под кључем који представља технолошку адресу коју клијент слуша. Посао провајдера је да креира клијента када се нека класа пријави за слушање методом *addValueListener*, да сачува тог клијента у хешмапи и прослиједи *listener* клијенту преко којег ће клијент обавјештавати слушаоца о промјенама. У случају да клијент који слуша одређену технолошку адресу већ постоји у мапи, само се просљеђује *listener* постојећем клијенту.

ValueClient садржи листу *listener*-а који слушају његову технолошку адресу. Његов посао је да креира инстанцу *ValueComponentClient*-а који ће преко *ValueManager*-а да комуницира са *SCADA*-ом.*Widget* имплементира *ValueChangeListener* што му омогућава слушање промјене вриједности са технолошке адресе коју корисник одабере. Ове компоненте и њихове везе приказане су на слици 8.

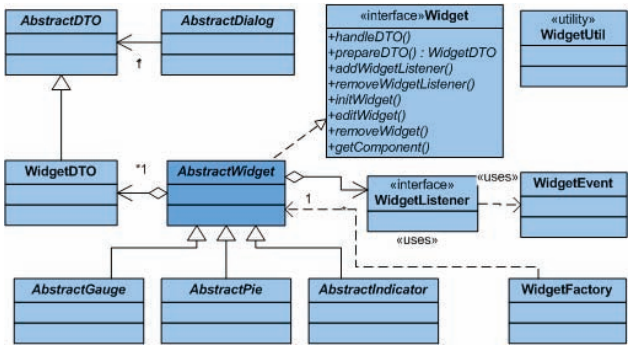


Слика 8. *ValueProvider компонента*

4.2. *Widget* компонента

У овом раду, *Widget*-и су графичке компоненте које омогућавају приказ различитих типова мјерења и трансформација вриједности добијених тим мјерењима. Функционалност коју ова компонента мора да обезбиједи прописана је *Widget* интерфејсом.

На слици 9 приказан је дијаграм *Widget* компоненте.



Слика 9. *Widget компонента*

Методе *initWidget* и *editWidget* омогућавају иницијализацију и промјену параметара *Widget* компоненте позивајући одговарајући дијалог. Метода *removeWidget* служи за уклањање *Widget* компоненте, одјаву *listener*-а и ослобађање свих ресурса које је ова компонента заузимала. Све три методе користе *WidgetListener* за комуникацију са *Dashboard*-ом.

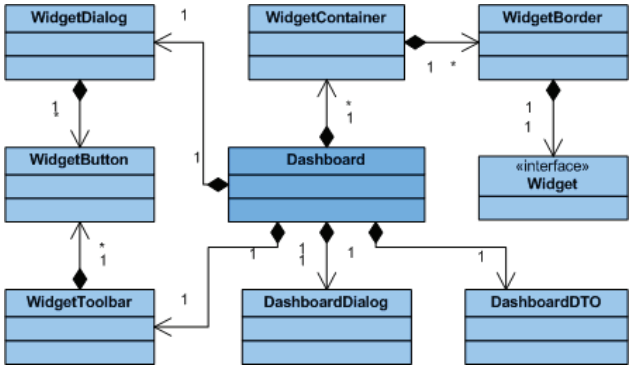
Методе *handleDTO* и *prepareDTO* омогућавају комуникацију између графичке компоненте (*view*) и *DTO* објекта (*model*).

За сваку *widget* класу постоји одговарајућа дијалог класа која кориснику омогућава промјену параметара карактеристичних за тај *widget*. Свака дијалог класа насљеђује *AbstractDialog* који имплеметира основне функције преноса параметара кроз *DTO* објекте и одговоре на корисникове *OK* и *CANCEL* акције затварања дијалога.

DTO компонента омогућава пренос свих параметара између графичке и дијалог компоненте *Widget*-а. Функције *Save*, *Load*, *Import* и *Export* користе ову компоненту да све параметре који описују *widget* претворе у *XML* стринг и назад

4.3. *WidgetContainer* компонента

Сваки *widget* је смјештен у свој *WidgetBorder* који омогућава промјену величине тог *widget*-а и његово помијерање по канвасу *Dashboard*-а, а канвас компоненту имплементира класа *WidgetContainer* која је уједно и један таб *Dashboard* апликације.



Слика 10. *WidgetContainer компонента*

WidgetDialog омогућава одабир типа *widget*-а кликом на дугме (*WidgetButton*) са одговарајућом сликом жељеног *widget*-а. Сваки *WidgetButton* садржи информације о *widget*-у којег представља (име, иконицу, *tooltip*) и имплементира *drag'n'drop* функционалност која омогућава превлачење *widget*-а са *WidgetToolbar*-а. Дијаграм ове компоненте приказан је на слици 10.

4.3. *WidgetFactory* компонента

За потребе подешавања стандардних параметара који описују једну врсту *widget*-а креиран је *property* фајл са *xml* структуром, *Widgets.xml*, који је смјештен у *dashboard.widgets* пакет. *WidgetFactory* представља компоненту која омогућава креирање *widget*-а на основу конфигурације коју прочита из *Widgets.xml* фајла, што је приказано на слици 11, уз помоћ механизма јава рефлексије који користи име класе за креирање конкретног објекта.



Слика 11. *WidgetFactory компонента*

5. ЗАКЉУЧАК

Програмско рјешење описано у овом раду зове се *Dashboard* и омогућава једноставан, јасан и конфигурабилан приказ стања и праћење измјерених вриједности физичких величина, за одабране елементе електричне мреже, а представља компоненту *SINAUT Spectrum* система.

Приликом развоја *Dashboard*-а коришћени су алати који симулирају пристигле податке са *SCADA*-е. Употребом ових алата много је олакшано тестирање функционалности и провјере тачности резултата.

Након имплементације прве верзије ова апликација је интегрисана у *SINAUT Spectrum* и њен даљи развој, тестирање и корекција грешака вршени су у склопу овог система и у складу са новим корисничким захтјевима. Апликација је успјешно интегрисана код крајњих корисника и у стварном окружењу испунила све захтјеве и остварила сва претходна предвиђања прилично олакшавајући рад оператеру. Оператер је овим алатом добио велику прилагодљивост интерфејса коју су му омогућили графички елементи – *widget*-и, омогућено му је да самостално креира приказ мјерења које жели да посматра и да укључи разне врсте аларма и обавијештења која су му омогућила да правовремено регује на промјене понашања електроенергетског система.

6. ЛИТЕРАТУРА

[1] Siemens AG (2006): Power System Control SCADA and Energy Management System SINAUT Spectrum

[2] Alan Dix et al. (2003): Human-Computer Interaction, third edition

[3] GUI widget, http://en.wikipedia.org/wiki/GUI_widget

Кратка биографија:



Горан Шмит рођен је 7. априла 1986. године у Брчком. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Примењене рачунарске науке и информатика одбранио је 2011. године.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 620.93

FARME VJETRENJAČA U VOJVODINI – PREGLED POGODNIH LOKACIJA

WIND FARMS IN AP VOJVODINA – REVIEW OF APPROPRIATE LOCATIONS

Vanja Radovanović, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Ovaj rad predstavlja pogodne lokacijama za izgradnju farmi vjetrenjača u Vojvodini, uvažavajući u obzir energetski potencijal vjetra, zatim izgrađenu infrastrukture, kao i uslove zaštite životne sredine, a posebno ptica.*

Abstract – *This paper presents most favorable sites for building wind farms in Vojvodina, taking into account all aspects, from the most important, wind energy resource, to the built infrastructure and environmental protection, especially birds.*

Ključne reči:- *energija vjetra,vjetroelektrane,lokacije.*

1. UVOD

Sve veća zabrinutost čovečanstva za životnu sredinu izražena je na više globalnih konferencija i konkretizovana je postavljanjem ciljeva za njeno održanje u više dokumenata, od kojih je najpoznatiji Kyoto protokol dogovoren 1997. god. na konfrenciji u gradu Kyoto u Japanu [1]. Mada ga je potpisala većina zemalja, ipak još je u fazi prihvatanja i ratifikacije u nekim državama sveta. Ovim dokumentom se predviđa značajno smanjenje emisije štetnih gasova u atmosferu, posebno CO_2 i time ublažavanje efekta „staklene bašte“, koji je već počeo da dovodi do značajnih klimatskih promena na Zemlji.

Kao ključna akcija u Evropi u tom smislu je proces postepenog gašenja električnih centrala na ugalj ili naftu, te uvođenje mera za smanjenje emitovanja dima i štetnih čestica korišćenjem elektrostatskih filtera, kvalitetnijih tipova uglja ili tehnika odsumporavalja uglja. Takođe, uvode se mere za postepeno ograničavanje i smanjenje emisije štetnih gasova od strane automobila, kamiona i drugih transportnih sredstava, te uvođenje novijih modela koji koriste električnu energiju za pogon.

Istovremeno, smanjivanjem globalnih rezervi nafte, sve veće cijene ovog energenta motivišu čovječanstvo da tradicionalna fosilna goriva zamijene ekološki prihvatljivijim alternativama. Iako su fosilna goriva i dalje dominantna, poslednjih 20-ak godina pažnja se sve više okreće ka proizvodnji energije iz obnovljivih izvora, od kojih je najveći potencijal u energiji vjetra.

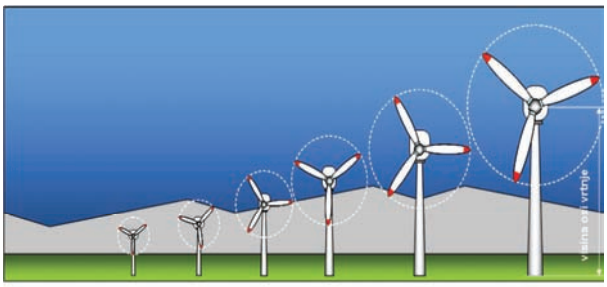
Energija vjetra je u značajnoj prednosti u odnosu na fosilna goriva, jer se proizvodna cijena električne energije iz vjetra može dugoročno predvidjeti, s obzirom da je vjetar, kao primarni energent, u osnovi besplatan.

NAPOMENA:
Ovaj rad je proistekao iz diplomskog-master rada čiji je mentor dr Vladimir Katić, red. prof.

Kroz dosadašnji brzi tehnološki razvoj korišćenja energije vjetra i njenu sve veću primjenu u proizvodnji električne energije, ona postaje vodeća među obnovljivim izvorima energije, naročito zbog svoje uloge u zaštiti okoline.

2. ENERGIJA VJETRA

Kao i svi drugi oblici obnovljive energije (osim energije talasa i geotermalne energije) i energija vjetra je transformisani oblik sunčeve energije. Oko 1-2% te dozračene energije na Zemlji se pretvara u ovu energiju. Linearna kinetička energija vetra koristi se tako što se transformiše u rotaciono kretanje vjetro-turbina, koje su preko multiplikatora brzine povezane sa osovinom električnog generatora. Danas se vetro-generatori prave sa instalisanim kapacitetom od 2-5 MW, sa planovima daljih povećanja na 8MW ili čak 10 MW [2]. Na slici 1 prikazan je razvoj vetrogeneratora, kao i njihove ključne osobine.



Godina	1980.	1985.	1990.	1995.	2000.	2005.
Nazivna snaga	30 kW	80 kW	250 kW	600 kW	1.500 kW	5.000 kW
Promjer rotora	15 m	20 m	30 m	46 m	70 m	115 m
Visina osi vrtnje	30 m	40 m	50 m	78 m	100 m	120 m
Godišnja proizvodnja	35.000 kWh	95.000 kWh	400.000 kWh	1.250.000 kWh	3.500.000 kWh	17.000.000 kWh

Slika 1. *Razvoj vjetrogeneratora.*

Količina energije koju vjetar prenosi na rotor vjetrogeneratora zavisi od površine kruga kojeg čine elise turbine u vrtnji, brzine vjetra kao i od gustine vazduha [3]. Gustina snage se definiše preko sledeće relacije:

$$P_{TURB.} = \frac{1}{2}\rho \cdot C_p(\lambda,\beta) \cdot v_{vetra}^3 \quad [W/m^2] \qquad (1)$$

gde su $C_p(\lambda)$ - koeficijent snage turbine, λ - odnos periferne brzine turbine i brzine vetra, β - ugao zakrenutosti elisa (eng. pitch angle), a v – brzina vetra.

Karakteristike vjetrova na nekoj lokaciji se najbolje mogu prikazati pomoću ruža vjetrova, Weibull-ove krive i atlasa vjetrova [4]. Ruža vjetrova prikazuje raspodjele brzina i smjerova vjetrova, tj. strane sveta sa kojih duvaju vjetrovi i njihove intenzitete po pojedinim pravcima. Weibull-ova kriva grafički prikazuje vjerovatnoću pojavljivanja određene brzine vjetra na nekoj lokaciji, odnosno indirektno ukazuje na energiju vjetra. Atlas vjetrova predstavlja skup geografskih karata na kojima se prikazuje energetski potencijal vjetra na promatranoj

lokaciji, regionu, državi ili kontinentu, prikazan u vidu srednje godišnje brzine vjetra i srednje godišnje gustine snage (W/m^2) na različitim visinama u odnosu na tlo (25m, 50m, 80m, 100m, 200m).

Vetroelektane predstavljaju skup vetro-generatora, raspoređenih na optimalan način na određenoj lokaciji, tako da se snage vetroelektrana kreću od 20MW do 200MW ili više.

Često se vetroelektrane nazivaju farmama vetrenjača ili parkovima vetrenjača, a prave se na kopnu (*on-shore*) ili na moru (*off-shore*) [2-6].

Prilikom opredeljenja za izgradnju vetro-elektrane, najvažniji koraci su da se utvrdi energetski potencijal vjetra na datoj lokaciji, zatim da se procjeni udaljenost pomenute farme vjetrenjača od energetske mreže, kao i načinima priključenja na istu, oceni postojeća infrastruktura (putevi, pruge, vodni putevi i sl.), da se osigura potreban kapital, obezbjedi plasman proizvedene električne energije i svakako, da se izvrši procjena uticaja na životnu sredinu.

3. ATLAS VJETROVA VOJVODINE

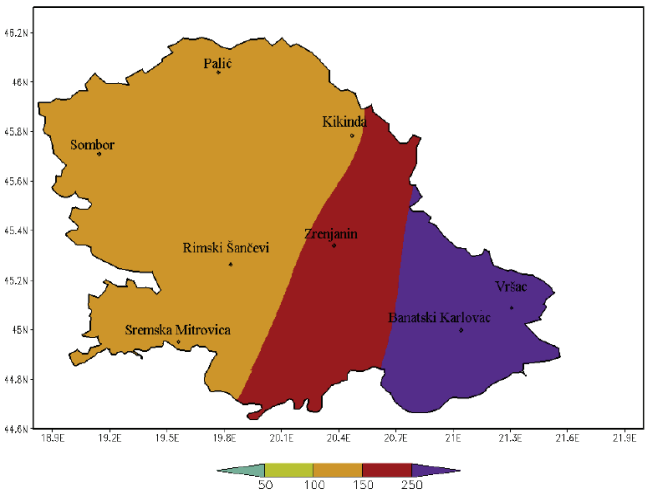
U dosadašnjem periodu urađeno je više studija, koje su dale podatke o energetskom potencijalu vjetra u Srbiji i Vojvodini, kao i konkretne vrednosti energije vjetra na pojedinim lokacijama u Vojvodini. Sve ove studije su manje-više dovodile do sličnih rezultata za potencijal energije vjetra na teritoriji Vojvodine [4].

Koristeći energiju vjetra Srbija bi mogla da proizvede 2,3 milijarde kilovat-časova električne energije godišnje, što je sedam odsto sadašnje proizvodnje. U jugoističnom Banatu vjetrovi duvaju više od 200 dana u godini i to jačinom koja bi zadovoljila sve potrebe da se na ovoj lokaciji podignu prvi vjetroparkovi u Srbiji koji bi proizvodili ekološku električnu energiju.

Istraživanja pokazuju da su, osim u jugoistočnom Banatu koji predstavlja najbolju lokaciju, veliki potencijali ove energije i na području cijele Vojvodine, kao i u zoni istočne i centralne Srbije. Prednost lokacija u Vojvodini je ta što je u najvećoj mjeri izgrađena putna, željeznička i vodna infrastruktura, u blizini je i visoko-naponska električna mreža, kao i veći gradovi (potrošači).

Takođe, najveća hidroelektrana u Srbiji „Đerdap“ može da obezbedi neophodnu balasnu energiju u slučaju većih i naglih varijacija brzine vetra. To su razlozi što se poslednjih godina istraživanja, a i krupni investitori, uglavnom fokusiraju na ovu teritoriju.

„Atlas vjetrova na teritoriji AP Vojvodine“ je do sada najobimnija studije energetskog potencijala za ovu pokrajinu, a baziran je na podacima dobijenim iz 8 meteoroloških stanica (Palić, Sombor, Kikinda, Zrenjanin, Novi Sad, Banatski Karlovac, Sremska Mitrovica, Vršac) [4].
Sastoji se od mapa brzina vjetrova u m/s i mapa srednjih gustina snaga vjetrova u W/m^2 na 10m, 25m, 50m, 100m i 200m. Na slici 2. je primer jedne mape na kojoj je prikazana srednja gustina snage vjetrova na 100m nadmorske visine.



Slika 2. *Srednja gustina snage vjetrova na 100m.*

4. VJETROELEKTRANE -FARME VJETRENJAČA

Početkom 20. vijeka masovno su korištene prve male vjetrenjače za proizvodnju električne energije, ali su mnoge od njih prestale sa radom tokom 30-ih godina prošlog vijeka zbog sve intenzivnijeg proširenja elektrodistributivnog sistema. Međutim, naftna kriza početkom 70-tih godina prošlog vijeka i nesreće u nuklearnim elektranama uslovilli su potrebu za korišćenjem obnovljivih izvora energije, pa time i energije vjetra. Krajem 80-tih i početkom 90-tih godina 20. vijeka, razvoj i primjena vjetrenjača za proizvodnju električne energije opet dobijaju na značaju. Poslednjih godina vjetroenergetika je značajno napredovala, kako u obimu proizvodnje, tako i u instaliranoj snazi. Na kraju 2010. god. u EU je bilo instalisano 84.278 MW, odnosno u Evropi 86.279 MW kapaciteta u vetroelektranama za proizvodnju električne energije [7].

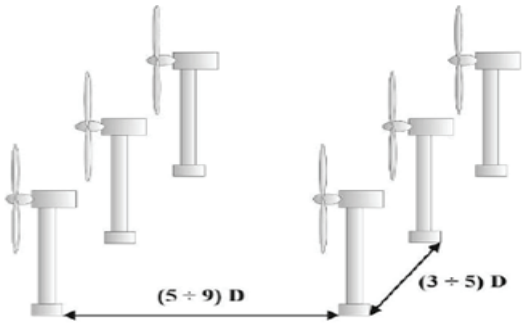
4.1. Tipovi i dijelovi vjetroturbina

Vjetroturbine se mogu podijeliti prema nekoliko različitih kriterijuma. Najvažnija podjela vjetroturbina je prema položaju osi turbinskog kola. U skladu sa to podjelom postoje vjetrenjače sa horizontalnom i vertikalnom osom vrtnje (Savonius, Darieus i H-tip). Pored ove, postoji i niz drugih podjela od kojih se može izdvojiti podjela prema broju lopatica (višelopatične, s nekoliko lopatica i sa jednom lopaticom).

Za vetroelektrane se danas uglavnom primenjuju turbine sa horizontalnom osom i tri kraka. Najvažniji dijelovi jedne vjetroturbine su: lopatice, rotor, kočnica, prenosnik snage, generator, kućište, stub, oprema za zakretanje, te upravljački sistem.

4.2. Pozicioniranje vjetrenjača u farmu

Mnogi tehnički problemi nastaju kada više vetrogeneratora (turbina) treba da se postavi jedna blizu druge u farmu vjetrenjača. Najvažnija pitanja su gdje postaviti turbine, te koliki razmak treba biti između njih. Generalno, vjetroturbine se nalaze na mjestima gdje ne postoje prepreke kao što su drveće, kuće i drugi predmeti koji uzrokuju turbulenciju. Pored lokacija na moru, najbolje lokacije su lokacije na vrhu brda, te lokacije na velikim ravnicama.



Slika 3. *Raspored vjetrenjača u farmi*

Prednosti povezivanja vjetrenjača u farmu vjetrenjača su, prije svega, cijena, te jednostavnija interkonekcija sa elektroenergetskim sistemom. Turbine u farmi nikada ne mogu dobiti 100% energije vjetra koju bi dobila usamljena turbina jer svaka turbina usporava vjetar iza sebe. Ovaj efekat je poznat kao „trag efekat“ dok, sa druge strane, iskoristivost zemljišta i cijena spajanja turbina na električnu mrežu, traže da ih smjestimo što bliže jednu drugoj. Zbog toga, prilikom izgradnje vjetroelektrane, treba naći kompromis između ova dva zahtjeva. Kompromis je nadjen tako da je udaljenost turbina u smjeru dolaska vjetra 3-5D, dok je udaljenost turbina u smjeru normalnom na smjer vjetra 5-9D („D“ predstavlja prečnik rotora vjetroturbine (Slika 3.)).

4.3. Prednosti i nedostaci farme vjetrenjača

Prednosti vjetrenjača su da prilikom njihovog rada ne dolazi do emisije bilo kakvih štetnih gasova, zatim to što se mogu smjestiti podjednako i na kopnu i na moru, i na obradivim i na neobradivim zemljištima. Osim toga, ispod stubova vjetroelektrane mogu se nesmetano obavljati poljoprivredni i stočarski radovi. U skladu sa tim, vjetroelektrane su poželjan oblik alternativnog izvora energije za razliku od elektrana na fosilna goriva.

Neki od nedostataka su to da imaju stohastičku proizvodnju, pri kretanju lopatica kroz vazduh javlja se buka, nije rješeno aktivno akumulisanje energije u periodima bez vjetra, te postoji štetan uticaj za ptice.

4.4. Troškovi izgradnje vjetroelektrane

Troškovi izgradnje modernih vjetroelektrana kreću se od 700-1100 €/kW za srednje i velike vjetroelektrane (snaga od 30-1500kW), pa sve do 1500-2500 €/kW za male vjetroelektrane (snaga manjih od 30kW). Od ukupnih troškova izgradnje skoro 75% se odnosi na troškove izgradnje same vjetroturbine, dok ostatak čine troškovi izgradnje temelja, priključka na EES, pravo građenja i druge. Troškovi izgradnje se mogu podijeliti na nekoliko grupa. To su kapitalni troškovi, balansni troškovi, troškovi finansiranja, troškovi održavanja, proizvodni troškovi i spoljni troškovi.

4.5. Propisi u Srbiji

Da bi se, uopšte, jedna vjetroelektrana u Srbiji mogla izgraditi i koristiti neophodno je da se ispune sledeći uslovi: Pribavljanje studije o procjeni uticaja na životnu sredinu, pribavljanje energetske dozvole, pribavljanje uslova za priključenja na mrežu, pribavljanje saglasnosti sa stanovišta bezbjednosti vazdušnog saobraćaja, pribavljanje lokacijske dozvole, pribavljanje građevinske dozvole, pribavljanje vodne dozvole, tehnički pregled objekta i pribavljanje upotrebne dozvole [8].

5.INFRASTRUKTURA

5.1. Električna mreža u APV

U Vojvodini postoji razvijena prenosna i distributivna električne mreža. Vojvodina se uglavnom snabdeva iz termo-elektrane „Nikola Tesla“ u Obrenovcu, hidro-elektrana na Drini, kao i tri lokalne termoelektrane-toplane u Novom Sadu, Zrenjaninu i Sr. Mitrovici. Prenos električne energije vrši se dalekovodima naponskih nivoa 110, 220 i 400 kV, ukupne dužine 2.885 km, dok je ukupna dužina srednje-naponske (35, 20 i 10 kV) i nisko-naponske (0,4 kV) distributivne mreže 25.260 km. Na teritoriji Vojvodine je i 10.610 transformatorskih stanica različitih nivoa, ukupne instalisane snage 8.924 MVA. Godišnja potrošnja u Vojvodini je 8.800 GWh, a vršno opterećenje je 1.724 MW. Elektrodistributivnom mrežom gazduje privredno društvo „Elektrovojvodina“, koje je u sklopu EPS-a, a formirano je 28.06.1958. kao jedinstveno preduzeće za distribuciju električne energije na području Vojvodine, dok prenosnom mrežom kompanija „Elektro mreže Srbije“ sa sedištem u Beogradu.

5.2. Saobraćajna infrastruktura

Teritorija AP Vojvodine se odlikuje veoma razvijenom mrežom puteva i pruga. Kroz Vojvodinu prolaze neke vrlo važne saobraćajnice, a prije svega autoput koji ide iz Centralne Evrope, pa preko Novog Sada do Beograda i dalje ka Nišu gdje se račva na 2 pravca (jedan ka Bugarskoj, a drugi ka Skoplju). Oko autoputa je izrazito razvijena i mreža lokalnih i regionalnih puteva.

Ukupna dužina pruga u AP Vojvodini je oko 1.580 km. Železnička mreža AP Vojvodine je dovoljnog kapaciteta, ali kako je prosječna starost magistralnih pruga u Vojvodini oko 35 godina, a prosječna starost vozova oko 40 godina, neophodna joj je modernizacija.

Ukupna dužina svih plovnih puteva u Srbiji, uključujući i kanal Dunav-Tisa-Dunav, iznosi 1677km. Sve plovne rijeke se većim svojim dijelom nalaze na teritoriji Vojvodine, pa riječni saobraćaj u Vojvodini ima izrazite potencijale. Osnovnu infrastrukturu vodenog saobraćaja čine rijeke Dunav, Sava, Tisa, kao i kanal DTD.

5.3. Ljudski potencijal

Stručnjaci iz oblasti elektrotehnike – elektroenergetike obrazuju se u srednjim elektrotehničkim školama za titulu tehničara (SSS), kao i na Fakultetu tehničkih nauka u Novom Sadu za titulu inženjera strukovnih studija ili diplomiranog inženjera akademskih studija. Godišnje diplomira oko 50 studenata. Istraživanja se vrše u sklopu Laboratorije za obnovljive i distribuirane izvore električne energije, kao i Centra za obnovljive izvore i kvalitet električne energije FTN-a.

6. UTICAJ VJETROELEKTRANA NA OKOLINU

6.1. Uticaj na ljude

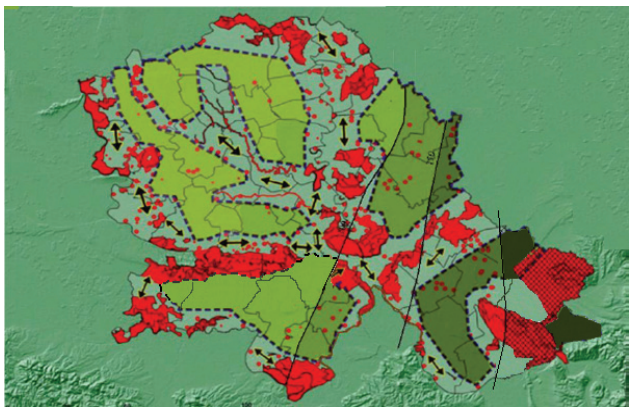
Bilo kakav građevinski rad, pa tako i postupak izgradnje vjetroelektrane, predstavlja sigurnosnu opasnost. Pored ovoga, kvar na lopaticama, kao i padajući led i požari na

vjetroturbinama predstavljaju opasnost po ljude koji se nalaze ili kreću u blizini vjetroelektrana. Pored ovih, potencijalnih posledica, prilikom izgradnje vjetro-elektrane treba voditi računa i o drugim efektima, prije svih, efektima buke, estetike i odsjaja koji se mogu smanjiti adekvatnim načinom gradnje.

6.2. Uticaj na ptice i ostalu floru i faunu

Vjetroelektrane imaju relativno mali uticaj na biljni i životinjski svijet. Tokom same izgradnje vjetroelektrane može doći do toga da se privremeno poremeti aktivnost životinja na tom području, ali će nakon uspostavljanja normalnog režima rada vjetroelektrane, ove životinje opet redovno koristiti to stanište.

Ugroženost ptica je često glavna zamjerka protiv instalacije vjetroturbina. Postoje mnogi radovi o uticaju vjetroelektrana na ptice i na to kolika je njihova smrtnost na područjima gdje su instalirane vjetroelektrane. Oni pokazuju da je relativno mali uticaj vjetroelektrana na smrtnost ptica. Ipak, za određivanje lokacije smještaja vjetroelektrane potrebno je izbjegavati migracione koridore, kao i staništa retkih ptica, kao i ona koja su od međunarodnog značaja za ptice. Zavod za zaštitu prirode Srbije, odnosno Vojvodine postavio je kartu migracionih područja ptica za teritoriju APV, kao i područja zaštićenih prirodnih dobara (Nacionalni parkovi, Parkovi prirode, Predeli izuzetnih odlika i druge zone zaštite) [9]. Na slici 4 migracioni koridori su obeleženi dvostranim strelicama, a područja zaštićenih prirodnih dobara crvenom bojom. Područja obeležena zelenom bojom i uokvirena isprekidanim linijama, osim crvenih tačkica na njima, su zone u kojima je mali uticaj na život ptica.



Slika 4. *Pregled pogodnih lokacija*

7. PREGLED POGODNIH LOKACIJA

Na slici 4. se uočavaju najpovoljnije lokacije za izgradnju farmi vjetrenjača u Vojvodini. Slika je načinjena, uzimajući u obzir uslove koji su bitni za izgradnju jedne farme. Pored zona zaštićenih prirodnih dobara, na slici su označene i linije nivoa srednje gustine snage vjetrova. Najpovoljnija područja su označena tamno zeleno i nalaze se na krajnjem jugo-istoku Vojvodine, oko Bele Crkve i severo-zapadno od Vršca. Pored ovih lokacija, povoljne su i one ok Nacionalnog parka „Delibatska peščara“, kao i istočno od Zrenjanina. Za još kvalitetnije određivanje, potrebno je preklopiti mape za izgrađene infrastrukturu (elektro mreža, putevi, pruge,...), kao i podatke iz prostornog plana Srbije, odnosno Vojvodine.

8. ZAKLJUČAK

Teritorija Vojvodine ima veliki potencijal u iskorištavanju energije vjetra, ali do sada nije podignut još nijedan vjetropark. Razlog može da bude visina potrebnih investicija (1 MW košta oko 1,2 miliona evra). Međutim, kao glavna smetnja navode se vladini propisi prema kojima sami investitori trebaju ulagati u gradnju infrastrukture, dok ulagači smatraju da je to posao za državu. Osim toga, procedura za dobijanje dozvola je jako komplikovana, a ukupno ih je potrebno 26.

9. LITERATURA

[1] Kyoto Protocole to the United Nations Framework Convention On Climate Change, http://unfccc.int/resource/docs/convkp/kpeng.html

[2] V. Katić, I. Kapetanović, V. Fuštić, “Obnovljivi izvori električne energije“, skripta, EU TEMPUS-CEFES Project JEP-18126-2003, Novi Sad, 2007

[3] G.H. Johnson, “*Wind energy system – Electronic Edition*“, Manhattan, KS, USA, Oct.2006, http://www.eece.ksu.edu/~gjohnson/Windbook.pdf

[4] V. Katić i saradn., “*Atlas vetrova na teritoriji AP Vojvodine*“, Studija za Pokrajinski sekretarija za energetiku i min. sirovine APV, Fakultet tehničkih nauka, Novi Sad, 2008, www.psemr.vojvodina.gov.rs

[5] S.Jerković, V.Katić, „Primer izgradnje vetroelektrane u Vojvodini”, *Zbornik rad.Fakulteta tehničkih nauka,* God.25, Br.19, 2010, pp.4093-96. www.ftn.uns.ac.rs

[6] J. Kartal, V. Katić, „Farme vetrenjača na moru“, *Zbornik radova Fakulteta tehničkih nauka,* God. 26, 2011, (prihvaćen za publikovanje), www.ftn.uns.ac.rs

[7] „Wind in Power - 2010 European Statistics“, The European Wind Energy Assoc., 2011, www.ewea.org

[8] B. Lepotić-Kovačević, B.Lazarević, „Izgradnja vetro-elektrana i proizvodnja el. energ. iz energije vetra u Srbiji-Vodič za investiture”, 2009, www.mem.gov.rs

[9] Zavod za zaštitu prirode Srbije, www.natureprotection.org.rs

Kratka biografija:



Vanja Radovanović rođen je u Tuzli 04.08.1987.god. Diplomski-master rad na Fakultetu tehničkih nauka u Novom Sadu iz oblasti Elektrotehnike i računarstva odbranio je 2011.god.



Vladimir Katić je doktorsku disertaciju odbranio je na Univerzitetu u Beogradu 1991. Redovan profesor Fakulteta tehničkih nauka u Novom Sadu je od 2001. za oblast Energetska elektronika, mašine i pogoni. Prodekan je FTN-a i šef katedre za energetsku elektroniku i pretvarače. Oblasti interesovanja su energetska elektronika, kvalitet električne energije i obnovljivi izvori električne energije.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

EKSPERIMENTALNA EVALUACIJA RADIOFREKVENCIJSKE METODE ZA ODREĐIVANJE ORIJENTACIJE DALJINSKOG UPRAVLJAČA

Miloš Vasić, *Fakultet tehničkih nauka, Novi Sad, milos.vasic@ieee.org*
Dragan Samardžija, *Fakultet tehničkih nauka, Novi Sad, dragan.samardzija@rt-rk.uns.ac.rs*
Miodrag Temerinac, *Fakultet tehničkih nauka, Novi Sad, miodrag.temerinac@rt-rk.uns.ac.rs*

Sadržaj – *Ovaj rad prikazuje ideju o uparivanju uređaja u namenskoj bežičnoj komunikacionoj mreži na osnovu orijentacije daljinskog upravljača – upaliti lampu u pravcu koje pokazuje daljinski upravljač, bez uticanja na stanje ostalih lampi. Predložene su i evaluirane dve radiofrekvencijske metode za određivanje orijentacije daljinskog upravljača: metoda fazne razlike i metoda ortogonalne polarizacije antena.*

1. UVOD

Problem će biti opisan pomoću sledećeg primera: izvestan broj lampi se nalazi u sobi. Usmeravanjem daljinskog upravljača u pravcu željene lampe, korisnik može da je uključi, isključi, ili promeni intenzitet njene svetlosti (tj. priguši je). Daljinski upravljač će određenu lampu kontrolisati (tj. upariti se sa njom) samo na osnovu položaja daljinskog upravljača, tj. pravca u kojem daljinski upravljač pokazuje. Ne postoje eksplicitne komande daljinskog upravljača za svaku lampu pojedinačno, nego postoji samo kontrola za jedno svetlo. Na primer, na Slici 1, daljinski upravljač pokazuje u pravcu lampe 4, uključuje je i isključuje, bez da utiče na stanje ostalih lampi. Isti daljinski upravljač, ukoliko se usmeri u pravcu druge lampe, koristeći istu kontrolnu dugmad, kontrolisao bi novousmerenu lampu.



Slika 1: Odabir lampe na osnovu pozicije daljinskog upravljača

Kako bi se rešio prethodno opisani problem, neophodno je da svaka lampa izmeri radio signal koji šalje daljinski upravljač i odredi njegove karakteristike. Rezultati merenja se razmenjuju između prisutnih lampi i posredstvom procedure odlučivanja donosi se odluka o tome u koju je od lampi sa najvećom verovatnoćom uperen daljinski upravljač. Komunikacija između uređaja može se implementirati koristeći IEEE 802.15.4 standard.

Uopštenje prethodno opisanog primera sa svetlima i daljinskim upravljačem je uparivanje uređaja u bežičnoj komunikacionoj mreži. Trenutne implementacije komunikacionih protokola kao što su *Bluetooth* ili *ZigBee* podrazumevaju da uređaj koji započinje komunikaciju zna adresu prijemnika. Na primer, kod *Bluetooth*-a, adresa prijemnika se otkriva prilikom pronalaska uređaja (*eng. device discovery*), kada predajnik šalje zahtev za otkrivanje uređaja u okolini. Uređaji, čije je otkrivanje omogućeno, periodično skeniraju zahteve, i ukoliko prime zahtev, šalju svoju adresu i lokalno vreme uređaju koji šalje zahtev. Po prijemu ta dva parametra, moguće je uspostaviti *Bluetooth* vezu [1]. Ideja ovog rada je da omogući uparivanje uređaja, tj. uspostavljanje veze između njih, bez poznavanja njihovih adresa unapred, nego samo na osnovu usmerenja uređaja.

2. METODOLOGIJA

Obe metode predstavljene u ovom radu podrazumevaju korišćenje dve predajne antene ugrađene na daljinski upravljač.

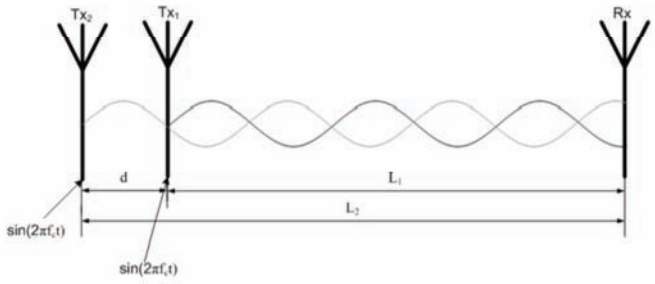
Ukoliko predajnik šalje beskonačni prostoperiodični sinusni signal $\sin(2\pi f_c t)$, gde je f_c frekvencija nosioca i iznosi 2,4 GHz (amplituda signala je jednaka jedan, a fazni pomeraj nula), prijemna strana registuje signal $r(t)$ koji je usled uticaja prenosnog kanala promenjene amplitude. φ je fazni pomeraj koji registruje prijemnik i direktno je proporcionalan rastojanju između predajnika i prijemnika L, a obrnuto propocijalan talasnoj dužini λ.

$$r(t) = A \cdot \sin(2\pi f_c t + \varphi) \qquad (1)$$

U okviru demodulacije signala na prijemnoj strani, primljeni signal $r(t)$ se množi sa $\sin(2\pi f_c t)$ i $\cos(2\pi f_c t)$ i time se translira u osnovni opseg. Niskopojasno filtriranje je neophodno kako bi se uobličio signal u osnovnom opsegu i uklonile komponente koje su izvan njega [2]. Time se dobijaju I i Q komponente (od *eng. inphase* i *quadrature)*. U I i Q komponentama ne figuriše frekvencija nosioca f_c, nego samo amplitudno slabljenje i fazni pomeraj, koje unosi bežični kanal.

A. Metoda fazne razlike

Dve predajne antene postavljaju se na daljinski upravljač na međusobnu udaljenost d (Slika 2), koja treba da bude manja od talasne dužine radio talasa λ. Idealno, udaljenost d iznosi pola talasne tužine, odnosno 6,25 cm na frekvenciji od 2,4 GHz. Jedna antena se postavlja na prednji a druga na zadnji kraj daljinskog upravljača. Prijemna lampa meri faznu razliku signala primljenih sa ove dve antene.



Slika 2: Propagacija prostoperiodičnog signala sa dve antene

NAPOMENA:
a) Ovaj rad proistekao je iz diplomskog-master rada Miloša Vasića. Mentor je bio dr Dragan Samardžija.
b) Rad je prethodno publikovan na konferenciji ETRAN, Banja Vrućci, juni 2011.

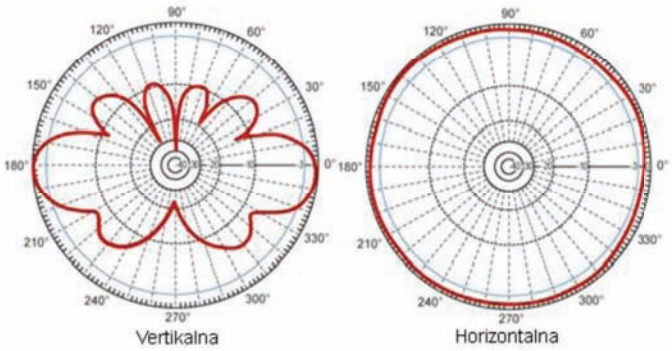
Uvrštavanjem izraza $\varphi = L/\lambda \cdot 2\pi$ u jednačinu (1), dobijaju se izrazi za signale poslate sa prednje i zadnje antene daljinskog upravljača, respektivno, a koje registruje prijemnik:

$$r_1(t) = A_1 \cdot \sin\left(2\pi f_c t + \frac{L_1}{\lambda} 2\pi\right),$$
$$r_2(t) = A_2 \cdot \sin\left(2\pi f_c t + \frac{L_1 + \Delta L}{\lambda} 2\pi\right) \quad (2)$$

Iz izraza (2) sledi da je fazna razlika proporcijalna razlici u pređenom putu ΔL. Uz pomoć fazne razlike može da se odredi usmerenost daljinskog upravljača. Daljinski upravljač je usmeren u pravcu određenog prijemnika ako fazna razlika koju detektuje taj prijemnik odgovara udaljenosti između antena d. Fazna razlika od 180° odgovara udaljenosti između antena od 6,25 cm.

B. Metoda ortogonalne polarizacije

Polarizacija antene predstavlja orijentaciju električnog polja $\vec{E}$ u odnosu na zemljinu površinu. Određena je fizičkom stukturom antene i njenom orijentacijom u prostoru. Slika 3 ilistruje jedan primer polja zračenja antene.



Slika 3: Polarizacija antena

Dve antene na predajniku moguće je postaviti pod uglom od 90°, tj. smestiti u dve ortogonalne ravni, i time ih suprotno polarizovati. Na taj način značajno se menja snaga signala primljenog sa antene koja ima polarizaciju suprotnu od polarizacije antene prijemnika, te je na osnovu te informacije moguće doneti zaključke o usmerenosti daljinskog upravljača. Rotiranjem daljinskog upravljača, tj. njegovim usmeravanjem pod uglom od 90° u odnosu na prijemnik, menja se polarizacija antena, pa druga antena postaje polarizovana isto kao i antena prijemnika.

3. EKSPERIMENTALNO ISPITIVANJE

Za eksperimentalno ispitivanje korišćen je SDR (*eng. Software Defined Radio*), svojevrsna kombinacija fizičke arhitekture i programske podrške namenjena da podrži raznovrsne radio protokole i algoritme [3]. Kako se radi o asinhronom sistemu, gde predajnik i prijemnici nisu sinhronizovani, potrebno ih je sinhronizovati i razlikovati trenutke kad primljeni podaci potiču sa prve antene od trenutka kad potiču sa druge antene. U tu svrhu izabrane su *Golay* sekvence [4]. Par komplementarnih sekvenci ima osobinu da zbir njihovih aperiodičnih autokorelacija iznosi nula, ukoliko autokorelacije nisu u fazi.

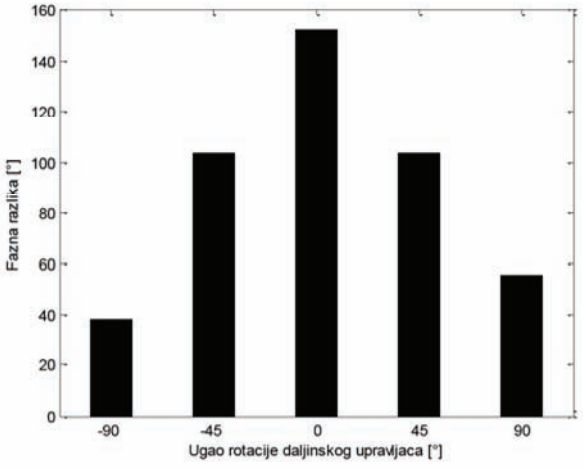
SDR predajnik poseduje dve antene, od kojih se jedna proglašava za prednju a druga za zadnju. Njima se dodeljuje komplementarni par *Golay* sekvenci. Antene naizmenično šalju signal, a na korišćenoj frekvenciji odabiranja od 960 kHz, aktivna antena se smenjuje svakih 1,067 ms. Antene korišćenog SDR predajnika nalaze se na međusobnoj fizičkoj udaljenosti od 5,1 cm, pa fazna razlika koja odgovara tom rastojanju iznosi 146,9°. Predajnik ima ulogu daljinskog upravljača. SDR prijemnik, koji ima ulogu lampe, poseduje jednu antenu i prosleđuje primljeni signal na mrežnu spregu sa računarom. Na računaru se primljeni paketi analiziraju upotrebom programskog alata Matlab.

Za metodu fazne razlike, relevantna karakteristika signala je fazna razlika primljenih signala sa prednje i zadnje antene daljinskog upravljača. Rezultati eksperimentalnih merenja prikazani su u Tabeli 1.

Tabela 1: Rezultati eksperimentalnih merenja za metodu fazne razlike

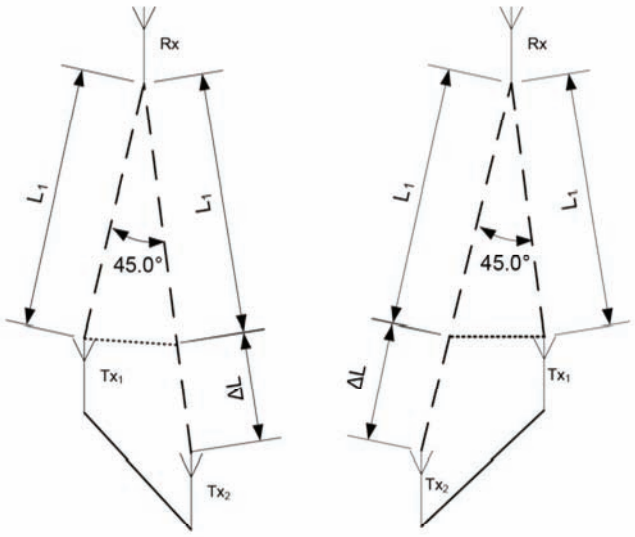
Ugao rotacije daljinskog upravljača [°]	-90	-45	0	45	90
Srednja vrednost fazne razlike [°]	38,2	103,6	151,7	103,5	55,9

Kada se daljinski upravljač usmeri u pravcu lampe koja meri karakteristiku njegovog signala (ugao rotacije od 0°), srednja vrednost fazne razlike iznosi 151,7°, što je veoma blizu teorijskoj faznoj razlici od 146,9°. Za ugao od 90°, talas se prostire preko SDR kutije, što bi mogao da bude razlog za razliku u odnosu na ugao od -90°, gde talas do prijemnika stiže direktno. Slika 4 grafički prikazuje srednju vrednost fazne razlike.



Slika 4: Srednja vrednost fazne razlike u funkciji od ugla rotacije daljinskog upravljača

Metoda fazne razlike ne može da razlikuje rotaciju daljinskog od 45° i -45°, zato što su pređeni putevi signala za ta dva slučaja jednaki (Slika 5).



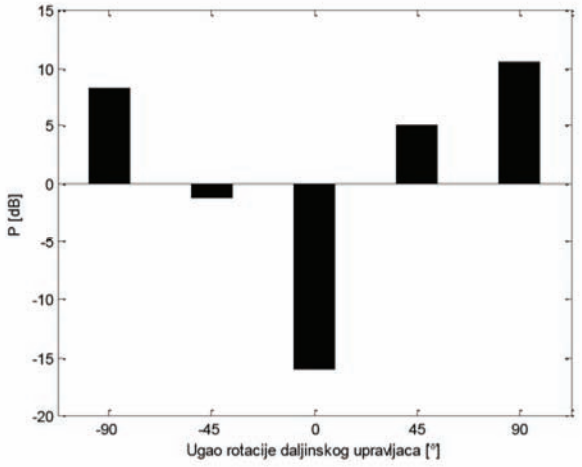
Slika 5: Neodređenost položaja metode fazne razlike

Za metodu ortogonalne polarizacije antena, relevantna karakteristika signala je odnos snaga primljenih signala sa prednje i sa zadnje antene daljinskog upravljača. Slika 6 prikazuje relativni odnos srednje vrednosti snage $10 \cdot \log_{10} \frac{\sum_i P_{1i}}{\sum_i P_{2i}}$ u decibelima, za sva merenja.

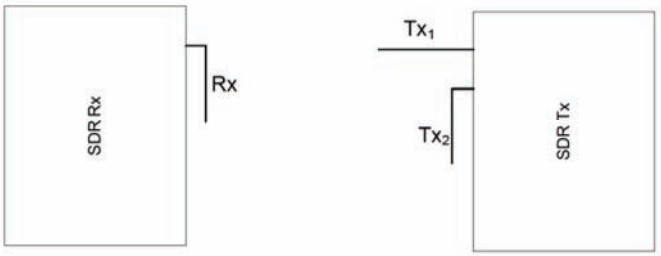
Tabela 2: Rezultati eksperimentalnih merenja za metodu ortogonalne polarizacije antena

Ugao rotacije daljinskog upravljača [°]	-90	-45	0	45	90
Relativni odnos snage [dB]	8,31	-1,21	-15,93	5,03	10,58

Kada se daljinski upravljač usmeri u pravcu lampe koja meri karakteristiku njegovog signala (ugao rotacije od 0°, Slika 7), srednja vrednost relativnog odnosa snage P_1 i P_2 iznosi -15,93 dB, što ukazuje na to da je snaga signala primljenog sa druge antene P_2, koji ima istu polarizaciju kao i antena prijemnika, 39 puta jača od snage signala primljenog sa prve antene P_1. Slika 6 grafički prikazuje srednju vrednost odnosa snage.



Slika 6: Relativni odnos srednje vrednosti snaga u zavisnosti od ugla rotacije daljinskog upravljača



Slika 7: Položaj antena za ugao rotacije od 0°

Za ugao od -90° dominira antena Tx_1 (antena Tx_2 je ortogonalno polarizovana), dok je za ugao od -45° snaga antene Tx_2 tek neznatno jača. Za ugao od 90° antena Tx_1 dominira. Razlika u odnosu snaga za uglove rotacije od 90° i -90° proističe iz karakteristike antene (Slika 3– vertikalna polarizacija), gde se vidi da antena ima mnogo manje slabljenje u smeru njenog vrha u odnosu na njen zadnji deo.

Odlučivanje u kom pravcu je usmeren daljinski upravljač se ostvaruje na sledeći način. Ukoliko se pet lampi postavi na međusobnoj udaljenosti od po 45° po zamišljenoj kružnici u čijem centru stoji korisnik sa daljinskim upravljačem, možemo da ih označimo brojevima od jedan do pet. Kada korisnik usmeri daljinski upravljač u pravcu određene lampe i aktivira neku od komandi, radio prijemnici na lampama prikupljaju merenja. Tri lampe registruju pozitivan odnos snage signala, što znači da je snaga signala koji dolazi sa predajne antene Tx_1 jača od snage signala koji dolazi sa antene Tx_2. Jedna lampa registruje odnos snaga od -1,21 dB, dok lampa u koju je usmeren daljinski upravljač registruje odnos od -15,93 dB. Smatra se da je daljinski upravljač usmeren u neku lampu, ako antena Tx_1 pokazuje u njenom pravcu i ima ortogonalnu orijentaciju u odnosu na nju, dok je antena Tx_2 orijentisana isto kao i antena prijemnika Rx (Slika 7). Na osnovu ovih merenja, algoritam odlučivanja procenjuje da je daljinski upravljač usmeren u pravcu određene lampe i donosi odluku da se ta lampa, npr., uključi.

U cilju poboljšanja verovatnoće tačnog odlučivanja, usrednjavanje rezultata merenja radi se u skladu sa teorijom frekvencijsko-prostorne diversifikacije. Srednja vrednost odnosa snage se računa kao količnik sume snage signala svih tonova poslatih sa prve antene i sume snage signala svih tonova poslatih sa druge antene daljinskog upravljača.

4. ZAKLJUČAK

Odstupanje rezultata merenja od teorijskih vrednosti treba tražiti u načinu prostiranja talasa u realnim uslovima u kojima su eksperimentalna merenja izvršena (odbijanje i difrakcija talasa). U odsustvu gluve sobe (*eng. anechoic chamber*), talas se odbija od raznih prepreka, kao što su zidovi, podovi, radijatori i sl. Dodatno, po Hajgensovom principu, svaka tačka pogođena talasom postaje izvor novog talasa, pa dolazi do efekta poznatog pod imenom difrakcija [5]. Prijemnik ne može da razlikuje osnovni talas od odbijenih, već ih prima sve zajedno, superponirane. To bi mogao da bude glavni razlog zašto je za ugao od -90° detektovana fazna razlika od 38,2° a ne očekivanih 0°. Međutim, kako svrha daljinskog upravljača i nije korišćenje u gluvoj sobi nego u realnim uslovima, merenja u realnim uslovima predstavljaju pravi izbor za procenu efikasnosti metoda.

Obe metode su u stanju da jasno odrede slučaj kada je daljinski upravljač usmeren u ciljnu lampu. Potencijalni problem može da nastane ukoliko se lampe nalaze previše blizu jedna drugoj. Takođe, čovekova ruka koja upravlja

daljinskim upravljačem predstavlja važan faktor, čiji uticaj na prostiranje signala nije ispitan u okviru ovog rada.

Smatra se da je metoda ortogonalnih polarizacija antena, kao jedna od mogućnosti iskorišćenja oblika polja zračenja antene, robustnija od metode fazne razlike. Snaga signala je manje osetljiva na odbijanje i difrakciju talasa u odnosu na fazu talasa. Moguća alternativa ortogonalnoj polarizaciji antena je korišćenje kružnih antena, zato što je kružna polarizacija otpornija na smetnje koje nastaju odbijanjem radio talasa.

ZAHVALNICA

Ovaj rad je delimično finansiran od Ministarstva prosvete i nauke Republike Srbije, projekat 36029, od 2011. godine.

LITERATURA

[1] Ryan Woodings, Derek Joos, Trevor Clifton, Charles D. Knutson: *Rapid Heterogeneous Connection Establishment – Accelerating Bluetooth Inqury Using IrDA*, Brigham Young University, Utah, 2002

[2] Dragan Samardžija: *Bežični komunikacioni sistemi, Digitalne modulacije – skripta*, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, 2010

[3] Radio Technology Systems: *Software Defined Radio*, Datasheet

[4] Matthew G. Parker, Kenneth G. Patersony, Chintha Tellambura: *Golay Complementary Sequences*, Royal Holloway University of London, 2004

[5] W. C. Jakes: *Microwave Mobile Communications*, John Wiley and Sons, New York, 1974

Abstract – The idea of this publication is device pairing in an ad-hoc wireless communication network, based on orientation of a remote controller – turn on a lamp, in whose direction the remote controller is pointing to, without affecting states of other lamps. Two radio-frequency methods have been proposed and evaluated: the phase difference method and the orthogonal antenna pattern method.

EXPERIMENTAL EVALUATION OF RADIO-FREQUENCY METHOD FOR DETERMINING REMOTE CONTROLLER ORIENTATION

Miloš Vasić, Dragan Samardžija, Miodrag Temerinac

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.317:616-7

PROJEKTOVANJE EMBEDDED DIGITALNOG ELEKTROKARDIOGRAFA

DESIGN OF EMBEDDED DIGITAL ELECTROCARDIOGRAPH

Mirjana Vulin, *Fakultet tehničkih nauka, Novi Sad*

Oblast –ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu su prikazane osnove elektrokardiografije, kao i karakteristični oblici EKG signala. Dat je opis hardvera zasnovanog na mikrokontroleru 18F8520 koji je korišćen u realizaciji projekta, kao i projektovani firmver. Funkcionalnost je testirana korišćenjem pretvarača.*

Abstract – *The paper presents the basics of electrocardiography, and explains typical forms of ECG signals. There is a description of hardware-based microcontroller 18F8520, which was used in the project and the designed firmware. The functionality was tested using the converter.*

Ključne reči: *Mikrokontroler, mikroC PRO for PIC, BigPIC5, elektrokardiografija, pojačavač, merni sistem.*

1. UVOD

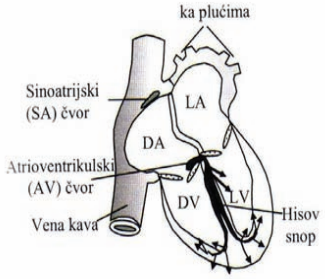
Elektrokardiogram (EKG) je bioelektrični signal nastao aktivnošću srca. Predstavlja najbolji način dijagnostikovanja abnormalnosti u srčanom ritmu, zbog čega je mogućnost dugotrajnog snimanja EKG signala od neizmerne važnosti u kardiološkoj dijagnostici i praćenju stanja pacijenta. Takođe, predstavlja i jednu od najstarijih dijagnostičkih metoda u kardiologiji, koja dolaskom novijih metoda, ne samo da nije izgubila na značaju, već je postala nezamenljiva u dijagnostici mnogih bolesti srca.

2. ELEKTROKARDIOGRAFIJA

Promene biopotencijala koji se javljaju u ćelijama, tkivima, i organima, se nazivaju akcioni potencijali. Registrovanje promena ovih potencijala u vremenu, kao i njihova analiza, daju dragocene podatke o funkcionisanju organa ili tkiva. Elektrokardiografija je metoda registrovanja električne aktivnosti koja prati rad srca. Uređaj koji beleži električnu aktivnost srca u vremenu, i iscrtava elektrokardiogram, je elektrokardiograf. Impulsi koji prolaze kroz srce, šire se i u okolna tkiva, a mali njihov deo dolazi i do površine tela, tj. do kože. Elektrokardiograf je specijalan uređaj osetljiv na električne signale na površini tela. Postavljanjem elektoda uređaja na površinu tela, na dve suprotne strane u odnosu na srce, moći ćemo da registrujemo impulse koji se u srcu javljaju.

2.1. Fiziologija srca

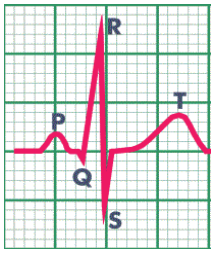
Ljudsko srce se može posmatrati kao dvostuka pumpa, koja se sastoji od četiri komore: leve i desne pretkomore, i leve i desne komore, što je i prikazano na slici 1. Desna pretkomora prima krv iz tela kroz gornju šuplju venu, kontrahuje se i pumpa krv u desnu komoru. Kontrakcijom desne komore pumpa se krv u plućni krvotok. U plućima se krv oksigenizuje, a zatim vraća u levu pretkomoru. Kontrakcija leve pretkomore potiskuje krv u levu komoru, koja se kontrahuje i pumpa krv u sistemski krvotok i to prvo kroz aortu, arterije i arteriole, zatim kroz kapilare svih organa i na kraju kroz vene odakle se vraća u desnu pretkomoru. Ove kontrakcije komora prati depolarizacioni talas, i nakon svakog ovakvog ciklusa dolazi do repolarizacije mišića i njihove relaksacije. Svaki mišić srca može da se kontrahuje samo pod dejstvom električne struje koja kroz njega protiče. To znači da se praćenjem i registrovanjem vremenske promene potencijala mogu dobiti dragoceni podaci o mehaničkom funkcionisanju srca.



Slika 1. *Šematski prikaz srca*

EKG signal se sastoji iz nekoliko karakterističnih talasnih oblika koji se periodično ponavljaju. Karakteristični delovi EKG signala (Slika 2.) su: P-talas, QRS kompleks (sastoji se od Q, R i S talasa) i T talas, gde:

- P talas odgovara depolarizaciji pretkomora.
- QRS kompleks odgovara depolarizaciji komora.
- T talas odgovara repolarizaciji pretkomora i komora.



Slika 2. *Karakteristični delovi EKG signala*

2.2. Istorijski pregled

Osnove za razvoj elektrokardiografije je postavio italijanski fizičar Luiđi Galvani, kada je 1783. godine otkrio da mišići i nervne ćelije stvaraju elektricitet. Tada je osnovni problem bio nepostojanje uređaja osetljivih na

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Zoran Mitrović, vanred.prof.

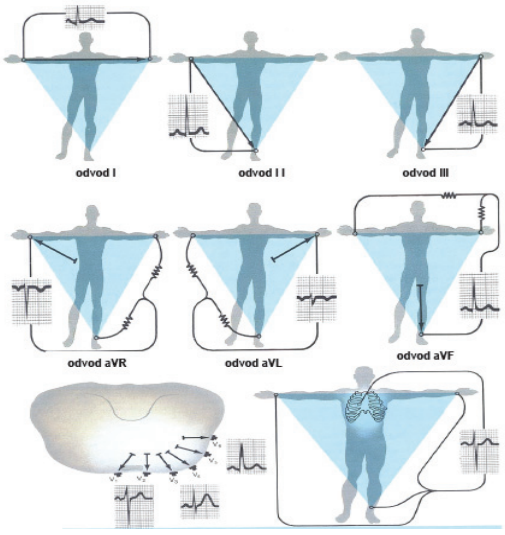
te slabe signale. Upravo iz tog razloga se o počecima elektrokardiografije govori tek u 19. veku, jer je to period u kome se prvi put pojavljuju instrumenti osetljivi na slabe električne signale na površini tela, kakav je i EKG signal. Prvi elektrokardiograf je konstuisao Vilijam Ajnthoven 1903. godine. Uređaj je koristio namotaj veoma tanke žice koja je prolazila između polova potkovičastog magneta. Pri prolasku struje kroz namotaj, elektromagnetno polje bi pomeralo taj namotaj, i žica bi imala mali otklon u odnosu na prvobitni položaj. Izvor svetlosti se nalazio sa jedne strane magneta i bio je paralelno postavljen žici i objektivu fotoaparata (koji se nalazio ispod žice). Snop svetlosti je bio usmeren na žicu, kako bi ona pravila senku nad objektivom fotoaparata, i na taj način je nastajao EKG snimak. Snimak EKG-a je morao posle da se razvija na fotografksom papiru. Orginalna mašina je zahtevala vodeno hlađenje za jake elektromagnete, i bila je teška oko 280 kg. Bilo je teško raditi sa njom, jer je za rad zahtevala prisustvo petoro ljudi. Ipak, ovaj uređaj je predstavljao veliko naučno dostignuće, jer je povećao osetljivost galvanometara toga doba, tako da se električna aktivnost srca mogla meriti na koži.

2.3. Principi elektrokardiografije

Svaki elektrokardiograf sadrži 'odvode', koji predstavljaju električne spojeve između udova i elektrokardiografa. „Odvod“ nije jedini kabelski spoj sa telom, već kombinacija dva kabla i njihovih elektroda, što sa elektrokardiografom zatvara ceo krug.

Postoji ukupno dvanaest EKG odvoda prikazanih na slici 3., i to su:

- Standardni bipolarni (Ajnthovenovi): I; II; III
- Pojačani ekstremitetni (unipolarni): aVR; aVL; aVF
- Prekordijalni: V_1-V_6



Slika 3 *Odvodi elektrokardiografa*

Standardni bipolarni odvodi registruju potencijalnu razliku između dva ekstremiteta, i to:

I odvod: Negativan pol elektrokardiografa se spaja sa desnom rukom, a pozitivan sa levom rukom. Kada mesto na kome se spaja desna ruka sa grudnim košem postane elektronegativno, u odnosu na mesto gde se spaja leva ruka sa grudnim košem, elektrokardiograf će pokazivati pozitivan otklon, i obrnuto.

II odvod: Negativan pol elektrokardiografa se spaja sa desnom rukom, a pozitivan sa levom nogom. Kad desna ruka postane elektronegativna u odnosu na levu nogu, otklon je pozitivan.

III odvod: Negativan pol elektrokardiografa se spaja sa levom rukom, a pozitivan sa desnom rukom. Elektrokardiograf će pokazati pozitivan otklon kada je leva ruka negativna u odnosu na levu nogu.

Ukoliko pri merenju izaberemo jedan potencijal kao referentni, i potom vršimo merenja u odnosu na taj potencijal, merenje nazivamo unipolarnim. Referentni potencijal je uvek povezan na invertovan ulaz ulaznog predpojačavačkog dela EKG uređaja. Zajednički odvod za dve elektrode nazivamo indiferentni odvod, a potencijali koje merimo se označavaju sa aVR, aVL i aVF, gde slovo "a" (od reči "augmented") označava povećan signal u odnosu na standardne odvode.

Postavljanjem elektroda bliže srcu smanjuje se "šum" koji potiče od električne aktivnosti mišića koji se nalaze između mernih tačaka, jer se artefakti pokreta smanjuju. Iz tog razloga se koriste odvodi sa grudnog koša, odnosno prekordijalni odvodi.

3. HARDVER

Osnovni elementi u lancu biomedicinskih merno-akvizicionih sistema su:

- Senzorsko-pojačavački deo;
- Deo za upravljanje i obradu podataka.

3.1. Senzorsko-pojačavački deo

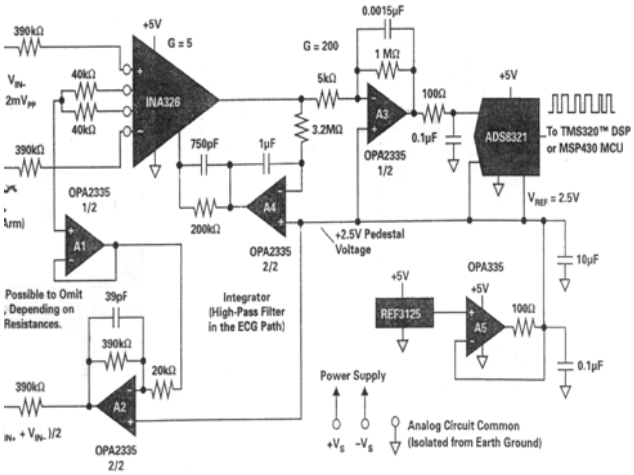
Ovaj deo sastoji se iz (uslovno rečeno) dva dela: dela zaduženog za registrovanje potencijala (elektrode (Slika 4.)), i njegov prenos do elektrofiziološkog pojačavača (oklopnjeni kablovi (Slika 5.)); dok drugi deo čini elektrofiziološki pojačavač (Slika 6.).



Slika 4. *Elektroda*



Slika 5. *Oklopljeni kabel*

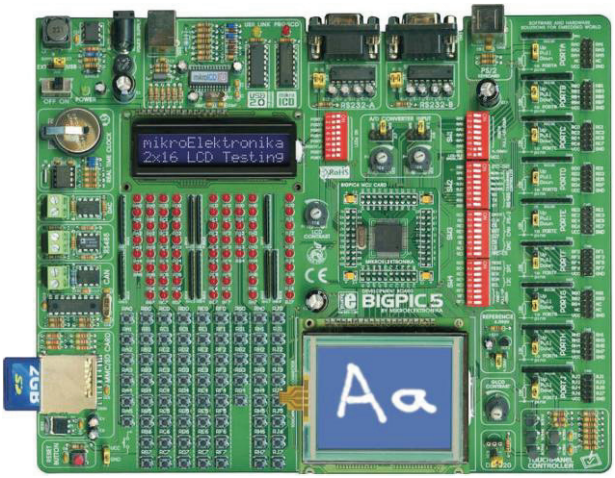


Slika 6. *Elektrofiziološki pojačavač*

Naime, ulazni napon registrovan elektrodama prenosi se oklopljenim kablom do elektrofiziološkog pojačavača. Na njegovom ulazu se nalazi predpojačavač. Čini ga instrumentacioni pojačavač, zbog potrebe za visokom ulaznom impedansom i potiskivanjem zajedničkog napona. Pojačavačko kolo ima pojačanje G=200, ali pored

pojačanja signala vrši i NF filtriranje, i antialiasing filtriranje. Sam pojačavač sadrži dva operaciona pojačavača, tako da je na ovaj način pojačanje razdvojeno na tri dela (ne može se napraviti jedno pojačavačko kolo koje će imati veliko pojačanje, veliku ulaznu impedansu i istovremeno vršiti filtriranje signala).

3.2. Razvojni sistem BigPIC5



Slika 7. *Razvojni sistem BigPIC5*

Mikroelektronikin razvojni sistem BigPIC5 je veoma pogodan za programiranje velikog broja Microchip-ovih PIC 64-pinskih ili 80-pinskih mikrokontrolera. Pruža velike mogućnosti kako studentima tako i inženjerima, da s lakoćom isprobaju i istraže mogućnosti PIC mikrokontrolera. Kao dodatna opcija, omogućeno je povezivanje PIC mikrokontrolera sa eksternim kolima i perifernim uređajima. Korisnik se može u najvećoj meri posvetiti razvoju softvera. Na slici 7. prikazan je razvojni sistem BigPIC5. Kao što se na slici može i videti, pored svake komponente na ploči postoje identifikacione oznake, kako sa gornje tako i sa donje strane. Ove oznake daju opis načina povezivanja sa mikrokontrolerom, radnih modova i takođe pružaju dodatne korisne informacije. Pošto se sve relevantne informacije mogu naći na samoj ploči, gotovo da ne postoji potreba za dodatnim šemama. Prilikom realizacije projektnog zadatka, korišćeni su sledeći moduli razvojnog sistema:

1) MCU (18F8520)
2) AD konvertor
3) GLCD modul
4) MMC/SD modul

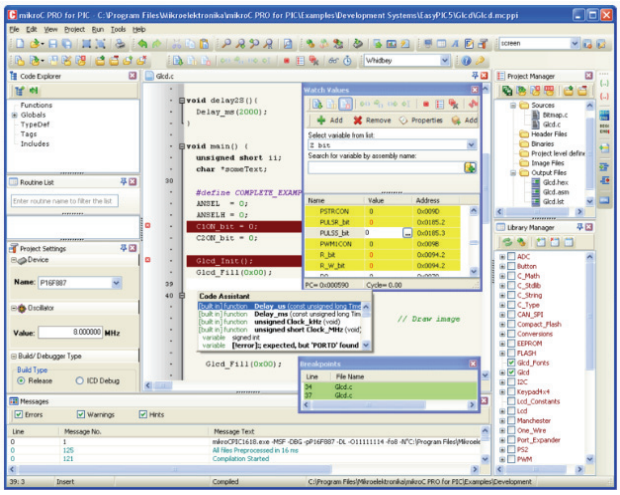
4. FIRMVER

U ovom poglavlju biće reči o razvojnom okruženju u kom je firmver pisan, dosta pažnje biće posvećeno planiranju i pisanju firmvera, kao i problemima prilikom njegove realizacije.

4.1. mikroC PRO for PIC

Razvojno okruženje "mikroC PRO for PIC" je izuzetno moćan, opcijama bogat razvojni alat za PIC mikrokontrolere. Dizajniran je s ciljem da programeru obezbedi najlakši mogući put ka razvoju aplikacija za ugrađene (embedded) sisteme, bez pravljenja kompromisa na štetu performansi ili kontrole nad sistemom. Na slici 8. je dat prikaz razvojnog okruženja, sa nekim od brojnih opcija.

PIC i C su savršena kombinacija, jer je PIC kao najpopularniji 8-bitni čip u svetu našao svoju primenu u brojnim rešenjima, a C, kojem se odaje priznanje zbog svoje efikasnosti, je logičan izbor za razvoj ugrađenih (embedded) sistema.



Slika 8. *mikroC PRO okruženje*

Razvojno okruženje "mikroC PRO for PIC" obezbeđuje uspešno slaganje visoko naprednih IDE (integrated development environment), ANSI kompatibilan kompajler, širok skup hardverskih biblioteka, sveobuhvatnu dokumentaciju, kao i mnoštvo već gotovih primera.

4.2. Razvoj firmvera

U cilju boljeg razumevanja generalne ideje po kojoj je firmver realizovan neophodno je napraviti kratak osvrt na projektni zadatak. Naime, zahtevi projekta su sledeći:

1) Sistem treba da meri potencijale sa površine tela koji potiču od rada ćelija srca.
2) Rezultati se prikazuju grafički na GLCD-u, u talasnom obliku, i u realnom vremenu (što je bitno napomenuti).
3) Rezultati merenja se takođe smeštaju na MMC karticu, u tekstualni fajl.
4) Fajl treba da sadrži podatke o vrednostima napona svakog od karakterističnih EKG talasnih oblika.

Imajući u vidu zahteve projekta, jasno je da se pisanje firmvera odvijalo u etapama. Da bi merni sistem funkcionisao na željeni način bilo je potrebno napisati delove koda koji upravljaju radom određenih hardverskih modula (A/D konvertor, GLCD, i MMC). U skladu s tim, prvi korak podrazumevao je pisanje zasebnih softverskih modula za rad sa pomenutim delovima hardvera. Veliko olakšanje u radu predstavljaju brojni primeri, kao i protipi pojedinih funkcija, dati u sklopu razvojnog okruženja.

Kao logičan nastavak, nakon izrade zasebnih modula za A/D konverziju i GLCD, pojavila se potreba za integrisanjem istih u nešto kompleksnije celine, u cilju uspešnog ispunjavanja projektnog zadatka. Tako da je moj sledeći korak bio spajanje koda koji vrši A/D konverziju, sa kodom koji upravlja radom GLCD-a. Kako se na izlazu A/D konvertora dobijaju vrednosti u opsegu

od 0 do 1023, bilo je neophodno u samom kodu ih obraditi tako da se dobiju vrednosti koje su pogodne za grafički prikaz na GLCD.
Po završetku pisanja firmvera za upravljanje radom A/D konvertora i GLCD-a usledio je razvoj firmvera za upravljanje radom MMC kartice. Sada je bilo potrebno vrednosti sa A/D konvertora, koje se nalaze u opsegu od 0 do 1023, prebaciti u opseg od 0 mV do 3 mV, u kojem se kreću vrednosti karakterističnih talasa EKG signala, i potom ih upisati na MMC karticu. Ovde su se javili problemi. Naime, vrednost karakterističnog S talasa nalazi u opsegu od 0 mV do 0.99 mV, što znači da je bilo potrebno neku od vrednosti iz ovog intervala upisati na MMC karticu. Iz nepoznatog razloga na MMC kartici se dobijala samo nula, nedostajao je broj iza zareza, što nije bio slučaj sa vrednostima većim od 1 mV. Problem sam rešila množeći vrednost sa 100, i upisujući prvo nulu, zatim zarez, i na kraju broj iza zareza (vrednost S talasa pomnoženu sa 100).
Na samom kraju razvoja firmvera ostalo je da objedinim dve potpuno funkcionalne softverske celine (MMC i GLCD+A/D konvertor) u jedinstven programski kod. Najbitniji deo posla je obavljen završetkom pisanja kodova za upravljanje zasebnim hardverskim modulima. Integracija istih je protekla bez većih problema, tako da je finalizacija projekta uspešno izvršena.

5. ZAKLJUČAK

Zahvaljujući zahtevima projektnog zadatka upoznala sam potpuno novu oblast primene mikroprocesorskih mernih sistema. Od samog početka projekta, tema mi se činila krajnje interesantnom, prvenstveno jer se bavi primenom elektrotehnike u oblasti medicine, odnosno medicinskom instrumentacijom. Rad na ovom projektu doveo me je do zaključka, da oblast elektrokardiografije predstavlja plodno tle za dalji, kako tehnički razvoj, tako i naučno-istraživački rad.

6. ZAHVALNOST

Na posletku bih se zahvalila svima, onima koji su, na bilo koji način pomogli realizaciju ovog projekta. Najveću zahvalnost dugujem predmetnom asistentu Platonu Sovilju, na nesebičnoj pomoći i brojnim upustvima, pa mu se ovom prilikom, u to ime zahvaljujem.

7. LITERATURA

[1.] Metting van Rijn A.C., Peper A., Grimbergen C. A. High-quality recording of bioelectric events. Part 1 interference reduction, theory and practice. Med.& Bio Eng. & Comput., 28 (1990), 389-397.
[2.] Fizika ljudskog organizma, Slobodanka Stanković, Univerzitet u Novom Sadu, Prirodno-matematički fakultet, 2004.
[3.] http://en.wikepedia.org/wiki/Electrocardiogram,
[4.] Mikroelektronika, *BigPIC5 User's Manual,* http://www.mikroe.com/
[5.] 2004 Microchip Technology Inc. , *PIC18F6520/8520/6620/8620/6720/8720 Datasheet,* http://www.microchip.com/
[6.] Mikroelektronika, *PICFlash User Manual,* http://www.mikroe.com/

Kratka biografija:

Mirjana Vulin rođena je u Bosanskom Petrovcu 1986. godine. Opštu gimnaziju je završila u Novom Sadu 2005. godine. Od školske 2005/06. je student Fakulteta tehničkih nauka u Novom Sadu.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.317:616-7

DVOKANALNI DIGITALNI ELEKTROENCEFALOGRAF

TWO-CHANNEL DIGITAL ELECTROENCEPHALOGRAPH

Mladen Karaklić, *Fakultet tehničkih nauka, Novi Sad*

Oblast –ELEKTROTEHNIKA I RAČUNARSTVO

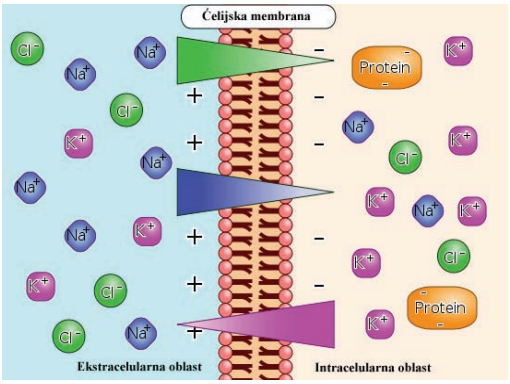
Kratak sadržaj – *U ovom radu opisan je postupak realizacije mikrokontrolerskog sistema za digitalno merenje EEG signala. Za tu svrhu iskorišćen je mikrokontroler PIC18F8520 iz familije PIC mikrokontrolera. Digitalno merenje se vrši internim A/D konvertorom mikrokontrolera. Mikrokontrolerski sistem putem RS232 serijske veze šalje rezultate merenja ka računaru. Za razvoj firmvera korišćen je razvojni sistem mikroC IDE, a za razvoj PC aplikacije programski paket Delphi 6.*

Abstract – *This project describes the realization of microcontroller system for digital measurement of EEG signals. For this purpose, was used microcontroller PIC18F8520 from PIC microcontroller family. Digital measurement is done with internal A/D converter of the microcontroller. Microcontroller system via RS232 serial connection sends measurement results to the computer. For the development of firmware was used developing system mikroC IDE, and for development of PC application was used software package Delphi 6.*

Ključne reči: *Digitalno merenje, Elektroencefalografija, Mikrokontrolerski sistemi, Merenje u realnom vremenu, Merno-akvizicioni softver.*

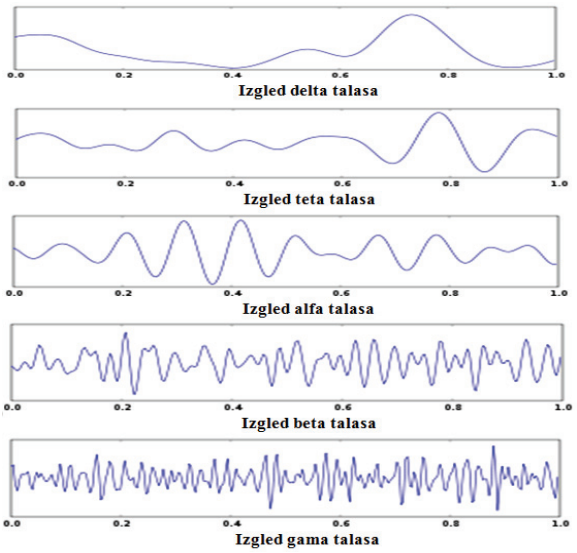
1. ELEKTROENCEFALOGRAFIJA

Elektroencefalografija predstavlja zapisivanje električnih aktivnosti kore velikog mozga. Ta aktivnost potiče od bioelektričnih procesa koji se javljaju kao rezultat životnih funkcija. Svaka funkcija ljudskog organizma ili njegovih delova praćena je odgovarajućom promenom rasporeda naelektrisanja. Ta promena se može prostirati kroz nervni sistem u vidu električnog signala. Biološke ćelije koriste bioelektricitet za skladištenje metaboličke energije, u svrhu obavljanja rada, aktivacije unutrašnjih promena ili međusobne komunikacije. Membranski potencijal (slika 1) je razlika električnog potencijala između unutrašnjosti i spoljašnjosti ćelije. On nastaje interakcijom jonskih kanala i jonske pumpe ugrađene u membranu, koje održavaju različite jonske koncentracije sa intracelularne i ekstracelularne strane membrane. Kod neeksitabilnih i eksitabilnih ćelija membranski potencijal se održava na relativno stabilnoj vrednosti, koja se zove potencijal mirovanja. Akcioni potencijal je kratkotrajan događaj u kojem se električni membranski potencijal ćelije brzo podiže i pada po određenom ustaljenom obrascu. Nastajanje akcionog potencijala odvija se po principu „sve ili ništa“.



Slika 1. *Membranski potencijal*

Da bi se pojavio, potrebno je da spoljni nadražaj ima intenzitet iznad praga nadražaja i dovoljno vreme trajanja. Ako su ti uslovi ispunjeni, akcioni potencijal će imati uvek jednaku amplitudu i isti oblik. Akcioni potencijali kod neurona igraju centralnu ulogu u komunikaciji između ćelija i nazivaju se i nervni impulsi. Mozak sadrži nekoliko milijardi neurona koji generišu i propuštaju električne signale. Ukupna električna aktivnost rezultira signalima koji se mogu detektovati i zabeležiti izvan mozga. Amplituda EEG signala se kreće u intervalu od 5 μV do 200 μV, a frekvencija od 0 Hz do 100 Hz. Prema vrednosti frekvencije, EEG signali se najčešće dele na: delta (do 4 Hz), teta (4 – 8 Hz), alfa (8 – 13 Hz), beta (13 – 30 Hz) i gama talase (od 30 Hz do 100 Hz) (slika 2). Elektroencefalogram se može koristiti za praćenje stanja bolesnika pri operaciji, u slučajevima kada se ne može snimati EKG, kao i za praćenje nivoa anestezije. EEG i ERP (*Event Related Potentials*) snimanja se intezivno koriste u neuronauci, kongitivnoj nauci, kongitivnoj psihologiji i psihofiziološkim istraživanjima.

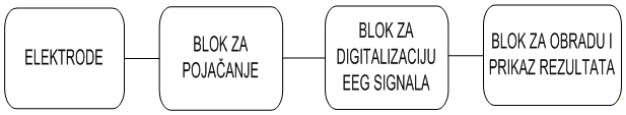


Slika 2. *Vrste EEG signala*

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Zoran Mitrović, vanred.prof.

2. MERNI EEG SISTEMI

Merni EEG sistemi prve generacije su analogni i sastoje se iz elektroda, pojačavača, i iz dela za zapisivanje EEG signala. Za zapisivanje rezultata merenja EEG signala kod ovakvih uređaja koristi se pisaljka i pokretni papirni grafički šablon. Druga generacija mernih EEG sistema se, kao i prva generacija, sastoji iz elektroda, pojačavača i dela za zapisivanje. Razlika je što se za zapisivanje koristi magnetna traka na koju frekvencijski modulisani analogni rekorder snima EEG signal. EEG instrumenti treće generacije zovu se i digitalni merni EEG sistemi jer osim elektroda i pojačavača sadrže i digitalni deo, odnosno blok za digitalizaciju EEG signala, pa je i zapisivanje kod ovih sistema drugačije. Blok šema ovakvog uređaja je prikazana na slici 3.



Slika 3. *Blok šema digitalnog EEG sistema*

2.1. Elektrode

Merenje biopotencijala se vrši pomoću električnih instrumenata koji su sa biološkim sistemom povezani preko elektroda. Elektrode mogu biti različitog oblika ili sastava. Da bi se mogla izmeriti potencijalna razlika u nekom biološkom sistemu potreban je par elektroda koje sačinjavaju dve metal-elektrolit polućelije. Elektrode se najčešće prave od metala koji su provodnici prvog reda, dok tkiva predstavljaju provodnike drugog reda. Na dodirnoj površini između metalnih elektroda i tkiva javljaju se nepoželjni efekti koji utiču na tačnost merenja. To su kontaktni potencijal i polarizacija elektroda. Potencijalana razlika između površine metala i formiranog sloja pozitivnih jona naziva se kontaktni potencijal. Razlika kontaktnih potencijala dve elektrode predstavlja grešku pri merenju. Da bi izmerili pravu vrednost potencijalne razlike potrebno je kontaktni potencijali na elektrodama budu jednaki. To se postiže korišćenjem elektroda od istog metala. Prilikom prolaska struje kroz elektrolit dolazi do elektrolize koju prati izdvajanje njenog produkta na elektrodama. Taloženje tog produkta može promeniti osobine elektroda i dovesti do njihove polarizacije. Struja polarizacije, koja se pri tome javlja, utiče na tačnost merenja biopotencijala. Za otklanjanje ovog nepoželjnog efekta koriste se elektrode napravljene od srebrohlorida nanesenog na pločicu od srebra. Za što bolji kontakt između elektrode i kože koriste se provodne paste kojima se smanjuje impedansa kontakta i obezbeđuje fiksiranje elektroda. Pasta mora da bude hipoalergijska i adhezivna. Kod dugotrajnih merenja, elektrode se dodatno pričvršćuju medicinskim flasterima. Za merenja sa velikim brojem elektroda koriste se kape sa elektrodama koje olakšavaju fiksiranje i prostorno raspoređivanje elektroda. Najčešće se koristi standardni sistem za postavljenje elektroda 10-20.

2.2. Pojačavači EEG signala

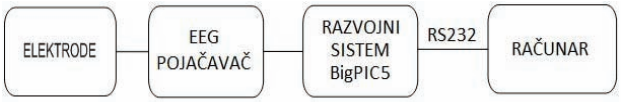
EEG signali su vrlo male amplitude, pa ih je potrebno pojačati, a neželjene smetnje eliminisati. Za pojačanje ovako malih signala koriste se pojačavači visokog pojačanja, preko deset hiljada puta. Osim pojačanja uloga pojačavača je i smanjenje uticaja šuma na koristan signal. U ljudskom telu i kablovima EEG uređaja se indukuju smetnje koje potiču od mrežnog napajanja i drugih okolnih uređaja (fluoroscentnog svetla, mašina, kompjutera i slično). Te smetnje se indukuju zbog kapacitivne sprege između ljudskog tela i kablova EEG uređaja i vodova mrežnog napajanja. Jedna od posledica ove sprege je stvaranje zajedničkog napona koji po amplitudi može biti znatno veći od merenog signala. Za otklanjanje ovih smetnji koriste se diferencijalni pojačavači, ali zbog malih vrednosti signala, upotreba klasičnih diferencijalnih pojačavača nije pogodna. Umesto njih koriste se instrumentacioni pojačavači. Ulazna impedansa ovih pojačavača ima veoma veliku vrednost, reda GΩ, a faktor potiskivanja zajedničkog napona (CMRR) iznosi 100 dB. Kablovi EEG uređaja se dodatno štite oklopljavanjem. Još jedan način za otklanjanje zajedničkog napona je korišćenje DRL (*Driven Right Leg*) kola. DRL kolo „preuzima“ zajednički napon sa ulaza u instrumentacioni pojačavač, invertuje ga i preko elektrode vraća na pacijenta. Invertovani zajednički napon se poništava sa zajedničkim naponom na telu pacijenta. Smetnje prouzrokovane kapacitivnom spregom sa mrežnim napajanjem se eleminišu i korišćenjem notch filtra. Zbog neophodnosti velikog pojačanja signala pojačavački blok se obično realizuje iz više delova da bi se obezbedila linearnost. Prvi deo je zaštitno kolo koje štiti pacijenta kao i elektronska kola od elektrostatičkog pražnjenja. Drugi deo je predpojačavač, koji ima ulogu da eliminiše zajednički napon i pojača diferencijalni napon. Treći deo je glavni pojačavački stepen, on pojačava signal i na njegovom ulazu i izlazu se postavljaju VF filtri niske granične frekvencije da bi se eliminisala jednosmerna komponenta koja može dovesti pojačavač u zasićenje. Četvrti deo je izlazni pojačavački stepen, osim što pojačava analogni signal on ga i priprema za A/D konverziju. U sklopu pojačavačkog bloka se nalazi i sistem za auto-kalibraciju EEG uređaja, sa čijeg se kraja signal za kalibraciju dovodi na ulaze pojačavača.

2.3. Digitalizacija i obrada EEG signala

Blok za digitalizaciju kod digitalnih EEG sistema ima ulogu da konvertuje analogni EEG signal u digitalni i pripremi ga za dalju obradu. Konverzija se vrši A/D konvertorima. Dobijeni digitalni signal se dalje, najčešće šalje ka računaru radi čuvanja, dodatne obrade i prikaza rezultata merenja. Operacije koje se vrše prilikom A/D konverzije su uzorkovanje, kvantovanje i kodovanje. Uzorkovanje se vrši u jednakim vremenskim intervalima. Za frekvenciju odabiranja se uzima vrednost koja je najmanje dva puta veća od najveće frekvencije posmatranog EEG signala. Ako taj uslov nije zadovoljen može doći do pojave aliasinga, odnosno preklapanja spektra signala, pa rezultati merenja ne bi bili merodavni. Da bi se obezbedilo da ne dođe do pojave aliasinga potrebno je da signal bude frekvencijski ograničen. To se postiže korišćenjem NF filtra, odnosno anti-aliasing filtra čija je granična učestanost jednaka najvećoj frekvenciji EEG signala koji se posmatra. Mikrokontroleri koji se danas mogu naći na tržištu najčešće imaju u sebi ugrađen A/D konvertor, što može u velikoj meri uprostiti izradu digitalnog dela EEG uređaja. Uloga računara u digitalnim EEG sistemima je memorisanje rezultata merenja, njihova digitalna obrada i vizuelni prikaz.

3. STRUKTURA REALIZOVANOG SISTEMA

U ovom radu najviše pažnje je posvećeno realizaciji bloka za digitalizaciju EEG signala kao i bloka za obradu i prikaz rezultata merenja (slika 3). Blok šema realizovanog sistema prikazana je na slici 4.



Slika 4. *Blok šema projektnog zadatka*

Zadatak ovog rada je razvoj dela za digitalizaciju i obradu signala digitalnog EEG sistema. Deo za digitalizaciju se sastoji od mikrokontrolera, u čijem se sastavu nalazi A/D konvertor, i RS232 modula za komunikaciju sa računarom. Na ulaz A/D konvertora se dovode simulirani signali, jer je senzorsko-pojačavački deo izostavljen. Digitalni odmerci sa izlaza A/D konvertora se pomoću mikrokontrolera i RS232 modula šalju ka računaru. Po prijemu, podaci se konvertuju u realne brojeve koji predstavljaju ulazne vrednosti u pojačavač.

3.1. Razvojni sistem BigPIC5

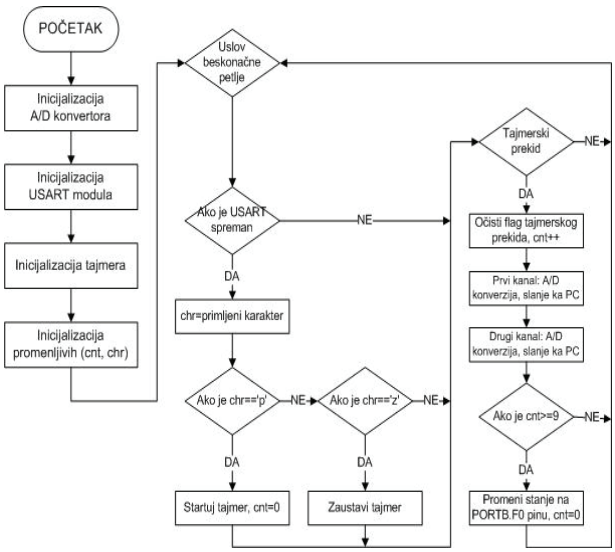
Razvojni sistem BigPIC5 predstavlja koristan alat za programiranje skoro svih Microchip-ovih PIC 64-pinskih ili 80-pinskih mikrokontrolera. Osim u edukativne svrhe, može se koristiti i u naučno-istraživačkim radovima za testiranje PIC-ovih mikrokontrolera, za testiranje softvera za mikrokontrolere, kao i za programiranje mikrokontrolera. Ovaj razvojni sistem omogućava povezivanje PIC mikrokontrolera sa eksternim komponentama (LCD, GLCD, eksterni A/D konvertor i slično) i perifernim uređajima. Razvojni sistem BigPIC5 može koristiti jedan od dva izvora napajanja – napajanje sa kompjutera preko USB kabla (uobičajeno) ili eksterno napajanje sa adaptera (jednosmerno ili naizmenično). Centralnu jedinicu razvojnog sistema BigPIC5 predstavlja slot za MCU karticu sa 64-pinskim ili 80-pinskim PIC mikrokontrolerom. Pomoću ovog slota, mikrokontroler je povezan sa ostalim modulima na ploči. Prilikom izrade ovog projekta korišćena je MCU kartica sa mikrokontrolerom PIC18F8520 i oscilatorom 10 MHz. Mikrokontroler u sebi sadrži A/D konvertor i potrebno je obezbediti vezu ulaza A/D konvertora sa simuliranim signalom. Jedan od komunikacionih modula razvojne ploče BigPIC5 je i RS232 modul pomoću koga se uspostavlja veza između mikrokontrolera i računara. S obzirom da naponski nivoi mikrokontrolera i računara nisu direktno kompatibilni jedan sa drugim, mora se koristiti uređaj za prilagođavanje naponskih nivoa, kao što je MAX232.

3.2. Komponente razvojnog sistema BigPIC5

Najbitnije karakteristike mikrokontrolera PIC18F8520 su posedovanje Timer0 modula (8-bitni/16-bitni tajmer/brojač), dva USART modula sa mogućnošću adresiranja (podržava RS485 i RS232) i 10-bitni, 16-kanalni A/D konvertor sa mogućnošću konverzije u *Sleep* modu. Tajmer predstavlja važan deo mikrokontrolera. Većina tajmera se može konfigurisati da generišu interapt signal kada dosegnu određen broj (obično kada se desi prekoračenje, *overflow*). Interapt signal se može iskoristiti da mikrokontroler izvrši preciznu vremenski-zavisnu naredbu. Zahtevi projekta bili su da frekvencija semplovanja merenog signala bude 256 Hz i generisanje kalibracionog signala frekvencije 14 Hz. Ovi zahtevi su rešeni korišćenjem tajmera. Jednačina kojom se izračunava željena frekvencija tajmera je $\Delta f=(f_{\mu C}/4)/PRESCALER$, gde $f_{\mu C}$ predstavlja frekvenciju mikrokontrolera, a PRESCALER broj pomoću koga se podešava frekvencija tajmera. Vreme za koje se sadržaj u registru tajmera inkrementira za jedan je $\Delta t=1/\Delta f$. T0CON registar je registar iz koga mogu da se čitaju i u koji mogu da se upisuju vrednosti bita koji kontrolišu sve parametre Timer0 modula. Režim tajmera se bira setovanjem bita T0CS na „0“. TMR0 interapt se generiše kada dođe do prekoračenja (*overflow*) TMR0 registra prilikom prelaska vrednosti sa FF u heksadecimalnom sistemu na 00 u heksadecimalnom sistemu u 8-bitnom modu ili sa FFFF heksadecimalno na 0000 heksadecimalno u 16-bitnom modu. Ovo prekoračenje setuje TMR0IF bit na „1“. Pre ponovnog omogućavanja ovog interapta, TMR0IF bit mora da se setuje na „0“ u softveru. PIC18F8520 mikrokontroler sadrži 10-bitni A/D konvertor modul i on omogućava konverziju ulaznog analognog signala, koji odgovara desetobitnom digitalnom broju. Izlaz kola za *sample and hold* je ulaz za A/D konvertor, koji rezultat dobija putem sukcesivnih aproksimacija. Opseg analognog signala koji se konvertuje mora biti od 0 V do 5 V i on ulazi u opseg prihvatljiv za PIC mikrokontroler. Za skladan i sinhron rad delova mikrokontrolera koriste se ujednačeni impulsi koji se generišu u oscilatoru. Stabilizacija frekvencije se najčešće obezbeđuje korišćenjem kristala kvarca ili keramičkog rezonatora. Instrukcije u mikrokontroleru se ne izvršavaju brzinom koju diktira oscilator, već nekoliko puta sporije. Razlog za to je što se svaka instrukcija izvršava u nekoliko koraka. Prilikom izrade rada korišćen je kvarcni rezonator koji obezbeđuje frekvenciju oscilatora od 10 MHz.

4. FIRMVER

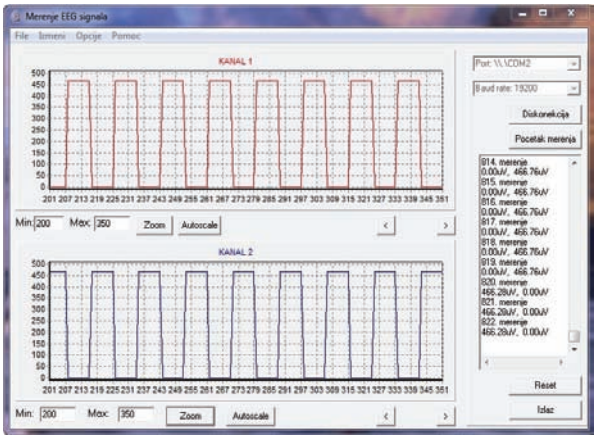
Source kod firmvera, kao i sam firmver realizovani su u razvojnom okruženju mikroC for PIC. Razvojno okruženje mikroC for PIC je moćan i opcijama bogat razvojni alat za PIC mikrokontrolere. Dizajniran je tako da pruži korisniku najlakše moguće rešenje za razvoj aplikacije za ugrađene (*embedded*) sisteme, bez kompromisa na štetu performansi ili kontrole sistema. Razvojno okruženje mikroC for PIC generiše čitljivu asemblersku verziju koda, kao i HEX verziju, kompatibilnu sa svim programatorima. Programator PICflash sa mikroICD podrškom predstavlja izuzetnu alatku za programiranje mikrokontrolera iz familija PIC16F i PIC18F kompanije Microchip. Prilikom realizacije firmvera potrebno je proći kroz nekoliko faza. Prvu fazu predstavlja definisanje zahteva koje mikrokontroler treba da ispuni. Ti zahtevi su da mikrokontroler prihvati analogni signal sa dva kanala, da ih konvertuje u digitalni i zatim prosledi ka računaru. Frekvencija A/D konverzije je 256 Hz, a kalibracionog signala pravougaonog oblika 14 Hz. Sledeću fazu realizacije firmvera predstavlja testiranje rada pojedinačnih modula mikrokontrolera. Testirani su Timer0 modul, A/D konvertor i USART modul. Naredna faza realizacije firmvera je integracija prethodnih test programa u jednu celinu. Uprošćena verzija algoritma source koda je prikazana na slici 5.



Slika 5. *Algoritam sorce koda firmvera*

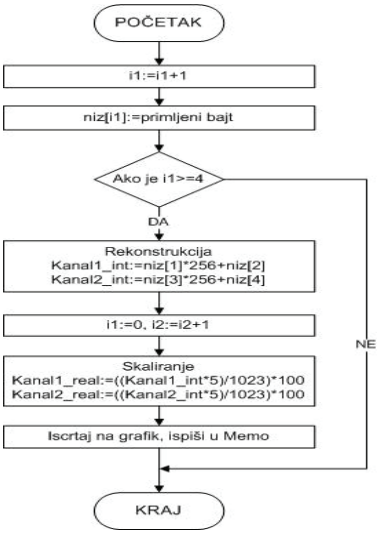
5. PC APLIKACIJA

PC aplikacija realizovana je u programskom paketu Delphi 6. Programski paket Delphi je strogo tipiziran objektno orijentisan profesionalni programski jezik, relativno lak za upotrebu i zasniva se na dobro dizajniranom programskom jeziku Pascal. Sastavni deo programskog paketa Delphi je VCL (*Visual Component Library*), uključujući većinu izvornog koda. Zadatak PC aplikacije u ovom projektu je prikupljanje podataka koje šalje mikrokontroler, njihova obrada i grafički i numerički prikaz rezultata merenja. Zahtevi koje PC aplikacija (slika 6) treba da ispuni su uspostavljanje komunikacije sa mikrokontrolerom, rekonstrukcija prispelih podataka, njihova obrada i prikaz. Za uspostavljanje komunikacije računara sa mikrokontrolerom potrebno je definisati komunikacioni port i njegove parametre kako ne bi došlo do greške prilikom prenosa podataka. Najvažniji parametri koje treba definisati su port na koji se povezuje mikroračunar i brzinu prenosa podataka (*baud rate*). Klikom na dugme „Konekcija“ uspostavlja se veza sa mikrokontrolerom.



Slika 6. *Izgled prozora PC aplikacije*

Klikom na dugme „Pocetak merenja“ pokreće se tajmer mikrokontrolera. Prilikom interapta, koji generiše tajmer, mikrokontroler započinje A/D konverziju, i po završetku konverzije šalje podatke, bajt po bajt, ka računaru. Veoma je bitno znati ovu informaciju, jer se na osnovu nje u PC aplikaciji piše kod koji prispele podatke pretvara u brojeve koji odgovaraju vrednosti A/D konverzije. Uprošćena verzija algoritma ovog dela koda PC aplikacije prikazana je na slici 7.



Slika 7. *Algoritam dela koda PC aplikacije u kome se primaju i obrađuju podaci*

6. ZAKLJUČAK

Ponovio bih da prilikom izrade ovog rada nije razmatran ceo digitalni sistem za merenje EEG signala, već samo njegovi delovi za digitalizaciju signala i prikaz rezultata merenja. Zbog izostanka elektroda i pojačavača, za testiranje realizovanog sistema koriste se simulirani signali. Rezultati merenja dobijeni na ovaj način su više nego ohrabrujući i u potpunosti odgovaraju očekivanim rezultatima pri korišćenju simuliranog signala. Dalji rad u ovom pravcu predstavlja izazov, jer analiza ovih signala danas sve više nalazi primenu u medicinskoj dijagnostici i naučnim istraživanjima.

Zahvaljujem se svima koji su na bilo koji način doprineli realizaciji ovog projekta. Najveću zahvalnost dugujem predmetnom asistentu Sovilj Platonu za nesebičnu pomoć, savete i smernice prilikom izrade ovog rada.

7. LITERATURA

[1] S. Stanković, „*Fizika ljudskog organizma*“, PMF, Novi Sad, 2006.

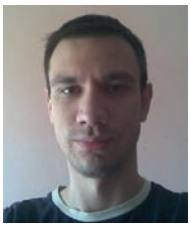
[2] P. Sovilj, „*Stohastičko digitalno merenje EEG signala*“, doktorska teza, FTN, Novi Sad, 2010.

[3] http://en.wikipedia.org/wiki/Electroencephalography

[4] „*BigPIC5 User's Manual*“, Mikroelektronika

[5] M. Cantu, „*Mastering Delphi 6*“, SYBEX

Kratka biografija:



Mladen Karaklić rođen je u Užicu 1985. Srednju tehničku školu u Užicu završio je 2004. Od školske 2004/05 student je Fakulteta tehničkih nauka u Novom Sadu.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 502

SOLARNI IZVORI ELEKTRIČNE ENERGIJE - PREGLED REŠENJA ZA MALE POTROŠAČE

SOLAR ELECTRIC POWER SOURCE - SURVEY OF SOLUTIONS FOR SMALL LOADS

Slobodan Lolić, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj: *Ovaj rad opisuje sistem koji omuogućuje napajanje potrošača malih snaga iz solarnih izvora. Realizovan je u cilju određivanja karakteristika i parametara fotonaponskih (FN) panela, pri radu u realnim uslovima. Pokazane su mogućnosti pri ostrvskom radu i pri priključenju na javnu mrežu, kao i potencijalni AC ili DC potrošači. Urađena je i kratka analiza isplativosti kroz ekonomske pokazatelje. Praktičan deo rada je pumpa za navodnjavanje kap-po-kap.*

Abstract: *This paper describes the system that enables low power supply of consumers. The system is implemented in order to determine the characteristics and parameters of photovoltaic (PV) panels, which operates in real conditions. Options are shown for islanding operation, connection to public grid, as well as loads which are of AC or DC characters. Feasibility studies have been done through economic indices. As a practical part the irrigation pump drop-by-drop is realized.*

Ključne reči: *Fotonaponski panel, invertor, regulator, baterija, LED dioda,*

1. UVOD

Sunčeva energija je obnovljiv i praktično neograničen izvor energije. Sunčevo zračenje, uz sekundarne solarne izvore kao što su vetar, energija talasa i biomasa, se računa u najčešće dostupne izvore energije na Zemlji. Trenutno se upotrebljava samo mali deo solarne energije od ukupne raspoložive količine.

2. PREGLED I SISTEMATIZACIJA ZA MALE POTROŠAČE

Osnovni elementi fotonaponskog (FN) sistema su: FN paneli, akumulator (baterija), invertor, regulator (kontroler), potrošač. Potrošači mogu biti AC i DC karaktera. Jedan povezan FN sistem prikazan je na slici 1.
Fotonaponski - FN (Photovoltaic - PV, engl.) paneli su uređaji za konverzija energije sunčevog zračenja u električnu. Solarni panel je u stvari niz, koji se dobija spajanjem velikog broja FN ćelija, koje kada se izlože svetlosti na svojim krajevima daju određeni jednosmerni napon. Obično je veličina jedne ćelije nekoliko kvadratnih centimetara i daje snagu do jednog vata. Solarni panel definiše se kao grupa od nekoliko modula, koji su električno povezani u serijsko-paralelnoj kombinaciji, a koji je snage 160-250 W.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Vladimir Katić.

Pretvaranje energije Sunca u električnu energiju omogućeno je primenom fotoelektričnog efekta, po čemu su ove ćelije i paneli i dobili ime.
Dobijena električna energija iz sunčanih ćelija akumulira se u olovnim akumulatorima sa elekrolitom u obliku gela jer su oni predviđeni da rade u režimu dubokog pražnjenja za razliku od akumulatora koji se koriste u automobilima. Akumulator služi za korišćenje energije u noćnim satima ili u vremenskim intervalima kada fotonaponski panel nema dovoljnu snagu da napaja potrošače.
PV modul daje DC napon i zbog AC potrošača potreban je DC/AC transformacija. Invertor je pretvarač na čiji ulaz se dovodi DC napon, a na izlazu on proizvodi željeni AC. Invertor se karakteriše preko koeficijenta efikasnosti η.
loga invertora je da stvara konstantan napon na AC strani i da pretvara snagu na ulazu P_{in} u snagu na izlazu P_{out} sa maksimalnom mogućom efikasnošću. To nam pokazuje izraz 1.

$$\eta = \frac{P_{out}}{P_{in}} = \frac{V_{ac} I_{ac} \cos\varphi}{V_{dc} I_{dc}} \quad (1)$$

Kontrola punjenja se koristi da reguliše tok energije u PV sistem, baterije i potrošače prikupljajući informacije o naponu baterije i poznavajući prihvatljive maksimalne i minimalne vrednosti za napon baterije. Postoje dva glavna načia rada kontrolera:

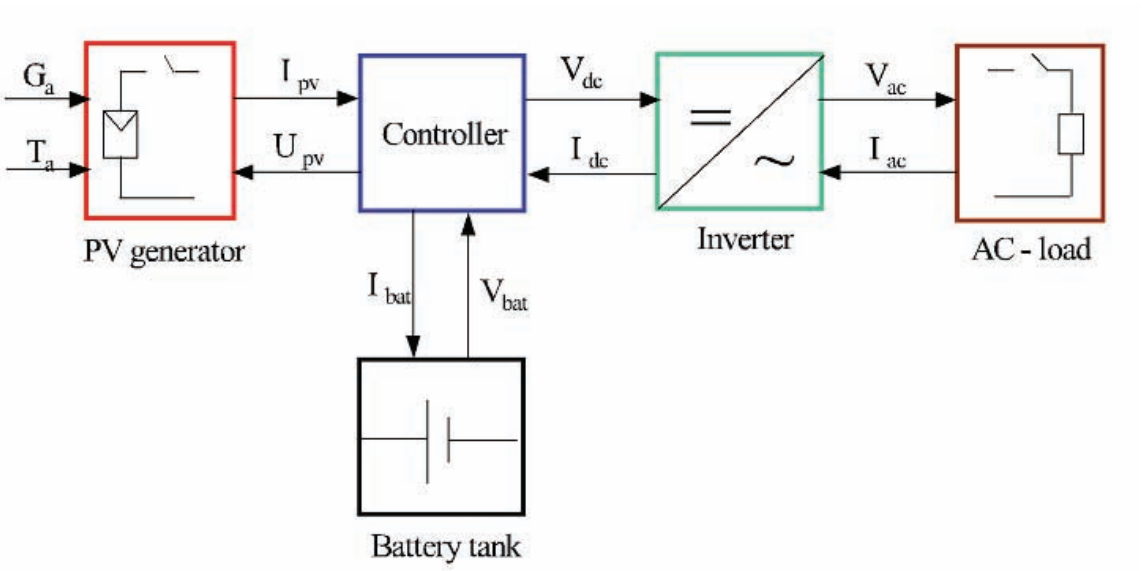
- Normalni radni uslovi, kada se napon baterije nalazi u dozvoljenim granicama
- Stanje prepunjenosti i prepražnjenosti, koji se javljaju kada baterija dostigne kritične vrednosti

Potrošač priključen na PV sistem može biti DC ili AC. Ukoliko je potrošač naizmeničan sistem zahteva uvođenje invertora koji će pretvoriti proizvedenu DC snagu u naizmeničnu koja odgovara potrošaču.

2.1 Sistem za navodnjavanje kap po kap

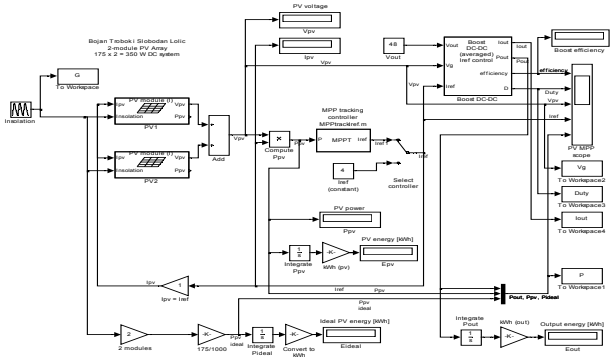
Ulazni parametri sistema su sunčevo zračenje i izlazni napon. Izlazni napon je postavljen na vrednost 96V, a za sunčevo ozračenje su uzimane minimalne i maksimalne vrednosti u toku godine koje se javljaju pri čistom nebu ili u proseku.

Svi slučajevi su testirani za solarni modul koji je postavljen sa fiksnim optimalnim uglom u odnosu na horizont i za solarni modul koji ima mogućnosti rotiranja kako bi na što veću efektivnu površinu padali sunčevi zraci.



Slika 1. *Uopštena šema fotonaponskog sistema*

Princip rada ovog sistema je sledeći: sunčevi zraci padaju na fotonaponski panel, u kome se pomoću fotoelektričnog efekta vrši konverzija energije Sunca u električnu energiju, koja pokreće pumpu. Sistem sadrži regulator MPPT 500, koji radi na principu traženja maksimalne tačke radne snage FN panela. Delovi sistema su: bunar dubine 25m, rezervoar od 1000l postavljen na visinu od 2,5m iznad zemlje, potopna pumpa, plovak koji se postavlja u rezervoaru, 2 solarna panela snage 175W, solarni regulator MPPT 500, 3 sonde potpoljene u bunar, grebenasti prekidači, osigurači, ampermetar, voltmeter, vatmetar, piranometar.



Slika 2. *Simulink model fotonaponskog sistema*

Tri sonde se spuštaju u bunar, i kablom PPL 3x2,5 mm^2 povezuje sa nivostatom. Sonde će na osnovu stanja u kome se nalaze (mokre/suve) slati nivostatu informaciju da li u bunaru ima dovoljno vode. Plovak, koji se nalazi u rezervoaru, povezan je na nivostat kablom PPL 2x2,5 mm^2.

U zavisnosti od položaja plovka i od informacija dobijenih od sondi, nivostat uključuje/isključuje pumpe na napon od 96V DC. Kada je pumpa priključena na napon, rezervoar će se puniti vodom. Takođe, u okviru ovog sistema nalaze se solarni paneli, koji predstavljaju izvore napajanja. Ukoliko je dan sunčan, solarni panel će prikupljenu energiju pretvoriti u električnu, i kablovima P/F 6 mm^2 (provodnik plave boje na - pol, provodnik crvene boje na + pol) proslediti u razvodni orman.

2.2. Primena FN panela u javnoj rasveti

Solarna rasveta nalazi svoju primenu u osvetljavanju ruralnih područja, bezbedonosnom i sigurnosnom osvetljenju, osvetljavanju dokova, parkova, tranzitnom osvetljenju, osvetljavanju otvorenih prostora, osvetljavanju objekata vojne i civilne bezbednosti. Sastavni delovi kola su:LED svetiljke (P=30W), monokristalni solarni panel (P=185W), GEL baterija 120Ah/12V (potrebne su dve baterije), kontroler, stub lampe, montažni matrijal i kablovi.

Troškovi konvencionalnog postavljanja javne rasvete, kopanje kanala i postavljanje podzemnih kablova čine sistem solarne rasvete u mnogim slučajevima ekonomski daleko povoljnijom opcijom.



Slika 3. *Primena FN panela u javnoj rasveti*

2.3. Primena FN panela u saobraćaju

Za napajanje saobraćajnog znaka potrebni su sledeći elementi energetske elektronike: fotonaponski paneli, invertori, akumulatori, regulatori, diode. Diode su neophodne zbog osvetljenja samog objekta, tj. saobraćajnog znaka.

Zadatak podrazumeva, pre svega energetsku efikasnost i ekonomičnost, tako da odabrano rešenje bude optimalno kako po pitanju kvaliteta osvetljaja, tako i po pitanju ukupnih troškova.

Parametar koji je primaran za svaki izvor svetlosti je osvetljenje izraženo u lumenima. Lumen (lm) je jedinica iz SI sistema, određena različitom osetljivošću oka s obzirom na talasne dužine svetlosti. Efikasnost osvetljenja je odnos emitovane energije vidljive svetlosti i ukupne ulazne energije u sijalicu.



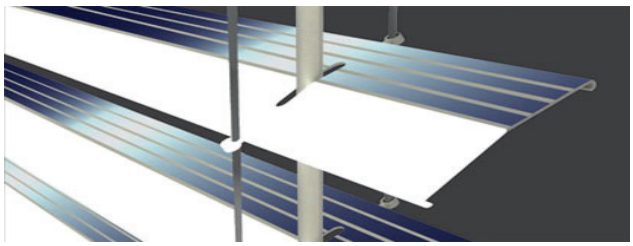
Slika 4. *Napajanje saobraćajnog znaka pomoću FN panela*

2.4. Ekološki venecijaneri

Ekološki venecijaneri su još jedna ideja koja se može realizovati u praksi. Umesto da samo blokiraju sunčevu svetlost oni i prikupljaju energiju preko dana, pomoću thin-filma, da bi je preko noći iskoristili za osvetljavanje, na primer stona lampa.

Solarni paneli su postavljeni na celoj površini venecijanera, baterija se nalazi u postolju, a za osvetljenje se koristi novi super efikasni electroluminescentni sloj sa druge strane venecijanera.

Održavanje ne traži velike zahteve. Prašina se može obrisati suvom krpom, samo treba biti obazriv jer je reč o jako tankom materijalu, koji je sklon grebanju, pa može dovesti do smanjenja efikasnosti.
Na slici 5 prikazano je postavljanje thin film-a na venecijanere, koji ima ulogu sakupljača sunčeve energije, koja se može iskoristiti preko noći za osvetljavanje, npr. stona lampa.



Slika 5. *Ekološki venecijaneri*

2.5. Primena FN panela za napajanje lovačkih kuća i vikendica

Objekti bez električne energije gotovo u nezamislivi u danjašnje vreme. Ali, električna energija nije dostupna baš svakom domaćinstvu, odnosno objektu. Razloga postoji puno, a najčešći su udaljenost od javne mreže, mali broj domaćinstava, česte migracije stanovništva, ekološki uslovi. Jednostavno kompletna infrastruktura ne omogućava dotok električne energije. Samim tim, potrebno je naći aletrnativni način za napajanje takvih objekata. Napajanje solarnim panelima jedan je od načina koji to omogućavaju.



Slika 6. *Napajanje lovačkih kuća i vikendica pomoću FN panela*

Unutar objekta mogu se nalaziti mnogi električni uređaji neophodni za boravak. Najčešće su to potrošači malih snaga, kao što su mini frižider, rešo, TV prijemnik, radio, ventilator, fen, kao i osvetljenje. Dakle, vidi se da su to većinom uređaji naizmeničnog karaktera. Za dobijanje električne energije potrebne za napajanje ovih uređaja neophodni su nam sledeći elementi: fotonaponski paneli, akumulator, invertor, regulator, kao i kablovi kojima se povezuju ovi elementi u celinu.

3. ZAKLJUČAK

U ovom radu opisan je princip rada solarnih sistema, kao i svih njegovih elementarnih delova (solarnih panela, baterijskog sistema, regulatora napona, invertora).

Solarna energija se pretežno koristi tamo gde je napajačka mreža nedostupna ili tamo gde je nemoguće napajati se iz nekog drugog izvora energije. Tržište fotonaponskih solarnih ćelija unazad nekoliko godina eksponencijalno

raste. Solarna energija ima niz prednosti besplatna je, upotreba fotonaponskih modula ne zagađuje okolinu, postrojenja nakon postavljanja rade i uz malo održavanja. Nedostaci su mala iskoristivost (<30%) i potreba sistema za akumulaciju energije, što povećava cenu.

4. LITERATURA

[1] Branko L. Dokić, Energetska elektronika, Elektro-tehnički Fakultet, Banja Luka, 2000

[2] http://www.amazon.co.uk/Planning-Installing-Photovoltaic-Systems-Installers

[3] http://www.solarnipaneli.org

[4] A.Lugue, S.Hegedus, Hand book of photovoltaic science and engineering, Wiley, 2003

Kratka biografija:



Slobodan Lolić rođen je u Šapcu 1987. godine. Master rad na Fakultetu tehničkih nauka iz oblasti elektrotehnike i računarstva -energetske elektronike odbranio je jula 2011. godine.



Vladimir Katić je redovni profesor Fakulteta tehničkih nauka u Novom Sadu za oblast Energetska elektronika, mašine i pogoni. Prodekan je za finansije FTN-a i šef Katedre za energetsku elektroniku i pretvarače. Oblasti interesovanja su energetska elektronika, kvalitet električne energije i obnovljivi izvori električne energije.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004

JEDNOSTAVNI SCENARIJI ZA TESTIRANJE PERFORMANSI DMS SOFTVERA

SIMPLE SCENARIOS FOR PERFORMANCE TESTING OF DMS SOFTWARE

Dragoljub Grubović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu opisan je način testiranja performansi softvera uopšte. U radu je objašnjena uprošćena arhitektura DMS softvera i PMT alata koji je korišćen za testiranje performansi. Prikazani su scenariji koji su korišćeni za testiranje, kao i rezultati koji su dobijeni.*

Abstract – *This paper describes one of the many ways for testing performance of the software. Also it explains simple architecture of the DMS software and PMT tool which is used for conducting our tests. This methodology is conducted for performance testing of DMS software. It explains results which are obtained by testing.*

Ključne reči: *elektrodistributivni system, softver, testiranje, performanse*

1. UVOD

U današnje vreme dostignut je nivo kada očekujemo da se programski zahtevi izvršavaju u deliću sekunde. Svi su zainteresovani za razvoj softverskih sistema sa vrlo visokim performansama u cilju zadovoljavanja zahteva kupaca. Zbog toga, zahtevi za poboljšanjem performansi dobijaju sve veći značaj u razvoju industrijskog softvera.

Ranije, kada se nije pridavalo toliko značaja testiranju performansi organizacije su kupovale skupe hardvere da bi rešile probleme vezane za performanse. U današnje vreme kompanije ulažu značajna sredstva za testiranje performansi i počinju da vrše testiranja u ranim fazama razvoja softvera. U određenom krugu ljudi i dalje vlada mišljenje da testiranje performansi predstavlja samo korišćenje alata za simulaciju velikog opterećenja i identifikaciju vremena odziva raznih transakcija. Ipak, to nije tako jednostavno kao što se misli.

Testiranje performansi je drugačiji način posmatranja sistema. Važan preduslov za sprovođenje testiranja je njegova funkcionalna stabilnost odnosno, da bi testiranje performansi imalo smisla, softver treba da bude u završnoj fazi razvoja. Testiranje performansi ne bi imalo smisla ukoliko se kod tokom testiranja menja i ukoliko sistem ne sadrži implementirane osnovne funkcionalnosti.

Rad sadrži opis metodologije merenja performansi softvera kao i osnovne karakteristike i arhitekturu DMS softvera i PMT alata. DMS softver je softver za praćenje, kontrolu i analizu sistema za distribuciju električne energije. Za testiranje DMS softvera korišćen je PMT (*Performance Measurment Tool*) alat koji je za potrebe testiranja performansi razvijen u kompaniji Telvent DMS. Pored toga, rad sadrži i jednostavne scenarije koji su sprovedeni radi testiranja performansi DMS softvera, kao i analizu rezultata koji su dobijeni [1].

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Aleksandar Erdeljan, docent.

2. TESTIRANJA PERFORMANSI SOFTVERA

2.1. Uvod u metodologiju testiranja

Testiranje performansi predstavlja proces procene softvera ili komponenata hardvera i sprovodi se radi određivanja skalabilnosti aplikacije, odnosno njene sposobnosti da se nosi sa dodatnim opterećenjem bez negativnih uticaja na performanse. Takođe, procena se vrši radi potvrđivanja usaglašenosti performansi testiranog sa ranije postavljenim ciljevima. U slučaju neslaganja sledi optimizacija performansi. Da bi se u potpunosti postigao cilj testiranja potrebno je sprovesti ga po tačno isplaniranom planu rada.

Sa testiranjem performansi često se poredi funkcionalno testiranje. Međutim, treba praviti razliku između te dve vrste testiranja. Testiranje prvog zavisi od uspešnosti testiranja drugog odnosno, softver mora biti funkcionalno stabilan da bi testiranje performansi bilo validno. Ako svi preduslovi nisu ispunjeni treba sačekati sa testiranjem performansi dok se to ne dogodi.

Testiranje performansi je esencijalna aktivnost u razvoju softvera. Vrlo je važno da se izbegnu iznenađenja u zadnjem trenutku, tj. nakon što se aplikacija postavi u radno okruženje. Svi zastoji u radu koje primeti krajnji korisnik predstavljaju prednost za naše konkurente [3].

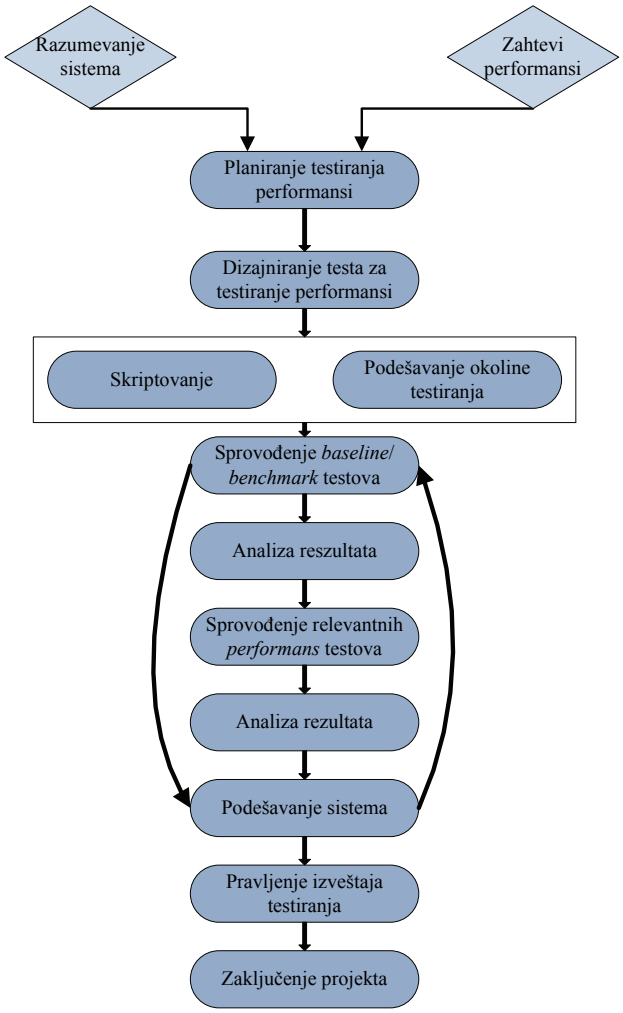
2.2. Menadžment testiranja performansi

Postoje dva pristupa testiranju performansi: reaktivan i proaktivan pristup.

U reaktivnom pristupu se ne vodi briga o performansama od samog početka razvoja softvera. Obično se čeka neki pad u performansama da bi se pristupilo njihovom testiranju. Čekanje da se problemi kod performansi pojave i da se tek onda rešavaju pokazao se kao lošiji pristup testiranju i u ovakvom slučaju je očekivano da se ne dobiju najbolji rezultati. Ovaj pristup je više „sredi kasnije“ pristup i nije efikasan.

Nasuprot njemu, proaktivan pristup jako dobro predviđa probleme koji bi mogli da se jave i adaptira tehnike koje koristi da bi se što bolje ublažili problemi. Proaktivan pristup podrazumeva da se performanse softvera testiraju i podešavaju u ranim fazama razvoja softvera. Kod ovog pristupa testiranje performansi se smatra važnim za vreme svih faza razvoja sistema [2].

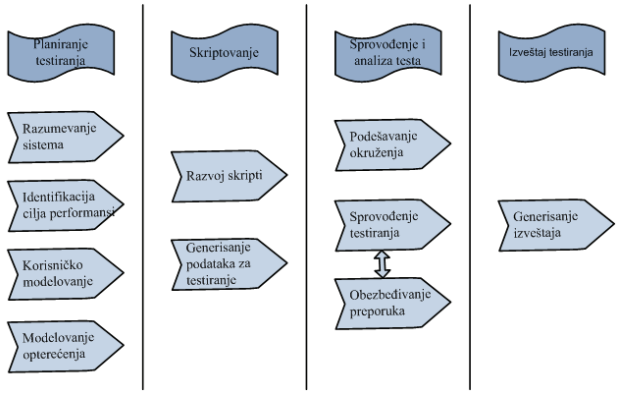
Kod oba pristupa testiranja treba da postoji proces koji će sadržati tačno definisane korake u testiranju performansi. Postoji više pristupa za sprovođenje ovog testiranja. Na slici 1. dat je primer jednog od najuspešnijih pristupa koji je sproveden u testiranju DMS softvera.



Slika 1. *Pristup testiranju performansi*

Planiranje testiranja performansi je faza u kojoj se proučavaju svrha i različite funkcionalnosti sisitema. Cilj testiranja performansi nekog sistema mora biti dobro shvaćen da bi se testiranje što bolje sprovelo.

Skriptovanje je faza u kojoj se scenariji testiranja performansi, koji su nastali tokom faze planiranja, skriptuju. Test skripte se generišu korišćenjem alata za testiranje performansi. Skripta simulira korisnički pristup sistemu u realnom vremenu.



Slika 2. *Faze u testiranju performansi*

Izvršavanje testa je faza u kojoj se planirani testovi izvode i performanse sistema potvrđuju. Pre nego što se izvrše planirani testovi preporučuje se sprovođenje *baseline* testa za jednog virtualnog korisnika. Cilj je potvrđivanje test skripte i prikupljanje metrike sistema za opterećenje jednog korisnika.

Pravljenje izveštaja testiranja je faza tokom koje se vrše analize testiranja, a preporuke se dokumentuju u razumljivom formatu i prezentuju zainteresovanim stranama. Vrlo je važno da izveštaj bude prezentovan detaljno i praćen objašnjenjem za tzv. uska grla koja su identifikovana tokom ciklusa testiranja kao i preporukama [3].

Sve gore navedene faze predstavljaju životni ciklus testiranja performansi, kao što je prikazano na slici 2.

3. ALAT ZA TESTIRANJE PERFORMANSI

Performance Measurment Tool (PMT) je alat koji obezbeđuje informacije o performansama DMS softvera. Svrha ovog alata je da meri metrike performansi koje su od značaja i koje mogu ukazati na postojanje određenih uskih grla, odnosno uzročnika lošeg ponašanja sistema. PMT je implementiran kao GUI i konzolna aplikacija. Za potrebe verifikacije performasi DMS softvera definisan je konfiguracioni fajl, nazvan profil. Jedan profil sadrži metrike za verifikaciju željenih performansi DMS softvera. Dodatno je definisan i profil za dimenzionisanje hardvera i skalabilnosti, isto kao i za generalnu svrhu merenja performansi.

Postoje tri nivoa (stanja) prezentacije za svaku kategoriju merene performanse: zeleni, žuti i crveni nivo. Zeleni nivo predstavlja stanje softvera u kome merenja zadovoljavaju zahteve. Žuti nivo predstavlja stanje softvera u kome je njegovo ponašanje zadovoljavajuće ali odredjeni parametri pokazuju da je sistem opterećen preko stepena dozvoljenog za zeleni nivo (eng. *yellow threshold*) definisanog u konfiguracionom fajlu. Ako se posmatra iskorišćenje procesora to je interval od 50 do 80%. Crveni nivo predstavlja stanje softvera u kome su performanse ozbiljno degradirane.

4. DMS SOFTVER

DMS (*Distribution Management System*) softver je osnovni proizvod kompanije „Telvent DMS“. Funkcija DMS softvera je praćenje, kontrola i analiza sistema za distribuciju električne energije. Da bi DMS softver vršio svoju funkciju svi njegi delovi moraju efikasno da izvršavaju svoje pojedinačne funkcije. Prilikom rada DMS softvera očekivano dolazi do opterećenja sistema što može izazvati pad performansi te je značajno merenje performansi tokom njegovog razvoja.

DMS softver je složen, sastavljen od većeg broja komponenti organizovanioh u slojeve. U osnovi, softver ima troslojnu arhitekturu koja sadrži:

- prezentacioni (klijenti),
- poslovni (servisi) i
- sloj baze podataka.

Svaki sloj arhitekture sadrži međusobno povezane komponente gde svaka od njih ima svoje funkcionalnosti.

Testiranje performansi DMS softvera sprovodi se radi postizanja ranije definisanih zahteva za performansama. Na taj način se eliminiše kasnija degradacija performansi.

U ovom radu su opisani rezultati testiranja iskorišćenja CPU-a. Prikazani su rezultati opterećenje celog sistema, mada su merena i opterećenja pojedinačnih komponenti.

5. SCENARIJI

Kod DMS softvera mogu nastati dve promene i to analogna i diskretna promena. Analogna promena nastaje prilikom prijema nekog merenja u mreži, dok diskretna promena nastaje kod prijema stanja nekog prekidača. Za rad je interesantan uticaj obe vrste promena na performanse, tj. na iskorišćenje CPU-a. Osmišljeni su jednostavni scenariji koji će dati odgovore o uticaju prethodno navedenih promena na posmatrane performanse.

Logika merenja u svim slučajevima je sledeća: prva dva minuta nema opterećenja DMS softvera, zatim se simulira scenario sledećih 15 minuta, nakon čega se čeka još 5 minuta pre nego što prestane merenje performansi. Ovakva logika je sprovedena da bi se uočile razlike u opterećenju CPU-a kada je softver u stanju neopterećenosti, zatim kada je opterećen, kao i ponašanje performansi nakon prestanka simuliranja scenarija. Učestanost ponavljanja scenarija se može menjati po potrebi.

Sprovedeni su sledeći scenariji:

- Menjanje stanja prekidača koji isključuje 16 potrošača.
- Menjanje merenja struje u određenoj podstanici i fideru.
- Menjanje stanja dva prekidača koji se nalaze na različitim korenima mreže.

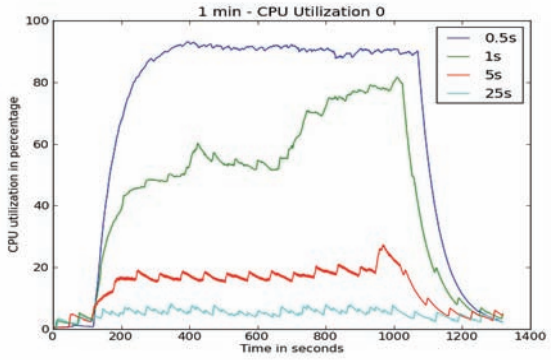
U ovom slučaju posmatrano je ukupno opterećenje CPU-a po jezgrima.

U nastavku rada data je analiza rezultata dobijenih merenjem performansi DMS softvera za sva tri scenarija.

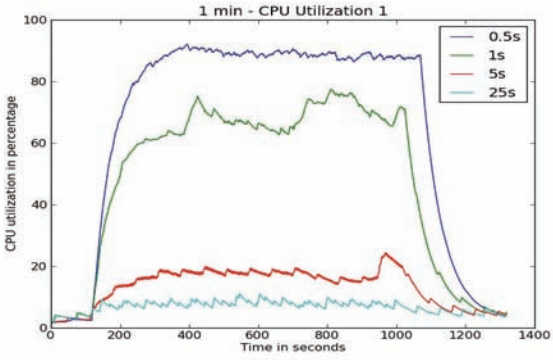
5.1 Analiza rezultata dobijenih menjanjem stanja jednog prekidača

Menjanje stanja prekidača dovodi do nastanka diskretne promene u DMS softveru. Kada dođe do promene stanja prekidača pokreću se proračuni tokova snaga za deo mreže na koji ta promena utiče. Sprovedena su merenja za promenu stanja na 0.5, 1, 5 i 25 sekundi čiji su uporedni grafici dati na slikama 4. i 5.

Za promene na 5 i 25 sekundi iskorišćenje CPU-a za oba jezgra nije prešlo 20% pa samim tim ni prag žute zone (50%). Merenje sa promenom stanja na 1 sekundu relativno brzo prelazi prag od 50% i ulazi u žutu zonu. Najbrža promena stanja (0.5 sekundi) opterećuje najviše CPU. Opterećenje raste jako brzo dok ne pređe prag crvene zone (80%) nakon čega opterećenje ulazi u saturaciju i ostaje takvo do prestanka simulacije scenarija.



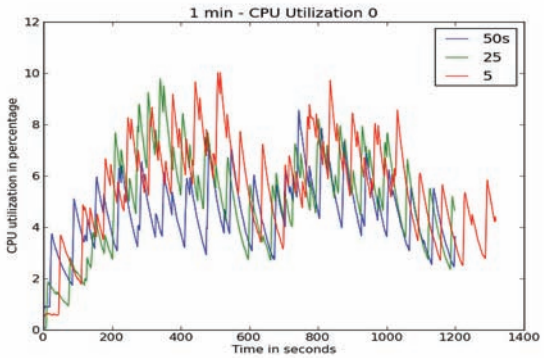
Slika 4. *Iskorišćenje CPU-a za prvo jezgro, prvi scenario*



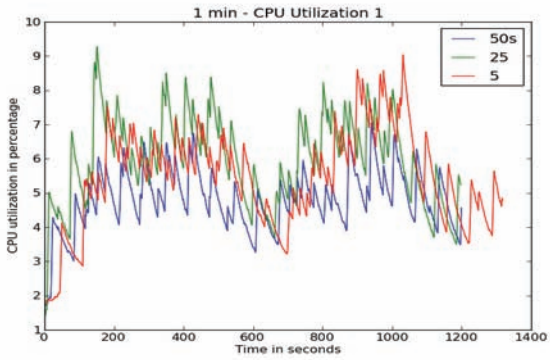
Slika 5. *Iskorišćenje CPU-a za drugo jezgro, prvi scenario*

5.2 Analiza rezultata dobijenih menjanjem merenja struje u određenoj podstanici i fideru.

Ovaj scenario predstavlja uvid u iskorišćenje CPU-a u slučaju analogne vrste promene kod DMS softvera. Kada dođe do promene merenja, DMS vrši estimaciju stanja mreže.



Slika 6. *Iskorišćenje CPU-a za prvo jezgro, scenario drugi*



Slika 7. *Iskorišćenje CPU-a za drugo jezgro, scenario drugi*

Ovaj scenario predstavlja promenu merenja (jačine struje) na izabranoj postanici i fideru (TSH12 i *Feeder_55*) gde je estimirana vrednost bila 1660A. Sprovedena su merenja za analogne promene na 5, 25 i 50 sekundi. Pored toga, vršena je i promena u amplitudi u intervalu od 1200 do 2100A sa raskorakom promene jednakim ritmu promene (5, 25, 50A).

Na slikama 6. i 7. prikazana su iskorišćenja CPU-a za sva tri merenja.

Kod sva tri merenja primećuje se da iskorišćenje CPU-a po jezgrima ne prelazi 10%. Zaključuje se da analogna

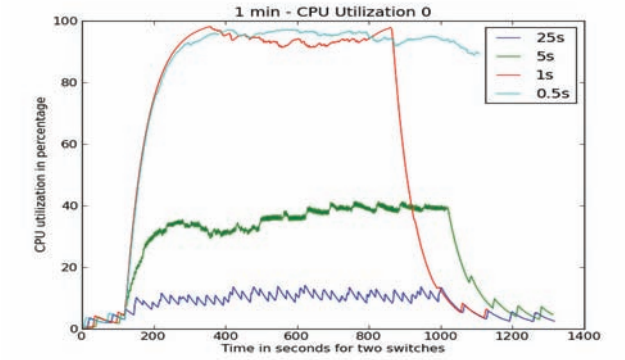
vrsta promene manje degradira performanse sistema za razliku od diskretne promene. Na primer, ako se uporede promene na 5 sekundi za prethodni i ovaj scenario vidi se da kod diskretne promene dolazi do duplo većeg iskorišćenja CPU-a (20%).

5.3 Analiza rezultata dobijenih menjanjem stanja dva prekidača na različitim korenima.

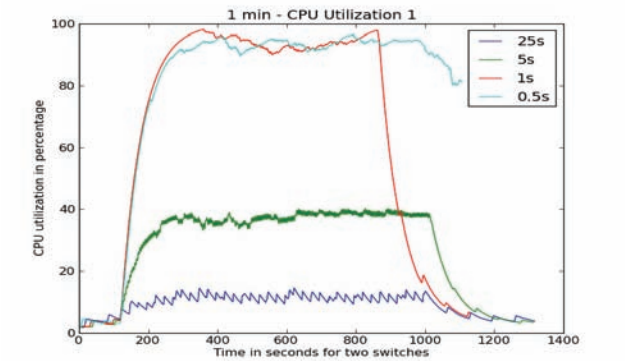
Treći scenario predstavlja nešto složeniji oblik prvog. U ovom slučaju dolazi do promene stanja dva prekidača koji se nalaze na različitim korenima. Kada ih istovremeno menjamo vrše se proračuni tokova snage za dve podmreže. Za rad je bilo interesantno kako će ove promene uticati na performanse. Kao i kod prvog scenarija sprovedena su četiri merenja za promene stanja prekidača, i to na 0.5, 1, 5 i 25 sekundi.

Na slikama 8. i 9. dati su uporedni grafici za sva četiri merenja.

Za promene stanja na 5 i 25 sekundi iskorišćenje CPU-a neće preći prag žute zone (50%) ali se primećuje veće opterećenje nego kod prvog scenarija. Na primer, za promenu stanja na 5 sekundi iskorišćenje CPU-a je za dva puta veće nego u prvom scenariju (tj. teži ka 40%). Za merenja promene na 0.5 i 1 sekunde primećuje se brz prelazak praga crvene zone (80%), a iskorišćenje ostaje iznad 90%. Za najbržu promenu (0.5 sekundi) nastavlja se opterećenje CPU-a i nakon prestanka simulacije scenarija.



Slika 8. *Iskorišćenje CPU-a za prvo jezgro, scenario treći*



Slika 9. *Iskorišćenje CPU-a za drugo jezgro, scenario treći*

5. ZAKLJUČAK

Rezultati primene ovih testova ukazali su na potencijalne ranjivosti jednog realnog sistema, te stoga mogu pomoći da se izbegnu otkazi prilikom upotrebe softvera, nastalih kao posledica otkrivenih ranjivosti. U ispitivanom slučaju pokazane su promene u iskorišćenju CPU-a kada je izložen određenoj vrsti opterećenja, kao i u kojim slučajevima dolazi do ozbiljne degradacije posmatrane performanse.

Treba napomenuti da je ovo testiranje validno za određeni hardver računara na kom se nalazio DMS softver.

LITERATURA

[1] E.J.Weyuker, F.I.Vokolos, *Experience with performance testing of software systems: issues, an approach, and case study*, IEEE Transactions on Software Engineering, Vol 26 (12), 2000, pp. 1147 - 1156

[2] Johann du Plessis, *Performance Testing Methodology* (whitepaper), Micro to Mainframe, 2008

[3] Ramya Ramalinga Moorthy, *Software Performance Testing Handbook - A Comprehensive Guide for Begineers*, ProcessWox, Bangalore, India, 2005



Dragoljub Grubović, rođen u Beranama 1986. godine. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacije odbranio 2011. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.391

NAGIB SPEKTRA SIGNALA SRČANOG RITMA

SPECTRUM SLOPE OF HEART RATE SIGNAL

Anđelko Pikula, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj - *Cilj rada jeste provera mogućnosti dobijanja značajnih dijagnostičkih parametara određivanjem nagiba spektra signala srčanog ritma. Signal je dobijen sa odraslih mužjaka Wistar pacova i granično hipertenzivnih pacova. Na pacovima je izvršena hirurška intervencija (ugradnja transmitera u trbušnu aortu životnje, u cilju merenja kardiovaskularnih parametara). Date životinje su zatim bile izložene uticaju dva tipa stresa: stresu protresanja i stresu ograničenja i mjereni su njihovi kardiovaskularni parametri. Rezultati mjerenja (u vidu signala) su dalje filtrirani i obrađeni u programskom okruženju Matlab.*

Abstract – *The aim of this paper is to check the possibility of diagnostic feature extraction by estimating the spectrum slope of heart rate. The signal was recorded from the adult male Wistar rats and Borderline Hypertensive rats. The rats were submitted to surgical intervention (insertion of transmitter in the abdominal aorta of the animal, in order to measure cardiovascular parameters). These animals were then exposed to two types of stress: shaker stress and restraint stress and their cardiovascular parameters were measured. The measurement results (in the signal form) are further filtered and processed in Matlab programming surrounding.*

Ključne riječi: *PSV, nagib spektra, srčani ritam, stres protresanja i stres ograničenja*

1. UVOD

Od osamdesetih godina prošlog vijeka pa do danas uočena je značajna povezanost između autonomnog nervnog sistema i smrtnosti zbog kardiovaskularnih problema, uračunavajući tu i smrt usljed iznenadnog prestanka rada srca. Podaci, do kojih se došlo eksperimentalnim putem, a koji ukazuju na povezanost između sklonosti ka fatalnim aritmijama i znakova o povećanju simpatičke ili smanjenju vagalne aktivnosti, podstakli razvoj kvantitativnih markera autonomne aktivnosti. Promjenljivost srčanog ritma (HRV- *Heart Rate Variability*) predstavlja jedan od najznačajnih markera. Srčani ritam HR obrnuto je srazmeran trajanju srčanog otkucaja, odn. vrednosti rastojanja između uzastopnih maksimuma signala pritiska (PI), odnosno između uzastopnih R vrhova elektrokardiogramskih signala (RR).

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je dr Dragana Bajić, red.prof.

Posljednjih godina velika pažnja bila je posvećena obliku spektra dugoročnih snimaka (24h) varijabilnosti srčanog ritma. Izvesna spektralna gustina snage invezno proporcionalna frekvenciji (1/f model), uočena je na raznim poljima fizike i biologije, da bi u poslednje vreme bila povezana i sa haotičnom dinamikom, osobinom samosličnosti i drugim osobinama nelinearnih i kompleksnih fenomena. Pošto su mehanizmi koji su uključeni u generisanje HRV-a, kompleksni, i pošto se određene nelinearnosti moraju razmotriti, parametri koji mere osobine determinističkog haosa RR serija su uneti u proučavanja autonomnih mehanizama za kontrolu srčanog ritma. Ovi novi parametri dali su interesantne rezultate u istraživanjima u vezi kardiovaskularnih bolesti kao što su esencijalna hipertenzija, kod koje autonomni nervni sistem izgleda igra važnu ulogu.

Ovaj rad bavi se ponašanjem spektralne gustine snage signala i pokušajem njegove aproksimacije korišćenjem 1/f modela.

2. MATERIJALI I METODE

2.1. Životinje

U eksperimentu su korišćeni Wistar pacovi i granično hipertenzivni pacovi, koji su F1 potomci Wistar ženki i spontano hipertenzivne serije. Pacovi su odgajeni na Institutu za farmakologiju pri Medicinskom fakultetu Univerziteta u Beogradu. Svi pacovi su muškog pola kako bi se izbegle hormonske interakcije. Pacovi su težine 330 ± 20 g i pojedinačno su odgajani u kavezima od pleksiglasa u kontrolisanim laboratorijskim uslovima (temperatura 21 ± 1 ° C, relativna vlažnost vazduha 60 ± 5%, 12h svetla i 12h mraka).

2.2. Operacija

Deset dana pre eksperimenta pacovi su bili izloženi teleometrijskoj operaciji

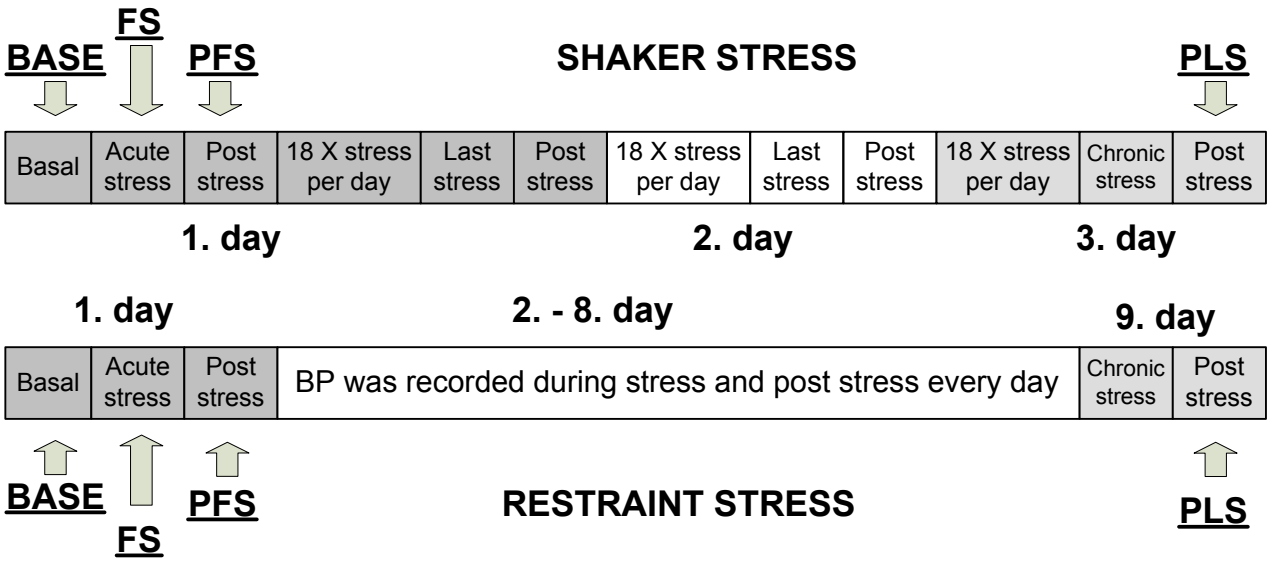
Telemetrija-ubacivanje transmitera u trbušnu aortu, zahvaljujuci čemu mogu da se registruju kardiovaskularni parametri. Sonde su ugrađene u abdominalnu aortu (TA11PA – C40, DSI, Transoma medical). Tri dana pre i posle operacije, pacovi su tretirani *gentamicinom* (25 mg/kg i. m). Posle operacije, pacovi su primili po jednu injekciju *metamizola* (200 mg/kg i. m) protiv bolova, i ostavljeni su da se potpuno oporave.

Signal arterijskog krvnog pritiska (BP) je digitalizovan na 1000 Hz i povezan sa PC-em opremljenim sa Dataquest

A.R.T. 4.0. softverom za analizu kardiovaskularnih signala.

Životinje su bile izložene uticaju dva tipa stresa: stresu protresanja (eng. *shaker stress*), sa pacovima pozicioniranim na platformi koja se tresla 200 ciklusa/min, i stresu ograničenja (eng. *restraint stress*), sa pacovima smještenim u pleksiglas ograničenoj cijevi (prečnika 5.5cm sa porama) u tromom položaju. Krvni pritisak (BP) je sniman pre stresa (BASE), tokom prvog izlaganja stresu (FS) i posle prvog izlaganja stresu (PFS). Nakon toga životinje su izložene hroničnom stresu, nasumičnim potresima 18 puta po danu tokom 3 dana, ili 60 minuta ograničenja tokom 9 dana, kao što je prikazano na Fig. 1. Period posle stresa kod izlaganja u poslednjem danu, takođe je bio snimljen (PLS). Dva tipa pacova (BHR i NRM) su bili izloženi uticaju dva tipa stresa (protresanju i ograničenju), iz kojih su proistekle 4 različite eksperimentalne grupe Shaker-NRM (SN), Shaker-BHR (SB), Restraint-NRM (RN), Restraint-BHR (RB) i, uključujući vremenom formirane signale stresa (BASE, FS, PFS i PLS) 16 različitih eksperimentalnih ishoda. Pulsni intervali (PI) od otkucaja do otkucaja izvedeni su iz arterijskog signal pritiska (BP) kao interval između maksimuma pulsnog talasnog signala BP-a. Ovi vremenski nizovi pažljivo su vizuelno pretraživani i filtriran zbog artefakata, nadostajućih otkucaja, koji bi mogli da utiču na spektralnu estimaciju, u većini slučajeva na automatski način. Proces filtriranja je brisao otkucaje, dok su signali od pacova sa tragovima nestabilnog zdravlja bili kompletno isključeni.

Konačan broj preostalih životinja po eksperimentalnoj grupi se kretao od 6 do 7. To je bilo zadovoljavajuće prema promenljivosti parametara u kontrolisanoj grupi pacova (statistički softver "Power Sample Size Calculation").



Slika 1. *Hronološki prikaz izlaganja stresu i njegovog snimanja*

2.3. Velčov metod

Metode za estimaciju spektralne gustine snage (eng. *Power Spectral Density,*PSD) su razne. Neparametarske metode, se odlikuju estimacijom PSD-a direktno iz signala.

Najednostavnija od takvih metoda je perdiogram. Velčova (eng. *Welch*) metoda je alternativni postupak za poboljšanje karakteristika perdiograma.

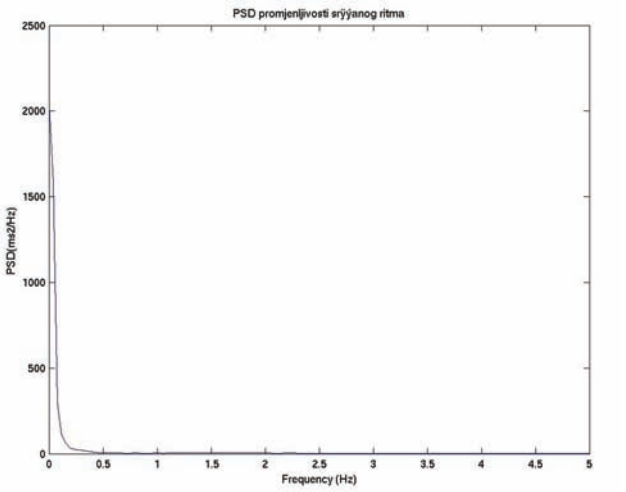
Za razliku od ostalih neparametarskih metoda kod Velčovog metoda se prilikom podele sekvence na segmente dozvoljava da se segmenti preklapaju. Kod nje se takođe uvodi i prozorska funkcija u izračunavanju perdiograma, a kao rezultat se dobija modifikovani perdiogram.

Prvi korak je detrendovanje pulsnih intervala (PI) , odnosno oduzimanje srednje vrijednost da bi se odstranila velika DC komponenta, odnosno sporo promenljiva komponenta – trend.

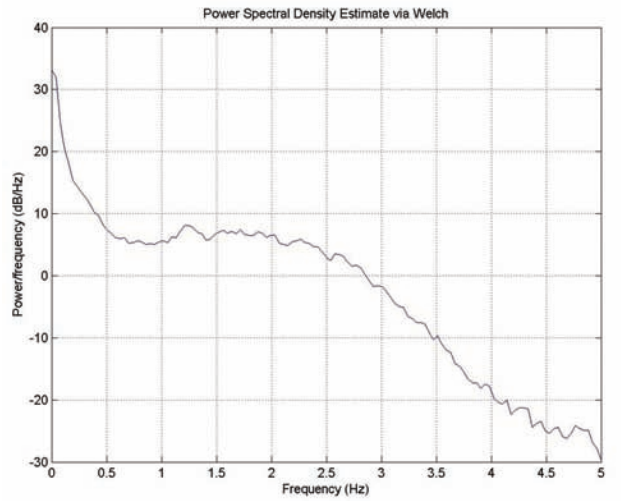
Zatim se od tako dobijene vrijednoti koriste za dobijanje vrijednosti u ekvidistantnim trenutcima. Da bi iz tih rezultata dobili PSD koristimo Velčov metod sa dužinom prozora od 256 i preklapanjem od 75 %.

3. Rezultati

Na slikama 2,3 i 4 prikazan je primjer dobijanja nagiba spektra srčanog signala pacova tipa BHR prije izlaganja stresu (BASE) ograničenja (Restraint).

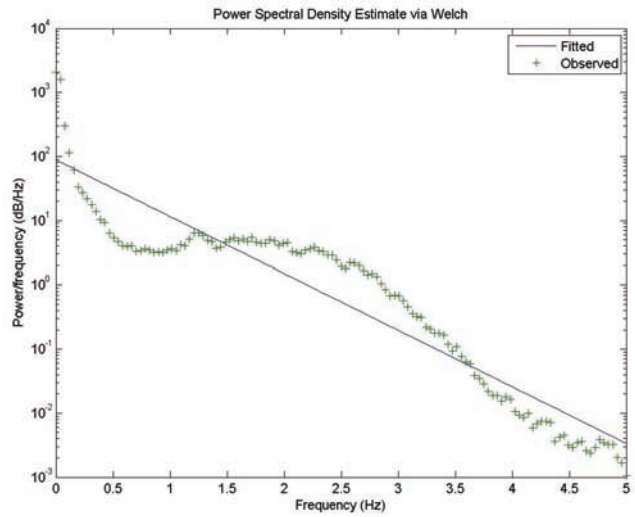


Slika 2. *PSD promjenljivosti srčanog ritma S_BHR_Baseline*

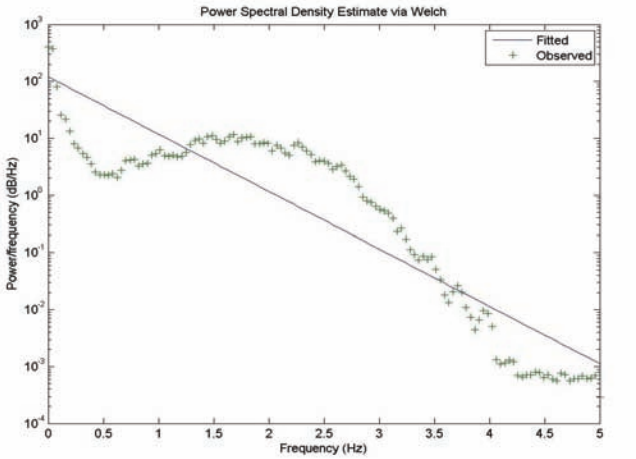


Slika 3. *PSD promjenljivosti srčanog ritma po semi-log skali S_BHR_Baseline*

Korišćen je semi-log prikaz u cilju jasnijeg prikaza 1/f oblika spektra. Nagib spektra je estimiran je korišćenjem polyfit i polyval funkcija programskog okruženja *Matlab*.

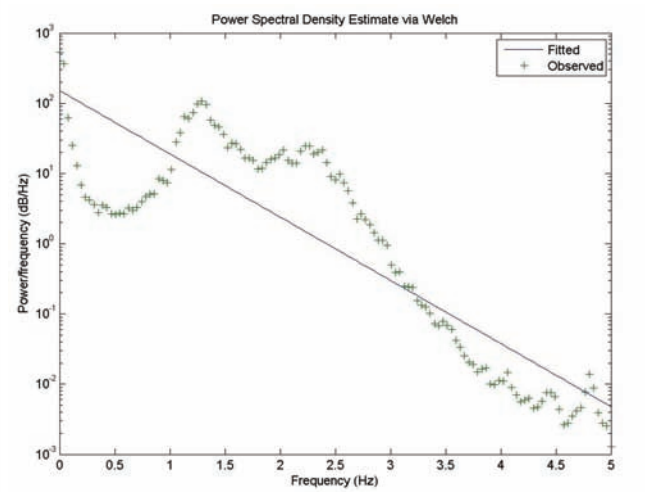


Slika 4. *PSD nagib S_BHR_Baseline*



Slika 5. *PSD nagib R_BHR_FS*

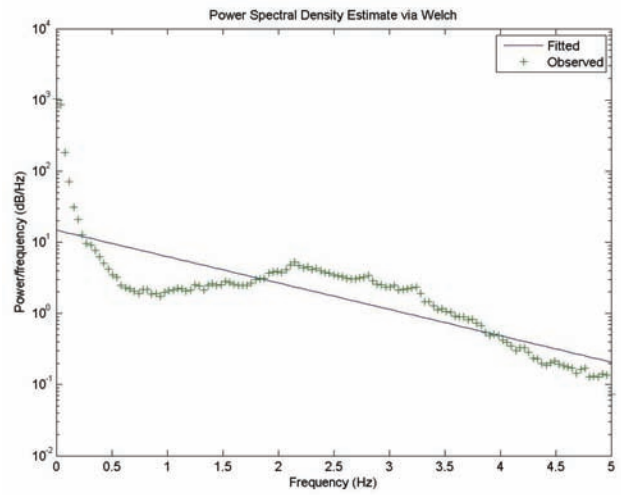
Na slici 5. prikazan je dobijeni nagib za istog pacova posle prvog stresa kao primer.
Tabela 1. prikazuje promjene nagiba, preko njegovih srednjih vrijednosti i standardnih devijacija u zavisnosti od tipa stresa perioda u procesu izlaganja stresu.



Slika 6. *PSD promjenljivosti srčanog ritma R_BHR_Baseline*

Tabela 1. *Srednjih vrijednosti i standardnih devijacija*

	Baseline	FS	PFS	PLS
R_BHR	-2.9336±0.1505	-1.7995±0.5605	-1.6449±0.4878	-1.7103±0.3055
S_BHR	-3.0395±0.3837	-2.8247±0.6088	-2.9372±0.5109	-3.0385±0.3812
R_NORM	-2.9772±0.4654	-1.0917±0.3003	-1.8413±0.3258	-2.4742±0.4892
S_NORM	-2.4436±0.5721	-2.1197±0.4998	-2.3977±0.3466	-2.4956±0.7786



Slika 7. *PSD promjenljivosti srčanog ritma R_BHR_FS*

4. **ZAKLJUČAK**

Iz tabele 1. kao i sa slike 5. date za primjer može se zaključiti da Shaker, poslije prvog stresa (FS), ne mjenja značajno parametre nagiba. Dok su kod Restraint stresa parametri nagiba znatno izmjenjeni. I tokom daljeg vremenskog perioda značajnije promjene su se dešavale kod Restraint stresa pa je i veća pažnja data tim rezultatima. Na slici 6. i 7. date su kao primjer evidentne promjene nagiba.

Iz daljih rezultata kroz period procesa izlaganja stresu zaključak je da se poslije prvog stresa (PFS), normotenzivni pacov počinje vraćati u normalno stanje (iako je zadržao neka razlike u odnosu na početno stanje), dok se to nije desilo i sa BHR pacovom.

Poslije poslednjeg stresa (PLS) normotenzivni pacov se vratio u normalno stanje, a BHR pacov, pacov podložniji hipertenziji se nije vratio.

Krajnji zaključak je da na granično hipertenzivne životinje stres ima produženo dejstvo, odnosno da je oporavak usporen.

Testiran je 1/f matematički model na jednočasovnim signalima pacova. Dobijeni rezultati (ukoliko posmatramo Shaker stres) pokazuju da kod hipertenzivnih subjekata dolazi do značajnog povećanja vrednosti spektralnog nagiba (*n* koeficijent), koji predstavlja obeležje dugoročnih kontrolnih mehanizama HRV signala.

Ovi podaci ukazuju na to da 1/f model dobro predstavlja ukupni spektar HRV-a i u mogućnosti je da razlikuje subjekte sa normalnom i patološkom kardiovaskularnom kontrolom.

5. ZAHVALNICA

Zahvaljujem se prof. dr Nini Japundžić-Žigon sa Medicinskog fakulteta u Beogradu, u čijoj laboratoriji su snimljeni signali nad kojim je urađena ova studija.

6. LITERATURA

[1] G Calcagnini Jr, S Lino, S Strano, S Cerutti, G Calcagnini: *1/f Spectrum slope of 24 Hour Heart Rate Variability Signal in Normal and Hypertensive Subjects*, University of Rome 'La Sapienza', Rome, Italy

[2] Booth, J. (1977): *A short history of blood pressure measurement*; Proceedings of the Royal Society of Medicine, Volume 70, Issue 11: 793-799;

[3] Reckelhoff, J. F. (2001): *Gender Differences in the Regulation of Blood Pressure*; Hypertension, Volume 37, Issue 5: 1199-1208;

[4] http://www.nhlbi.nih.gov/health/prof/heart/hbp/hbp_ped.pdf; *Blood Pressure Tables for Children and Adolescents;* National Heart, Lung and Blood Institute;

[5] http://circ.ahajournals.org/cgi/content/full/93/5/1043; *Heart Rate Variability Standards of Measurement, Physiological Interpretation, and Clinical Use*

[6] http://www.biocomtech.com/hrv-science/heart-rate-variability; *Heart Rate Variability*

Kratka biografija:

Anđelko Pikula rođen je u Gacku 1987. god. Bečelor rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Obrada biomedicinskog signala odbranio je 2011. god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 620.93

IDEJNO REŠENJE HIBRIDNOG VETRO-FOTONAPONSKOG SISTEMA NA ZGRADI NTP FTN

CONCEPTUAL SOLUTION OF HYBRID WIND-PHOTOVOLTAIC SYSTEM ON THE NTP FTN BUILDING

Milan Milanko, Vladimir Katić, Zoltan Čorba, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Ovim radom se daje idejno rešenje hibridnog vetro-fotonaponskog sistema na zgradi NTP FTN. Za sistem su izabrani pretvarači određene snage, invertori i odgovarajuća prateća i zaštitna oprema. Na osnovu procenjenog energetskog potencijala sunčevog zračenja i brzine vetra proračunat je položaj postavljanja sistema panela, odabrana odgovarajuća vetroturbina i izračunata očekivana proizvodnja električne energije na godišnjem nivou iz svakog od navedenih sistema pojedinačno. Na kraju se daje šema povezivanja odabranih elemenata i grafički prikaz postavljanja sistema na zgradu.*

Abstract – *The paper provides a conceptual solution of hybrid wind-photovoltaic system on the NTP FTN building. For this purpose certain converters, inverters and appropriate protective and supporting equipment were selected. Based on estimated energy potential of solar radiation and wind speed, position of setting solar panels and expected annual energy production from the each of individual systems were calculated and appropriate wind turbine was selected. Finally, the paper provides a connection scheme of selected devices and graphical overview of abandoned system on the building.*

Ključne reči: *Obnovljivi izvori energije, hibridni vetro-fotonaponski sistem, vetroturbina, solarni panel.*

1. UVOD

U poslednje vreme se sve češće naglašava značaj obnovljivih izvora energije i energetske efikasnosti. Jedna od ideja je da se globalna potrošnja energije smanji na taj način što bi se smanjila potrošnja energije svakog pojedinačnog objekta. Pored kvalitetnije izolacije, koja umanjuje ukupnu utrošenu energiju na grejanje, odnosno hlađenje, i inteligentnog upravljanja potrošnjom, nije redak slučaj da se na objekte ugrađuju uređaji koji bi proizvodili energiju iz obnovljivih izvora i na taj način smanjivali ukupno utrošenu energije. Uređaji koji se postavljaju na objekte i koji se koriste za pretvaranje obnovljivih izvora energije u korisnu energiju su najčešće bazirani na pretvaranju energije sunca i vetra u električnu energiju. Za to se koriste fotonaponske (FN) ili solarne ćelije i vetro-generatori.

NAPOMENA:
Ovaj rad je proistekao iz master rada čiji je mentor bio prof. dr Vladimir Katić.

Solarna ćelija, koja je u stvari poluprovodnička dioda, apsorbuje sunčevo zračenje, a fotonaponskim efektom se na njenim krajevima dobija napon, odnosno ona postaje izvor električne energije. Izlazne karakteristike solarne ćelije zavise direktno od intenziteta sunčevog zračenja i od temperature okoline. Više ćelija povezanih redno ili paralelno čine solarne module, a niz modula se povezuju u panele, koji se postavljaju u određena standardna kućišta snaga oko 200-250 W.

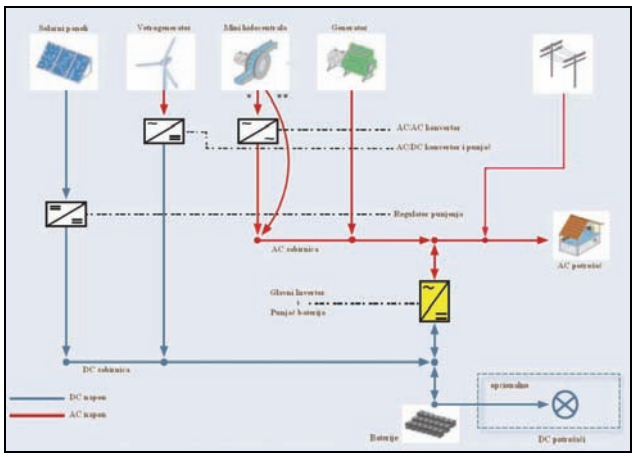
Energija vetra se iskorištava tako što se pomoću vetroturbine pretvara u rotacionu mehaničku, a zatim pomoću električnog generatora montiranog na istu osovinu, u električnu energiju. Količina raspoložive energije uglavnom zavisi od brzine vetra (kubna zavisnost), površine koju opisuje rotor turbine, ali i od gustine vazduha, na koju utiču temperatura, vazdušni pritisak i nadmorska visina. Snage vetrogeneratora variraju od 500 W do 5 ili više MW.

2. HIBRIDNI VETRO-FOTONAPONSKI SISTEMI

Sistemi, koje u sebi sadrže dva ili više tipova obnovljivih izvora energije (FN paneli, vetrogeneratori, mini hidrocentrale i dr.), nazivaju se hibridni sistemi. Postoje različiti tipovi hibridnih sistema, koji omogućavaju lako upravljanje proizvedenom energijom i njenom potrošnjom. Najopštija podela je na samostalne hibridne sisteme i hibridne sisteme povezane na distributivnu električnu mrežu [1].

Samostalni hibridni sistemi nisu povezani na distributivnu mrežu, zbog čega je neophodno da se proizvedena električna energija skladišti ili odmah potroši. Samostalni sistemi se najčešće koriste u delovima gde nema distributivne mreže ili gde bi dovođenje distributivne mreže bilo neekonomično. Sistemi povezani na mrežu se koriste u urbanim delovima gde već postoji distributivna mreža.

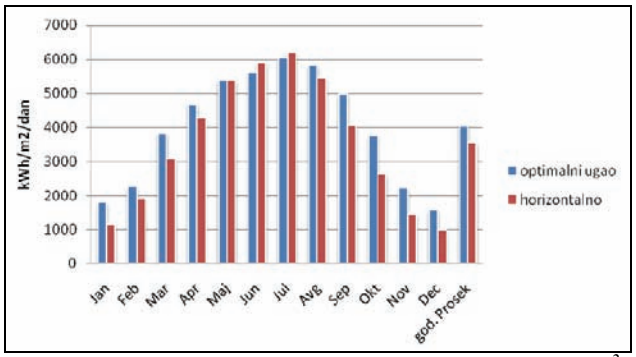
Hibridni sistem, koji je predmet ovog rada, ima strukturu datu na slici 1, a zamišljen je tako da vrši samostalno napajanje potrošača zbog čega su potrebni uređaji za skladištenje energije (baterije). Iako su baterije uređaji koji su karakteristični za samostalne sisteme, predviđeno je da ovaj sistem bude priključen na distributivnu mrežu radi dopunjavanja baterija za slučaj kada ne postoji dovoljna proizvodnja energije bilo iz fotonaponskog ili vetroenergetskog sistema. Pored toga za primenjeni sistem je karakterističan i komunikacioni uređaj, koji omogućava monitoring sistema, daljinsku diagnostiku, skladištenje podataka i njihovu vizuelizaciju. Proizvođač preporučuje da ovaj uređaj da bude priključen na mrežu (230V, 50Hz).



Slika 1. Hibridni sistem povezan na mrežu

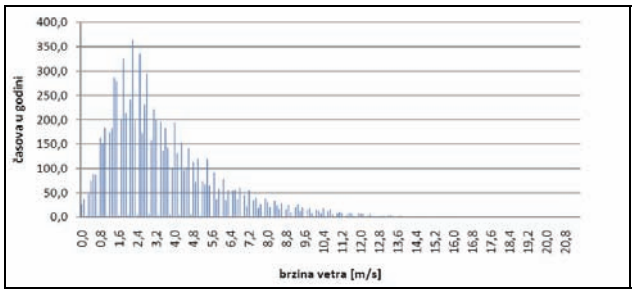
3. PROCENA ENERGETSKOG POTENCIJALA SUNČEVOG ZRAČENJA I BRZINE VETRA ZA OKOLINU NOVOG SADA

Na osnovu urađene analize sunčevog zračenja za širu okolinu Novog Sada pomoću softvera koji se nalazi na sajtu Razvojnog Centra Evropske Komisije (Fotonaponski geografski informacioni sistem, PVGIS) dobijeni su podaci o dnevnoj i mesečnoj sumi zračenja [9]. Ovi podaci ne moraju biti sasvim tačni, ali su dovoljno precizni da bi se mogli smatrati referentnim za dalje proračune. Takođe su određeni i optimalni uglovi za svaki mesec pojedinačno, odnosno optimalni godišnji ugao za sistem panela koji je fiksiran i on iznosi 34º. Na slici 2 je dat grafik na kome je, na osnovu gore navedenog softvera, predstavljeno prosečno dnevno ozračenje za pojedine mesece, kao i prosečno godišnje ozračenje za dva položaja panela horizontalan i nagnut pod optimalnim uglom. Kada se posmatra prosečna vrednost u toku jedne godine, na horizontalnu površ panela dnevno stigne 3,54 kWh/m^2. Ukoliko je panel postavljen pod optimalnim uglom ta gustina energija iznosi 4,03 kWh/m^2. U radu je takođe proračunato i prosečno mesečno ozračenje. Najveći iznos sunčeve energije je na raspolaganju u julu, 187 kWh/m^2, a najmanje u decembru, 49 kWh/m^2. Godišnji prosek za period od mesec dana iznosi 121 kWh/m^2.



Slika 2. Procenjeno dnevno i prosečno godišnje u Wh/m^2 ozračenje za okolinu Novog Sada

Analiza energetskog potencijala vetra je urađena na osnovu sedmogodišnjeg merenja brzine i pravca vetra u metrološkoj stanici u okolini Novog Sada (Rimski Šančevi) i to u periodu od 2001-2007. godine [1]. Ovi zvanični meteorološki podaci predstavljaju jednočasovno usrednjavanje brzine i pravca vetra. Da bi se što preciznije proračunala stvarna proizvedena energija na godišnjem nivou potrebno je da se merenje intenziteta i pravca vetra vrši što bliže predviđenom mestu postavljanja vetroturbine. Na žalost ovakvi podaci nisu raspoloživi, pa su korisćeni podaci iz pomenute meteorološke stanice mereni na visini od 10m iznad površine zemlje. Na osnovu podataka o brzini vetra na nekoj visini moguće je preračunati brzinu vetra na željenu visinu. U radu je izvršeno preračunavanje na visinu 37m na kojoj će se nalaziti rotor vetrogeneratora. Na slici 3 je data estimirana raspodela srednjih brzina vetra na visini od 37m.



Slika 3. Estimirana raspodela srednje brzine vetra MS Novi Sad za visinu 37m

Sredjne brzine vetra se menjaju u toku godine. Najbrži vetrovi za okolinu Novog Sada se javljaju iz pravca severo-zapada (brzina 3,6 m/s), odnosno jugo-istoka (brzine 3,3 m/s), a ujedno ovi pravci su i najčešći. U posmatranom vremenskom intervalu, a posmatrajući srednje mesečne vrednosti brzine, može se reći da su najveće brzine vetra u januaru, februaru, martu i aprilu. Nešto manje vrednoti se javljaju u jesenjim mesecima, dok je jačina vetra najslabija u junu, julu i avgustu sa prosečnim brzinama od oko 2 do 2,5 m/s.

4. ODABIR UREĐAJA HIBRIDNOG VETRO-FOTONAPONSKOG SISTEMA

Odabir uređaja hibridnog sistema se vrši uz uvažavanje procenjenog energetskog potencijala sunčevog zračenja i brzine vetra. Zadatak ovog rada je odabir fotonaponskog sistema i odgovarajuće vetroturbine, a zatim i odgovarajućh invertora i ostale neophodne opreme.

Za fotonaponski sistem su izabrani paneli SCHOTT Solar EFG 165 Wp od visokokvalitetnih polikristalnih ćelija. Solarni deo hibridnog sistema se sastoji od 8 fotonaponskih panela i ima ukupnu instalisanu snagu 1320Wp [7].

Komercijalne vetroturbine male snage uglavnom se konstruišu za nominalnu brzinu vetra reda 10m/s. Ako se uzme u obzir srednja brzina vetra na mesečnom nivo za odabrano područje postavljanja, vidi se da se nominalna brzina komercijalnih vetroturbina postiže tek u pojedinim kraćim vremenskim intervalima. Zbog toga se za kriterijum izbora vetroturbine uzima brzina pri kojoj vetroturbina počinje da proizvodi korisnu energiju, ili polazna (*cut-in*) brzina tubine. Zbog veoma niske brzine vetra poželjno je da ova vrednost bude što je moguće niža. Kao vetrogenerator za ovaj sistem se odabran je AeroCraft AC 752 nominalne snage 750W sa polaznom brzinom obrtanja od 3 m/s i izlaznim nominalnim DC naponom od 12V [8]. Na slici 4 predstavljeni su odabrani fotonaponski panel i vetrogenerator.



Slika 4. Vetrogenerator AeroCraft AC 752 i solarni panel SCHOTT Solar EFG 165

Za kompletiranje sistema, uz odabrani panela i vetrogeneratora izabrani su sledeći uređaji hibridnog sistema:

Regulator PSR 60V-20A – uređaj (ispravljač) koji se koristi za povezvanje trofaznog generatora nominalnog napona 48V i maksimalne snage 1000W sa invertorom. Ovaj uređaj omogućuje efikasnu zaštitu od previsokog napona. Ukoliko se desi da jednosmerni izlazni napon premaši vrednos 58V vrši se usporavanje generatora kratkim spajanjem njegovih krajeva [8].

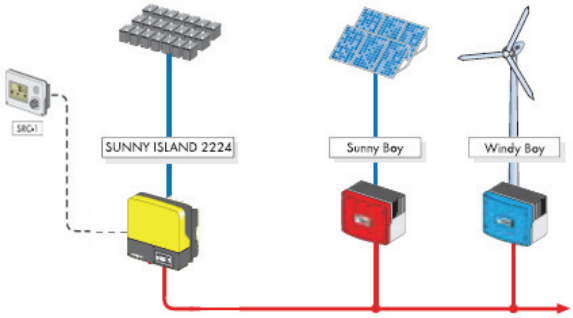
Invertor SMA Windy Boy 1100LV - standardni industrijski invertor pomoću kojeg je moguće povezivanje malog vetroenergetskog sistema na postojeću samostalnu mrežu, distributivnu mrežu ili povezivanje u hibridni sistem. Na ulaz invertora se dovodi napon sa sinhronog generatora sa permanentnim magnetima koji se prethono ispravi odgovarajućim ispravljačem. Ulazni napon varira u zavisnost od brzine obrtanja generatora, a na izlazu invertora dobijamo standardan mrežni napon 230V i frekvencije 50Hz [5].

Invertor SMA Sunny Boy 1200 – standardni industrijski invertor koji pretvara jednosmerni napon generisan na fotonaponskim panelima u mrežni naizmeničan napon 230 V frekvencije 50Hz. Invertor omogućava maksimalno iskorišćenje snage sistema fotonaponskih panela pomoću funkcije MPPT (*Maximum Power Point Tracking*). Izlaz iz invertora povezuje se na zajedničku AC sabirnicu na kojoj se u ovom slučaju nalazi i odabrani Windy Boy invertor [5].

Sunny Island 2224 – dvosmerni pretvarač (ispravljač-punjač bateri i invertor) koji predstavlja centralnu jedinicu hibridnog sistema. Invertor na izlazu daje standardni mrežni napon 230V i frekvencije 50Hz. Ima mogućnost da napaja samostalnu mrežu, a eventualnim viškom energije koju dobija sa AC strane da puni baterije. Sunny Island 2224 takođe ima mogućnost da automatski prekine napajanje nekog potrošača ukoliko baterije ne sadrže dovoljno energije za njegovo napajanje i na taj način štiti baterije [5].

Baterije – da bi obezbedili pouzdan rad samostalnog sistema, SMA Solar Tehnology preporučuje da najmanji kapacitet baterija za Sunny Island 2224 (24V) bude 175 Ah za C_{10}. Ukoliko se vetro- fotonaponski sistem poveže na javnu mrežu, SMA Solar Tehnology preporučuje da kapacitet baterija bude 200 Ah po 1kW AC nominalne snage fotonaponskog invertora, za slučaju sistema bateruja nominalnog napona 24 V. U ovom sistemu su odabrane baterije nemačkog proizvođača HOPPECKE [6].

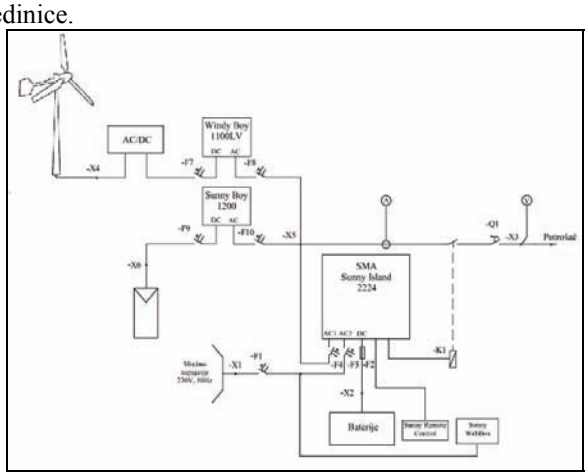
Sunny WebBox – komunikacioni uređaj visokih performansi koji omogućava monitoring sistema, daljinsku dijagnostiku, skladištenje podataka i njihovu vizualizaciju. Kao centralni komunikacioni interfejs, Sunny WebBox povezuje fotonaponski sistem sa operaterom [5].



Slika 5. Blok šema hibridnog sistema sa odabranim invertorima

5. JEDNOPOLNA BLOK ŠEMA HIBRIDNOG VETRO-FOTONAPONSKOG SISTEMA

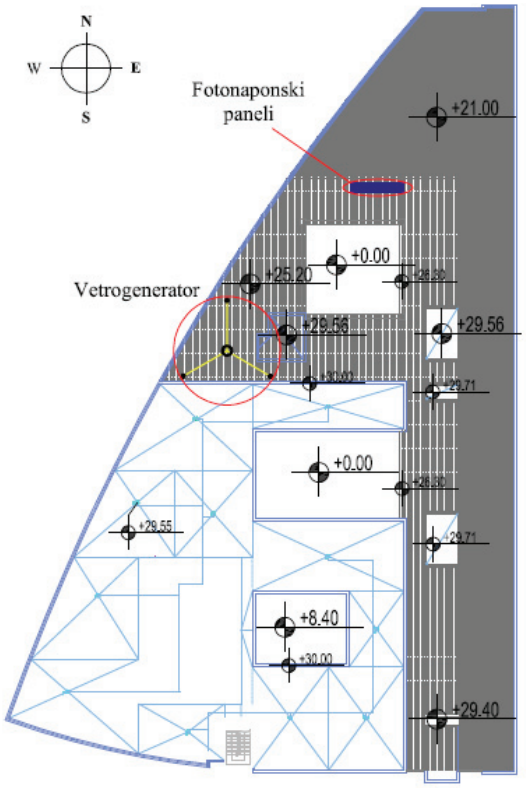
Postavljanje glavnog razvodnog ormana u koji se smeštaju invertori i ostala zaštitna i komunikaciona oprema se predviđa u labaratoriji za energetsku elektroniku i pretvarače. Na *slici 5.* se nalazi uprošćena jednopolna šema sistema. Električna energija generisana na vetroturbini se dovodi na ulazne stezaljke X6, zatim se ona vodi na Regulator PSR 60V- 20A gde se napon ispravlja i tako ispravljen vodi na ulaz Windy Boy LV 1100 invertora. Energija dobijena sa fotonaponskih panela vodi se na ulaz Sunny Boy 1200 invertora. Izlaz Sunny Boy invertora se priključuje na AC sabirnicu zajedno sa izlazom Windy Boy i vodi na stezaljke X3 koje su predviđene za priključenje potrošača, odnosno na AC1 ulaz Sunny Island 2224 jedinice. AC2 ulaz Sunny Island 2224 jedinice se priključuje na mrežni napon i na taj način je moguće dopunjavanje baterija iz mreže ukoliko su vremenski uslovi takvi da ne obezbeđuju dovoljnu proizvodnju energije iz vetrogeneratora i sistema solarnih panela. Na DC priključak Sunny Island 2224 se povezuju baterije. Pomoću kontaktora K1 može se vršiti uključenje i isključenje potrošača a njegovo upravljane je ostvareno pomoću *load shedding* kontakta Sunny Island 2224 jedinice.



Slika 6. Jednopolna blok šema hibridnog vetro-fotonaponskog sistema

6. POLOŽAJ FOTONAPONSKIH PANELA I VETROTURBINE NA KROVU ZGRADE NTP-FTN

Zbog konstrukcije objekta praktično je neizvodivo da se fotonaponski paneli i vetroturbina postave na površinu zgrade koja je najviša. Zbog toga se fotonaponski paneli postavljaju na nižu površinu, naznačenu na slici 7, ali je tad neophodno da se obrati pažnju na efekat senke viših delova objekta, odnosno na turbolenciju vazduha u slučaju vetroturbine.

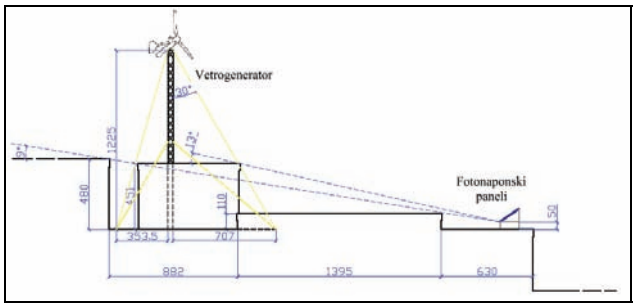


Slika 7. Raspored fotonaponskih panela i vetrogeneratora na budućoj zgradi NTP FTN

Minimalni upadni ugao sunčevih zraka u toku godine se ima u decembru i on iznosi 26º. Fotonaponski paneli se postavljaju tako da im je minimalni mogući upadni ugao sunčevih zraka manji od gore navedenog ugla i u ovom slučaju on iznosi 13º. Fotonaponski paneli se postavljaju na aluminijumsku konstrukciju koja je nagnuta prema jugu za vrednost godišnje optimalnog ugla proračunatog softverom PVGIS koji iznosi 34º za okolinu Novog Sada.
Kod montaže vetrogeneratora na površinu zemlje osnovu za postavljanje čine betonski blokovi zapremine $1m^3$. Ovakav način postavljanja na krovu zgrade nije moguć, pa bi se za postavljanje vetroturbine verovatno koristila montažna čelična konstrukcija sa zateznim sajlama. Problem turbolencije vazduha u slučaju vetroturbine je rešen na taj način što je vetroturbina postavljena na stub visine 12,25m, odnosno 7,45m iznad najviše tačke objekta.
Na slici 8 daje grafički prikaz dela zgrade na koji bi se postavili fotonaponski paneli i vetrogenerator

7. ZAKLJUČAK

U radu je dato idejno rešenje jednog hibridnog vetro-fotonaponskog sistema za buduću zgradu NTP FTN snage 2 kW. Sa aspekta slobodnog prostora postoji izvesna mogućnost proširenja fotonaponskog sistema do 5 kWp, ali je prethodno neophodno obratiti pažnju na efekat senke viših delova objekta.



Slika 8. Grafički prikaz dela zgrade na koji su postavljeni fotonaponski paneli i vetrogenerator

8. LITERARURA

[1] Zoltan Čorba, "Mogućnosti primene hibridnog sistema za pretvaranje energije vetra i sunca u električnu", Magistarska teza, FTN, Novi Sad, 2009.

[2] G. Vamović, V. Katić, Z. Čorba, „Projekat hibridnog sistema obnovljivih izvora energije i realizacija solarnog dela", Zbornik radova Fakulteta tehničkih nauka, God. 24, Br.4, 2009, pp.1294-1297.

[3] "Small Wind Electric System" A U.S Consumer's Guide, U.S Departmant of energy, 2005.

[5] http://www.sma.de

[6] http://www.hoppecke.com/

[7] http://www.schottsolar.com

[8] http://www.aerocraft.de/ac752daten.html

[9] http://re.jrc.ec.europa.eu/pvgis/

Kratka biohrafija:



Milan Milanko je rođen 1985. god u Somboru. Diplomski-Master rad na Fakultetu tehničkih nauka u Novom Sadu, iz oblasti Elektrotehnike i računarstva, na smeru Energetska elektronika i električne mašine odbranio je u julu 2011. god.



Vladimir Katić je doktorirao na Univerzitetu u Beogradu 1991. Redovni profesor Fakulteta tehničkih nauka u Novom Sadu je od 2002. za oblast Energetska elektronika, mašine i pogoni. Prodekan je FTN-a i šef Katedre za energetsku elektroniku i pretvarače.



Zoltan Čorba je magistrirao na Univerzitetu u Novom Sadu 2009. Radio je u razvojnom timu Novkabela, a od 1992. kao šef laboratorije za energetsku elektroniku i laboratorije za obnovljive izvore energije.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

IZRADA SCADA APLIKACIJE, MATEMATIČKO MODELIRANJE I RAZVOJ FUZZY REGULATORA VODOVODNOG SISTEMA

DEVELOPMENT OF SCADA APPLICATION, MATHEMATICAL MODEL AND FUZZY CONTROLLER IN WATER INDUSTRY SYSTEM

Igor Bende, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U prvom delu rada opisan je postupak izrade SCADA aplikacije za nadzor i akviziciju podataka vodovodnog sistema. Nakon toga izrađen je matematički model i na osnovu učitanih podataka određeni su parametri sistema. Na kraju je izvršena simulacija, gde je održavanje pritiska u sistemu vršeno prvo PID, a zatim naprednim fuzzy regulatorom.*

Abstract – *This work shows development of SCADA application for supervision and data acquisiton of water industry system, mathematical modelling of this system and experimental parameter determining based on read data, and at the end, it is simulated constant pressure controlling with PID and advanced fuzzy controller.*

Ključne reči: *klijentska SCADA aplikacija, vodovodni sistem, fuzzy regulator*

1. UVOD

SCADA (*Supervisory Control And Data Acquisition*) predstavlja sistem za merenje, praćenje i kontrolu uglavnom industrijskih sistema.
Obuhvataju širok spektar opreme, podsistema i tehničkih rešenja koji omogućavaju prikupljanje i obradu podataka o procesima (udaljenim sistemima) i reagovanje na adekvatan način.
Mogu se upotrebiti od jednostavnog praćenja temperature, vlažnosti vazduha, pritiska, do veoma kompleksnog praćenja i kontrole proizvodnih procesa fabrike ili npr. železničkog saobraćaja. [1]
Matematički model podrazumeva formalno matematički zapis ponašanja i/ili osobina posmatranog fizičkog sistema. Ukoliko je poznat matematički model, znači da u potpunosti poznajemo ponašanje i osobine fizičkog sistema. Često je nemoguće izraditi apsolutno tačan matematički model i tada se pristupa određenim aproksimacijama i zanemarivanjima malo uticajnih osobina ili uticaja.
Poznavanjem matematičkog modela sistema pruža nam se mogućnost da ih predstavimo u digitalnom obliku, tj. na računaru, što nam omogućava izradu raznih simulacija, eksperimenata i razvoj novih teorija upravljanja bez ugrožavanja postojećeg sistema i bezbednosti ljudi.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Velimir Čongradac, docent.

2. SCADA SISTEMI

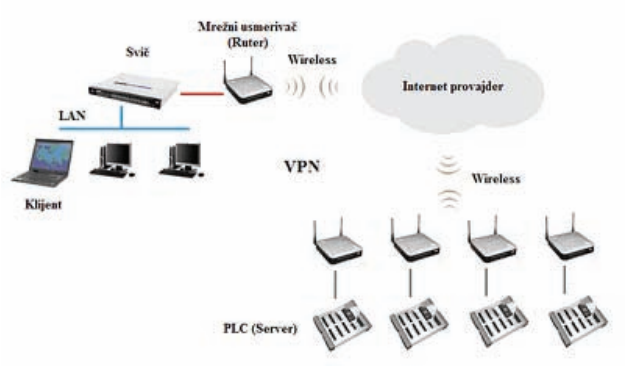
SCADA mreža se sastoji od jednog ili više MTU (*Master Terminal Unit*), koji su zapravo računarske stanice opremljene odgovarajućim softverom i operativnim sistemom. Ove stanice operateri koriste za praćenje i kontrolu jedne ili više RTU (*Remote Terminal Unit*). RTU je takođe računarski uređaj koji je obično namenjen za rad u industrijskim uslovima ili čak u ekstremnim kao što je npr. okruženje u vasioni. Njegov zadatak je da prikuplja informacije sa raznih digitalnih i analognih senzora i da prosleđuje komande uređajima koji na neki način menjaju stanje upravljanog sistema (razni step motori, generatori, releji i sl.). Često se za njihovu realizaciju koriste razni tipovi PLC-a (*Programmable Logic Controller*).
Za prenos podataka između MTU i RTU se koristi neki od standardnih ili specijalizovanih računarskih mreža i protokola kao što su Ethernet, Modbus, ProfiNET... Optički kablovi su idealni za realizaciju SCADA mreže jer nude potpunu zaštitu od emisije raznih zračenja koja bi mogla da dovede do prenosa pogrešnih informacija. U idealnom SCADA sistemu svi elementi su redundantni, što dostupnost i pouzdanost sistema podiže na visok nivo.
Velika je raznolikost u SCADA implementacijama i dešava se da se funkcije MTU, RTU i ostalih delova sistema prepliću tako da je teško razlikovati specifične komponente. Ipak, jedan od delova, tzv. HMI (*Human Machine Interface*) je svakako veoma lako prepoznati, jer je njegova uloga da prezentuje informacije korisnicima SCADA pa je to najčešće računar sa softverom koji na jasan način prikazuje sve informacije o sistemu i realizuje njegovu upravljivost [1].

3. REALIZACIJA SCADA SISTEMA

3.1. Arhitektura reallizovanog sistema

Na slici 1. prikazana je arhitektura realizovanog SCADA sistema. Klijentski računar, na kom je instalirana SCADA aplikacija, nalazi se u dispečerskom centru i predstavlja centralnu stanicu. Preko sviča (uređaj koji upravlja protokom podataka u LAN mreži) povezan je u lokalnu računarsku mrežu sa ostalim računarima, kao i serverima sistema. Svič je preko rutera (mrežnog usmerivača) povezan na Internet, tj. preko njega je ostvarena bežična veza sa Internet provajderom. Internet provajder daje mogućnost korišćenja virtuelne privatne mreže. Stvara se utisak da se svi korisnici nalaze u istoj lokalnoj mreži. Četiri PLC-a predstavljaju udaljene stanice, a zaduženi su za kontrolu sistema automatskog upravljanja u četiri objekta (CSV Horgoš, UPOV Horgoš, CSV Kanjiža,

UPOV Kanjiža). Oni su takođe preko rutera bežično povezani na Internet i nalaze se u istoj virtuelnoj privatnoj mreži kao i centralna stanica. Između SCADA aplikacije i PLC-a ostvarena je klijent-server komunikacija, gde PLC ima ulogu servera, koji klijentu po zahtevu šalje podatke (informacije) [2].



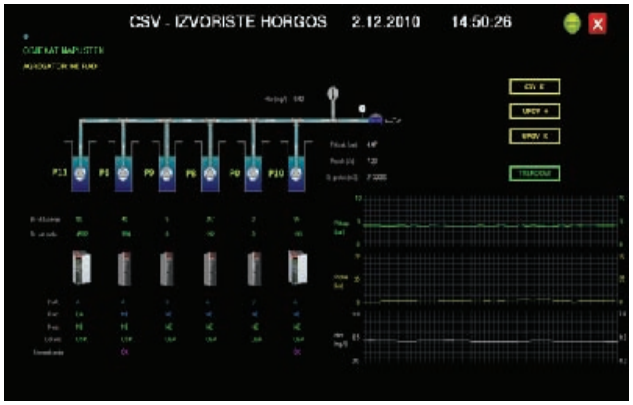
Slika 1. *Arhitektura SCADA sistema*

3.2. Klijentska SCADA aplikacija

Klijentske SCADA aplikacija rađena je u programskom jeziku *Borland C++ Builder 5* [2], a od nje se zahteva da:

- održava kontinualnu komukaciju preko Interneta sa četiri PLC-a (na mestima CSV Horgoš, UPOV Horgoš, CSV Kanjiža i UPOV Kanjiža)
- vizuelno prikaže elemente sistema automatskog upravljanja
- prikaže vrednosti promenljivih koje se menjaju (pritisak u cevima, protok, količina hlora, procenat kiseonika, nivo, stanje pumpi-)
- prikaže prisutnost u objektu kao i karakter (ovlašćen/neovlašćen ulazak)
- stanje o napajanju sistema (ispad, rad agregatora)
- datum i vreme
- grafički prikazuje promenu najbitnijih parametara u određenom vremenu (pritisak, protok, nivo)
- prikaže alarmantne događaje
- vrši neprestanu akviziciju podataka
- omogući prikazivanje trendova
- ima zaštitu od pada softvera ili komunikacijske veze

Na slici 2. vidi se prikaz stanja u crpnoj stanici vode u Horgošu.



Slika 2. *SCADA – CSV Horgoš*

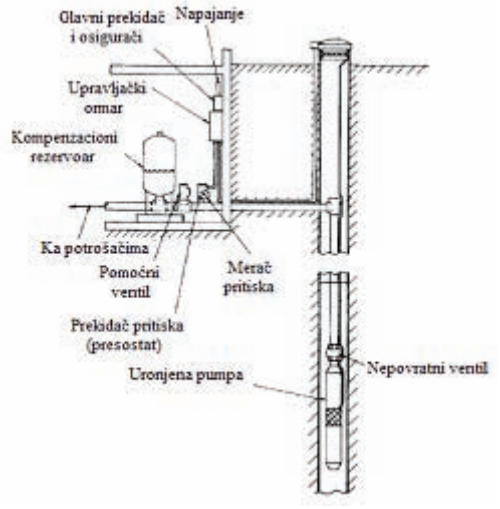
4. MATEMATIČKI MODEL SISTEMA

4.1. Matematički model sistema automatskog upravljanja

Da bi se razumela dinamika i projektovalo upravljanje za neki kompleksan sistem, prvo mora biti poznat njegov matematički model. Pošto su razmatrani sistemi u prirodi dinamički, za njihovo opisivanje se koriste sistemi diferencijalnih jednačina (DJ). Pri rešavanju sistema diferencijalnih jednačina pogodno je koristiti Laplasovu transformaciju (LT) koja pojednostavljuje određivanje rešenja. Ukoliko je sistem automatskog upravljanja (SAU) opisan sistemom nelinearnih DJ, pre primene LT je potrebno izvršiti linearizaciju.
Nakon završenog modeliranja, sistem automatskog upravljanja je opisan sistemom linearnih DJ. Na kraju se na osnovu postavljenog modela, primenom LT, određuje ponašanja sistema u različitim uslovima i za različite pobude [3].

4.1. Matematički model vodovodnog sistem sa jednom pumpom

Na slici 3. dat je prikaz jednog vodovodnog sistema sa jednom pumpom. Uronjena pumpa sa nepovratnim ventilom crpi podzemnu vodu i ubacuje je u vodovodni sistem. Na početku vodovodnog sistema nalazi se kompenzacioni rezervoar koji služi za kompenzaciju nagle promene pritiska ukoliko dođe do nagle promene protoka. Dalje se voda vodi ka potrošačima. U upravljačkom ormaru nalazi se električna oprema koja obezbeđuje nadzor i upravljanje.



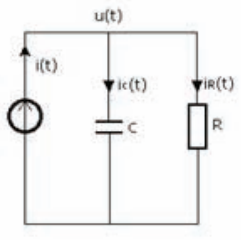
Slika 3. *Vodovodni sistem sa jednom pumpom*

Elektromehaničke analogije se uvode zato što se određeni sistemi i njihove komponente ponašaju slično kao električni. Svaki takav sistem moguće je svesti na ekvivalentan električni. Prednost toga je što se tada za njihov proračun primenjuju teorije električnih kola koje su razvijenije, dobro proučene i lakše za razumevanje. [4]
Za vodovodni sistem sa jednom pumpom moguće je primeniti sledeće analogije:

- zapreminski protok ≈ struja
- pritisak ≈ napon
- pumpa ≈ strujni ili naponski izvor
- kompenzacioni rezervoar ≈ kondenzator

- otpornost cevi i otvorenost svih ventila ≈ otpornik

Na slici 4. prikazan je ekvivalentan električni sistem sa jednačinama.



$$u(t) = u_R(t) = u_C(t)$$
$$i(t) = i_R(t) + i_C(t)$$
$$i_R(t) = \frac{u(t)}{R}$$
$$i_C(t) = C\frac{du_C(t)}{dt} \quad (1)$$

Slika 4. *Ekvivalentni sistem u el. domenu*

Oznake u vodovodnom sistemu:

- zapreminski protok – q
- pritisak – p
- hidraulička kapacitivnost rezervoara – C
- hidraulička otpornost cevi i otvorenost potrošačkuh ventila – R
- frekvencija napona motora pumpe – f
- koeficijent upravljanja – K

Jednačine električnog i njemu ekvivalentnog vodovodnog sistema:

$$i(t) = i_C(t) + i_R(t) = C\frac{du(t)}{dt} + \frac{u(t)}{R(t)} \Rightarrow \frac{du(t)}{dt} = \frac{1}{C}i(t) - \frac{u(t)}{R(t)C}$$
$$\approx$$
$$Kf(t) = C\frac{dp(t)}{dt} + \frac{p(t)}{R(t)} \Rightarrow \frac{dp(t)}{dt} = \frac{K}{C}f(t) - \frac{p(t)}{R(t)C} \quad (2)$$

Matematički model vodovodnog sistema sa jednom pumpom:

$$\frac{dp(t)}{dt} = \frac{K}{C}f(t) - \frac{p(t)}{R(t)C} \quad (3)$$

5. REGULACIJA

5.1. Regulacija u automatici

Svaki sistem automatskog upravljanja se odlikuje određenim zakonom ili zakonima upravljanja. Zakon upravljanja predstavlja matematičku zavisnost na osnovu koje upravljački uređaj obrađuje relevantne signale (informacije) i generiše odgovarajuća upravljačka dejstva. Najčešća forma ovakvih upravljačkih uređaja naziva se regulator. Bez obzira na tip regulatora i način njegove realizacije, osnovni zahtevi koji se postavljaju pred svaki regulisani sistem su: *stabilnost*, *tačnost* i *brzina odziva*. Tri navedena zahteva su i osnovni problemi regulacije pa teorija automatskog upravljanja prvenstveno treba da odgovori na ta pitanja [5].

5.2. PID regulatori

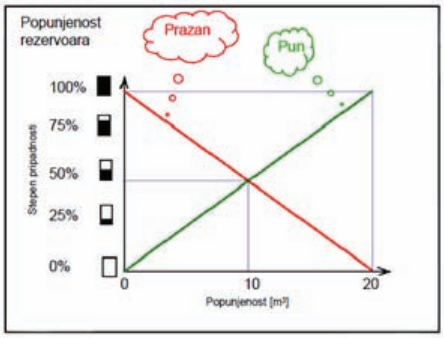
Postoji veliki broj različitih tipova regulatora, ali ako se razmatraju regulatori u kontinualnim linearnim sistemima, onda se njihova klasifikacija najčešće vrši na osnovu vrste zakona upravljanja. Kod osnovnih zakona upravljanja, njihovo dejstvo može linearno da zavisi od greške, njenog integrala ili prvog izvoda greške po vremenu. Na bazi ove činjenice sledi podela regulatora na proporcionalni, integralni i izvodni (diferencijalni) regulator. Pomoću navedenih osnovnih regulatora moguće je formirati složenije proporcionalno-integralne i proporcionalno-diferencijalne regulatore, kao i najsloženije proporcionalno-integralno-diferencijalne regulatore.

Kombinovanjem sva tri osnovna zakona upravljanja se dobija PID regulator čije se ponašanje može opisati sledećom jednačinom:

$$u(t) = K_p e(t) + K_i\int_0^t e(t)dt + K_d\frac{de(t)}{dt} = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}\right) \quad (4)$$

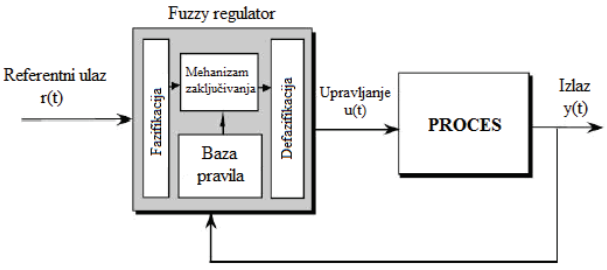
5.3. Fuzzy regulatori

Fuzzy regulatori su regulatori bazirani na fuzzy logici. Osnovni elemenat za predstavljanje i obradu nepreciznosti u fuzzy tehnologijama je fuzzy skup. Fuzzy skup predstavlja skup elemenata sa sličnim svojstvima, dok je diskretan (klasičan) skup skup elemenata sa istim svojstvima. U diskretnom skupu svaki elemenat pripada tom skupu sa stepenom pripadnosti 1, dok u fuzzy skupu svaki element pripada tom skupu u izvesnom stepenu. Na slici 5. dat je primer jednog fuzzy skupa (popunjenost rezervoara).



Slika 5. *Primer fuzzy skupa*

Na slici 6. prikazana je struktura fuzzy regulatora.



Slika 6. *Struktura fuzzy regulatora*

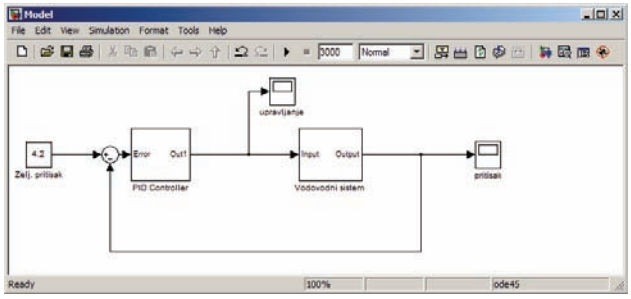
- **Baza pravila** sadrži znanje o tome kako najbolje kontrolisati sistem, i to u formi skupa logičkih (if – then) pravila.
- **Mehanizam zaključivanja** je mehanizam za procenjivanje koja kontrolna pravila su relevantna za trenutno stanje sistema i odlučuje logičkim sklopom kakav će biti upravljački signal, tj. ulaz u proces.
- **Fazifikacija** naprosto modifikuje signale ulaza tako da mogu biti pravilno protumačeni i upoređeni sa pravilima u bazi pravila. *Crisp* signal pretvaramo u adekvatan *fuzzy* oblik.

- **Defazifikacija** transformiše zaključak interfejsa u takav oblik signala da ovaj može biti signal koji predstavlja ulaz u proces. Ovo je transformacija fuzzy oblika u crisp oblik signala, koji je „razumljiv“ procesu.

U osnovi, na fuzzy regulator treba gledati kao na veštačkog donosioca odluke koji radi u sistemu sa zatvorenom spregom u realnom vremenu. On „sakuplja“ podatke izlaza procesa, upoređuje ih sa referencom i onda na način svojstven fuzzy logici „odlučuje“ šta u tom trenutku treba da bude ulaz procesa, i to tako da se zadovolje željene performanse i zadati ciljevi specifikacije [6].

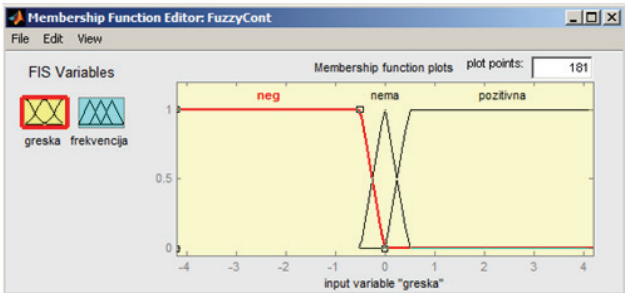
6. MODELIRANJE I SIMULACIJA SISTEMA U MATLAB-U

Na slici 7. prikazan je model realnog sistema, gde se upravljanje vrši pomoću PID regulatora. Cilj je održavanje željenog pritiska u sistemu. Model je izrađen u MATLAB-ovom **Simulink**-u, koji omogućava jednostavnu implementaciju i simulaciju raznih modela i sistema. [7].



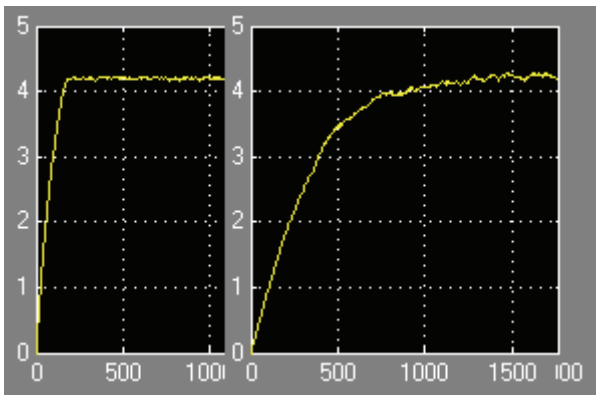
Slika 7. *Model sistema – upravljanje PID regulatorom*

Model sistema sa fuzzy regulatorom identičan je prikazanom na slici 7, s tim što se umesto PID nalazi blok sa fuzzy regulatorom. Realizovan jednostavan kontroler sa dve lingvističke promenljive, jedna je ulazna i predstavlja grešku, tj. razliku između željene i stvarne vrednosti, a druga izlazna i predstavlja upravljačku frekvenciju. Definisanje LP, kao i zadavanje *ako-onda* pravila, vrše se u MATLAB-ovom **FIS Editor**-u. Postupak je prikazan na slici 8.



Slika 8. *Definisanje ulazne LP (greška)*

Na slici 9. prikazano je poređenje kontrole pritiska fuzzy (levo) i PID (desno) regulatorom. Parametri PID regulatora identični su onima iz realnog sistema.



Slika 9. *Kontrola pritiska fuzzy (levo) i PID (desno) regulatorom*

7. ZAKLJUČAK

SCADA aplikacija uspešno je realizovana i testirana, trenutno je u funkciji. Matematički model predstavlja uprošćen model sistema, pošto se eksperimentalno određivanje parametara vršilo oko radne tačke, čime se vršila linearizacije nelinearnog sistema. PID regulacija u potpunosti zadovoljava potrebe upravljanja kod ovakvih sistema, međutim, nedostatak je da se pri odabiru parametara mora ostvariti određeni kompromis, pošto su promene u sistemu različitog karaktera. Fuzzy regulator se u ovom slučaju pokazao kao bolji, naime, potrebno mu je mnogo manje vremena da sistem dovede u radnu tačku, a opseg odstupanja vrednosti upravljane promenljive od željene vrednosti (tj. greške) mnogo je manji.

8. LITERATURA

[1] sr.wikipedia.org/sr-el/SCADA
[2] Branka Šuka, Nadzor rada u kotlarnici realizovan upotrebom client/server aplikacije u programskom okruženju Delphi, Biblioteka Fakulteta tehničkih nauka , Novi Sad, 2001
[3] Milić Stojić: Kontinualni sistemi automatskog upravljanja; Nauka, Beograd
[4] Roland S. Burns, "Advanced Control Engineering", Butterworth-Heinneman 2001
[5] Karl J. Astrom, Bjorn Wittenmark, "Computer-Controlled Systems – Theory and Design", 3rd Edition, Prentice-Hall 1997
[6] Kratak uvod u fuzzy logiku i upravljanje, FTN Novi Sad – Inteligentni upravljački sistemi
[7] MATLAB SIMULINK – Simulation and Model Based Design

Kratka biografija:



Igor Bende rođen je u Novom Kneževcu 1986. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Automatika i upravljanje sistemima odbranio je 2011.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

PRIMER REALIZACIJE FUZZY REGULATORA ZA UPRAVLJANJE SISTEMOM GREJANJA/HLAĐENJA U STAMBENOM OBJEKTU

EXAMPLE OF REALIZATION THE FUZZY CONTROLLER FOR MANAGEMENT SYSTEM OF HEATING / COOLING IN THE RESIDENTIAL AREA

Borivoj Jovišić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu su prikazani principi na kojima se zasniva rad Barix kontrolera, Modbus protokola kao i osobine RS-485 serijske komunikacije. Takođe je predstavljen i fuzzy logički regulator , karakteristike i način rada ovog projektovanog fuzzy logičkog regulatora za upravljanje sistemom grejanja/hlađenja u stambenom prostoru.*

Abstract – *In this paper are presented principles on which is based the work of the Barix controllers, Modbus protocol and characteristics of the RS-485 serial communication. Also, here is presented and the fuzzy logic controller, characteristics and the way of operating this designed fuzzy logic controller for management system of heating / cooling in the residential area.*

Ključne reči: Barix, MODBUS, RS-485, Fuzzy regulator, kućna automatika.

1. UVOD

Globalna svetska politika i iscrpljenost izvora fosilnih goriva uzrok su imperativne strategije svake zemlje-energetska nezavisnost i program korištenja obnovljivih izvora energije. U skladu sa ovim, projektuju se i grade kuće koje koriste najjeftinije energente i imaju mogućnost kombinovanog korištenja raznovrsnih energenata. "Pametna kuća" je pojam koji podrazumeva projektovanje u skladu sa prirodom, ulaganje u gradnju koja pruža komfor, toplotnu, zvučnu i fizičku zaštitu. "Pametna kuća" svakako zahteva veća ulaganja od klasično projektovanog objekta, no u fazi korištenja, ta se investicija višestruko isplati**.** "Pametna kuća" je sistem koji je po meri korisnika programiran i njemu prilagođen [1][2].

2. PAMETNE KUĆE

Sistem kućne automatike (takođe nazvan pametna kuća) je sistem koji u najširem smislu podrazumeva integraciju raznih, pre svega električnih uređaja u domaćinstvu u sistem kojim se može upravljati na različite načine.
Kuća je "pametna" jer se "prilagođava" trenutnoj aktivnosti, raspoloženju, navikama i životnom stilu svakog ukućana, a pritom ostvaruje energetske uštede. Pored životnog stila, komfora, bezbednosti i sigurnosti stanara, racionalnog korištenja uvek oskudnih vremenskih resursa, koncept podrazumeva i izgradnju i prisustvo svesti o potrebi za pažljivim i racionalnim iskorištavanjem energetskih resursa, čija će ekonomska cena biti sve veća, a globalna raspoloživost sve manja, u novom veku. Zadovoljiti potrebe za kvalitetnijim životom savremenog domaćinstva, a pritom ostvariti značajne energetske uštede, prevashodni je zadatak kućne automatike [2].
Postoji čitav niz tehnoloških standarda i protokola kojim je omogućeno upravljanje uređajima u kući. Neki od njih su: Echelon LonWorks (LON), EIB-KNX, Bac-Net, X10, Modbus, itd. Najčešće korišten protokol je X10, zbog mogućnosti da se sistem integriše na već postojeću električnu instalaciju bez građevinskih radova, koji je praćen SWT (Security Wireless Technology) standardom. Kompletna komunikacija između uređaja i upravljačkih modula funkcioniše preko postojećih električnih instalacija. Sistem je potpuno otvoren za eventualnu dalju nadogradnju u slučaju novih potreba, prilagođavanja, reprogramiranja, tako da je obezbeđen uvek 100% stepen iskorištenja i uvek optimalan odnos između cene i korisnosti sistema.

2.1. Načini upravljanja

Sistemom kućne automatike upravljamo na 4 načina: upravljanje univerzalnim daljinskim upravljačem, upravljanje mobilnim ili fiksnim telefonom, upravljajne računarom, upravljanje pomoću touch-screen-a. Ova četiri načina upravljanja ne idu nužno zajedno, već se koriste načini upravljanja koje korisnik preferira i za kojima ima potrebe [3].
Najčešći način upravljanja kućnom automatikom je upravljanje univerzalnim daljinskim upravljačem (IR/RF). Grupe rasvete možemo uključivati, isključivati i dimovati, dok se ostali kućni aparati mogu uključivati ili isključivati. Kada radi u RF (Radio frequency) modu, daljinski upravljač, upravlja "kroz zidove" kućnom automatikom, dok pri radu u IR (Infrared), daljinski upravljač upravlja u jednoj prostoriji i zamenjuje sve postojeće daljinske upravljače sa IR prijemnikom [3]. Sistemom kućne automatike može se daljinski upravljati i preko mobilnog ili fiksnog telefona, sa bilo koje lokacije koja je pokrivena telefonskom mrežom (GSM-Global System for Mobile Communications ili fiksnom telefonijom). Funkcija upravljanje i kontrolisanja kućne automatike telefonom najčešće se koristi za kontrolu klime, grejanja, bojlera, saune, bazena i ostalih većih potrošača kojima je potrebno određeno vreme da dostignu odgovarajuće temperature i da "pripreme" kuću za dolazak stanara [3].

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Velimir Čongradac, docent.

Sistemom kućne automatike moguće je jednostavno daljinski upravljati preko internet mreže kućnim ili dislociranim (prenosnim računarom).
Jednostavnim komandama korisnik upravlja automatizovanim uređajima i po želji menja preprogramiranu logiku sa drugog dislociranog računara kojim je pristupio internet mreži. Ovaj vid upravljanja, svoju najintezivniju primenu nalazi kod korisnika koji veliki deo radnog i slobodnog vremena provode na putovanjima, koji žive i rade u inostranstvu, koji svakodnevno koriste prenosne računare i kojima je slobodno vreme kritičan resurs [3].
PLC (Programabilni logički kontroler) dodajemo kada korisnik želi da upravlja kućnom automatikom preko touch-screen-a u boji ili želi da kontroliše osim standardnog upravljanja u uključi/isključi režimu i neke analogne veličine (temperaturu u svakoj prostornoj celini, dimovanje rasvete, itd.).
Za razliku od standardne funkcionalnosti sistema, za ovu opciju upravljanja potrebno je postaviti odgovarajuće žičane instalacije [3].

3. BARIX KONTROLERI

Barix Barionet 100 je potpuno programabilni mrežni kontroler zasnovan na IP komunikaciji. Sa Barionet 100 kontrolerom svaki uređaj automatskog sistema može da se umreži i lako kontroliše preko standardnog web browsera. Barionet 100 je modularna komponenta koja funkcioniše samostalno ili sa drugim jedinicama (web serverima i sistemima kontrole) i na taj način pruža mogućnost izgradnje velikog, inteligentnog, distribuiranog sistema za širok spektar komercijalnih kontrola, praćenja i primena u kućnoj automatizaciji. Brojni ulazi i izlazi čine uređaj fleksibilnim i prilagodljivim za bilo koju vrstu korisničkih aplikacija [4].
Barix modul X8 je I/O (ulazno/izlazni) modul koji se koristi za razne primene u kućnoj automatizaciji. Koristi industrijski standardan dvožični RS-485 Modbus protokol. Uređaj može da se kontroliše sa bilo kog uređaja. Preko RS-485 Modbus konvertora svaki I/O port može da se konfiguriše kao ulaz ili izlaz [4].
BCL (Barix Control Language) je jezik visokog nivoa koji se koristi za programiranje određenih Barix kontrolera. Jezik ima ugrađene podrške za razne ulazno/izlazne module i za različite mrežne protokole. BCL programi mogu biti pisani u bilo kom tekstualnom editoru koji podržava standardnu ASCII (American Standard Code for Information Interchange) datoteku. Primer takvog editora je Notpad aplikacija koja se isporučuje sa Microsoft Windows operativnim sistemom. BCL datoteke sa izvornim kodom imaju **.bas** ekstenziju. BCL programski prevodilac izvršava programe u Barix TOK formatu.
Tokenizer alatka se koristi za konverziju ASCII koda BCL programa u Barix **TOK** format. Alatka **web2cob** se koristi za pakovanje sadržaja direktorijuma sa svim potrebnim fajlovima u jedan **COB** fajl koji se može direktno učitati na Barix uređaj.
Navedena veličina cob fajla može biti učitana na stranicu flash memorije odgovarajućeg Barionet uređaja pomoću TFTP (Trivial File Transfer Protocol) protokola ili preko web interfejsa [4].

4. KOMUNIKACIJA

4.1. MODBUS protokol

Modbus interfejsi su jednostavni za implementaciju i zbog toga su korišteni od strane mnogih proizvođača. Zbog toga je Modbus postao prvi široko prihvaćeni Fieldbus standard. Modbus je komunikacioni interfejs zasnovan na master/slave (vodeći/prateći) arhitekturi. Komunikacija između modbus čvorišta se obavlja porukama. Prvi Modbus interfejsi su radili pomoću RS-232 serijske komunikacije, ali većina kasnijih Modbus implementacija koristi RS-485 serijsku komunikaciju jer omogućava veće razdaljine (do 1200 m), veće brzine (do 35 Mbps). Dok se ranije Modbus uglavnom koristio na žičane serijske komunikacione linije, postoje i proširenja standarda za bežične komunikacije i TCP / IP mreže [5].
Modbus serijske veze mogu da koriste dva osnovna načina prenosa: ASCII ili RTU (Remote Terminal Unit). Prenosnim modom u serijskoj komunikaciji definiše se način na koji su poruke kodirane. U Modbus/ASCII režimu poruke su u čitljivom ASCII formatu. Modbus/RTU format koristi binarno kodiranje koji čini poruku nečitljivom prilikom praćenja, ali smanjuje veličinu svake poruke koja omogućava razmenu podataka. Svi čvorovi na jednom Modbus mrežnom segmentu moraju da koriste isti serijski režim prenosa. Svaka Modbus poruka ima istu strukturu. Četiri osnovna elementa su prisutna u svakoj poruci. Redosled ovih elemenata je isti za sve poruke, da bi se lako identifikovao sadržaj poruke [5].
Prva informacija u svakoj Modbus poruci je adresa prijemnika (**Device address**). Važeće adrese su u opsegu od 0...247. Vrednosti 1...247 su dodeljene pojedinačnim uređajima a 0 se koristi kao broadcast adresa (šalje se svim uređajima). U okviru Modbus uređaja holding registrima (registrima za čuvanje podataka), ulazima i izlazima su dodeljeni brojevi između 1 i 10000.
Drugi parametar u svakoj Modbus poruci je kod funkcije (**Function code**). Kod funkcije definiše tip poruke i vrstu akcije koja se zahteva od slave uređaja. Važeći funkcijski kodovi su u opsegu od 1..255. Najčešći kodovi su: **01**-Read coil status, **02**-Read input status, **03**-Read holding registers, **04**-Read input registers, **05**-Force singl coil, **06**-Prest singl coil, **07**-Read exception status, **15**-Force multiple coils, **16**- Preset multiple registers, **17**-Report slave ID.
Treći parametar u svakoj Modbus poruci su podaci sa dodatnim informacijama (**Data)**.
Četvrti parametar je deo za proveru ispravnosti poruke. (**Error check**) je obavezan i sprovodi se nezavisno o odabiru paritetne provere znakova. Zavisno od varijante prenosa, ispravnost poruke proverava se pomoću LRC (Longitudinal Redundancy Check, za ASCII varijantu prenosa) odnosno CRC (Cyclic Redundancy Check, za RTU varijantu prenosa) [5].

4.2. RS-485 serijska komunikacija

RS-485 je trenutno široko korišten komunikacioni interfejs za akviziciju podataka i upravljanje aplikacijama gde više čvorova komuniciraju jedni sa drugima. RS-485 signali se prenose preko signalne **+ linije** i signalne **- linije**. Najbolji rezultati se postižu ako su signalne linije upredene.RS-485 je jedini interfejs koji je u stanju da

poveže više predajnika i prijemnika u istoj mreži (do 256 uređaja). Takođe, pomoću RS-485 ponavljača (ripitera) moguće je taj broj proširiti na nekoliko hiljada, podeljanih u više kilometara. RS-485 se koristi kao fizički sloj za mnoge poznate komunikacione interfejse, uključujući Profibus i Modbus [6].

5. OPIS REŠENJA PROBLEMA

Pri rešenju problema potrebno je realizovati sistem "pamente kuće" sa osnovnim funkcijama kao što su: regulacija temperature u objektu (grejanje i hlađenje), upravljanje rasvetom, upravljanje ventilacijom i upravljanje roletnama baziranim na Barix kontrolerima. Pri realizaciji problema korištena je RS-485 komunikacija kao i Modbus protokol za rezmenu poruka. Kao odgovarajući kontroleri izabrani su kontroleri Barix Barionet 100 i ulazano/izlazni modul X8

Na početku je potrebno podesiti odgovarajuće parametre Barionet kontrolera. Barionet 100 je IP adresabilni kontroler, pa je neophodno dodeliti odgovarajuću IP adresu kontroleru (u našem slučaju uzeta je IP adresa 192.168.1.210 i Net maska 255.255.255.0).

Pošto je modul X8 ulazno/izlazni modul potrebno je podesiti da se modul koristi kao ulazni modul (8 ulaza) kao i parametre Modbus protokola i RS-485 komunikacije. To se vrši podešavanjem registara (8-bitni registri) 274, 275, 276 modula X8. Registar 274 se koristi za podešavanje serijske komunikacije. Bit 0 označava brzinu komunikacije gde vrednost 0 datog bita predstavlja brzinu od 19200 bit/s dok vrednost 1 predstavlja brzinu od 9600 bit/s. Bit 7 se koristi za izbor pariteta gde vrednost 0 označava deljiv paritet dok vrednost 1 označava da nema pariteta. Ostali biti nisu od interesa. Kao parametri RS-485 komunikacije izabrani su sledeći parametri (po default-u): Tip interfejsa: RS-485 (2 žice), brzina interfejsa: 19200 bit/s, biti podataka: 8 bita, paritet: omogićen (deljivi paritet), stop bit: 1 bit. Registar 275 se koristi kako bi se dodelila Modbus adresa modulu X8. Vrednosti adresa su u opsegu od 0-255, gde je adresa 255 broadcast adresa, što znači da je moguće adresirati 256 modula. U našem slučaju izabrana je Modbus adresa 255. Registar 276 se koristi da bi se odredilo da li se modul koristi kao ulazni ili izlazni modul. Vrednost 0 određenog bita označava da se dati I/O koristi kao ulaz dok se vrednost 1 označava da se odgovarajući I/O koristi kao izlaz. U našem slučaju svi I/O se koriste kao izlazi. Kao Modbus prenosni režim izabrana je Modbus/RTU varijanta prenosa. Pomoću BCL (Barix Control Language) napisan je programski kod pomoću kog se vrši upravljanje određenom grupom uređaja. Datoteke sa programskim kodom su pomoću tokenizer, web2cob i tftp alata prevedene u format koji je razumljiv samom kontroleru i spušten na njega. Programiranje u programskom jeziku BCL i spuštanje koda na sam kontroler objašnjeno je u radu u poglavlju « BCL (Barix Control Language)». Korisnička aplikacija koja omogućava upravljanje sistemom krajnjem korisniku, napravljena je u programskom jeziku HTML. Interakcija BCL programskog koda sa web aplikacijom se odvija preko HTML tagova. Ispisivanje vrednost BCL promenljivih na web stranici vrši se pomoću taga &LBAS. Sintaksa ovog taga je: **&LBAS (1,<format >,<naziv_promenljive>);.** Tag vraća vrednost **[NO_VAR]** ako ne postoji promenljiva sa tim nazivom. Zadavanje vrednost BCL promenljivoj pomoću korisničke aplikacije sa web stranice vrši se pomoću skripta **basic.cgi.** Sintaksa ovog taga je:

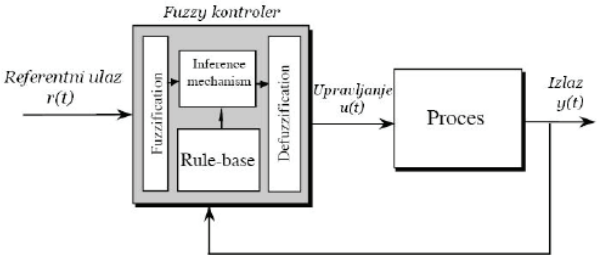
<a href=/basic.cgi?promenljiva=xxx,y,z /a>

gde je: *basic.cgi*- ime skripta, *promenljiva* –ime promenljive kojoj se zadaje vrednost , *xxx*- modbus adresa uređaja , *y*-broj promenljive, *z*- vrednost promenljive. Primer basic.cgi skripta je:

<a href=/basic.cgi?output=255,5,1 /a > što znači: na uređaju čija je Modbus adresa 255 stanje izlaza broj 5 postaviti na vrednost 1. Regulacija temperature u prostoriji je realizovana pomoću fuzzy logičkog regulatora čiji je postupak projektovanja opisan u radu u poglavlju «Fuzzy upravljanje» - «Projektovani fuzzy logički regulator».

6. FUZZY UPRAVLJANJE

Cilj *fuzzy* pristupa je da, umesto da jezikom matematike pokuša da što bolje reši problem upravljanja sistemom, omogući implementaciju inženjerskog iskustva o procesu u sam algoritam kontrolera [7][8]. Fazi regulator je pogodan za upravljanje sistemima koji se vrlo teško mogu modelovati. Kod projektovanja fazi regulatora redosled operacija je sledeći: 1. Analiza procesa, 2. Dobijanje pravila od strane eksperta, 3. Simulacija fuzzy regulatora. Realizacija fuzzy regulatora u digitalnom sistemu upravljanja se svodi na pisanje programskog koda koji na osnovu zadate vrednosti dostiže tu vrednost i održava je sve dok se u sistemu ne dogodi zahtev za nekom drugom vrednošću koju postavlja korisnik. Da bi se napisao program za fuzzy regulator potrebno je na neki način uvesti skup fazi pravila koja opisuju akciju eksperta. Situacija u sistemu upravljanja se može opisati na sledeći način: ekspert posmatra sistem (npr. koliko trenutna vrednost temperature odstupa od zadane) i na osnovu toga donosi odgovarajuću odluku koliko dugo grejati [7][9].



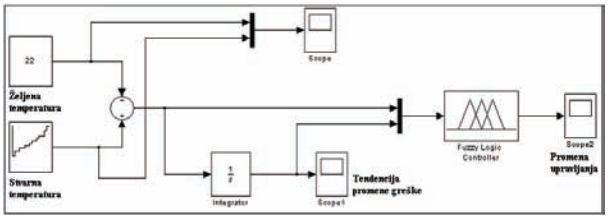
Slika 1**.** *Struktura fuzzy regulatora*

6.1. Projektovani fuzzy regulator

Za regulaciju temperature u objektu implementiran je fuzzy regulator. Model fuzzy regulatora koji simulira regulaciju temperature u objektu napravljen je u programskom jeziku Matlab i simuliran u Simulink-u. Kao uređaj kojim se reguliše temperatura uzeta je fan coil jedinica. Na slici 2. prikazan je model fuzzy regulatora u Simulinku.

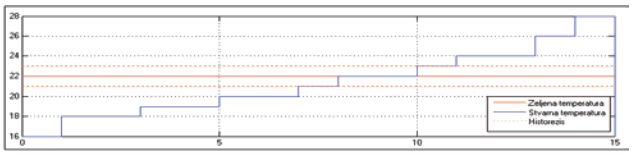
Na ulaz fuzzy regulatora se dovode rezlika između željene i trenutne vrednosti temperature kao i tendencija greške koja se dobija integraljenem pomenute razlike. Fuzzy regulator na osnovu tih ulaza i na osnovu definisanih

fuzzy pravila definiše odgovarajuću promenu upravljanja koja utiče na brzinu i režim rada fan coil jedinice.



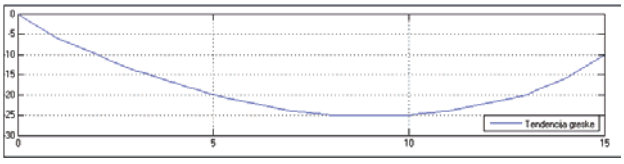
Slika 2. *Model fuzzy regulatora u Simulinku*

Pri definisanju greške temperature kao funkcije pripadnosti fuzzy skupovima korištene su trapezoidne funkcije pripadnosti. Za opseg greške uzete su vrednosti od -8 do 8, sa obzirom da ne bi bilo poželjno da temperatura opada ispod 14 °C kao i raste iznad 30 °C. Opseg je podeljen u nekoliko delova na osnovu kojih se određuje željena brzina i režim rada fan coil jedinice. Prilikom definisanja optimalnog opsega temperature u obzir se uzeo ±1 stepen histerezisa. Režimi rada fan coil-a su : -1 (hlađenje) ,1 (grejanje) a brzine rada su 1, 2, 3. Upravljanje režimom ventila realizovano je po principu ON/OFF (0-100%). Primer rada fuzzy regulatora je sledeći: ako je greška u opsegu [-5, -4] a tendencija je u rastu tada će režim rada biti grejanje a brzina 3. Međutim, ako je u ovom opsegu tendencija u opadanju režim će isto biti grejanje ali će brzina biti 2. Na sledećoj slici je prikazan primer razlike temperature u određenom vremenskom intervalu.



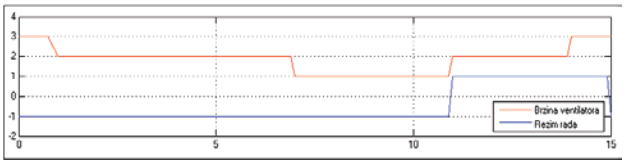
Slika 3. *Regulacija temperature u određenom vremenskom interval pomoću projektovanog fuzzy regulatora*

Na sledećoj slici je prikazana tendencija greške za gore prikazanu promenu temperature.



Slika 4. *Tendencija promene greške*

Na sledećoj slici je prikazana promena brzina rada Fan coil jedinice kao i njen režim rada za gore prikazanu promenu temperature.



Slika5. *Promena brzine fan coil-a i promena režima rada*

7. POREĐENJE RADA PROJEKTOVANOG FUZZY REGULATORA I TERMOSTATA iT8

iT8 termostat u sebi ima ugrađen PI regulator, koji poseduje dobre osobine za regulaciju temperature. Raznim metodama može se vršiti podešavanje parametara regulatura sa ciljem da se dobiju što bolji rezultati. Fuzzy upravljanje danas sve češće predstavlja uspešnu alternativu klasičnom upravljanju posebno u slučajevima kada se objekat upravljanja opisuje nelinearnim matematičkim modelom ili se čak matematički model uopšte ne može postaviti. Prednost fuzzy regulacije sastoji se u tome što zahteva slabije numeričke resurse a fleksibilna struktura samog regulatora omogućava vrlo jednostavnu i brzu nadogradnju funkcija. Što se tiče konkretno našeg problema, s obzirom da se radi o jednostavnom problemu upravljanja, gde je zakon upravljanja jednostavan i može lako da se modeluje, oba oblika upravljanja mogu da se koriste i lako da se implementiraju. Fuzzy regulator predstavlja savremenije i jednostavnije rešenje upravljanja sa mogućnošću jednostavnijeg modifikovanja i fleksibilnosti upravljanja, i zbog tog razloga opredelili bi se za upravljajne i regulaciju pomoću fuzzy regulatora.

8. ZAKLJUČAK

Prednosti pametnih kuća se ogledaju u prilagodljivosti, štedljivosti, racionalnosti, ali pre svega u ispravnom izboru energije koju koriste. Fuzzy upravljanje danas sve češće predstavlja uspešnu alternativu klasičnom upravljanju posebno u slučajevima kada se objekat upravljanja opisuje nelinearnim matematičkim modelom ili se teško može modelovati. S obzirom da se našem problemu radi o jednostavnom upravljanju, gde zakon upravljanja može lako da se modeluje, oba gore pomenuta načina upravljanja mogu da se koriste i lako da se implementiraju. Zbog savremenijeg i jednostavnijeg rešenja upravljanja opredelili smo se za upravljanje i regulaciju pomoću fuzzy regulatora. Projektovani fuzzy regulator je pokazao dobre osobine u pogledu projektovanja, modifikovanja i eventualne nadogradnje sistema.

9. LITERATURA

[1] http://bms.co.rs/
[2] http://www.antel.rs/pametne-kuce-2.html
[3] http://www.pametnakuca.rs/live/nacini_upravljanja
[4] http://www.barix.com
[5] http://www.lammertbies.nl/comm/info/modbus.html#fc01
[7] http://www.mikroe.com/sr/magazine/3broj/3broj1.htm
[8] *Oscar Castillo, Patricia Melin, „Fuzzy Logic: Theory and Applications“, 2008.*
[9] *Timothy J. Ross, „Fuzzy logic with engineering applications“, 2004.*

Kratka biografija:



Borivoj Jovišić rođen je u Mrkonjić Gradu 1986. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektro-tehnike i računarstva – Automatika i upravljanje sistemima odbranio je 2011.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

LON IP GATEWAY KAO KOMUNIKACIONI INTERFEJS KA SISTEMIMA KUĆNE AUTOMATIKE

LON IP GATEWAY AS A COMMUNICATION INTERFACE TO HOME AUTOMATION SYSTEMS

Nebojša Apostolovski, *Fakultet tehničkih nauka*, *Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U okviru rada je izvršeno upoznavanje sa LON protokolom i mogućnostima njegovog proširenja ka IP mrežnim sistemima. Napravljena je web aplikacija za nadzor i upravljanje parametrima datog sistema (DataPoint-ima kao svojstvenim promenljivama LON sistema). U okviru aplikacije omogućen je direktan pristup željenim promenljivama, kao i mogućnost kreiranja uređaja koji se vezuju za prostor u kome se uređaj nalazi i koje opisuju naziv DataPoint-a na koji se uređaj povezuje. Spisak komandi koji je moguće obaviti nad određenim DataPoint-om određivao se na osnovu tipa promenljive prilikom kreiranja uređaja.*

Abstract – *Doing this work, I acquainted myself with LON protocols and the possibilities of its extension to IP network systems as well. It is created a web application to control and operate the system parameters (DataPoint specific for LON systems). Within that application, it is enabled a direct access to desired variables and the possibility of creating a device which is linked to the space where that device is and describe the name of DataPoint to which the device is connected. The list of commands we can take over some DataPoint is determined by virtue of the type of the variable during creation of the device.*

Ključne reči*: iLON100 server, SOAP poruka, LonWorks, Data point, IP.*

1. UVOD

LON protocol, koji je ustanovljen u ranim 1990-im, nije se promenio. Uređaji izrađeni tada će zajednički raditi na istoj mreži kao i uređaji ugrađeni danas. Zvanično ime protokola je ANSI CEA 709.1 ili EN- 14908 protokol i nalazi se pod kontrolom SAD i evropskih standardizacionih tela. Osnovno pravilo je da, u cilju sprovođenja protokola, ceo protokol mora biti realizovan u skladu sa specifikacijama. Ne postoje „opcioni“ elementi protokola, koji bi mogli učiniti implementaciju različitih programa ne interoperabilnom [1],[2].
Tehnički, LON standard pokriva svih sedam slojeva OSI (Open Systems Interconnect) referentnog modela od fizičkog interfejsa, kao što su žičani, power line ili RF, pa sve do aplikacionog sloja i svih slojeva između. Glavna razlika između LON protokola i drugih jezika jednako priznatih, kao što su BacNet, ModBus i Zigbee, je da je LON dizajniran od dna pa naviše kao platforma sa kontrolom komunikacije.

LON protokol nije ograničen na određenu oblast primene kao što je kontrola u zgradama ili HVAC (*Heating, Ventilation and Air Conditioning* - HVAC). Koncept se proteže od komercijalnih objekata do industrijskih postrojenja, preko kućne/apartmanske automatizacije, komunalnog i energetskog nadgledanja, transporta, ulične rasvete, maloprodaje, fast food restorana, vojne primene i još mnogo toga. Iz toga se može videti da je potencijal za usvajanje LON protokola širom sveta u mnogim oblastima primene, a ne samo u upravljanju zgradama [3].

IP Gateway za bilo koju tehnologiju otvara širok spektar novih mogućnosti, proširujući postojeću tehnologiju njegovim prisustvom na internetu. Ovim proširenjem LON mreža može da pruži veći konfor, uštedu energije i smanjenje troškova održavanja.[4] Kao dobar primer uređaja koji uvodi LON u svet IP mreža i interneta jeste Echelon iLON100 server. I u daljem delu rada biće korišćen ovaj uređaj za testiranje mogućnosti koje uvodi komunikacioni interfejs između dveju tehnologija u sistemima kućne automatike.

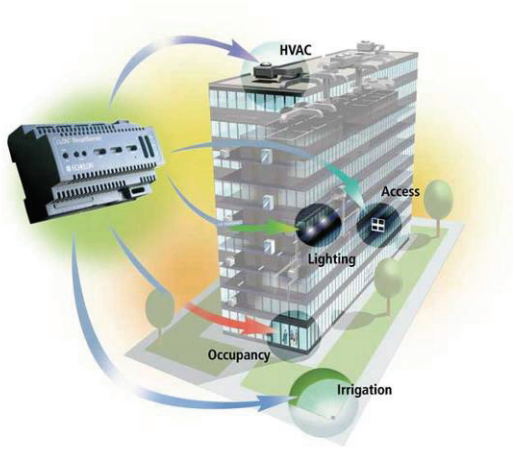
2. iLON 100 internet server

iLON 100 e3 internet server je kontroler, ruter i Web server preko kog se LonWorks, M-Bus i Modbus uređaji spajaju na zajedničke IP mreže ili na internet. iLON 100 server je svojevrstan server koji omogućava Web pristup svim podacima kojim upravlja, kao i unutrašnjim alarmnim, planskim i aplikacijama za skladištenje podataka. Poseduje i mogućnost za Web binder (vezu) za povezivanje više LonWorks domena, kao i mogućnost SOAP/XML Web servisnih interfejsa za upotrebu namenski urađenih Web stranica [5].

Primena iLON servera sa LonWorks mrežnom platformom ima savršene rezultate u kontroli sistema unutar gradova, zgrada, kuća i industrije. Omogućena je upotreba raznovrsnih elektronskih uređaja koji se prostim umrežavanjem preko LonWorks-a koriste za nadzor i upravljanje u jednom sistemu. Primenom iLON servera sistem je potpuno nezavisan u radu od PC-a ili nekog drugog kontrolera (Slika 1.).

Dok su druge mreže daleko bolje u prenosu velike količine podataka ekstremnim brzinama, LonWorks mrežna platforma se odlično pokazala u sistemima kontrole sa „prostim“ podacima.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Velimir Čongradac, doc.



Slika 1. *Prikaz sistema za kontrolu i upravljanje „pametnom“ zgradom uz pomoć iLON 100 servera*

3. SOAP poruke

SOAP (Simple Object Access Protocol) je protokol baziran na XML-u (Extensible Markup Language) kao svom formatu poruke. On definiše set pravila koja se upotrebljavaju za određivanje strukture poruka. Takve poruke se mogu koristiti za prosto jednosmerno slanje, ali se uspešno koriste i za zahtev-odgovor dijaloge. SOAP nije vezan posebno ni za jedan transportni protokol, premda je HTTP (HyperText Transfer Protocol) najpopularniji za prenos poruka. Takođe, nije vezan ni za jedan operativni sistem ili programski jezik, tako da, teoretski, klijent i server (između kojih se vrši dijalog) mogu biti pokretani na bilo kojoj platformi napisanoj na bilo kom jeziku. Sve ovo važi dok i jedan i drugi mogu da formulišu i razumeju SOAP poruke. Kao takav, SOAP protokol je veoma koristan pri razvoju aplikacija koje će se koristiti kao servis na internet-u.

Ovaj XML bazirani protokol sastoji se iz tri dela:

- Envelop (omotnice) – koja definiše šta se nalazi u poruci i kako je treba procesirati
- Set pravila za enkodovanje
- Pravilo za predstavljanje „*call*“ i „*response*“ procedura [6]

Pošto je SOAP po prvi put predstavljen javnosti, vremenom je postao pozadinski nivo mnogo kompleksnijih Web servisa, baziranih na WSDL-u (Web Service Description Language). iLON 100 server poseduje WSDL fajl koji se koristi pri izradi Web stranica.

4. Kreiranje aplikacija za nadzor i upravljanje parametrima LON mreže

Pre bilo kakvog rada potrebno je iskonfigurisati iLON100 server. Ako je moguće povezati se sa iLON100 serverom koristeći Internet Explorer i LonMaker, može se preskočiti naredni korak. U slučaju da pristup nije moguć, neophodno je pristupiti konfiguracionom fajlu pomoću HyperTerminala i podesiti tražene parametre.

Preporučuje se da se koristi serijski interfejs između PC-a i iLON100 servera da bi se potvrdila podešavanja, naročito IP adresa. Ovo je privremena veza. Koristi se za snimanje osnovnih parametara kao što su IP adresa, subnet mask, gateway, host name…

ILON100 server poseduje i web stranicu pomoću koje je moguće jednostavno pristupiti DataPoint-ima preko Web browser-a ukoliko imamo instaliran Microsoft Internet Explorer 6 i noviji. Kucanjem njegove IP adrese, i ne previše komplikovanim kretanjem kroz organizacionu strukturu servera, moguće je pristupiti definisanim DataPoint-ima.

Postoji još nekoliko načina za kreiranja sopstvene, namenski rađene Web stranice za čitanje i upis novih vrednosti DataPoint-a. Oni nam nisu preterano interesantni zbog neophodnosti kombinovanja dva programa (iLon Vision i Macromedia Contribute) i robusnosti istih, iako nam radeći na ovaj način nije potrebno poznavanje JavaScript-a ili HTML-a.

Ipak je odlučeno da se razvija potpuno nova aplikacija u Eclipse programskom paketu posebno prilagođena za Magiesta okruženje namenjeno za upravljanje sistemima kućne automatike.

4.1. Kreiranje komunikacije između aplikacije i iLON100 servera

Svaki iLON 100 server poseduje WSDL (Web Service Description Language) fajl. Ovaj fajl definiše iLON100 SOAP/XML interfejs, i sadrži sve informacije koje će aplikaciji trebati za upotrebu SOAP/XML interfejsa.

Kada se piše ovakva aplikacija, neki alati imaju mogućnost da “pokupe” iLON 100 WSDL fajl i da automatski naprave klasu za slanje i primanje SOAP poruka. Pošto Eclipse ne ostavlja takvu mogućnost, za potrebe generisanja datih klasa bilo je neophodno napraviti potprojekat za generisanje java klasa iz WSDL fajla koji su eksportovani u jar fajlove i kasnije korišćeni u daljoj realizaciji projekta.

U fazi realizacije došlo je do problema prilikom korišćenja izgenerisanih klasa na osnovu WSDL fajla pri pozivu funkcija za izmenu DataPoint-a. Izgenerisane klase su pravile SOAP poruku u kojoj se nalazilo samo zaglavlje sa praznim telom poruke. Ovo je na serverskoj strani izazivalo grešku i aplikaciji je dalje prosleđivana poruka da nedostaju podaci pri pozivu date funkcije i da sintaksa nije ispoštovana. Ovaj problem je prevaziđen pravljenjem sopstvene funkcije za kreiranje SOAP poruke specificirane u dokumentaciji iLON100 servera i njihovim slanjem na zadato odredište [7].

Primer - XML fajl zahtev za čitanje vrednosti promenljivih sa iLON100 servera. Prikazana poruka je konkretno vezana za zahtev za čitanjem DataPoint-a sa nazivom “NVL_Switch” i poljem “state”. U opštem slučaju izgled poruke zahteva će se razlikovati samo u vrednostima ova dva parametra.

```
<iLONDataServer>
 <NVL>
<UCPTpointName>NVL_Switch</UCPTpointName>
<UCPTfieldName>state</UCPTfieldName>
```

```
 </NVL>
</iLONDataServer>
```

Odgovor na ovaj zahtev je oblika:

```
<iLONDataServer>
 <NVL>
 <UCPTindex>310</UCPTindex>
<UCPTpointName>NVL_Switch</UCPTpointName>
<UCPTpointUpdateTime>2011-06-09T00:17:37.870-
07:00</UCPTpointUpdateTime>
<UCPTformatDescription>SNVT_switch</UCPTform
atDescription>
 <UCPTvalue>100.0 1</UCPTvalue>
 <UCPTvalueDef>ON</UCPTvalueDef>
 <UCPTunit></UCPTunit>
<UCPTpointStatus>AL_NO_CONDITION</UCPTp
ointStatus>
 <UCPTpriority>255</UCPTpriority>
 </NVL>
</iLONDataServer>
```

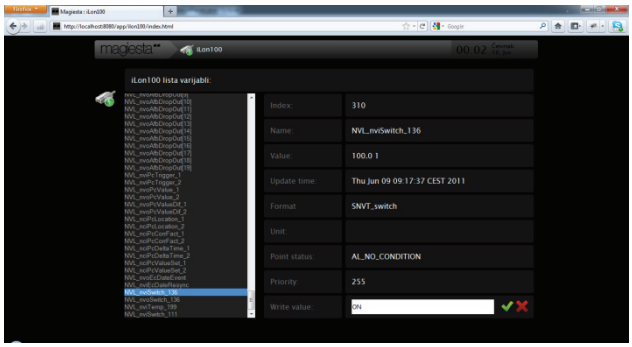
<DPType> : specifikacija tipa DataPoint –a (NVL, NVC ili NVE),
<UCPTindex> : broj koji predstavlja indeks dodeljen DataPoint -u,
<UCPTpointName> : ime DataPoint -a,
<UCPTpointUpdateTime> : vreme poslednjeg ažuriranja,
<UCPTformatDescription> : format promenljive na osnovu koje je kasnije moguće odrediti spisak mogućih akcija nad DataPoint-om,
<UCPTvalue> : trenutna vrednost,
<UCPTvalueDef> : vrednost u predefinisanom format,
<UCPTpointStatus> : status DataPoint-a. Njega setuje alarm generator ili dojava alarmnog stanja,
<UCPTpriority> : nivo prioriteta trenutno zadatog DataPoint-a (0-255). On odlučuje koja aplikacija ima pristup. Najviši nivo je 255 i ako se eksplicitno ne navede drugi nivo, uzimaće se on. [8]

Kao što vidimo, ovakav način razmene poruka, ne samo da nije preporučljiv, nego je i vizuelno loš. U porukama koje dobijamo kao odgovor ima mnogo linija sa različitim informacijama, koje su nama nepotrebne, jer je naša želja da se usredsredimo samo na vrednosti data point-a.

Naša aplikacija će biti u stanju da uradi sledeće: da isčitava sve data pointe koji se nalaze na iLON100 serveru (posebno za svaki tip: NVL, NVE i NVC); da pročita zahtevane vrednosti data point-a po našem izboru; i da izmeni vrednosti data point-a po našem izboru (Slika 2.).

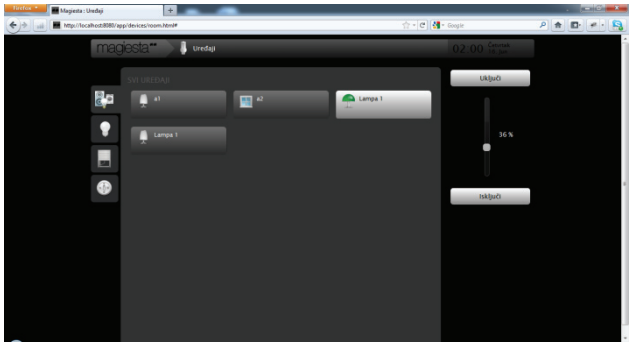
Da bismo bili u mogućnosti da ovo ostvarimo, potrebno je da budemo konektovani na internet (kada je serveru dodeljena javna IP adresa), ili na lokalnu mrežu (ako je serveru dodeljena lokalna IP adresa), preko kojih ćemo pristupati iLON100 serveru koristeći se njegovom IP adresom ili hostname –om.

Pored ovog direktnog pristupa željenim DataPoint-ima, moguće je kreirati i uređaje kojima možemo pridružiti DataPoint kao odredišnu adresu, i kojima je kasnije moguće obavljati niz akcija dozvoljenih datom uređaju na osnovu polja *<UCPTformatDescription>* specificiranom u odabranom DataPoint-u.



Slika 2. *Izlistavanje data point-a i izbor data point-a na kom se izvršava operacija*

Ukoliko je data promenljiva npr. tipa SNVT_Switch nad njom je dozvoljeno obavljanje samo funkcije UKLJUČI/ISKLJUČI, dok je kod dimabilnih uređaja pored ovih funkcija omogućen i odabir vrednosti dimovanja.



Slika 3. *Vizuelni prikaz mogućih akcija nad definisanim uređajem*

Ovakav način nadgledanja i promene vrednosti željenih data point-a je veoma brz i jednostavan. Kao što je rečeno u prethodnom tekstu, konfigurisanje iLON 100 servera je najbolje, najlakše i najsigurnije izvoditi pomoću originalnih (proizvođačevih) Web stranica, dok je za nadgledanje i promene vrednosti data point-a, najjednostavnije kreirati sopstvenu aplikaciju.

5. ZAKLJUČAK

U narednom periodu nastaviće se rast cena energije i energenata i to će direktno uticati na porast troškova života i poslovanja. Stoga se prioritet daje održivoj potrošnji energije, kroz racionalno planiranje same potrošnje i implementaciju mera energetske efikasnosti.

Takođe, sve više će se posvećivati pažnja stvaranju bezbednog, komfornog, produktivnog i ekonomičnog radnog ili životnog okruženja, uz istovremenu štednju energenata, smanjivanje troškova održavanja i širenje svesti o zaštiti životne sredine.

Jedna od mogućnosti za dobijanje sistema koji ispunjavaju ovakve uslove jeste integracija sistema kućne automatike i LonWorks mrežne platforme zajedno sa nekim od inteligentnih SmartServer-a. Mogućnost pristupanja raznovrsnim elektronskim uređajima umreženim preko LonWorks-a sa velikih udaljenosti i njihov nezavisan rad od PC-a ili nekog drugog kontrolera,

još su neke od prednosti korišćenja ovakvih sistema. U okviru ovog rada korišćen je primer iLON100 servera, ali se zaključci mogu pripisati celoj familiji ovakvih uređaja.

Nažalost, širu primenu ovakvih sistema sprečava nedostatak finansijskih sredstava za ulaganje u takve tehnologije, kao i nedovoljna informisanost. Međutim, trendovi pokazuju da se ovakve tehnologije veoma brzo razvijaju, da se cene istih smanjuju iz dana u dan i da se investicija isplati već kroz nekoliko godina, što nam daje za pravo da kažemo da će se ovakva rešenja uskoro pronaći u znatno široj upotrebi, i da će zasigurno opravdati uložena sredstva.

6. LITERATURA

[1] K. W. Song, G. S. Choi, G. H. Choi, and J. S. Kim, "LonWorks-based virtual device network" *Int. J. of Applied Electro-magnetics and Mechanics*, vol. 18. no. 1-3, pp. 67-79, 2003

[2] Gi Sang Choi, "LonWorks/IP", Journal of the Institute of Industrial Technology, Vol. 12, pp. 9～16, 2004

[3] Ron Bernstein, "The LON protocol", Info 4 Security Magazine, pp. 11～12, feb 2008

[4] Save energy and improve operations with the i.LON SmartServer, ECHELON

http://www.echelon.com/solutions/eem/default.htm

[5] i.LON® 100 e3 User's Guide, ECHELON

http://www.echelon.com/support/documentation/manuals/cis/078-0310-01B.pdf

[6] What is SOAP

http://www.soapuser.com/basics1.html

[7] Axis User's Guide

http://axis.apache.org/axis/java/user-guide.html

[8] i.LON® 100 e3 Programmers Reference, ECHELON

http://www.echelon.com/support/documentation/manuals/cis/078-0250-01E.pdf

Kratka biografija:



Nebojša Apostolovski rođen je 11.04.1988. godine u Pančevu. Završio srednju Elektrotehničku školu "Nikola Tesla" u Pančevu 2006. godine i osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu, smer Računarstvo i automatika, Automatika i upravljanje sistemima 2010. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.395.721.5

UPOTREBLJIVOST VIRTUELNIH TASTATURA NA MOBILNIM UREĐAJIMA SA EKRANIMA OSETLJIVIM NA DODIR

USABILITY OF VIRTUAL KEYBOARDS ON MOBILE DEVICES WITH TOUCHSCREEN

Srđan Farkaš, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Pregled tehnologija ekrana mobilnih uređaja sa stanovišta prikaza, reagovanja na dodir i fidbeka. Upoznavanje sa aktuelnim platformama za pametne mobilne telefone. Prikaz i testiranje fabričkih i aplikativnih rešenja virtuelnih tastatura na osnovu kojih su predložena dva nova dizajna.*

Abstract – *Overview of display technology for mobile devices from acpects of view, reaction to touch and feedback. Introduction to current platforms for smart mobile phones. Overview and testing of the factory and software solutions for the virtual keyboards which was used for proposal of two new designs.*

Ključne reči: *virtuelna tastatura, fidbek, veličina tastera, pametni mobilni telefon (smartphone)*

1. UVOD

Personalno računarstvo je do pre par godina podrazumevalo samo desktop računar i noutbuk (notebook, laptop). PDA (Personal Digital Assistant) [1] uređaji u početku, a zatim i pametni telefoni su spadali u drugu kategoriju budući da im je hardver bio skromnih mogućnosti, a funkcije softvera prilagođene korisnikovim potrebama u pokretu.

Pojava netbuka (netbook) 2007. godine (Eee PC kompanije Asus) nije unela neke značajnije novine u IT svet budući da je to bio samo umanjeni noutbuk. Bitno je da je ovim računarom korisnicima stavljeno do znanja da je tehnologija dovoljno napredovala tako da je uz smanjivanje dimenzija samog uređaja moguće održati određen stepen performansi istog.

Iste 2007. godine kada se pojavio Ajfon (iPhone) prve generacije, kompanija Epl (Apple) započinje značajne promene u IT industriji redefinisanjem modela kako bi pametni telefon trebao da izgleda i funkcioniše. 2010. godine izbacivanjem na tržište Ajpeda (iPad), kompanija Epl nije samo oživela stari koncept tzv. tablet računara [2] nego započinje novu eru personalnog računarstva.

Ajfon i Ajped je bilo potrebno pomenuti zato što dele veoma važnu zajedničku karakteristiku, a to je da je korisnički interfejs potpuno prilagođen korišćenju putem ekrana osetljivog na dodir. Oba spadaju u mobilne uređaje s tim što je ekran Ajpeda mnogo veći (10 inča u odnosu na 3.5 inča kod Ajfona). Ostali proizvođači hardvera i softvera su krenuli istim stopama kao i Epl, nudeći telefone sa velikim ekranima osetljivim na dodir (do 4.3 inča), a od nedavno su se u prodaji našli i tablet uređaji (ekrani od 7 do 10 inča). Kako su ovi uređaji mahom bez fizičkih tastatura, pojavio se problem unosa teksta. Isti je rešen virtuelnim tastaturama koje će biti predmet ovog rada. Ukratko, virtuelne tastature su tastature koje se prikazuju na ekranu uređaja i služe za unos teksta.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Dragan Ivetić, r.prof.

2. TEHNOLOGIJE EKRANA

Razvoj displeja kroz istoriju obeležile su dve tehnologije: CRT (Cathode Ray Tube) [3] i LCD (Liquid Cristal Display) [4]. CRT tehnologija je korišćena kod televizora i monitora, a pored velike potrošnje, zajednička osobina im je bila ta da su teški i kabasti. LCD tehnologija se pojavila kao alternativa za male i prenosive uređaje sa malom potrošnjom poput satova, digitrona i ostalih uređaja koji su trebali da prikažu uzak skup informacija, imaju dugu autonomiju rada na bateriji, a da im se pritom ne poveća masa i veličina. Vremenom je CRT tehnologija dostigla svoj zenit dok je LCD konstantno napredovala. Ovo je dovelo do toga da LCD polako ali sigurno postaje dominantna tehnologija prikaza u svim uređajima. Kasnije su se u komercijalnoj primeni pojavile tehnologije kao što su Plasma [5], OLED (Organic Light-Emitting Diode) [6] i elektronski papir (E-Ink) [7]. Kod mobilnih uređaja visoke klase (pametnih telefona i tablet računara), pojavljuju se unapređene varijante pomenutih tehnologija i to S-LCD (Super-LCD) [8], AMOLED (Active-Matrix OLED) [9] i S-AMOLED (Super-AMOLED) [10].

Ekran osetljiv na dodir nije posebna vrsta ekrana. U pitanju je klasičan ekran preko kojeg se postavljaju naročiti slojevi, odnosno folije koji omogućavaju detekciju dodira. Postoji nekoliko impementacija koje ekran čine oseljivim na dodir, a danas se za prenosne uređaje poput mobilnih telefona i tablet računara najviše koriste dve i to, rezistivna [11] i kapacitivna tehnologija [11].

3. EKRANI NA PAMETNIM MOBILNIM TELEFONIMA

Ekrani pametnih mobilnih telefona [12] su oduvek bili veći i sa većom rezolucijom u odnosu na ekrana klasičnih mobilnih teledona. Do pojave iPhone-a, najčešće korišćena rezolucija je bila 240 x 320, a dijagonale nisu prelazile 7.7 cm. Pomenuta rezolucija se i danas koristi za najjeftinije smart telefone, a razlog ovome je što predstavlja najbolji kompromis između kvaliteta slike i

veličine grafičkih elemenata na ekranu. Zbog malih dijagonala, koristi se rezistivna touch-screen tehnologija, kako bi korisnik mogao da koristi stilus za upravljanje telefonom.
Što se rezolucije tiče, ako je ona veća, slika na telefonu je oštrija i dopadljivija korisniku. Posle pojave iPhone-a, dijagonale počinju da rastu, kao i rezolucije. Tako se danas sreće HTC HD7 koji ima dijagonalu 4.3 inča rezolucije 480 x 800 piksela ili iPhone 4 koji na svega 3.5 inča poseduje rezoluciju od čak 640 x 960 piksela odnosno 326 PPI. Podsećanja radi, čovekovo oko može u idealnim uslovima da percipira do 300 PPI.
Na prikaz boja utiče i to koju vrstu ekrana osetljivog na dodir poseduje telefon. Zbog činjenice da rezistivni ekrani imaju foliju koja štiti ekran, ona ujedno povećava refleksiju i ti ekrani su teže vidljivi na suncu. Proizvođači to rešavaju specijlnim premazima ekrana koje imaju za cilj da smanje refleksiju. Primer su telefoni SonyEriccson-a M600i i P1i koji poseduju isti rezistivni ekran ali s tom razlikom što je kod P1i upotrebljen drugi premaz koji ga je učinio bolje vidljivim na suncu.
Kako kapacitivni ekrani osetljivi na dodir nemaju dodatnih folija preko istog, na njima je slika u startu čistija i bolje vidljiva na suncu ali to već zavisi i od panela koji koriste. Jedino se u trenucima kada ekran nema pozadinskog osvetljenja, pod određenim uglom, mogu primetiti provodne trake koje služe za detekciju dodira.

3.1. Fidbek ekrana osetljivih na dodir

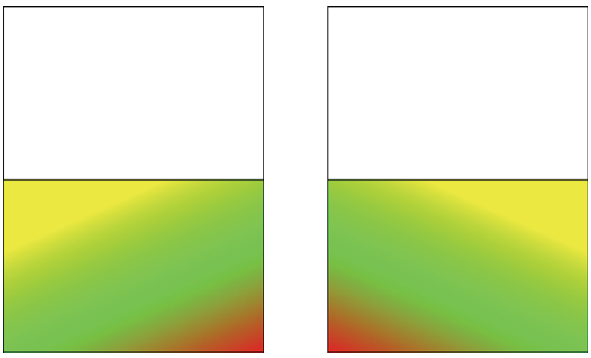
Fidbek (feedback) tj. povratna informacija je reakcija nekog entiteta (u ovom slučaju tastera ili ekrana) na korisnikovu aktivnost. U slučaju virtuelnih tastatura predstavlja krucijalan nedostatak u odnosu na klasične fizičke tastature. Osnovna karakteristika fizičkih tastatura je taktilan osećaj koji korisnik ima pod prstom kada dodirne taster. Oblik, veličina tastera, prostor između njih se tačno može osetiti. To nije slučaj kod virtuelnih tastatura gde se oni prikazuju na ravnom ekranu koji je za prst potpuno jednoličan na dodir. Samim tim, tzv. slepo kucanje je praktično nemoguće na virtuelnim tastaturama. Virtuelne tastature poseduju 3 vrste fidbeka. To su: vibriranje pri dodiru ekrana (tzv. haptic feedback), simuliranje zvuka tastera koji se dobija pri kucanju i vizuelni fidbek. Vizuelni fidbek podrazumeva promenu prikaza na ekranu tako da korisnik vidi rezultat svoje akcije i nije vezan samo za virtuelne tastature.

4. APLIKATIVNA REŠENJA VIRTUELNIH TASTATURA

- **8Pen** predstavlja radikalno drugačiji pristup u dizajniranju kako izgleda tako i funkcionalnosti lejauta. Za razliku od virtuelnih QWERTY tastatura koje zahtevaju jednostavno pointiranje, 8Pen se potpuno oslanja na prevlačenje prsta po ekranu po određenim pravilima.
- **MessagEase** je rešenje koje se odlikuje inovativnošću i optimizovanim lejautom. Poseduje osobine tastature i metode za unos pomoću gestova.
- **Graffiti** tastatura koristi mehanizam za unos koji je jedan od najstarijih kada su u pitanju uređaji sa ekranom osetljivim na dodir. U pitanju je klasično ispisivanje gestova koji predstavljaju slovo, cifru ili interpunkciju. Kako je u ovom rešenju striktno definisino kako treba da izgleda gest za određeni karakter, ono neće biti nazvano mehanizmom prepoznavanja rukopisa. Popularnost je steklo na Palm OS uređajima, ali se sreće i kod Sony-Ericsson pametnih telefona koji su posedovali ekran osetljiv na dodir.
- **SlideIT** je jedno od dva najpopularnija aplikativna rešenja za unos teksta sa Android Marketa. Popularnost je steklo na iOS-u, mada svoje korene vuče još sa Windows Mobile CE platforme gde je poznato kao Swype. Pretstavlja hibrida gde se nad QWERTY tastaturom reči unose gestovima umesto pritiskanjem pojedinačnih tastera.
- **Thick Buttons** je zanimljiv pristup baziran na mehanizmu predikcije koji smanjuje uloženi napor tako što menja dimenziju ciljnog tastera. Kako se počne kucati određena reč, algoritam predviđa koji od karaktera mogu biti sledeći otkucani i povećava njihovu veličinu na lejautu dok ostala smanjuje.
- Ideja iza **SwiftKey** rešenja je podizanje nivoa predikcije na jedan mnogo viši nivo nego što je slučaj sa bilo kojim drugim rešenjem. Algoritam u pozadini uči navike korisnika i sklopove reči koje često koristi. Posle nekog vremena, čitave rečenice se mogu otkucati bez potrebe za dužim kucanjem karaktera jer je sve već predloženo i potrebno je samo pritiskati razmak (način za unos preporučene reči). Zaista impresivno. Čak i u slučaju unosa nekog nepovezanog sklopa reči, relativno brzo se dobija tačna predikcija.
- Svako od prethodnih 6 rešenja je sa sobom donelo neku specifičnost bilo po pitanju lejauta i/ili načinu unosa teksta. **Better Keyboard** igra na drugu kartu, a to je broj opcija za podešavanje kojih brojčano ima više nego kod prethodnih šest rešenja zajedno. To znači da je korisniku ostavljena mogućnost da podesi većinu parametara funkcionalnosti tastature prema svojim potrebama. Po svemu ostalom predstavlja kopiju HTC Sense rešenja, tako da funkcionalnost tastera neće biti ponovo opisana za predstavljanje ovog rečenja.

5. REZULTUJUĆA TASTATURA

Ergonomija virtuelne tastature je veoma osetljiva tema, budući da direktno zavisi od veličine ekrana ali takođe i od veličine šake korisnika, kao i načina na koji on drži telefon u ruci. Generalno, palcu desne ruke odgovaraju potezi severo-istok (SI) – jugo-zapad (JZ) odnosno četvrt kružnice od 270 do 360 stepeni dok levom palcu odgovaraju potezi severo-zapad (SZ) – jugo-istok (JI) tj. od 0 do 90 stepeni. Što se palac više savija, potezi u tom smeru su manje prijatni za korisnika, JI za desni palac i JZ za levi. Zanimljiva činjenica u vezi tih poteza je da iako su manje prijatni, oni su precizniji od recimo SZ za desni i SI za levi palac. To je iz prostog razloga što pri savijanju palca korisnik sve više koristi vrh prsta dok kod ostalih, više dostupnih koristi jagodice, smanjivajući odnosno povećavajući dodirnu površinu prsta [13]. Na slici 1 su zone dodira grafički predstavljene za levi i desni palac, uzimajući u obzir deo ekrana koji bi zauzimala tastatura. Zelenom bojom je označena optimalna zona, žutom zona koja odgovara palcu ali je manje precizna i crvenom je označena zona koja nije prijatna ali je preciznija od žute.

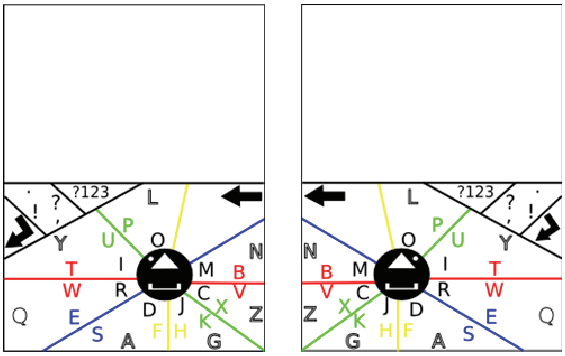


Slika 1. Zone dodira za desni i levi palac na parčetu ekrana koji odgovara tastaturi

5.1. X2 dizajn

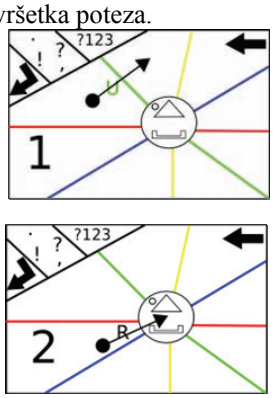
Izgled X2 lejauta (slika 2) za desnoruke korisnike je najlakše obajsniti ako se ouče 3 zone koje se podudaraju sa slikom 1. Prva zona je najmanja i zauzima gornji levi ugao tastature sa crnim graničnim linijama. Poseduje 3 tastera: Enter, taster za interpunkciju i ?123 taster. Druga zona je najveća i podeljena je na 4 dela tj. tastera. Ona počinje od prve zone i završava se plavom graničnom linijom. Kao što je rečeno u delu 5.1 to su tasteri koji se nalaze na idealnoj poziciji za desni palac ali koji se dodiruju jagodicama zbog čega su najveći. Treća zona je ona okarakterisana kao najmanje prijatna ali koja se dodiruje vrhom prsta, zbog čega je manjih dimenzija od druge, takođe podeljena na 4 tastera. Oblik tastera diktiraju ivice ekrana i razdelne linije. Između druge i treće zone je crni krug koji ima funkciju razmaka tj. Space taster. Ovaj taster ima i funkciju Shift koja se poziva prevlačenjem prsta iz kruga prema gore. Dva takva poteza aktiviraju Caps Lock funkciju. Prevlačenjem prsta prema dole deaktivira Shift/Caps Lock funkcija. Svaki taster (osim tastera Enter, ?123 i Space) poseduje 4 funkcije. Jedna se dobija direktnim pritiskom na taster dok se ostale 3 pozivaju prevlačenjem prsta u određenom pravcu. Dugim pritiskom se pozivaju sekundarne funkcije tastera. Backspace funckija se nalazi na gornjem desnom tasteru.

Razlog zbog kojeg su za tastere sa karakterima izabrane zelena i crvena zona (slika 1) umesto zelene i žute kao logičnijeg izbora je preciznost na uštrb udobnosti. Sa ovakvom idejom podele ekrana i funkcionisanja tastature, tasteri u gornjem levom uglu (lejaut za dešnjake) ne bi bili dovoljno veliki, jer za razliku od tastera iste veličine u crvenoj zoni, oni bi se dodirivali većom površinom prsta (palca). Tasteri koji bi u tom slučaju bili ispod plave granične linije, nalazili bi se na nezgodnom mestu koje bi bilo između crvene i zelene zone što bi značilo da nijedan taster nije na idealnoj poziciji. Sve ovo bi rezultovalo manjom preciznošću unosa. Sa predlogom koji je dat, 4 tastera su na idealnim pozicijama, dok su ostala 4 na pozicijama gde se može računati na najveću preciznost dodira.



Slika 2. X2 lejaut za desnoruke i levoruke korisnike

Svakom tasteru se može dodeliti jedan karakter (funkcija) koji se dobija pritiskom na taster i on je na lejautu označen belom bojom sa crnim ivicama. Ostala 3 karaktera (funkcije) se dobijaju prevlačenjem prsta iz sredine tastera u smeru granične linije uz koju stoje odnosno ka centralnom krugu. Karakteri su u istoj boji kao granična linija odnosno krug prema kojima prst mora da se prevuče. Na slici 3 se mogu videti 4 scenarija za unos pojedinih slova. Scenario označen brojem 1 je način unosa slova “U”, dok je scenario broj 2 unos slova “R”. Kružić na liniji označava početak poteza, dok je strelica smer i mesto završetka poteza.



Slika 3. Scenariji korišćenja X2 rešenja

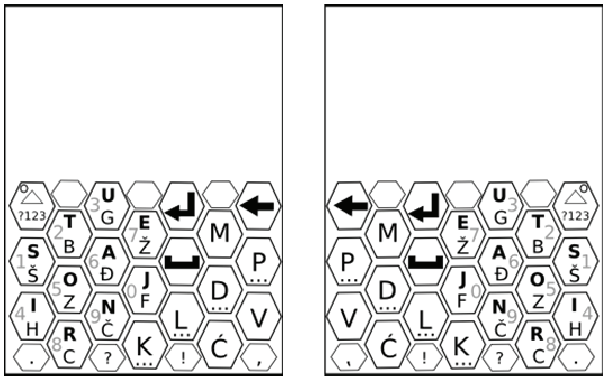
5.2. HexaPen dizajn

HexaPen poseduje šestougaone tastere i predstavlja pandan Compact QWERTY tastaturama (slika 4). U cilju povećanja veličine tastera njihov broj je redukovan što je izvršeno na osnovu frekventne karakteristike pojave slova u srpskom jeziku.

Način grupisanja tastera je izvršen na sledeći način:

- Slova Lj, Nj i Dž su izostavljena jer predstavljaju kombinaciju dva slova u latiničnom pismu i za njih nema posebnih tastera.
- Slova latinice X, Y, Q, W su izostavljena jer se ne koriste u srpskom jeziku. Ona su dostupna kao sekundarne funkcije kod tastera sa slovima K, L, D, P i unose se dugim pritiskom na taster.
- Najčešće korišćena slova se nalaze na levoj strani tastature i na tasterima su označena zatamljenim slovima. Njihov unos se vrši jednim pritiskom na taster. Najmanje korišćena slova se nalaze kao sekundarna i unose se dvostrukim pritiskom na taster. Ima ih ukupno 10 tako da se kao sekundarne funkcije tastera unose cifre.
- Tasteri sa desne strane su oni koji poseduju slova čije su frekfencije pojave između često i retko korišćenih slova.
- Taster u gonjem levom uglu je kombinovani Shift – ?123 taster. Pritiskom na njega se poziva Shift funkcija, dugim ili dvostrukim pritiskom Caps Lock funkcija, dok se dregovanjem prsta u nekom smeru dobija numeričko-znakovni lejaut. U meniju za podešavanja će korisniku biti ostavljena mogućnost da sam podesi kako će se ove 3 funkcije pozivati.

- Pri sredini lejauta se nalaze taster za unos novog reda odnosno Enter i taster Space (jedan ispod drugog). U desnom gornjem uglu je taster Backspace. Korisniku će biti ostavljena mogućnost za raspodelu funkcija ova 3 tastera.
- U donjem delu ekrana su 4 sitnija tastera i to za tačku, upitnik, uzvičnik i zarez. To su najčešće korišćeni znaci interpunkcije.
- Za 3 tastera pri vrhu, funkcija nije unapred predodređena. Korisniku će biti ostavljena mogućnost da im dodeli funkcije poput: Cut, Copy, Paste i/ili ispis nekog znaka interpunkcije itd.



Slika 4. HeksaPen lejaut za desnoruke i levoruke korisnike

Deset alfabetnih tastera u levom delu tastature (slika 4, lejaut za desnoruke korisnike), nalaze se u žuto-zelenoj zoni (slika 1). Ostalih sedam se nalaze u zeleno-crvenoj zoni. Pri rasporedu alfabeta postojala je nedoumica da li kao sekundarne alfabete postaviti one srednjih vrednosti frekfencije pojave ili one sa najnižim vrednostima. Kao sekundarni, bili bi na pozicijama koji više odgovaraju ergonomskim zahtevima, a i putanje prsta bi se smanjile budući da bi se svi najčešće korišćeni alfabeti nalazili na malom prostoru. Međutim, povećala bi se potreba da korisnik neki od tih tastera pritiska po dva puta što u slučaju ekrana osetljivih na dodir nije poželjno. Zato je odluka bila da se alfabeti sa najmanjim vrednostima pojave postave kao sekundarni. Ovako se dobija tastatura sa sedamnaest direktno dostupnih karaktera koji su uz to i najčešće korišćeni tako da se upotrebljivost tastature podiže na viši nivo.

Šestougaoni tasteri su izabrani iz više razloga. Prvi je oblik i način uklapanja tastera. Pošto su raspoređeni poput saća, tasteri poseduju veću širinu u centralnom delu nego što bi imali pravougaoni tasteri, a takođe je postignuta veća visina jer redovi zalaze jedan u drugi. U poređenju sa tasterima Compact QWERTY i MessagEase rešenja, HexaPen ima tastere manje površine. U odnosu na Compact QWERTY rešenje (nadalje CQ), prednost HexaPen-a (nadelje HX) je što ima sedamnaest u odnosu na CQ-ovih četrnaest direktno dostupnih karaktera. Pored toga, HX poseduje optimizovan raspored alfabeta, dok kod CQ to nije slučaj i primarni alfabet (onaj koji se dobija jednim pritiskom na taster) nije određen frekfentnom karakteristikom nego je predodređen QWERTY raspodelom. U odnosu na MessagEase rešenje (nadalje ME), prednost HX-a je što ima skoro duplo više direktno dostupnih alfabeta (sedamnaest nasuprot samo devet). ME kao i HX poseduje optimizovan raspored alfabeta. U cilju smanjivanja putanja koje prelazi prst pri unosu, centralni taster kod ME poseduje osam sekundarnih karaktera koji se dobijaju prevlačenjem. Ergonomija je takođe slična pošto se devet alfabetnih tastera ME rešenja nalaze u žuto-zelenoj zoni (slika 1). Prema tome, oba rešenja su poprilično izjednačenja, a presudu može doneti afinitet korisnika, tj. da li preferira kraće putanje prsta sa manje direktno dostupnih alfabeta uz korišćenje prevlačenja ili duže putanje kretanja sa više direktno dostupnih alfabeta i povremenu upotrebu dvostrukog pritiska tastera.

6. ZAKLJUČAK

Kao sledeću fazu, autor rada ima implementaciju predloženih lejauta kako bi isti mogli biti testirani od strane što većeg broja korisnika. Sledi dodatna optimizacija prvenstveno u pogledu rasporeda karaktera. Dalji razvoj rešenja će biti planiran u zavisnosti od povratnih informacija korisnika, njihovih predloga i sugestija.

7. LITERATURA

[1] Wikipedia, Personal Digital Assistant (PDA), 2010, Wikipedia, http://en.wikipedia.org/wiki/PDA

[2] Wikipedia, Tablet Personal Computer (Tablet PC), 2010, Wikipedia, http://en.wikipedia.org/wiki/Tablet_pc

[3] Wikipedia, CRT – Catode Ray Tube, 2010, Wikipedia, http://en.wikipedia.org/wiki/Cathode_ray_tube

[4] Vladimir Kovačević i Branislav Atlagić, Logičko projektovanje računarskih sistema, 2007, FTN Izdavalaštvo

[5] Wikipedia, Plasma display, 2010, Wikipedia, http://en.wikipedia.org/wiki/Plasma_display

[6] Wikipedia, Organic light-emitting diode, 2010, Wikipedia, http://en.wikipedia.org/wiki/Organic_light-emitting_diode

[7] Wikipedia, E ink, 2010, Wikipedia, http://en.wikipedia.org/wiki/E-ink

[8] InfoSync, Super LCD screen technology explained, 2010, InfoSync, http://www.infosyncworld.com/news/n/11153.html

[9] Wikipedia, Active-Matrix OLED (AMOLED), 2010, Wikipedia, http://en.wikipedia.org/wiki/Active-matrix_OLED

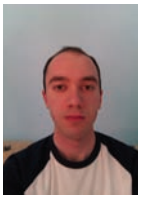
[10] OLED-Info.com, Super AMOLED, 2010, OLED-Info.com, http://www.oled-info.com/super-amoled

[11] Vojislav Mihajlović, Ekrani koji se pipkaju, 2011, Svet kompjutera

[12] Wikipedia, Smartphone, 2011, Wikipedia, http://en.wikipedia.org/wiki/Smartphone

[13] Jochen Rick, Performance optimization od virtual keyboards for stroke-based text entry on a touch-based tabletop

Kratka biografija:



Srđan Farkaš rođen je u Novom Sadu 1986.god. Na Fakultet tehničkih nauka se upisao 2005. godine na odseku za Računarstvo i automatiku. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i Automatika odbranio je 2010.god

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.397.13:004.932

DIGITALNA ZEMALJSKA TELEVIZIJA

DIGITAL TERRESTRIAL TELEVISION

Darko Lauš, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu je predstavljena strategija prelaska sa analognog na digitalno zemaljsko emitovanje TV signala u Evropi. Opisane su prednosti koje pruža digitalna televizija, kao i raspodela frekvencijskih opsega namenjenih za digitalnu radiodifuziju.*

Abstract – *The paper presents the strategy of transition from analogue to digital terrestrial broadcasting TV signals in Europe. The benefits offered by digital television are described as well as distribution of frequency bands allocated for digital broadcasting.*

Ključne reči: *digitalna tranzicija, DVB, MPEG, frekvencijski opsezi*

1. UVOD

Dok je 50-ih godina prošlog veka radiodifuzija vršena isključivo zemaljskim putem, danas gledaoci televizijskim sadržajima mogu pristupati i putem satelitskih, kablovskih ili IPTV platformi zasnovanih na digitalnoj tehnologiji ili prenosu putem optičkih vlakana. Nove platforme su donele i nove mogućnosti kao što je veliki izbor televizijskih programa, ali i znatno bolji kvalitet prenosa u odnosu na terestričku platformu. Sve ovo je dovelo do gubitka nadmoći terestričkog radiodifuznog sistema, jer su se gledaoci sve više okretali ka drugim vidovima prenosa televizije kao primarnim. Jedan od najvećih nedostataka terestričke platforme bio je nedovoljan broj atraktivnih kanala koji se mogu omogućiti zemaljskim prenosom, što je povezano i sa nedostatkom broja slobodnih radio frekvencija, jer se kod analogne televizije po jednom tv kanalu može preneti samo jedan kompletan tv program. Analogna televizija je pokazala i ozbiljne nedostatke kada je u pitanju mobilni ili portabl prijem [4].

Sve ove činjenice uslovile su razvoj novog digitalnog sistema za terestričku radiodifuziju. Prelazak na digitalno emitovanje, pored racionalnijeg iskorišćenja RF kanala i boljeg kvaliteta signala, doneo je brojne prednosti koje će biti izložene u nastavku. U Evropi se od 1993. godine standardizacijom u oblasti digitalne radiodifuzije bavi DVB projekat, međunarodni konzorcijum sa više od 270 članova. Pored usvajanja digitalnih standarda za prenos televizijskog signala putem satelita (DVB-S) i kablovskih sistema (DVB-C), DVB projekat je razmatrao i emitovanje signala zemaljskim vezama u slobodnom prostoru, osvrćući se naročito na problem višestruke propagacije. Kao rezultat toga razvijen je složen način emitovanja digitalne zemaljske televizije putem DVB-T standarda (EN 300.744) usvojenog 1996.godine. Razvoj tehnologija koje čine osnovu uređaja i softverskih rešenja u televizijskom prenosu kao i savremeni zahtevi doveli su do nastanka složenijih, ali znatno efikasnijih i prilagodnijih rešenja u okviru DVB projekta. Tako je nastala i druga generacija DVB-T standarda (EN 302.755) koja je usvojena 2008.godine [2].

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Vojin Šenk, red.prof.

2. PRELAZAK SA ANALOGNE NA DIGITALNU TERESTRIČKU RADIODIFUZIJU.

2.1. Prelazak na digitalno emitovanje u Evropi

Analogni prenos signala često je izložen pojavi tzv. „senki“ u slici koje nastaju zbog mnogo različitih uzroka. Jedan od njih je svakako pojava refleksija signala od zgrade i prepreke, koje se dodaju glavnom signalu na prijemu ali imaju različita vremena kašnjenja u zavisnosti od pređenih putanja. Pored refleksije javlja se i efekat tzv. međukanalne interferencije, koji je naročito izražen tokom povećanog atmosferskog pritiska [1]. Analogni TV sistemi (PAL, SECAM) su veoma osetljivi na inter–ferenciju od drugih analognih TV signala i zahtevaju visoke međukanalne zaštitne odnose.

Prelaskom na digitalno emitovanje, koje je znatno manje osetljivo na šum i interferenciju, pruža se mogućnost rada sistema na niskom ERP nivou i na taj način se redukuje interferencija u odnosu na postojeće analogne servise. U toku prenosa na digitalan signal se takođe superponira šum i na ulaz prijemnika stižu i zakasnele komponente refleksije, ali digitalni prijemnik je u stanju da odvoji korisni signal od šuma i „senki“ i u potpunosti ga rekonstruiše. Dekodovanjem signala u formatu komponenti, digitalan signal će izbeći gubitak jasnoće koji je postojao kod analognog (PAL/NTSC) formata kodovanja. Rezolucija primljene slike neće varirati od protoka, ali će se u slici pojaviti artefakti ukoliko je protok isuviše mali za detalje koje program sadrži.

Proces prelaska sa analognog na digitalno emitovanje je vrlo složen i zahtevan korak, koji u mnogome zavisi od konkretnih faktora i preduslova svake zemlje. U zavisnosti od strukture postojeće mreže i teritorijalne pokrivenosti signalom, primenjuju se različite strategije za uvođenje digitalnih servisa. Kako bi se omogućio nesmetan proces tranzicije i maksimalono iskoristile sve prednosti digitalne televizije, zemlje u regionu moraju usaglasiti svoje akcione planove i preći na digitalno emitovanje na koordinisan način. Stoga je na regionalnoj konferenciji o radio komunikacijama, Međunarodne unije za telekomunikacije (ITU), održanoj u maju i junu 2006. godine u Ženevi (RRC06), predviđeno da uvođenje digitalne i potpuno gašenje analogne televizije u Evropi bude završeno najkasnije do 17. juna 2015. godine. Veliki

broj zemalja, među kojima je i Republika Srbija, je kao rok za prelazak na potpuno digitalno emitovanje odredio 2012. godinu.

Tabela 1. Rok za prelazak na digitalno emitovanje zemalja u Evropi [5]

Zemlja	Digitalna TV pokrenuta	Period prestanka analognog emitovanja
Velika Britanija	1998	2012
Švedska	1999	Završen
Španija	2000/2005	2010
Finska	2001	Završen
Švajcarska	2001	Završen
Nemačka	2002	Završen
Belgija	2002	Završen
Holandija	2003	Završen
Andora	2004	Završen
Italija	2004	2012
Francuska	2005	2011
Češka	2005	2011
Luksemburg	2006	Završen
Danska	2006	2009
Estonija	2006	2010
Austrija	2006	2010
Slovenija	2006	2011
Norveška	2007	2009
Litvanija	2008	2012
Mađarska	2008	2011
Portugalija	2009	2012
Hrvatska	2009	2011
Letonija	2009	2011
Poljska	2009	2013
Bugarska	2009	2012
Grčka	2010	2012
Irska	2010	2012
Slovačka	2010	2012
Rumunija	2010	2012
Kipar	2010	2012

2.2. Prednosti koje pruža digitalna televizija

Uprkos visokom kvalitetu slike koja nastaje u studiju, na strani gledaoca ona je često dosta slabijeg kvaliteta. Razlog tome jesu brojne smetnje koje se javljaju u procesu prenosa slike. Digitalna televizija u velikoj meri otklanja ove nedostatke i pruža znatno bolji kvalitet reprodukovane slike. Digitalna televizija omogućava višestruke pogodnosti kako gledaocima, tako i TV stanicama (emiterima). Pogodnosti sa stanovišta gledalaca ogledaju se kroz:

- Postojan kvalitet prijemnog signala
  Za razliku od analognog signala čiji kvalitet opada s udaljavanjem od predajnika, kvalitet digitalnog signala je postojan unutar zone prijema, a na njenim ivicama dolazi do naglog prekida prijema
- Relativno mala greška prijema
  Kod digitalne televizije slika ili postoji u zoni pokrivanja i tada je kvalitetna ili ne postoji. Izuzetak su ivice zone gde zbog mogućih fluktuacija u prijemu slika ili postoji ili je zamrznuta
- Veća imunost prijemnog signala na smetnje
  Kao što je već rečeno analogni TV signali su izuzetno osetljivi kada je u pitanju interferencija od drugih kanala, industrijske smetnje itd. Kod digitalne televizije ove smetnje su znatno manje izražene.
- Reprodukovana slika je bez refleksija
  Zahvaljujući primeni OFDM sistema moulacije i umetanju zaštitnog intervala tokom kog je prijemnik neaktivan, pa se na taj način ignorišu naknadno pristigli reflektovani signali, eliminisana je pojava refleksija u slici.
- Visok kvalitet reprodukovanog zvuka i slike
  Kvalitet zvuka koji se postiže digitalnim prenosom TV signala je na nivou CD kvaliteta i može biti stereofonski ili sa okružujućom reprodukcijom. Što se tiče kvaliteta slike on je gotovo na nivou studijskog, zahvaljujući velikoj otpornosti digitalnog signala na smetnje.
- Širi format reprodukovane slike
  Pored standardnog formata slike 4:3 (koji je aktuelan kod analogne televizije) digitalni sistemi omogućavaju i format 16:9. Ovaj znatno širi format posebno je pogodan kod određenih programskih sadržaja kao što su npr. prenosi sportskih događaja.
- Niži nivo prijemnog signala
  U odnosu na analognu televiziju, da bi se ostvario prijem digitalne televizije potreban je niži nivo prijemnog signala. Ovo zapravo znači da bi se ostvarilo isto teritorijalno pokrivanje potrebna je manja snaga u slučaju digitalnih sistema.
- Veći broj programa po jednom TV kanalu
  Rečeno je već da se u slučaju analogne televizije po jednom TV kanalu može preneti samo jedan TV program. Broj programa koji se može preneti po jednom TV kanalu u slučaju digitalne televizije je znatno veći. Tako u slučaju DVB-T standardne rezolucije (SDTV), uz primenu MPEG-2 kompresionog standarda, uobičajen broj programa je 4-6, u zavisnosti od definisanog protoka. Za DVB-T uz primenu MPEG-4.10 kompresije ovaj broj je dvostruko veći (8-10). Prelazak na sistem druge generacije (DVB-T2) broj programa se uvećava za još 30%. Kad je u pitanju televizija visoke rezolucije (HDTV) broj programa koji se može preneti u DVB-T sistemu (uz primenu MPEG-4.10) je realno 2-3. (u DVB-T2 i do 4)
- Nehijerarhijski/hijerarhijski mod rada
  DVB-T nudi mogućnost da se isti programski sadržaj emituje kao signal nižeg prioriteta (LP stream), ali višeg kvaliteta (koristi se 64-QAM ili 16-QAM modulacija) i kao signal višeg prioriteta (HP stream), ali slabijeg kvaliteta (koristi se QPSK modulacija isključivo), koji je znatno robustniji na smetnje. Niz višeg prioriteta se prenosi kao tzv. "ugrađeni" QPSK u 16-QAM ili 64-QAM.Kada su uslovi za prijem dobri prijemnik će primati oba niza, dok će u suprotnom primati samo HP niz. Na ovaj način postignuto je da sigurnost prijema kvalitetne slike bude viša, a prenos slike višeg prioriteta i dodatnih servisa bude brži.
- Dodatni sadržaji
  Digitalna televizija donosi sa sobom brojne dodatne mogućnosti kao što su superteletekst, elektronski programski vodič (EPG), interaktivni multimedijalni sadržaji, prevodi... Elektronski programski vodič omogućava pretraživanje čitavog televizijskog sadržaja koji smo primili, slično kao na internet portalima. Gledalac može između raznih sadržaja odabranih po nekom ključu izabrati onaj koji mu najviše odgovara, bez prethodnog pretraživanja svih raspoloživih TV programa. Takođe, DTV nudi mogućnost da prevod bude dodatna aplikacija, a ne

deo same slike. Potpuno nova usluga jeste interaktivni multimedijalni program.

- Nisu neophodna velika ulaganja gledalaca
  Korisnička oprema mora ne samo da primi televizijski signal, nego i da ga na odgovarajući način obradi kako bi se sadržaj prikazao na ekranu ili reprodukovao na zvučnicima. Naime, preko 95% TV prijemnika su klasični (analogni) i bilo bi neopravdano da korisnik nabavlja poseban TV prijemnik koji bi služio samo za prijem DVB-T. Stoga su proizvođači TV opreme na tržište izneli eksterni dodatak tzv. *set top box* (STB) koji se priključuje između postojećeg TV prijemnika i antene. STB je zapravo prijemnik (receiver) koji pretvara digitalni signal u oblik pogodan za prikazivanje na postojećim TV prijemnicima. Za prijem DVB-T nisu potrebne posebne prijemne antena, ukoliko zadovoljavaju uslove prijema analognog signala na bliskom TV kanalu sa iste lokacije. Dakle, postojeće antene i televizori biće neko vreme prihvatljivi za prijem signala i prikaz sadržaja uz dodatak STB. Alternativa (nešto skuplja) ovome jeste kupovina novog TV prijemnika koji ima integrisan tj. u sebi sadržan DVB prijemnik (IDTV).

Posmatrano sa staništa TV stanica (tj. emitera) prednosti digitalne televizije su:

- Veća oblast pokrivanja
  Pri jednakoj izračenoj efektivnoj snazi digitalnim emitovanjem postiže se veća oblast pokrivanja nego u slučaju analognog signala, tj. ista zona pokrivanja se ostvaruje sa manjom efektivnom izračenom snagom (ERP), a samim tim i manjom izlaznom snagom digitalnog predajnika.
- Mogućnost rekonfiguracije i adaptacije sistema
  DVB-T standard , a naročito DVB-T2, nudi široku lepezu parametara čime se sistem uspešno prilagođava trenutnim i na taj način se ostvaruje optimalan prijem. Izborom ovih parametara moguće je definisati mod rada sistema, tip mreže, kvalitet signala, njegovu robusnost na smetnje itd.
- Emitovanje više TV programa po jednom kanalu
  Kompresija kod digitalnog prenosa signala uklanja one informacije koje čovek zbog nesavršenosti svojih čula ne može da registruje. Nije neophodno prenositi sve delove slike i zvuka u kontinuitetu, kao što se to radi kod analogne televizije, već se prenose samo oni elementi koji su se u međuvremenu promenili. Ovo omogućuje da se koristeći isti frekvencijski opseg emituje više programa, čime se poboljšava i spektralna efikasnost.
- Emitovanje dodatnih sadržaja
  Emiterima TV programa se pruža mogućnost emitovanja i dodatnih sadržaja kao što su vremenska prognoza, telešoping, video igre, "video-on-demand", internet servis, čitanje i slanje elektronske pošte, klađenje, glasanje i razne druge interaktivne usluge, čime se znatno unapređuje njihova ponuda.
- Niža cena eksploatacije emisionih uređaja
  Značajne uštede se npr. ostvaruju u pogledu korišćenja električne energije na predajnicima, pa samim tim su i manji troškovi prenosa signala [3].

2.3. Frekvencijski opsezi namenjeni za digitalnu radiodifuziju

Planiranje frekvencija koje će se koristiti za digitalnu radiodifuziju izvršeno je prema sporazumu usvojenom na Regionalnoj konferenciji o radio komunikacijama Međunarodne unije za telekomunikacije (RRC06) održanoj u Ženevi 2006. godine. Tada je usvojen GE06 međunarodni plan raspodele radiofrekvencija za potrebe prenosa digitalnog zemaljskog radio i televizijskog programa. Prema ovom planu predviđen je prelazak na digitalnu zemaljsku radiodifuziju u VHF opsegu III i UHF opsezima IV i V.

Dogovorom je utvrđeno da 17. jun 2015. godine bude datum nakon koga će GE06 biti potpuno raspoloživ, odn. datum kada se završava period tranzicije. Posle ovog datuma prestaje obaveza usaglašavanja planova analognog emitovanja programa sa zemljama koje vrše analogno emitovanje u regionu. Ovo praktično znači da zemlje koje nisu u mogućnosti da pređu na digitalno emitovanje programa mogu nastaviti rad u analognom sistemu, ali TV kanali na kojima će se emitovati analogna televizija neće biti međunarodno štićeni odgovarajućim planovima. Budući da je digitalna televizija manje osetljiva na analogne smetnje, nego u obrnutom slučaju, to znači da će svaka zemlja morati da počne digitalno emitovanje do tog perioda, jer će se u suprotnom suočiti sa smetnjama na prijemu i manjom zonom kvalitetnog pokrivanja.

Za emitovanje analogne zemaljske televizije širom sveta korišćeni su opsezi I, III, IV i V, odn. grupe frekvencija koje pripadaju tzv. VHF i UHF opsezima. Tako je npr. u V.Britaniji u početku emitovanje TV signala (BBC) vršeno čak u srednje-talasnom području (MF). Usvajanjem monohromatskog sistema sa 405 linija prešlo se u opseg I, međutim ovo je omogućavalo korišćenje svega pet radio frekvencijskih kanala za televizijsko emitovanje. Sa razvojem drugog Britanskog servisa (poznatog kao ITV) postalo je jasno da je opseg III mnogo bolji opseg za rad jer bi omogućio potpun razvoj dve nacionalne mreže u Britaniji. Konačno, prelaskom na PAL sistem (625 linija) 1985.godine migriralo se u opsege IV i V (UHF područje) [1].

Pregled frekvencija u kojima se prema planu GE06 predviđa digitalno emitovanje dat je u tabeli 2.
Iz tabele 2 vidimo da je za potrebe TV u VHF području (III opseg) nominalna širina kanala 7 MHz dok u UHF području (IV i V opseg) ona iznosi 8 MHz. Sporazum GE06 regulisao je raspodelu frekvencija za potrebe digitalnih radio i televizijskih servisa (T-DAB i DVB-T standardi), pa je tako VHF opseg (174-230 MHz) namenjen i jednom i drugom servisu, dok je UHF opseg (470-862 MHz) rezervisan za digitalno televizijsko emitovanje (DVB-T). Što se tiče primene DVB-T2 standarda, potrebno je reći da on zahteva iste uslove u pogledu opsega kanala koji multipleks zauzima.
Iako je korišćen za potrebe analogne televizije opseg I se napušta kada je u pitanju digitalno emitovanje Naime, ovaj opseg ima odlične karakteristike propagacije, ali je sklon problemima međukanalne interferencije i budući da je isuviše uzak za emitovanje mnogo kanala, neće biti iskorišćen za digitalnu televiziju, već će biti upotrebljen u druge svrhe.

Tabela 2. *Pregled frekvencija namenjenih za digitalno emitovanje [2].*

Опсег	Граница [MHz]	Број канала у опсегу	Редни број првог канала	Редни број последњег канала	Ширина канала у [MHz]	Намена канала
III (VHF)	174 - 230	8	5.	12.	7	DVB-T&T-DAB
IV i V (UHF)	470 - 862	49	21.	69.	8	DVB-T
IV	470-582	14	21.	34	8	
V	582-862	35	35.	69.	8	

3. ZAKLJUČAK

Pogodnosti koje donosi prelazak na digitalnu radiodifuziju su brojne. Građanima, sa jedne strane, digitalizacija će omogućiti bolji kvalitet zvuka i slike, raznovrsniji sadržaj, više radio i televizijskih programa, nove usluge za osobe sa invaliditetom i za starije osobe, unapređene dodatne usluge, portabl i mobilni prijem programa kao i konvergenciju usluga. Pružaocima usluga, sa druge strane, digitalizacija će dati mogućnost prilagođavanja sadržaja prema potrebama različitih ciljnih grupa, interaktivnost kao i mogućnost pružanja usluga na zahtev, niže troškove emitovanja. Na kraju, digitalizacija će državi omogućiti efikasnije korišćenje radio-frekvencijskog spektra, upotrebu oslobođenog dela spektra za nove usluge, promociju razvoja tehnologije i nova radna mesta, unapređenu konkurenciju. Takođe, prelazak na digitalno emitovanje uvešće red u etru budući da će mesto u multipleksima biti omogućeno samo emiterima sa važećim dozvolama za emitovanje programa.

Dakle, projekat digitalizacije je proces u kome aktivno moraju da učestvuju država, pružaoci usluga (TV stanice) i građani. Potrebna su velika finansijska ulaganja, koja država treba da obezbedi, kako u emisionu opremu tako i za subvencionisanje pretlatničke baze, odn. nabavke STB uređaja budući da država ima obavezu da svim građanima obezbedi nesmetano informisanje i prijem televizijskog programa. Od digitalne televizije se očekuje da pruži atraktivnu ponudu kako bi se što veći broj gledalaca vratio ovoj vrsti televizije kao primarnoj, a pri tome je neophodno poboljšati i samu naplatu pretplate. Takođe, potrebno je voditi i računa o nacionalnom i kulturološkom entitetu stanovništva i obezbediti više adekvatnih programa za manjinske zajednice.

Neophodno je i da digitalna televizija bude dovoljno promovisana, odn. da građani budu informisani o procesu digitalizacije i njenom značaju uopšte čime se podiže njihova tehnička svest na jedan viši nivo koji odgovara savremenom društvu 21. veka.

4. LITERATURA

[1] S. O'Leary , *Understanding Digital Terrestrial Broadcasting*, Artech House, London, 2000

[2] *Strategija prelaska sa analognog na digitalno emitovanje radio i televizijskog programa u Srbiji* http://www.digitalnaagenda.gov.rs/aktivnosti/digitalizacija-televizije/akcioni-plan/

[3] D. Marković, *Terestrička digitalna televizija*

[4] S. Maksimović-Moićević, *Analiza primene jednofrekvencijskih mreža u digitalnoj distribuciji TV signala*

[5] DigiTAG, *Understanding DVB-T2; Key technical, business, & regulatory implications,* DigiTAG, Ženeva, 2009

Kratka biografija:



Darko Lauš rođen je u Vrbasu 1986. god. Završio je gimnaziju *„J.J.Zmaj"* u N.Sadu i 2005. godine upisao prvu godinu studija na Fakultetu tehničkih nauka na odseku za energetiku, elektroniku i telekomunikacije. Bachelor rad iz oblasti telekomunikacionih mreža odbranio je 2010.god.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9

APLIKACIJA ZA ADMINISTRIRANJE SECURITY PODSISTEMA DMS SOFTVERA

APPLICATION FOR ADMINISTRATION OF DMS SOFTWARE SECURITY SUBSYSTEM

Aleksandra Krneta, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu opisana je primena Google Web Toolkit i OpenLDAP tehnologija na primeru dinamičke web aplikacije za administriranje security podsistema DMS softvera.*

Abstract – *This paper describes application of Google Web Toolkit and OpenLDAP technologies to implement a dynamic web system for security management in DMS software.*

Ključne reči: *DMS, GWT , OpenLDAP*

1. UVOD

U ovom radu je opisana primena ***Google Web Toolkit*** i ***OpenLDAP*** tehnologija na primeru aplikacije za administraciju ***security*** podsistema DMS softvera. U radu su opisane osnovne karaktiristike DMS softvera s posebnim osvrtom na na ***Security Management*** podsistem DMC aplikacije. Napravljena je kratka analiza korišćenih tehnologija i na kraju osvrt na postignute rezultate i mogućnosti daljeg usavršavanja realizovane ***web*** orijentisane aplikacije.

2. OPIS REŠAVANOG PROBLEMA

2.1. DMS softver

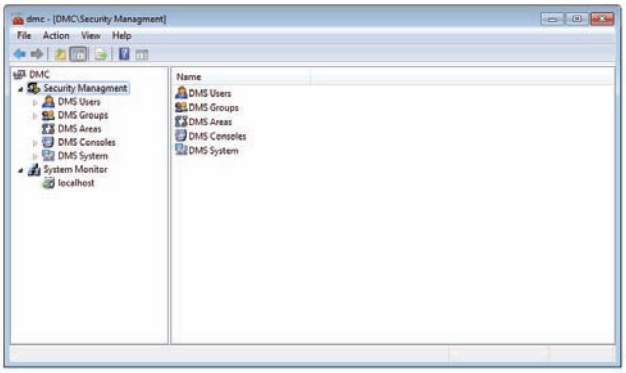
Kako bismo bolje razumeli rešavani problem potrebno je prvo upoznati DMS softver i njegovu arhitekturu. ***Distribution Management System*** (DMS) je savremen softverski sistem za upravljanje distribucionim sistemima, odnosno softver za upravljanje i optimizaciju potrošnje električne energije. DMS obezbeđuje centralizovano i daljinsko upravljanje elektrodistributivnom mrežom dispečerima zaposlenim u elektrodistribuciji u cilju što boljeg funkcionisanja mreže. Distribucija električne energije je pretposlednja faza u isporuci električne energije krajnjim potrošačima [1]. DMS je sistem sa troslojnom arhitekturom. Podsistemi koji ga čine su u velikoj meri nezavisni:

- Podsistem za interakciju sa korisnikom implementira funkcije korisničkog interfejsa.
- Podsistemi za implementaciju osnovnih funkcija sistema (implementiraju tzv. „poslovnu logiku“). Ovaj sloj može se dalje podeliti na sloj sistemskih servisa i aplikativni sloj.
- Podsistem za rukovanje podacima, pri čemu se pre svega misli na fizički smeštaj podataka (ovo je, zapravo, sistem za upravljanje bazama podataka).

2.2. DMS *Management Console*

Za administraciju i monitoring DMS softvera koristi se DMS ***Management Console*** (DMC). DMC je klijentska ***desktop*** aplikacija razvijena uz oslonac na ***Microsoft Management Console*** (MMC). Kao takva vezana je za ***Microsoft*** operativne sisteme što je čini neportabilnom. DMC konzola ima dve sekcije: ***Security Management*** i ***System Monitor***.

DMC konzola usko je vezana za ***User Server*** od koga dobija sve informacije potrebne za konstruisanje interfejsa i podataka ***security*** podsistema, te je stoga njeno dodatno ograničenje da instalacija i upotreba ove aplikacije može da se vrši isključivo na mašini na kojoj je podignut ***User Server***.



Slika 2.1 *Korisnički interfejs DMC aplikacije*

2.2.1. *Security Management*

Security Management bavi se administracijom resursa DMS softvera. U ovom radu diskutovane su mogućnosti DMC konzole na 2.74.1 verziji DMS softvera. ***Security Management*** na pomenutoj verziji softvera pruža sledeće usluge:

- Administracija korisnika,
- Administriranje organizacionih jedinica,
- Administracija korisničkih grupa i njihovih dozvola,
- Administracija zona na mreži
- Administracija konzola,
- Upravljanje vlasništvom sistema.

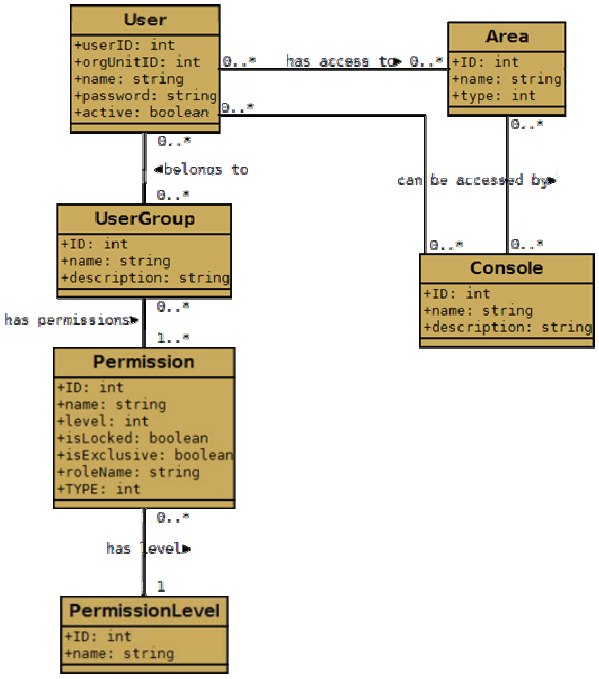
Administracija podrazumeva dodavanje, brisanje i izmenu gore navedenih resursa. Za pristup podacima DMC konzola obraća se ***User Server***-u preko USI i USCI

NAPOMENA: Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Hajduković, red.prof.

biblioteka. USI (***User Server Interface***) pruža neke od usluga potrebnih za administriranja korisnika i korisničkih grupa dok USCI (***User Server Client Interface***) pruža preostale usluge za administriranje. Upotreba ovih biblioteka ograničava implementaciju ***Security Management*** podsistema utoliko što serverski deo aplikacije mora biti instaliran na serverskoj mašini na kojoj je podignut i ***User Server***.

2.3. Šema DMS sigurnosnih podataka

Šema DMS sigurnosnih podataka se čuva u ***Directory Service***-u. Na slici 2.2 šema je prikazana konceptualnim modelom podataka. ***User*** entitet predstavlja korisnika DMS softvera. Zona, tj. entitet ***Area*** predstavlja segment elektrodistribucione mreže kojoj korisnik može da pristupi preko DMS softvera. ***UserGroup*** entitet predstavlja sigurnosnu grupu korisnika. Preko sigurnosnih grupa, korisnicima su dodeljene određene privilegije. Privilegije korisnika su predstavljene entitetom ***Permission***. Poslednji entitet u okviru šeme je ***Console*** entitet, koji predstavlja računare DMS sistema.



Slika 2.2 *Konceptualni model DMS sigurnosnih podataka*

2.4. Korisnički zahtevi

Obzirom na veliki skup klijenata DMS softvera sve češće se javalja potreba za sistemom za administriranje resursa kojem bi se lako i jednostavno moglo pristupiti putem Interneta i bez potrebe za instalacijom klijentskog softvera.

Cilj ovog master rada je da se realizuje rešenje koje će zadovoljiti potrebe klijenata. Potrebno je osmisliti i realizovati dinamičku ***web*** orijentisanu aplikaciju koja će pružati usluge administriranja ***security*** podsistema DMS softvera.

Usled nedostataka vezanih za ***User Server***, potrebno je naći adekvatno rešenje za dobavljanje i skladištenje sigurnosnih podatka putem Interneta. Za tu svrhu treba istražiti mogućnosti ***Directory Service***-a (DS) kao potencijalnog skladišta podataka. Korisnički interfejs je potrebno implemetirati uz pomoć ***Google Web Toolkit*** alata.

Aplikacija treba da omogući:

- DMS ***Users***:
  - Prikaz svih korisnika,
  - Dodavanje i brisanje korisnika,
  - Izmenu osnovnih atributa korisnika (ime, lozinka, deaktiviranje naloga),
  - Dodeljivanje pripadnosti korisnika korisničkim grupama,
  - Dodeljivanje pripadnosti korisnika zonama,
  - Dodeljivanje konzola korisniku,
- DMS ***Groups***:
  - Prikaz svih grupa,
  - Dodavanje i brisanje korisničkih grupa,
  - Izmenu osnovnih atributa korisničke grupe (ime i opis),
  - Dodeljivanje korisnika korisničkoj grupi,
  - Dodeljivanje dozvola korisničkoj grupi,
- DMS ***Areas***:
  - Prikaz svih zona,
  - Dodeljivanje korisnika pripadnika zone,
  - Dodeljivanje konzola koje mogu pristupiti zoni,
- DMS ***Consoles***:
  - Prikaz svih konzola,
  - Dodavanje i brisanje konzola,
  - Izmena osnovnih atributa konzole (ime i opis),
  - Dodeljivanje korsinika konzolama,
  - Dodeljivanje zona kojima konzola može pristupiti.

3. OPIS KORIŠĆENIH TEHNOLOGIJA I ALATA

3.1. Direktorijumski servis

Direktorijumski servis (***Directory Service***) je softverski sistem koji skladišti i organizuje podatke u direktorijumu i pruža podršku za njihovo dobijanje [2].

Sa aspekta računarskih mreža, direktorijumski servis je deljena informaciona infrastruktura za lociranje, upravljanje, administraciju i organizovanje mrežnih resursa i objekata kao što su: fajlovi, štampači, korisnici, grupe, uređaji, telefonski brojevi itd.

Direktorijum je set logički i hijerarhijski organizovanih objekata (***entry***) organizovanih u stablo [3]. Objekti sadrže uređen skup atributa. Atribut direktorijumskog objekta ima ime (tip ili opis) i jednu ili više vrednosti.

U dizajnu direktorijuma postoje pravila koja utvrđuju kako se mrežni resursi imenuju i identifikuju. Pravila nalažu da imena moraju biti jedinstvena i nedvosmislena. U X.500 seriji standarda i LDAP (***Lightweight Directory Access Protocol***) protokolu ime objekta je „jedinistveno ime“ (DN - ***distinguished name***) i koristi se za referenciranje na kolekciju atributa.

Server čuva stablo direktorijuma i direktorijumske objekte počevši od određenog objekta. Server takođe može da sadrži reference i na druge servere, tako da zahtev za određenim direktorijumskim objektom može da se završi vraćanjem informacija o objektu ili vraćanjem reference na server koji čuva taj deo stabla direktorijuma. Nije redak slučaj da određeni serveri podržavaju ulančavanje (***chaining***), tj. proces u kome će server

kontaktirati drugi server, koji sadrži željene podatke i vratiti ih klijentu bez potrebe obraćanja klijenta drugom serveru.

3.2. LDAP

Lightweight Directory Access Protocol (LDAP) je aplikacioni protokol za pretraživanje i modifikovanje direktorijumskih servisa koji se oslanja na TCP/IP (***Transmission Control Protocol/Internet Protocol***) protokol. Trenutna verzija datog protokola je v3. Specifikacija programskog interfejsa za pristup LDAP servisima definisana je u RFC-u 4510.

Klijent otvara LDAP sesiju konektovanjem na LDAP server, koji se zove ***Directory System Agent*** (DSA). Obično DSA sluša na TCP portu 389. Po konektovanju, klijent ima mogućnost da šalje operativne zahteve serveru.

3.3. *OpenLDAP*

OpenLDAP je besplatna, ***open source*** implementacija LDAP protokola razvijena od strane ***OpenLDAP Project*** organizacije [4]. ***OpenLDAP*** projekat predstavlja zajednički trud volontera da se razvije robustan, komercijalno upotrebljiv, besplatan LDAP softver koji sadrži skup aplikacija i alata za razvoj.

OpenLDAP softver sastoji se od tri osnovne komponente:

- ***slapd*** – samostalan LDAP server koji radi u pozadinskim servisima (***standalone*** LDAP ***daemon*** (server)),
- ***libraries*** – implementacija LDAP protokola,
- ***utilities***, ***tools and sample clients*** – alati i primeri korišćenja softvera.

Istorijski gledano arhitektura ***OpenLDAP*** servera (***slapd***) obuhvata:

- ***frontend*** deo – deo koji se bavi pristupom mreži i obradom protokola,
- ***backend*** deo – deo koji se bavi striktno sa skladištenjem podataka.

Arhitektura je modularna i mnogo različitih pozadinskih delova (***backends***) je na raspolaganju za povezivanje sa drugim tehnologijama, a ne samo za tradicionalne baze podataka. Između ***fronend*** i ***backend*** dela može se ubaciti međusloj, takozvani ***overlay***. Upotreba međuslojeva omogućava jednostavno proširivanje funkcionalnosti ***sladp*** servera bez potrebe da se pišu novi ***backend*** delovi. Tako se nove funkcionalnosti dodaju kao kompaktni moduli koji se na jednostavan način mogu analizirati i održavati. Za potrebe ovog rada, ***OpenLDAP*** server proširen je sa ***RefInt*** (***Referential Integrity***, ***slapo-refint***) ***overlay*** slojem koji je bio potreban kako bi ***OpenLDAP*** server bio svestan izmena naziva entiteta koji su referencirani u drugim entitetima.

3.4. *Ajax*

Ajax (***asynchronous JavaScript and XML***) predstavlja skup povezanih metodologija za razvoj koda klijentske strane ***web*** orijentisanih aplikacija [5]. U početku ***Ajax*** se sastojao od sledećih tehnologija:

- HTML ili XHTML i CSS za prezentaciju elemenata prikaza,
- ***Document Object Model*** (DOM) za dinamički prikaz i interakciju sa podacima,
- XML za razmenu podataka i XSLT za manipulaciju podacima,
- ***XMLHttpRequest*** objekat za asinhronu komunikaciju,
- ***JavaScript*** da objedini sve ove tehnologije.

Danas termin ***Ajax*** se koristi da obuhvati mnogo veći skup tehnologija. Upotrebom ***Ajax***-a ***web*** aplikacija može komunicirati u pozadini (***in the backround***) asinhrono s serverom, ne ometajući prikaz i ponašanje postojeće ***web*** stranice.

3.5. *Google Web Toolkit*

Google Web Toolkit je razvojno okruženje koji služi za brz i optimizovan razvoj kompleksnih ***browser*** orijentisanih aplikacija visokih performansi, bez potrebe za ekspertskim znanjem u domenu ***XMLHttpRequest***, ***JavaScript*** i nepredvidivih okolnosti koje se često vezuju za ***web browser*** [6]. Za potrebe ovog master rada korišćena je 2.3.0 verzija GWT alata.

U svojoj osnovi GWT je razvojno okruženje za ***Ajax***. GWT koristi ***Java*** programski jezik za pisanje ***web*** orijentisanih aplikacija zasnovanih na ***Ajax*** tehnologijama [7]. Osim nekoliko izvornih (***native***) biblioteka, sve je ***Java*** kod koji se može ***build***-ovati na svakoj platformi koja podržava ***Java***-u, uz prisustvo GWT ***Ant build*** alata.

GWT nije alat čija je jedina namena kreiranje korisničkog interfejsa. To je skup alata za kreiranje raznih funkcionalnosti visokih performansi na klijentskoj strani aplikacije, korišćenjem ***JavaScript***-a. Osim klijentskog koda koji se kompajlira u ***JavaScript***, u GWT-u se piše i serverska strana koda odnosno servleti s kojim klijentska strana komunicira asinhrono preko interfejsa. Za razliku od klijentskog koda, serverski kod koristi ***Java***-u tj. nije ograničen na upotrebu GWT biblioteka.

Najčešći oblik komunikacije između klijentske i serverske strane odvija se putem GWT ***remote procedure calls*** (GWT RPC) [8]. Za serverski kod koji poziva klijent koristi se termin „servis“ (***service***), a čin pozivanja koda sa klijentske strane naziva sa „pozivanje servisa“ (***invoking a service***).

4. OPIS REŠENJA PROBLEMA

Aplikacija je implementirana s visokim stepenom modularnosti pri čemu postoji izuzetno mala zavisnost između modula.

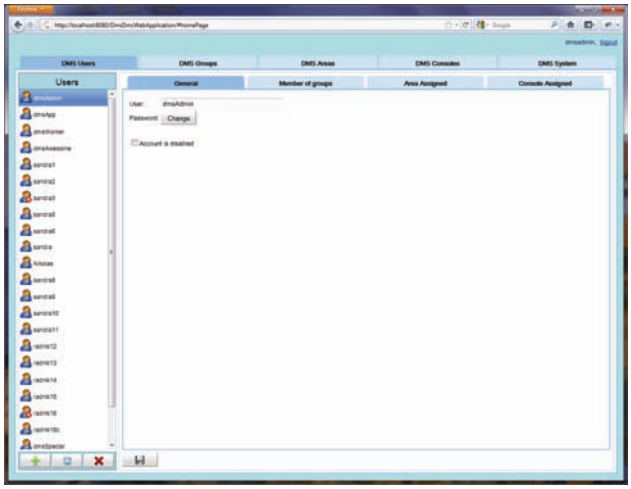
Po startovanju aplikacije u ***web browser***-u korisnik dobija „***login***“ dijalog u koji je potrebno uneti korisničko ime i lozinku. Ukoliko su parametri ispravni i konekcija s direktorijumskim servisom uspešna, korisnik dobija osnovnu stranicu na kojoj se može videti da je aplikacija organizovana u pet sekcija koje su raspoređene u kartice:

- ***DMS Users,***
- ***DMS Groups,***
- ***DMS Areas,***
- ***DMS Consoles,***
- ***DMS System***.

U daljem tekstu za podatke kao što su korisnici, grupe, zone i konzole, koristiće se termin „entiteti". Prve četiri sekcije imaju slično organizovan sadržaj, pri čemu s leve strane stoji lista entiteta date sekcije i kontrole vezane za nju, a s desne strane stoji deo sa detaljima označenog entiteta i komande za izmenu vrednosti označenog entiteta. Peta sekcija, ***DMS System***, još uvek je u fazi razvoja jer nisu ispunjeni svi uslovi za njenu implementaciju. Iznad svake liste entiteta stoji zaglavlje koje deklariše naziv grupe entita, a ispod liste nalazi se ***toolbar*** koji nudi tri funkcionalnosti:

- dodavanje entiteta,
- osvežavanje liste,
- brisanje entiteta.

Deo sa detaljima organizovan je u podsekcije čiji sadržaj zavisi od tipa i vrednosti označenog entiteta. Aplikacija je nazvana „***Web DMC***". Gore opisana organizacija korisničkog intefejsa može se videti na slici 4.1.



Slika 4.1 *Osnovna stranica Web DMC aplikacije*

Ukoliko su startovanje aplikacije i autentifikacija korisnika bili uspešni, korisnik će na raspolaganju imati osnovnu stranicu aplikacije. Administracija entita obavlja se na sekciji aplikacije koja nosi ime entiteta. Tako na primer administriranje korisnika (***users***), obavlja se na sekciji „***DMS Users***".

Ukoliko postoji bar jedan DMS korisnik, on će biti označen i deo sa detaljima biće popunjen podacima koji odgovaraju tom korisniku. Kada se obavi izmena podataka označenog korisnika, odnosno kada nastanu izmene u delu za detalje, potrebno je potvrditi izmene klikom na dugme „***Apply changes***" koje je predstavljeno plavom disketom. Ukoliko se snimanje ne obavi, prilikom prvog pokušaja navigacije s označenog korisnika na drugog DMS korisnika ili na neki drugu sekciju, korisnik aplikacije biće upitan da li želi snimiti nastale izmene.

Osim izmene podataka vezanih za korisnika, takođe je moguće dodati nove korisnike, obrisati postojeće i osvežiti listu korisnika. Mehanizmi za administraciju identično funkcionišu na svim sekcijama aplikacije.

5. ZAKLJUČAK

Pri realizaciji aplikacije ispunjeni su svi korisnički zahtevi navedeni u poglavlju 2.4.

U toku izrade ovog master rada zapaženo je da tvrdnja da se GWT aplikacije isto ponašaju u okviru svakog ***web browsera*** nije u protpunosti istinita. ***Widget***-i i paneli nisu se dosledno ponašali na različitim ***browser***-ima usled različitog tumačenja CSS standarda od strane tih ***browser***-a. S izuzetkom gore navedenih nedostataka, GWT se pokazao kao dobra tehnolohija za jednostavan i brz razvoj ***web*** aplikacija. GWT grafički elementi omogućili su da razvijena aplikacija ima pozitivne osobine dobre ***desktop*** aplikacije (prozori za dodavanje novih entiteta, dobra preglednost, pokretnost grafičkih elemenata) i dobre osobine ***web*** aplikacije (nema potrebe za instalacijom korisničkog softvera na klijentskoj mašini).

Direktorijumski servisi pokazali su se kao dobro skladište podataka, jednostavno za korišćenje i manipulisanje podacima.

Sama aplikacija, nezavisno od nedostataka korišćenih tehnologija, ima dosta prostora za usavršavanje:

- Potrebno je implementirati mehanizme za zaštitu podataka tokom komunikacije klijentske i serverske strane, primenom HTTPS (HTTP sa SSL/TLS) protokola.
- Potrebno je implementirati podršku za rad sa tastaturom (selekcija elemenata, aktivacija dugmića, navigacija itd).
- Treba implementirati sekciju „***DMS System***".
- Treba razmotriti mogućnost upotrebe ***hibernate*** funkcionalnosti za mapiranje modela podataka na entitete u direktorijumskim servisima.

6. LITERATURA

[1] M.Mirković, "*Basic concepts and architecture overview*", Telvent DMS
[2] *http://en.wikipedia.org/wiki/Directory_service, Directory Service*
[3] *http://en.wikipedia.org/wiki/LDAP, LDAP*
[4] *http://en.wikipedia.org/wiki/OpenLDAP, OpenLDAP*
[5]*http://en.wikipedia.org/wiki/Ajax_%28programming%29, Ajax (programming)*
[6] *http://code.google.com/webtoolkit/overview.html, Google Web Toolkit*
[7] *http://en.wikipedia.org/wiki/Google_Web_Toolkit, Google Web Toolkit*
[8]*http://code.google.com/webtoolkit/doc/latest/tutorial/RPC.html, GWT RPC*

Kratka biografija:



Aleksandra Krneta rođena je 1986. godine u Šibeniku. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo odbranila je 2011. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.39:004.7

АКВИЗИЦИЈА ПОДАТАКА БЕЖИЧНИМ ПУТЕМ КОРИШЋЕЊЕМ ПДА УРЕЂАЈА

WIRELESS DATA ACQUISITION USING A PDA

Дејан Бирманчевић, *Факултет техничких наука, Нови Сад*

Област – МЕХАТРОНИКА

Кратак садржај – *У раду је приказан развој апликација за прикупљање и обраду информација посредством бежичних технологија и њихове примене у реалним системима. На основу наведених захтева бежичне комуникације и манипулације подацима, креиране су две апликације (сервер - клијент) које у себи имају интегрисане модуле за коришћење RFID технологије и SQL базе података.*

Abstract – *The paper describes the development of applications for collecting and processing information through wireless technologies and their application in real systems. Based on these demands, wireless communication and data manipulation, we have created two applications (server - client) which themselves have integrated modules for the use of RFID technology and SQL database.*

Кључне речи: *RFID, PDA, базе података, бежичне технологије*

1. УВОД

Развој рачунарских технологија и стицање услова за њихову примену, допринело је бољом повезаношћу људи, доступношћу и брзином протока информација. Искоришћен је низ погодности са све чешћим утицајем на свакодневни живот човека као јединке, целокупног друштва и цивилизације.

Суштина и начини примена бежичних технологија у развоју апликација је објашњено у поглављу 2. У поглављу 3, представљена је технологија преноса података помоћу радио таласа, позната као Radio Frequency IDentification (RFID) технологија. У поглављу 4, представљен је PDA (Personal Digital Assistant) уређај и развојни алати (Visual Studio- Visal Basic 2008 и SQL (Structured Query Language)) који су коришћени током израде дипломског- мастер рада.

Структура и опис апликација које представљају део система који служи за манипулисање информацијама, као што су: складиштење, обрада, пријем, слање, прикупљање, итд., је описана у поглављу 5: "Аквизиција података бежичним путем коришћењем PDA уређаја".

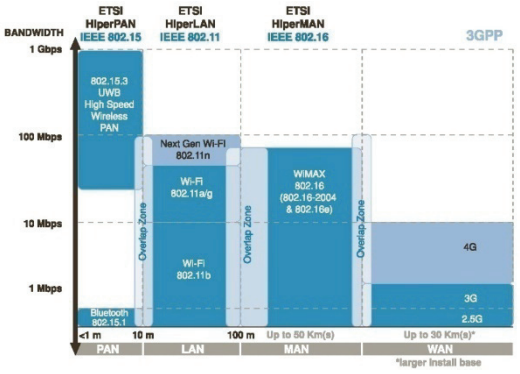
У основи, апликација представља спрегу између сервера (рачунара на ком је смештена база података) и мобилног уређаја (пример: PDA) путем бежичних технологија.

НАПОМЕНА:
Овај рад проистекао је из дипломског мастер рада, чији је ментор др Гордана Остојић, доц.

2. Бежичне технологије WLAN (Wi-Fi)

Бежична технологија (Wireless Technology) представља метод за пренос информација. Омогућава пренос информација између два или више система на удаљеност која варира од кратких растојања (неколико метара), до дугих релација (хиљадама или милионома километара, за радио комуникацују), без посредства електричних проводника или жица [1].

Обухвата разне врсте фиксних, мобилних и преносних двосмерних радио апарата, мобилних телефона, персоналних дигиталних уређаја (Personal Digital Assistant-PDA) и бежичних мрежа. Бежична локална мрежа WLAN (Wireless local area network) је технологија димензионисана за мале удаљености које у градским условима износе од неколико десетина до неколико стотина метара. На слици 1 је приказана подела бежичних мрежа према домету уређаја и брзине преноса података за RFID, Bluetooth, WLAN (Wi-Fi), WIMAX, UMTS (3G) мреже, итд...



Слика 1. *Подела бежичних мрежа према домету уређаја и брзине преноса података*

3. RFID технологија

RFID је бежична технологија за идентификацију и праћење објеката путем радио таласа[2].

У основне елементе RFID система спадају:

- **Читач**- представља фиксни или преносиви уређај чија је функција активирање тага и прикупљање информација које одашиљу тагови.
- **Контролер**- посредничка компонента која омогућава RFID систему да комуницира са читачем.
- **Антена-** служи као појачавач сигнала.

- **Таг (транспондер)**- носилац информација, чува и шаље податке.

Технологија која користи бежичну комуникацију путем радио таласа за размену података између читача и картице, тага или транспондера у циљу идентификације и праћења. Не захтева контакт читача и картице, а подаци могу бити очитани на удаљености од неколико метара, изван видокруга читача, као и кроз људско тело, одећу и неметалне материјале.

4. Примена SQL језика и програмског пакета Visual Studio

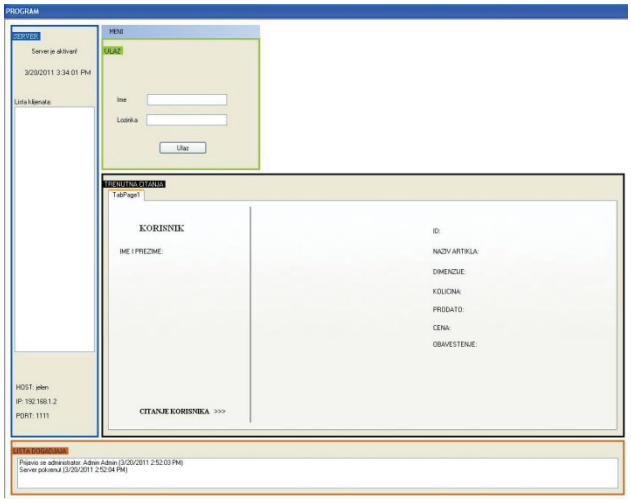
Један од најчешће коришћених стандардизованих језика за манипулисање базом података, у шта спада креирање, приступ, мењање, управљање базом података, итд. јесте SQL (Structure Query Language) [3]. Овај језик је заснован на упитима и намењен је искључиво за контролу података и за ефикасно управљање базама података.
Visual Studio је интегрисано развојно окружење (Integrated Development Environment- IDE) корпорације Microsoft, слика 20. Може се користити за развој конзола и графичких корисничких интерфејс апликација упоредо са Windows Form апликацијама, веб сајтовима, веб апликацијама и веб сервисима на платформама које подржавају Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight. Visual Studio садржи код Едитор, интегрисани дебагер чији је задатак отклањање грешака на машинском нивоу. Остали уграђени алати укључују облике: дизајнер за изградњу graphical user interface (GUI) апликација, веб дизајнер, дизајнер класа, као и дизајнер шема базе података.

5. Аквизиција података бежичним путем коришћењем PDA уређаја

У овом поглављу биће приказана: почетна идејна решења, смернице и правци развоја, структура и опис апликација које представљају софтверско решење система који служи за манипулисање информацијама, као што су: пријем, обрада, складиштење, слање, прикупљање, итд. Аквизиција података бежичним путем коришћењем PDA уређаја се у основи састоји од апликација које представљају спрегу између сервера (рачунара на ком је смештена база података) и мобилног уређаја (пример: PDA) путем бежичних технологија.

5.1 Сервер апликација

Апликација која је описана у овом поглављу, представља спрегу између корисничке апликације и базе података. Сервер апликација има могућност креирања базе података у коју ће се смештати подаци и све корисне информације везане за администраторе, кориснике, артикле и пропратне информације о систему.
Попуњавање и манипулисање базом података се врши преко апликације. У случају да база непостоји, што је карактеристично приликом првог покретања апликације, програм то детектује и даје могућност администратору да креира базу, са одговарајућим табелама за администраторе, кориснике и артикле. Визуелни приказ система се састоји од четири прозора (слика 2), који представљају радне станице са посебним функцијама. У зависности од тога да ли се желе пратити одређени параметри система, приказ броја корисника па све до манипулације подацима које су смештени у бази података, радне станице ће представљати користан алат у току рада. Свака радна станица је обележена оквиром различите боје, са циљем бржег распознавања, сналажења, као и због различитог приоритета операција током рада на радним станицама.



Слика 2. *Приказ радних станица апликације*

Постоје четири радне станице, а то су:

- **Главна** радна станица, преко које се врши манипулисање базом података.
- **Сервер** радна станица, на којој се приказује време, адреса, порт сервера и број корисника који су активни.
- **Листа догађаја**, радна станица која има функцију обавештавања о догађајима у систему.
- **Тренутна читања**, радна станица чија је намена визуелни приказ и праћење читања корисника.

Главна радна станица састоји се из три сегмента. Основна функција главне радне станице је спрега са базом података (први сегмент), визуелни приказ тренутно активног администратора (други сегмент) и манипулација радним станицама (трећи сегмент). Сервер радна станица се састоји од листе активних клијената (листа у коју се смештају активни клијенти са којима је успостављена бежична комуникација) и индикатора који приказују да ли је сервер активан, HOST на коме је сервер активан, тренутно време на серверу, IP адреса сервера и порт преко кога се врши комуникација са клијентима. Радна станица тренутна читања представља станицу која служи да визуелно прикаже тренутна читања клијената. Састоји се из два сегмента. Први сегмент радне станице служи да визуелно прикаже корисника који је извршио операцију читања. Његови основни подаци: име,

презиме и слика. Други сегмент радне станице представља део за визуелни приказ артикла који је прочитан од стране клијента. Идентификациони број, назив артикла, основне карактеристике и изглед артикла, представљају податке који се приказују у другом сегменту станице.
Радна станица- листа догађаја представља станицу која служи за визуелни приказ свих догађаја који се јаве у систему.

5.2 Корисничка (PDA) апликација

Апликација о којој ће бити речи у овом поглављу представља спрегу између RFID система и података који су уписани на RFID картице. RFID систем је представљен као интегрисани модул PDA уређаја за кога је апликација и намењена. Сходно томе пре самог описа апликације и њене намена, биће представљен модел PDA уређаја који је коришћен током израде дипломског рада.

5.2.1 M3 SKY RFID MC-7510S

Представља модел PDA уређаја који је коришћен током израде дипломског рада. Садржи интегрисани RFID читач, могућност бежичне комуникације и другу пратећу опрему и модуле који нису били значајни за израду пројекта, а представљени су на слици 3. Поред RFID читача и могућности бежичне комуникације, коришћена је база у виду dll фајлова, који представљају SDK за дотични уређај. Наведени фајлови представљају везу између програма који креира програмер и хардвера (у овом случају RFID читача), а такође представљају системске фајлове са готовим фунцијама, које ће бити позиване у зависности од задатка које RFID читач треба да обави.



Слика 3. *M3 SKY RFID MC-7510S*

5.2.2 Опис корисничке апликације

Корисничка апликација (креирана за оперативни систем Windows Mobile 5.0) састоји се из три сегмента:

- први сегмент представља везу апликације са RFID системом;
- други сегмент представља везу апликације са сервером.

Почетна форма корисничке апликације састоји се од визуелне индикације, у коју спада индикација верзије RFID читача, да ли је успостављена комуникација са RFID читачем, да ли је порт отворен, брзина преноса информација кроз порт и да ли је корисник успоставио везу са сервером (базом у којој су смештени подаци о кориснику и које корисник ишчитава) и приказана је на слици 4.



Слика 4. *Почетна форма корисничке апликације*

Командно дугме "Отвори порт" служи да се преко њега успостави комуникација између корисничке апликације и RFID читача, а притиском на командно дугме "Конектуј се", активира се форма у којој се уносе основни подаци за конекцију са сервером (IP сервера, корисничко име и лозинка).
Да би се извршио процес активирања RFID читача, потребно је отворити порт и успоставити комуникацију са RFID читачем. Наведена операција, се извршава притиском на командно дугме "Отвори порт", после чега се појављује обавештење да је порт отворен или је порт и даље затворен. По отварању порта појављује се могућност коришћења RFID читача, свих његових опција, у које спадају: могућност читања, уписивање, подешавање опција и приказ информација о типу и моделу RFID читача, слика 5.
На приказаној форми, слика 5, активиран је сегмент "Читај"(притиском на језичак "'Читај') активира се форма на којој се налазе командни елементи, преко којих је могуће вршити читање информација са RFID тага (картице). Сваки елемент форме има одређену функцију приликом вршења операције повећања или смањења вредности адресе блока, читања картица, чишћења листе прочитаних картица, заустављања операције читања, итд.
Командно дугме "Читај", има функцију да активира опцију читања RFID система. Део форме који је означен као "Блок" служи кориснику да дефинише са ког сегмента, блока картице ће се вршити читање. Елементи форме, текст поље за јединствени индентификациони број картице и листа података картице служе да прихвате наведене информације које су ишчитане са картице и дају визуелни приказ информација кориснику који врши читање.



Слика 5. *Активана форма "Читај"*

Командно дугме "Стоп", има функцију да заустави процес очитавања података са картица које се налазе у активном подручију читања RFID система. Командно дугме "Очисти", има функцију да чисти листу података. Сви подаци прочитани са картица, који су уписани у листу података, биће избрисани. Поред описане форме сегмента "Читај", постоје још две форме које се могу активирати притиском на језичак "Упиши"(уписивање података на таг) и "Верзија"(подаци о RFID читачу), под усовом да је успостављена комуникација са RFID читачем, у супротном то неће бити могуће.

5.2.3 Спрега и размена података међу апликацијама

Описане апликације, сервер и корисничка апликација, су међусобно спрегнуте како би обављале функцију протока информација и података у оба смера (Слика 41). Као што је објашњено, на сервер апликацији се налази база података и систем за прикупљање информација које јој шаље корисничка апликација и да на њих одговара повратном информацијом или податком, из базе података, који захтева корисничка апликација. Спрега је остварена бежичним путем, коришћењем Wi-Fi технологије, и тиме је обезбеђена већа мобилност и погодност бежичне технологије.
За правилно функционисање система, потребно је извршити:

- Активирање сервер апликације;
- Креирање базе података;
- Именовање администратора;
- Именовање корисника;
- Активиратње корисничке апликације;
- Успостављање везе са сервером;
- Пријављивање на сервер;
- Активирање RFID система;
- Активирање сегмента за читање;
- Читање;
- Слање информација и података на сервер.



Слика 6. *Спрега и размена података међу апликацијама*

6. ЗАКЉУЧАК

Систем за аквизицију података путем бежичне технологије коришћењем PDA уређаја, превасходно налази примену у опслуживању складишта свих величина, праћења предмета током процеса производње, и након процеса производње, присутност предмета у радном окружењу, сортирања и одлагања предмета на задату локацију, итд. Повезаност корисника са сервером, комуникација сервер-корисник, корисницима омогућава да располажу потребним информацијама у тренутку када је то неопходно. Применом система за аквизицију података путем бежичне тахнологије коришћењем PDA уређаја, смањује се време потребно за извршење операције лоцирања предмета рада, прикупљања основнох информација о предмету и обављања одређене акције на предмету рада.
Правци даљег развоја апликација, њихов међусобни утицај и већа поузданости целог система би се кретала у правцу примене технологија које би одговарале датим захтевима мрежне покривености. Коришћењем GPRS технологије, GSM система, омогућила би се мрежна покривеност у случају система који комуницирају на великим раздаљинама, где сваки део система представља засебан део једне целине.

7. ЛИТЕРАТУРА

[1] Powell, Steven; Shim, J.P. (Eds.): Wireless Technology: Applications, Management, and SecuritySeries Lecture Notes in Electrical Engineering, Vol. 44
[2] Gordana Ostojić, Stevan Stankovski: "Primena bežičnih tehnologija u sistemima za snabdevanje "Fakultet tehničkih nauka Novi Sad
[3] Mary E. Treseler: Learning SQL (2nd ed.). Sebastapol, CA, USA: O'Reilly. ISBN 978-0-596-52083-0.

Кратка биографија:

Дејан Бирманчевић рођен је у Шапцу 1985. год. Дипломски мастер рад на Факултету техничких наука из области Мехатронике – Мехатроника, роботика и аутоматизација одбранио је 2011.год.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 629.3.03

АНАЛИЗА ЗАЛЕТА ПОГОНА ТРАКАСТИХ ТРАНСПОРТЕРА

ANALYSIS OF THE RUNNING DRIVE BELT CONVEYORS

Ђорђе Васић, Растислав Шостаков, *Факултет техничких наука, Нови Сад*

Област – МЕХАТРОНИКА

Кратак садржај – *У раду је приказан преглед, анализа и поређење најчешће коришћених типова погона тракастих транспортера. За конкретан пример транспорта рафинисаног шећера извршен је прорачун тракастог транспортера и његовог погона. Приказан је прорачун времена убрзања погона као и степена сигурности против проклизавања између траке и погонског бубња за три карактеристична типа погона.*

Abstract – *The paper presents the review, analysis and comparison of the most common types of belt conveyor driving systems. For the concrete example of refined sugar transportation, the calculation of the belt conveyor and its driving mechanism has been made. The calculation of the start-up duration and the safety factor against slippage between the belt and the driving drum for three specific operation regimes are shown.*

Кључне речи – *тракасти транспортер, време убрзања, степен сигурности против проклизавања, прорачун.*

1. Увод

Савремена технологија транспорта материјала захтева примену комплексне механизације високе продуктивности. У таквим условима интенција је на увођењу континуалног процеса експлоатације, где тракасти транспортери са врло високим транспортним ефектима имају значајну улогу.

Тракасти транспортери се користе за транспорт насипног и комадног терета претежно у хоризонталном или мало нагнутом правцу (до 25º) помоћу траке као вучног и носећег елемента. Због својих изузетно повољних техноекономских карактеристика представљају најраспрострањеније уређаје непрекидног транспорта у експлоатацији.

Данас се у свету примењују тракасти транспортери са ширином траке и до 3200 mm, инсталисане снаге по погонској станици и до 12 MW, са брзином транспорта до 7.5 m/s, капацитета до 40.000 m^3/h.

На површинским коповима, како у нашој земљи, тако и у свету, још увек су доминантна класична транспортна постројења, али се мора нагласити да су у последње две деценије погонски и управљачки системи транспортера знатно унапређени.

Класична решења погона тракастих транспортера су базирана на:

- асинхроним кавезним електромоторима (АКМ) директно прикљученим на мрежно напајање,
- АКМ покретаним сниженим напоном системом "звезда-троугао" или помоћу софт-стартера,
- АКМ повезаним са редуктором преко хидродинамичке спојнице (ХДС) са константним или варијабилним пуњењем,
- асинхроним клизноколутним електромотором (АКЛМ) са покретањем роторским упуштачима и повезаним са редуктором преко еластичне спојнице.

Данас је у свету све израженији тренд да се погони трака тракастих транспортера изводе са променљивим брзинама заснованим на примени фреквентних претварача.

Основни проблем код погона транспортера великих капацитета је њихово покретање. Све масе потребно је из стања мировања покренути и убрзати до стационарне брзине. Од укупног времена рада транспортера, сам залет обухвата релативно кратко време, али са веома израженим напрезањима свих елемената транспортера. Непредвиђене ситуације као што су прекид напајања електричном енергијом или било какви кварови чији је резултат заустављање оптерећеног транспортера, могу код неправилно димензионисаних погона, да доведу до проблема при стартовању истог након нормализације напајања, односно отклањања кварова.

2. Анализа погона

Погоне тракстих транспортера је могуће класификовати на више начина, нпр. на електричне и електромеханичке системе у зависности од тога да ли поред електромотора садрже и неку механичку направу за промену брзине (хидродинамичка спојница, механички варијатор и сл.).

У овом раду су прво анализиране различите врсте погона базиране на примени асинхроног кавезног мотора, затим асинхроног мотора са клизним колутовима и на крају, као најзанимљивије решење, примена асинхроних кавезних мотора управљаних фреквентним претварачем.

3. Погони са АКМ

Предности АКМ, у односу на остале врсте електричних машина, су првенствено мања цена, једноставност конструкције, мањи момент инерције,

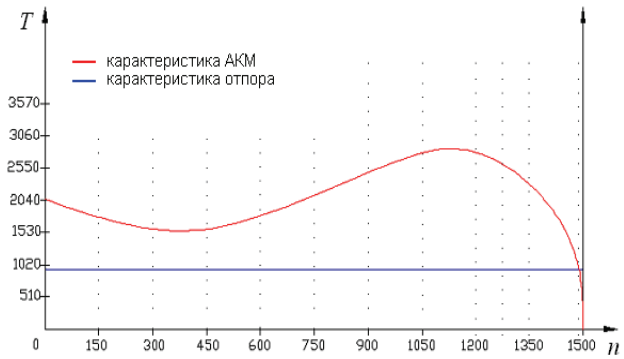
Напомена:
Овај рад је проистекао из дипломског-мастер рада чији је ментор био доц. др Растислав Шостаков.

робусност, поузданост и сигурност у раду, лако одржавање, док су недостаци везани углавном за услове покретања (полазна струја 3÷9 пута већа од назначене) и немогућност регулисања брзине обртања независно од отпора.

3.1. Погон са АКМ директно прикљученим на мрежу

За покретање тракастог транспортера потребна је одређена додатна снага, тачније обртни момент, који је у стању да савлада отпоре кретања транспортера и да убрза масе из стања мировања до номиналне брзине. Погони са константним радним отпором, као што су и тракасти транспортери, могу у околини тачке седла карактеристике мотора имати потешкоћа са залетом. Да би се побољшале особине кавезних мотора при залету, односно да би се смањиле полазне струје и повећао полазни моменат, много је рађено на конструкцији ротора и предлаган знатан број разних решења. Најбоље особине по овом питању имају мотори са двоструким и дубоким жљебовима.

Дијаграм погона за дати транспортер приказан је на слици 3.1.



Слика 0.1 Карактеристике АКМ и радног уређаја

На основу прорачунски добијеног времена трајања залета, које је у овом случају релативно велико (≈ 6 секунди), може се закључити да би АКМ био изложен великом термичком напрезању током залета, што би довело до смањења његовог века трајања, а у најгорем случају и до прегоревања намота.

.

Предности ове врсте погона тракастих транспортера се огледају у једноставности конструкције, док су мане:

- велика струја при залету, која оптерећује мрежу,
- велики полазни момент, који може да доведе до проклизавања између погонског бубња и траке,
- велико механичко напрезање елемената преноса и траке, што утиче на смањење века трајања и повећање трошкова одржавања.

Због свега горе наведеног, погон тракастих транспортера са АКМ директно прикљученим на мрежу се примењује код кратких транспортера мањих снага који ретко стартују.

3.2. Погон са АКМ и преклопком звезда-троугао

За залет АКМ већих снага се у случају слабијих мрежа и у циљу смањења полазне струје, често примењује укључивање мотора под сниженим напоном. Једна од метода базирана на овом принципу је покретање помоћу преклопке звезда-троугао где се врши ступњевита промена напона напајања, обзиром да је у споју звезда напон $\sqrt{3}$ пута мањи него у споју троугао, док се момент смањује са квадратном зависношћу од напона. Ово смањење момента у неким случајевима може бити врло значајна ограничавајућа околност за примену овог начина покретања код постројења какви су тракасти транспортери, нпр. при стартовању оптерећеног транспортера.

Због горе наведеног, погон АКМ са преклопком звезда-троугао се понекад примењује за покретање празног транспортера док знатно већу примену налази у погону компресора, вентилатора и других уређаја код којих је момент отпора зависан од брзине кретања.

3.3. Погон са АКМ и софт-стартером

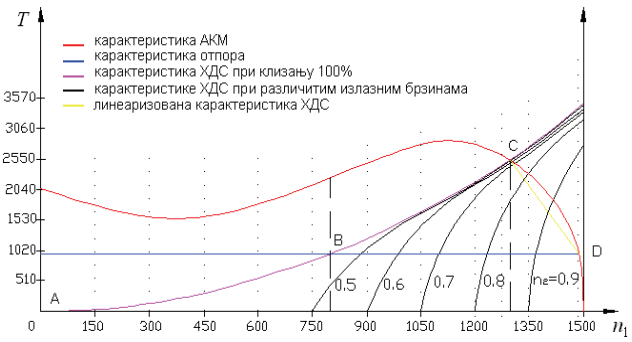
Рад софт-стартера заснива се на континуалној промени напона напајања АКМ. Применом софт-стартера у погону тракастих транспортера обезбеђује се минимални полазни момент који омогућава "меко" покретање траке. Континуалним повећањем напона прво се врши елиминисање зазора у елементима преноса, а затим постепено затезање траке и њено убрзавање до номиналне брзине. Резултат тога је смањење напрезања у елементима преноса, смањење силе затезања у траци као и спречавање проклизавања између бубња и траке приликом покретања. Мања затезна сила у траци у фази стартовања омогућује коришћење трака мање прекидне чврстоће, што се такође односи и на остале елементе транспортера.
За залет тракастих транспортера користе се софт-стартери који садрже регулатор струје помоћу кога се одржава полазна струја на сталној вредности. Најчешће је вредност полазне струје 3-4 пута већа од номиналне струје АКМ. Код покретања пуне траке или после дужег застоја у зимском периоду, или у сличним ситуацијама када су отпори кретања код поласка повећани, софт-стартер ће одржавати сталну полазну струју, а тиме и момент, услед чега може доћи до делимичног затезања траке, али не и њеног покретања. Овакве ситуације су честе у пракси и треба их имати у виду.

3.4. Погон са АКМ и хидродинамичком спојницом

У хидродинамичкој спојници се на примарном, пумпном колу механичка енергија, добијена од погонског мотора, претвара у струјну енергију радног флуида и на секундарном колу се поново врши трансформација у механичку енергију која се предаје радном уређају.

Код примене хидродинамичких спојница основни захтев је да АКМ, који у старту вуче из мреже струју вишеструко већу од номиналне, извршивши залет у кратком периоду, а затим почне са предајом обртног момента радној машини која лагано убрзава преко хидродинамичке спојнице.

Развој хидродинамичких спојница са константним пуњењем је текао у правцу продужења времена залета постројења са великим масама какви су и тракасти транспортери великих дужина и капацитета. Извршена је подела на хидродинамичке спојнице без преткоморе и са преткомором. Спојнице са преткомором су предности хидродинамичких спојница још додатно повећале. Посебно за тракасте транспортере великих дужина су развијене спојнице са релативно великим преткоморама које омогућавају стартно време транспортера и до 70 секунди.
Дијаграм погона са ХДС са константним пуњењем без преткомора за конкретан транспортер дат је на слици 3.2. Добијено време залета погона износи око 11 секунди док је динамички степен сигурности против проклизавања 0,40.
Време трајања залета АКМ у критичној области је мање од једне секунде, тако да нема опасности од термичког преоптерећења истог. Овако низак степен сигурности је последица избора обичне ХДС без преткомора. Усвајањем спојнице са преткоморама или са варијабилним пуњењем, време залета се може знатно продужити, а такође и спречити проклизавање између погонског бубња и траке.



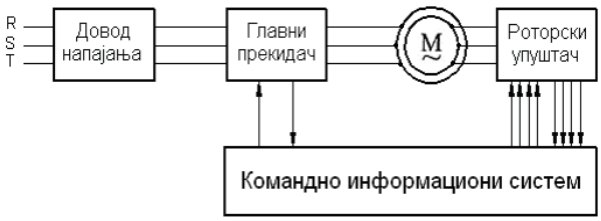
Слика 3.2 Дијаграм погона са ХДС

Уградњом хидродинамичке спојнице између електромотора и редуктора у погону тракастог транспортера постижу се веома позитивни ефекти:

- безударно, континуално убрзање великих маса,
- омогућена примена електромотора са кратко-спојеним ротором (кавезних мотора),
- мотор се растерећен стартује до високих обртаја,
- димензионисање електромотора према номиналном оптерећењу,
- ограничен полазни момент,
- веома добро пригушење торзионих удара и осцилација у току рада погона,
- сигурна заштита електромотора и трансмисије од превеликих оптерећења и заглављивања,
- изравнавање оптерећења код вишемоторних погона.

4. Погон са АКЛМ и роторским упуштачем

На слици 4.1 је приказан концепт погона код кога се команде за укључење мотора и управљање роторским упуштачем врше из рачунарски реализованог командно информационог система који управља радом целе погонске станице.



Слика 4.1 Шема погона АКЛМ са роторским упуштачем

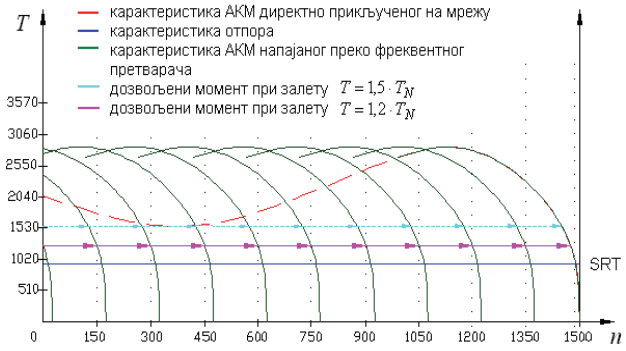
Недостаци АКЛМ се огледају у постојању контактних четкица и клизних прстенова који усложњавају конструкцију и смањују њену поузданост. Такође, у поређењу са са АКМ, мотори са клизним прстеновима имају веће габарите и вишу цену. У апликацијама у којима се захтева рад при смањеним брзинама, АКЛМ карактерише неекономичан рад и то из разлога јер се при прикљученим додатним отпорницима непотребан вишак енергије преводи у топлоту.

5. Погон са АКМ и фреквентним претварачем

Последњих година у развијеним земљама рударства све већи број транспортера са гуменом траком је опремљен погонима са променљивом брзином, базираним на примени фреквентних регулатора (претварача), апарата састављених од елемената енергетске електронике. У поређењу са стандардним моторима са константним напоном и константном фреквенцијом, ови погони, поред регулације брзине, омогућавају значајно смањење потрошње енергије и хабања механичких делова и траке.

Константно убрзање траке које је могуће остварити АКМ напајаним фреквентним претварачем, доприноси смањењу динамичких оптерећења траке и осталих елемената тракастог транспортера, као и избегавању проклизавања између погонског бубња и траке, што се и види на основу прорачунски добијеног степена сигурности, који за задати момент током залета ($1{,}2 \cdot T_N$) износи 1,13. На основу тога може се закључити да је при инсталацији ове врсте погона могуће усвојити траку мање прекидне чврстоће, а такође и остале елементе транспортера, изузев носеће конструкције која мора бити предвиђена за 100% искоришћења попречног пресека траке. Такође се мора нагласити да се при овом начину погона, као и код примене ХДС, могу усвојити јефтинији АКМ ниже класе ротора.

Дијаграм погона за конкретан пример тракастог транспортера приказан је на слици 5.1.



Slika 5.1 Dijagram pogona trakastog transportera sa frekventnim pretvaračem

6. Zaključak

На претходним странама дат је опис и анализа најчешће примењиваних типова погонских система, а за најинтересантније од њих је извршен и прорачун за конкретан тракасти транспортер.

Директни погон са АКМ представља најједноставније решење, али са великим недостацима. За АКМ је веома важно да изврши залет у кратком временском периоду како се не би термички преоптеретио услед вишеструко веће струје од номиналне коју повлачи током залета. На основу прорачунски добијеног времена трајања залета, које је у овом случају релативно велико, може се закључити да би АКМ био изложен великом термичком напрезању током залета што би довело до смањења његовог века трајања, а у најгорем случају до прегоревања намота.

Због своје повољне механичке карактеристике, ХДС омогућује краткотрајни залет АКМ уз брзи пролазак кроз, за њега, неповољни режим рада, уз лагани залет транспортера изложеног знатно умањеним силама инерције. Велика предност примене ХДС је и могућност употребе јефтинијих АКМ ниже класе ротора Коришћењем спојнице са додатним коморама, а посебно са променљивим пуњењем, време залета се може продужити и до неколико десетина секунди. Самим тим је убрзање знатно мање, а тиме и опасност од проклизавања између погонског бубња и траке.

У односу на остале врсте погона, управљачке карактеристике фреквентних претварача омогућују њихово интегрисање у систем управљања и надзора, чиме се добија врло висок ниво заштите, контроле и аутоматизације система.

У погледу трошкова одржавања и експлоатације, предности фреквентно регулисаних транспортера су веома изражене. Са аспекта инвестиционих улагања треба нагласити да брзи развој фреквентних претварача смањује њихову набавну цену, чиме разлика у инвестиционим улагањима постаје све мања.

Узимајући у обзир све наведене особине фреквентне регулације тракастих транспортера, као и могућност једноставне и брзе имплементације алгоритама управљања, долази се до закључка да ће у будућем развоју погона фреквентно регулисаних транспортера, тенденција бити повећање степена искоришћења и синхронизација транспортера и целокупног система.

7. Литература

[1] Владић Ј.: Механизација претовара II, Факултет техничких наука, Нови Сад, 1991.

[2] Шостаков Р.: Погонски системи (скрипта), Нови Сад, 2007.

[3] Шостаков Р., Узелац Д., Бркљач Н.: Методологија праћења рада погонског механизма са хидродинамичком спојницом и одређивање трајања његовог залета, Факултет техничких наука.

[4] Теодоровић В.: Електричне погонске машине I, Научна књига, Београд, 1978.

[5] Студија: Оправданост примене тракастих транспортера са фреквентном регулацијом брзине на површинским коповима ЕПС-а, Рударско-геолошки факултет, Београд, 2005.

[6] Рударске машине (скрипта), Рударско-геолошки факултет, Београд, 2009.

[7] Крсмановић Љ., А. Гајић: Турбомашине-хидродинамички преносници снаге, Универзитет, Београд, 2006.

[8] Радаковић Ј., Секер И., Тошин А., Бабић А., Жикић З.: Аутоматски роторски покретач за асинхроне клизноколутне моторе за транспортни систем, десети симпозијум енергетске електронике, Нови Сад, 1999.

[9] Л. Г. Шахмейстер, Г.И.Солод: *Подземные конвейерные установки*, Москва, 1976.

[10] Лекић З.: Регулациона турбоспојница у погонском систему тракастог транспортера, Научни рад, Косовска Митровица, 2005.

[11] Документација произвођача "*Dunlop*": *Conveyor belt Technique- Design and calculation.*

[12] Каталози произвођача: *ATB Sever, 14 Oktobar, Ivo Lola Ribar, Voith, Dunlop.*

[13] Смиљић Р.: Погон кретања моста дизалице са хидродинамичком спојницом, графоаналитичко одређивање трајања залета, Дипломски рад, Нови Сад, 2006.

Кратка биографија:



Ђорђе Васић је рођен 1983. год. у Зрењанину. Завршио је Електроехничку школу „Никола Тесла“ у Зрењанину. Факултет техничких наука је уписао 2002. год. Студент је смера Мехатроника у механизацији..



Растислав Шостаков рођен у Београду 1949. године. Докторирао на ФТН у Новом Саду 2007. године и од тада у звању доцента. Области интересовања: машине унутрашњег транспорта, погонска чврстоћа, електромоторни погони.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 658.52.011.56

OPTIMIZACIJA POTROŠNJE VAZDUHA POD PRITISKOM NA LINIJI ZA PROIZVODNJU LIMENKI

OPTIMIZATION OF COMPRESSED AIR CONSUMPTION IN CANS MANUFACTURING SYSTEM

Kosta Žarkov, Dragan Šešlija, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – *U ovom radu je dat kratak pregled metoda za uštedu energije u sistemima vazduha pod pritiskom. U eksperimentalnom delu su specificirani troškovi vazduha pod pritiskom u sistemu za proizvodnju limenki i izvršena optimizacija potrošnja vazduha pod pritiskom na liniji za oblikovanje vrata limenki. Izvršena su merenja potrošnje vazduha pod pritiskom na liniji za oblikovanje vrata limenki i predloženo idejno rešenje za smanjenje potrošnje vazduha pod pritiskom a odabrano rešenje je zatim i projektovano.*

Abstract – *In this document a brief overview of saving energy methods in compressed air systems is made. Scope of the experimental part is to determine costs of compressed air consumption in cans manufacturing system and the task is to measure compressed air consumption on can necks shaping production. Further, to optimize compressed air consumption on can necks shaping production line and propose solutions for reducing of compressed air consumption and to project chosen solution.*

Ključne reči: *Distribucija vazduha pod pritiskom, troškovi vazduha pod pritiskom*

1. UVOD

Povećanje energetske efikasnosti je opšti pravac razvoja u svetu. Značajan segment u tome zauzima povećanje energetske efikasnosti u industriji, a u okviru toga veoma je značajan udeo povećanja energetske efikasnosti pneumatskih sistema. U većini fabrika se efikasno iskoristi tek polovina ili čak i manje od ukupno proizvedenog vazduha pod pritiskom.

Vazduh pod pritiskom je jedan od najrasprostranjenijih oblika energije koja se koristi u industriji, a pneumatski sistemi se primenjuju, pored industrije, i u mnogim drugim oblastima ljudskog delovanja. Velik broj faza rada fabričkih postrojenja koristi vazduh pod pritiskom. Najčešći primeri sistema koji koriste vazduh pod pritiskom su sistemi pneumatskog transporta, obrada hrane, hemikalija i farmaceutskih proizvoda, pneumatski alati, bojenje sprejom, oblaganje površina praškovima, mašine za pakovanje i vazduh pod pritiskom za opšte namene u fabrikama i radionicama.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je dr Dragan Šešlija, red. prof.

Sistemi vazduha pod pritiskom su pouzdani, sigurni i veoma pogodni za obavljanje mnogih funkcija, ali se o njihovoj ekonomičnosti retko vodi računa. Oni su značajan prenosnik energije u industriji, pa se često vazduh pod pritiskom smatra četvrtim po značaju, posle električne energije, gasa i vode. Međutim, jedino se vazduh pod pritiskom proizvodi na licu mesta pa se time omogućava korisnicima potpuna kontrola nad celim procesom. Na žalost, pri tome ne postoji merni uređaj koji će pokazati koliko košta potrošeni vazduh pod pritiskom.

Prisutna je česta zabluda da je vazduh pod pritiskom besplatan ili da toliko malo košta da ne zaslužuje trošenje skupocenog menadžerskog vremena na rešavanje ove problematike. Međutim, vazduh je besplatan samo dok se ne sabije i ubaci u sistem vazduha pod pritiskom, a onda ima cenu koja može da doprinese da se neke aplikacije koje troše vazduh pod pritiskom preusmere na neki drugi izvor energije.[2]

2. TROŠKOVI VAZDUHA POD PRITISKOM I DISTRIBUTIVNI SISTEMI

2.1. Troškovi vazduha pod pritiskom.

Kod većine kompanija troškovi vazduha pod pritiskom imaju značajno učešće u ukupnim troškovima. Često kompanije plaćaju mnogo više za vazduh pod pritiskom nego što je neophodno. Razlog je što njihovi pneumatski sistemi ne rade na efikasan način. Zbog toga je potrebno sprovesti studiju o potrošnji vazduha pod pritiskom, koja bi obuhvatila određivanje potrebne količine i zahtevani kvalitet vazduha pod pritiskom kao i definisala ostale zahteve za funkcionisanje opreme kako bi se omogućilo podešavanje sistema za optimalni rad. Studije ukazuju na to da se prosečno od 30% do 35% proizvedenog vazduha pod pritiskom izgubi curenjem [1]. Curenje vazduha pod pritiskom indirektno dovodi do dodatnih. Variranje pritiska može dovesti neefikanog rada pneumatskih alata i do zastoja u proizvodnom procesu. Može doći i do skraćenog životnog veka opreme (čime se povećavaju investicioni troškovi) i učestalih postupaka održavanja.

Kao opšte pravilo trebalo bi usvojiti da se pri nabavci opreme umesto poređenja cene investicije, porede troškovi životnog ciklusa opreme u kome je nabavna cena samo jedan, najčše manji deo ukupnih troškova, kako je prikazano na slici 2.1. [2].

2.2. Sistem za distribuciju vazduha pod pritiskom

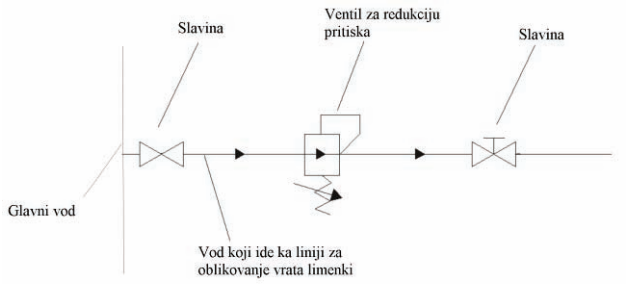
Sistem za distribuciju vazduha pod pritiskom vrši integraciju svih komponenti sistema. Sistemski pristup projektovanju je veoma bitan jer i pored najboljih

komponeti sistem neće ispravno raditi ako komponente nisu dobro postavljene i povezane i ako je distribucija vazduha pod pritiskom neodgovarajuća.
Cevovodi su složeni sistemi za transport fluida sastavljeni od strujnih cevi i cevne armature, koji mogu biti spojeni na različite načine. Spajanje cevovoda sa kompresorom i ostalim pratećim uređajima kao i na mestima gde se zahteva rastavljanje vrši se prirubnicama. Za razliku od hidrauličnih cevovoda pneumatski cevovodi ne zahtevaju povratni vod.

3.TEHNIČKI OPIS PROBLEMA I MERENJE

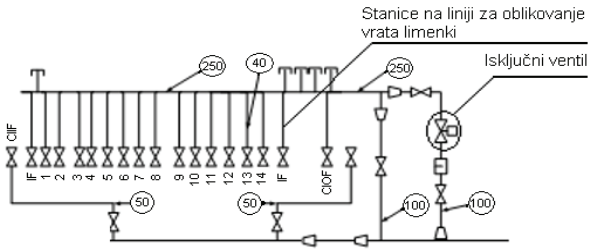
3.1 Tehnički opis distributivnog sistema i linije za oblikovanje vrata limenki

Distribucija vazduha pod pritiskom za celu fabriku Ball Packaging Euro Belgrade Ltd odvija se preko glavnog voda prstenastog oblika. Od glavnog prstenastog voda odvajaju se ogranci za različite podsisteme vazduha pod prtiskom. Ovakao projektovan distributivni sistem obezbeđuje ravnomerno snabdevanje svih potrošača vazduhom pod pritiskom. Pritisak vazduha u glavnom vodu iznosi 7-7,8 bara. Različiti podsistemi vazduha pod pritiskom u Ball Packaging Europe Belgrade Ltd rade na različitim radnim pritiscima. Stoga je neophodno svakom podsistemu vazduha pod pritiskom obezbediti odgovarajući radni pritisak.
Jedan od ogranaka sistema vazduha pod pritiskom, predstavlja linija za oblikovanje vrata limenki. Neophodan i dovoljan pritisak za rad linije za oblikovanje limenki je 2-2,5 bara. Radi uštede, vazduh pod pritiskom koji ide ka liniji za oblikovanje vrata limenki, se redukuje sa 7-7,8 bara (koliki je pritisak u glavnom vodu) na 2-2,5 bara. Grana koja od glavnog voda ide ka liniji za oblikovanje vrata limenki sadrži redom ventil, ventil za redukciju pritiska i isključni ventil, a simbolički je prikazana na slici 3.1.



Slika 3.1 *Simbolički prikaz voda koji ide ka liniji za oblikovanje vrata limenki*

Rešenje da se pritisak redukuje sa 7-7.8 bar-a na 2-2.5 bar-a predstavlja svakako uštedu u potrošnji vazduha pod pritiskom. Ipak vazduh treba proizvoditi sa najmanjim mogućim pritiskom koji zadovoljava potrebe potrošača. Uzimanjem u obzir potrošnje vazduha pod pritiskom za rad Nekera i planirana proširenja proizvodnog kapaciteta, bilo je neophodno projektovati sitem za distribuciju nižeg pritiska.
Nedostaci postojećeg sistema za snabdevanje vazduhom pod pritiskom na liniji za oblikovanje vrata limenki, ogledaju se prilikom prekidanja rada linije. Prilikom prestanka rada na liniji za oblikovanje vrata limenki zatvara se isključni ventil, koji se nalazi između glavnog voda i snabdevajuće cevi, slika 3.2. Sav vazduh pod pritiskom u sistemu za snabdevanje linije za oblikovanje vrata limenki se ispušta u atmosferu. Ni na jednoj od 17 cevi koje snabdevaju stanice linije za oblikovanje vrata limenke ne postoji ventil koji bi zaustavio isticanje vazduha pod pritiskom, usled smanjenog protoka limenki i usled uzimanja uzoraka radi kontrole. Kada se uzme u obzir da linija za oblikovanje vrata limenki u toku 24 h staje (i pokreće se) oko 60 puta, izračunato je da se iz distributivnog sistema u toku 24h oslobodi 163,8 Nm^3/dan, Dalje, kada se u proračun uzme broj radnih dana u godini, koji iznosi 335 dana (rade svakog dana u godini 24 h dnevno, sa tim što jednom godišnje imaju remont sistema koji traje 30 dana), dobijamo vrednost zapremine na godišnjem nivou od 54732,3 Nm^3/dan. Kako proizvodnja jednog normalnog metra kubnog vazduha košta 2,694 din/Nm^3 to bi iznosilo 147448,81 din/god, odnosno oko 1540 € uštede godišnje, što nije zanemarljiva vrednost.



Slika 3.2 *Stanice na liniji za oblikovanje limenki*

3.2 Merenje i oprema

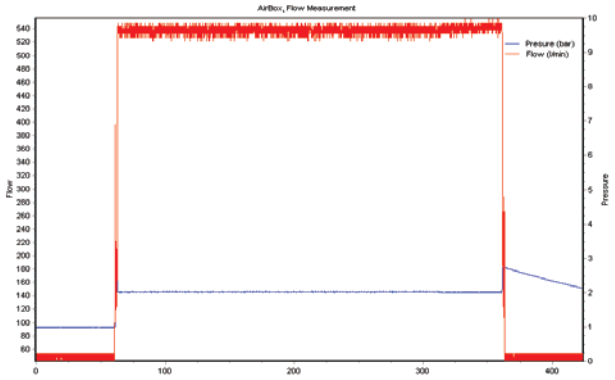
U cilju postavljanja odgovarajućih ventila na cevi za snabdevanje 17 stanica na liniji za oblikovanje vrata limenki izvršeno je merenje protoka vazduha pod pritiskom kroz cevi. Merenje protoka vazduha pod pritiskom izvršeno je pomoću uređaja AirBox, nemačke kompanije Festo. Dovod vazduha ka jednoj od stanica na liniji za oblikovanje vrata limenki je prekinuta i na to mesto je umetnut AirBOX, tako da vazduh na svom putu ka jednoj od stanica na liniji za oblikovanje vrata limenki prolazi kroz uređaj ne prekidajući time dovod vazduha pod pritiskom, kako je prikazano na slici 4.4. Sa jedne strane dovedeno je crevo pod pritiskom na ulazu u uređaj, dok je druga strana spojena sa sistemom kako bi se zatvorio krug. Puštanjem vazduha u sistem, vazduh prolazi kroz uređaj koji posedaje ugrađeni merač protoka. Na taj način beleži se potrošnja vazduha pod pritiskom.



Slika3.3 *Priključci AirBoxa na jednu od stanica*

Nakon priključivanja Airbox-a na dovod vazduha, započeto je sa merenjem protoka. Dobijeni rezultati predstavljeni su na diagramu na slici 3.4.

Sa dijagrama se može očitati protok i videti pritsak u jednoj od stanica za oblikovanje limenki. Dobijeni rezultati predstavljani su i u tabeli 3.1 i matematičkim putem određena je potrošnja vazduha pod pritiskom na jednoj od stanici za oblikovanje limenki, odnosno prosečna potrošnja vazduha pod pritiskom predstavlja aritmetičku sredinu svih rezultata marenja sa slike 3.4



Slika 3.4 *Izmereni protok i pritisak na jednoj od stanica*

Tabela 3.1 Prikazan je način dobijanja potrošnje vazduha pod pritiskom sa slike 3.4 i tabelarni prikaz rezultata merenja potrošnje vazduha pod pritiskom

vreme t (s)	protok q (l/min)	pritisak (bar)*	$t_i - t_{i-1}$	$q_i + q_{i-1}$	$(t_i - t_{i-1})\cdot(q_i + q_{i-1})$
202,605	538	2,02	0,049	1082	53,018
202,655	538	2,02	0,05	1076	53,8
202,705	533	2,01	0,05	1071	53,55
202,755	549	2,01	0,05	1082	54,1
203,602	527	2,02	0,05	1071	53,55
203,651	538	2,02	0,049	1065	52,185
203,701	533	2,03	0,05	1071	53,55
203,751	533	2,03	0,05	1066	53,3
203,801	544	2,02	0,05	1077	53,85
203,851	538	2,02	0,05	1082	54,1
203,9	538	2,02	0,049	1076	52,724
…	…	…	…	…	…
* Apsulutni pritisak			$\Sigma(t_i - t_{i-1})\cdot(q_i + q_{i-1})$		322822,891

Ukupna potrošnja vazduha za jednu od stanica se izračunava po sledećoj formuli:

$$Q_l = \frac{1}{2}\sum_{i=1}^{n}(t_i - t_{i-1})\cdot(q_i + q_{i-1}) = \frac{322822,891}{2*60} = 2690,19l$$

4. OPTIMIZACIJA DISTRIBUTIVNOG SISTEMA I LINIJE ZA OBLIKOVANJE VRATA LIMENKI

4.1 Izbor i proračun glavnog elemenata glavnog voda distributivnog sistema nižeg pritiska

U cilju povećanja energetske efikasnosti sistema u Ball Packaging Europe Belgrade Ltd pored postojećeg projektovan je još jedan distributivni sistem nižeg pritiska. Postojeći distributivni sistem snabdevaće pneumatske sisteme za čiji rad je potreban pritisak između 3 i 6 bar-a. Za podsisteme koji se snabdevaju sa distributivnog sistema višeg pritiska, a potreban im je radni pritisak manji od 6 bara, biće priključeni na vod koji ima ugrađen ventil za redukciju pritiska, kao što je prikazano na slici 3.1. Novo projektovani distributivni sistem nižeg pritiska snabdevaće pneumatske sisteme za čiji rad nije potreban pritisak veći od 3 bar-a, među kojima je i linija za oblikovanje vrata limenki, zbog koga je distibutivan sistem nižeg pritiska prvensteno i projektovan.

Za izbor dimenzija cevovoda za razvod vazduha pod pritiskom potrebno je uzeti u obzir protok, dužinu razvoda, dozvoljeni pad pritiska, vrednost radnog pritiska i broj i vrstu prigušenja u cevovodu.

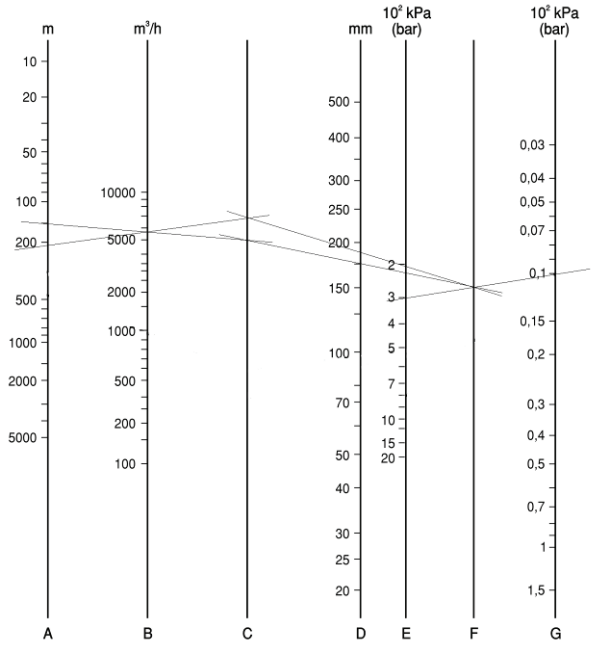
Merodavna vrednost za izbor prečnika glavnog cevovoda niskog pritiska pored postojeće potrošnje treba da uzme u obzir i planirano povećanje potrošnje vazduha pod pritiskom u narednih pet godina.

Za dozvoljeni pad pritiska se u mreži usvaja $\Delta p = 0,1$ bar, a za radni pritisak se u glavnom vodu nižeg pritiska, uzima p = 3 bar, dok se u ostalim vodovima uzima p = 2.5 bar-a.

Gubici koja vazdušna struja trpi pri promenama smera strujanja (na svakom skretanju račvanju, pri prolasku kroz ventile, zasune i druge elemente) se zamenjuju odgovarajućom dužinom prave cevi.

Za proračun prečnika u ovom slučaju biće korišćena grafička metoda pomoću nomograma i to tako da se prvo izračuna prečnik cevi bez prigušenja, a zatim se na osnovu tog prečnika i odgovarajućih dužina koje se dodaju za prigušenja računa stvarni prečnik cevi.

Potrebna dužina glavnog cevovoda niskog pritiska iznosi 140 m. Prečnik cevi novog sistema za distribuciju vazduha pod pritiskom od 3 bar-a, je određen pomoću nomograma sa slike 4.1. i iznosi 200 mm.



Slika 4.1 *Određivanje prečnika novog glavnog voda*

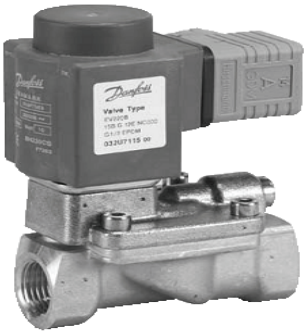
Planirano povećanje potrošnje vazduha pod pritiskom za narednih pet godina zahteva povećan prečnik glavog voda pa je investitor zahtevao da prečnik glavog voda bude 250 mm.

Predloženo rešenje izgeda glavog voda nižeg pritiska, predstavlja otvoreni cevovod koji se prostire od kompresorske stanice do linije za oblikovanje vrata limenki. Otvoreni cevovod iskorišćen je kao najednostavnija forma i zahteva najmanju dužina glavog voda. Veliki nedostatak otvorenog cevovoda predstavlja snabdevanje svakog narednog potrošača (udaljenijeg od izvora pritiska) i zavisi od potrošnje vazduha pod pritiskom prethodnog. Kako je projektovan kao namenski cevovod, gde će linija

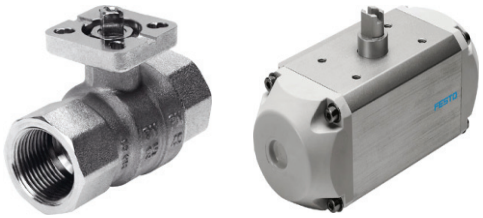
za oblikovanje vrata limenki biti najveći potrošač, ovakav nedostatak neće previše uticati na rad sistema.
Potreba za pritiskom od 7-7,8 bar-a u distributivnom sistemu višeg pritiska, prestala je da postoji projektovanjem novog distributivnog sistema nižeg pritiska i izdvajanjem najvećeg potrošača vazduha pod pritiskom. Takav pritsak od 7-7,8 bar-a u postojećem distributivnom sistemu bio je nephodan da se amortizuju varijacije pritisaka usled pokretanja mašina, pogotovo linije za oblikovanje vrata limenki. Varijacija pritiska, prilikom uključivanja i isključivanja linije za oblikovanje vrata limenki, uticala je na rad Body maker-a. Dovoljan pritisak za rad mašina koje se snabdevaju vazduhom pod pritiskom iz distributivnog sistema višeg pritiska je 6 bar-a.

4.2 Optimizacija potrošnje na liniji za oblikovanje vazduha pod pritiskom

Nakon merenja protoka i pritiska na stanicama linije za oblikovanje vrata limenki, generisano je idejno rešenje za optimizaciju potrošnje tako da se ugrađivanjem ventila zatvore stanice pre prekida rada na liniji za oblikovanje vrata limenki. Time bi vazduh pod pritiskom u toku zastoja ostao u sistemu. Takođe, prilikom svakog njegovog pokretanja ne bi postojale oscilacije pritiska i ne bi se gubilo vreme na uspostavljanje radnog pritiska.
Za ovu funkciju neophodni 2/2 monostabilni razvodnici i kuglasta slavina. Neophodno je da budu električno upravljani. Radiće na naizmeničnu struju od 24V i biće upravljani PLC-om (FEC FC660). Za cevi prečnika 40 mm izabran je razvodnik EV220B slika 4.2, proizvođača Danfoss, a za cevi prečnika 50 mm kuglasta slavina VAPB-2-F-25-F05 sa zakretnim pogonom DRD-4-F-05 slika 4.3, proizvođača Festo.



Slika 4.2 *Razvodnik Danfoss EV220B*



Slika 4.3 *Kuglasta slavina Festo (levo), sa zakretnim pogonom (desno)*

Investicija za nabavku predložene opreme na liniji za oblikovanje vrata limenki, radi povećanja energetske efikasnosti linije, isplatila bi se u naredne četiri godine.

5. ZAKLJUČAK

U radu je prikazana struktura pneumatskog sistema koji, pored povećanja energetske efikasnosti, omogućava bolji rad pneumatskih mašina.
Različiti podsistemi vazduha pod pritiskom mogu da rade na različitim radnim pritiscima. Stoga je neophodno svakom podsistemu vazduha pod pritiskom obezbediti odgovarajući radni pritisak. Na primeru, projektovanja distributivnog sistema nižeg pritiska u kompaniji Ball Packaging Europe Belgrade Ltd i povezivanje linije za oblikovanje vrata limenki i drugih podsistema na distributivni sistem nižeg vazduha pritiska, smanjilo se opterećenje u distributivnom sistemu višeg pritiska. Ovim se postiglo da se vazduh pod pritiskom proizvodi sa minimalnim potrebnim pritiskom koji zadovoljava potrošače.
Pored toga, najveći neregulisani potrošač u Ball Packaging Europe Belgrade Ltd je linija za oblikvanje vrata limenki. Postavljanjem odgovarajućih ventila ispred potrošača, zaustavljeno je direktno ispuštanje vazduha pod pritiskom u atmosferu. Takođe su u velikoj meri neutralisane varijacije pritiska, usled čestog pokretanja i zaustavljanja linije za oblikovanje vrata limenki, što je negativno uticalo na rad ostalih pneumatskim mašina.

5. LITERATURA

[[1] Šešlija, D.: *Proizvodnja, priprema i distribucija vazduha pod pritiskom, IKOS, Novi Sad, 2002*
[2] Šešlija, D.: *Povećanje energetske efikasnosti pneumatskih sistema*.
[3] Stanić A.: *Distribucija vazduha pod pritiskom u savremenim industrijskim sistemima*
[4] Dudić S., Ignjatović I., Šešlija D.:*Kvantifikacija gubitaka na curenju vazduha pod pritiskom u fabrici Knjaz Miloš*

Kratka biografija:



Kosta Žarkov rođen je u Vrbasu 1983 god. Apsolvent je na Fakultetu tehničkih nauka, odsek mehatronika, smer za mehatoniku, robotiku i automatizaciju

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 658.52.011.56

MODERNIZACIJA KOLONE I SISTEMA VAZDUHA POD PRITISKOM U RAFINERIJI

MODERNIZATION OF COLUMN AND AIR PREASURE SYSTEM IN RAFINERY

Milan Vasiljev, Dragan Šešlija, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

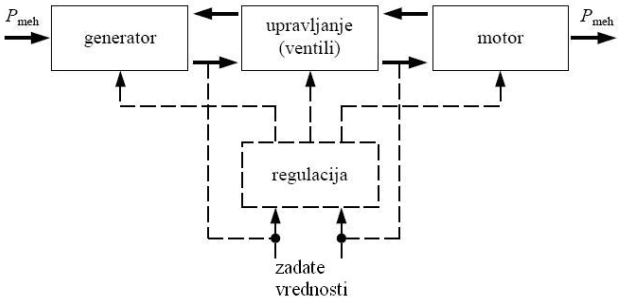
Kratak sadržaj – *Implementacijom pneumatike u neki industrijski sistem, može se potpuno ili delimično vršiti upravljanje tim sistemima. Da bi se ostvarilo upravljanje u pneumatskim sistemima, neophodno je vazduh prethodno pripremiti, jer se vazduh (ili neki drugi gas) koristi kao radni medij. Priprema se sastoji iz sušenja vazduha sušačima i filtriranjem vazduha od čvrstih čestica i ulja. U ovom radu predstavljena je automatizacija rada kolone za upravljanje prerade propana, realizacija uvođenja sušača vazduha u sistem za pripremu vazduha pod pritiskom pomoću kojeg mi upravljamo izvršnim organima unutar te kolone. Izvršeno je merenje klase kvaliteta vazduha, pre i posle uvođenja sušača u sistem za proizvodnju i pripremu vazduha pod pritiskom u jednoj od rafinerija za preradu gasa na području Vojvodine. Prikazani su rezultati dobijeni merenjem kvaliteta vazduha pre i posle adaptacije sistema i izvršena je provera curenja vazduha nakon završetka uvođenja sušača u sistem.*

Abstract – *With the implementation of the pneumatics in an industrial system it can be wholly or partially controedl. In order to achieve control in pneumatic systems, previously, it is necessary to prepare the air, because the air (or other gas) is used as the working fluid. The preparation consists of air drying with air dryers and removing solid particles and oil from the air. This paper presents automation of the column for control of propane processing and installation of air dryer in compressed air system and results of measuring of air quality before and after installation of air dryer. It shows testing results of air leak after putting into operation system in one refinery for processing gas in the region of Vojvodina.*

Ključne reči: *Sistem vazduha pod pritiskom, Sušač vazduha, Klasa vazduha, Curenje*

1. UVOD

Zadatak pneumatskog sistema uključuje pretvaranje, prenos i upravljanje energijom. Pneumatika kao radni medij koristi stišljivi fluid – gas, najčešće vazduh. Velika prednost pneumatskih sistema je nezapaljivost radnog medija, pa se često koriste u sistemima u kojima vlada visoka opasnost od izbijanja požara i eksplozija. U pneumatskim prenosnim sistemima povezani su kompresori, kao generatori vazduha pod pritiskom, izvršni elementi (cilindri, fluidni motori itd.) i upravljački elementi u zatvoreni sistem, u kojem radni fluid cirkuliše kroz zatvoreni sistemi cevi i creva prenoseći energiju (slika 1).



Slika 1. *Blok dijagram prenosa energije unutar pneumatskog sistema*

Da bi atmosferski vazduh mogao da postrane prenosnik energije zahvaljujući sposobnosti sabijanja i ekspanzije, potrebno ga je dovesti na viši potencijal sabijanja Takav sabijeni vazduh koristi se za pogon pneumatskih motora, pneumatskog alata i drugih pneumatskih komponenti ili u tehnološkim procesima. Vazduh je smeša gasova bez mirisa i ukusa i u prirodi je pun čestica kao što su prašina, pesak, čađ kao i vodene pare u određenoj količini, u zavisnosti od stanja u kome se nalazi. Količina vodene pare i zagađenost vazduha presudno utiču na proces sabijanja i na kvalitet vazduha koji isporučuje kompresor [1]. Kada se vazduh sabije, koncentracija vlage i drugih nečistoća se povećava. Kada bi se dozvolilo da ostanu u sistemu, ta korozivna mešavina bi štetno uticala na pneumatsku opremu, izazivajući neželjene zastoje u proizvodnji, kvarenje krajnjih proizvoda procesa u kojima se koristi i smanjenje radnog veka opreme. Da bi se sprečio štetan uticaj vlage, čvrstih cestica i ulja na sistem, u sistemu za pripremu vazduha postoje sušači vazduha za sušenje i filteri za odstranjivanje čestica i ulja iz vazduha.

2. AUTOMATIZACIJA KOLONE

Automatizacija upravljanja jedne od kolona i merenje određenih parametara rađeno je u jednoj od rafinerija na području Vojvodine. Za regulaciju protoka, pritiska, temperature i nivoa tečnosti u sudovima koriste se regulaciona kola sačinjena od grupe međusobno povezanih elemenata. Automatizacija regulacionih kola je veoma odgovoran zadatak, jer su kola delom pneumatska, delom elektropneumatska i mora se dovesti struja do elektropneumatskih elemenata unutar eksplozivne zone, kako bi ona funkcionisala. Tokom rada na projektu u rafineriji izvršena je automatizacija (jedne od 9 kolona) T-3 kolone. Radilo se na daljinskom upravljanju iniciranja propana u toj koloni.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je bio dr Dragan Šešlija, red.prof.

2.1. Stara regulacija kolone

Prethodna realizacija upravljanja protoka propana u T-3 koloni bila je neautomatizovana i sastojala od jednog indikatora protoka (*flow indikator)*. Indikatora protoka je bio sačinjen od čelične epruvete sa izrezom jednog dela epruvete, koja je bila staklene površi kako bi se moglo očitavati i podešavati željena vrednost protoka propana kroz cevovod u koloni, i lebdećeg plovka unutar epruvete. Indikatora protoka je bio povezan na ručno podešavajući ventil i tim ventilom se zadavala željena pozicija lebdećeg plovka, a samim tim i protok propana.

2.1. Modernizacija kolone

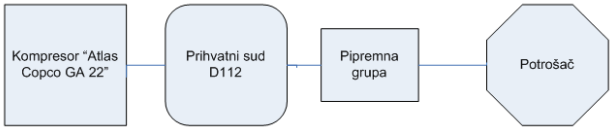
Po završetku rada na projektu u potpunosti je automatizovana regulacija protoka propana u T-3 koloni rafinerije. Potreba za ovakvim projektom se ogleda u funkcionalnoj efikasnosti same kolone kao i olakšanog pregleda, posmatranja i upravljanja iste. Upravljački krug se sastoji iz transmitera protoka proizvođača *Vortex Foxboro,* koji je montiran na samom vodu cevi u T-3 koloni. Transmiter se napaja naponom od 4 do 20 mA iz kontrolne sobe preko modula za napajanje koji je ujedno i pojačavač povratnog signala. Sačinjen je od kućišta koja ispunjava *Ex* standarde eksplozivne zaštite u nultoj zoni opasnosti i „viljuške" koja je u stvari senzorski deo transmitera. „Viljuška" je postavljena u prekidu cevi kojom se provodi propan tako da propan prvo dolazi u kontakt sa jednim krakom od dva a zatim do drugog kraka, i preko razlike pritiska, pre dotoka gasa do prvog kraka i pritiska izmedju dva kraka, meri se protok propana kroz cevovod. U ovoj realizaciji transmiter je fizički odvojen od indikatora *slika 2,* pa se signal od transmitera šalje putem naponskog signala kablovima do indikatora koji se nalazi odmah pored regulacionog ventila u postrojenju (eksplozivna zona). Pored toga signal se još šalje u kontrolnu sobu gde se nalazi još jedan indikator kako bi se moglo dvojako pratiti kontrola protoka propana što je ujedno i glavni motiv za uvođenje ovakvog regulacionog kola. Ovakvim slanjem i prikupljanjem podataka omogućeno je njihovo očitavanje napolju u pogonu kao i iz kontrolne sobe. Signal koji je poslat ka kontrolnoj sobi ide ka pojačavaču, a iz pojačavača do regulatora i pisača. Regulatorom *Foxbor T630* se pravi diferencijalna razlika između željenog i traženog pritiska i kao takav signal se vraća preko drugog pojačavača nazad u postrojenje do I/P pretvarača. Tu se naponski signal pretvara u signal pritiska kojim se upravlja radom regulacionog servo ventila (BS@B 70-13-550-10 radni pritisak od 0 do 1 bar) i time zatvara krug sa povratnom spregom.

3. SISTEM ZA PRIPREMU VAZDUHA POD PRITISKOM

Sistem za proizvodnju i pripremu vazduha pod pritiskom, u zavisnosti od zahteva sistema za koji je potreban taj vazduh, mogu sadržati sledeće elemente: kompresor, filtre, zauljivače, rezervoare za skladištenje vazduha, sušač vazduha, razvodne ventile i upravljačke cilindre itd.

3.1. Postojeći sistem

Sistem za pripremu vazduha pod pritiskom se, pre realizacije projekta, sastojao od sledećih elemenata prikazanih na slici 2.



Slika 2. *Blok dijagram postojećeg sistema za pripremu vazduha*

Automatizacijom kolone javila se potreba za uvođenjem sušača vazduha u sistem za pripremu i proizvodnji vazduha pod pritiskom. Naime, vazduh koji je puštan u sistem ka izvršnim organima nije bio dovoljno suv i predstavljao je problem za rad adaptiranog upravljačko - regulacionog sistema.

Takođe, vršeno je i utvrđivanje ulja u vazduhu pod pritiskom, u cilju provere da li postojeći sistem obezbeđuje potreban nivo kvaliteta vaduha pod pritiskom. Merenjem količine ulja, izmerene uređajem Air box, u vazduhu pod pritiskom, dobijeni su zadovoljavajući rezultati u prisutnosti ulja u vazduhu od 1 mg/m3.

3.2. Poboljšanje sistema i merenja karakteristika vazduha pre i posle njegove adaptacije

Na početku izvođenja projekta, poboljšanja kvaliteta pripremnog vazduha pod pritiskom, neophodno je bilo utvrditi klasu kvaliteta vazduha koji se dobijao iz prvobitnog sistema za pripremu vazduha. Pripremnom grupom, koju čine filtri za uklanjanje čvrstih čestica iz vazduha, ostvarena je zadovoljavajuća vrednost od 500 čvrstih čestica veličine od 1-5 µm po kubnom metru vazduha. Da bi se ustanovila tačka rose takvog vazduha i količina ulja neophodno je bilo merenje sa Air box mernim uređajem (slika 4).



Slika 3. *"Air box" merni uređaj*

Pomoću ovog uređaja određuju se karakteristične vrednosti vode i ulja sadržane u tom vazduhu. Protok i pritisak vazduha mere se senzorima koji se nalaze unutar mernog uređaja. Količina vode i ulja u vazduhu se mere tako što se na konektor 5 dovodi vazduh i unutrašnjim kanalima odvodi na izlaze 6, 7, 8 (slika 3). Na izlaze 6, 7 i 8 postavljaju se indikatorske cevčice za merenje vlage i ulja u vazduhu. Cevčice su staklenog valjkastog oblika i

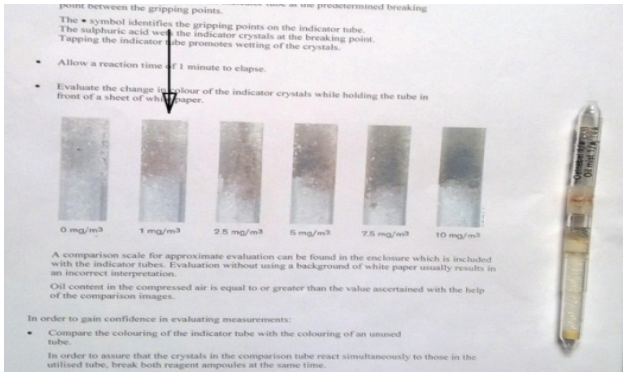
unutar njih se nalaze katalizatori (u kristalnom obliku) koji menjaju boju prilikom strujanja vazduha kroz njih. Vazduh prolazi od otvora 5 unutrašnjim kanalima ka otvorima 6, 7, i 8, zatim prolazi od donjeg kraja cevčice ka gornjem, napolje u atmosferu i pri prolasku kroz cevčice vazduh dolazi u reakciju sa katalizatorima, dovodeći do menjanja boje katalizatora.

Pre ugradnje sušača, uređaj *Air Box* je priključen odmah na izlazu iz kompresora (slika 4). Merenjem je ustanovljeno da količina vlage u vazduhu izlazi van opsega naših mernih mogućnosti. Nakon merenja katalizator je promenio boju duž cele cevčice i nemoguće je bilo očitati najveću vrednost jer ona nije relevantan podatak, moguće bi bilo da je cevčica sa više podeoka i duža ali to već svakako dokazuje da je vazduh isuviše vlažan.



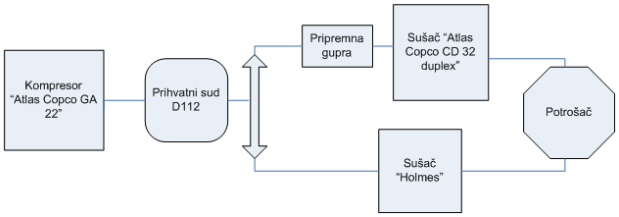
Slika 4. *Prikaz merenja vlage u vazduhu i povezivanja na sistem pre uvođenja sušača u sistem*

Tako vlažan vazduh nije zadovoljavao kriterijume sistema jer se prilikom podizanja pritiska vazduha kompresorom u sistemu izdvajala veća količina vode koja je imala štetan uticaj na elemente u pneumatskom sistemu. Istim merenjem pokazano je da je količina ulja u vazduhu bila zadovoljavajuća. Katalizator cevčice je bio boje koja je najviše odgovaralala drugoj u nizu od šest mogućih boja kao što se vidi sa slike 5.



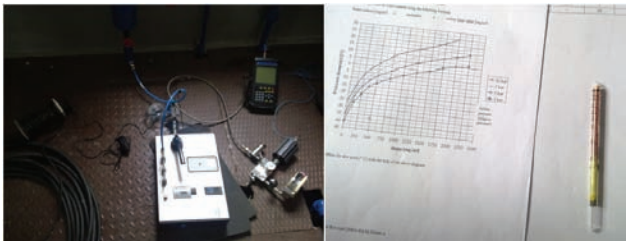
Slika 5. *Prikaz cevčice kojom je izmerena količina ulja u sistemu*

Prethodnim merenjima se utvrdilo da je količina ulja u vazduhu za sistem bila zadovoljavajuća i da je jedino vrednost za količinu vode u vazduhu bila kritično iznad dozvoljene. Kako bi se ovaj problem rešio, u sistem se uvode dva sušača. Novonastali sistem za proizvodnju i pripremu vazduha pod pritiskom, neophodnog za rad pogona, sada se sastoji od kompresora, rezervoara za skladištenje vazduha pod pritiskom D-112, sušača vazduha (jedan aktivan i jedan u rezervi), pripremne grupe, kao i sistema za distribuciju vazduha, odnosno cevovoda, koji pripremljen vazduh dalje odvodi ka izvršnim elementima (slika 6).



Slika 6**.** *Blok dijagram adaptiranog sistema za pripremu vazduha*

Uvođenjem sušača u sistem ponovo je izvršeno merenje vlage vaduha povezivanjem mernog uređaja posle sušača (slika 7).



Slika 7. *Prikaz merenja vlage u vazduhu i povezivanja na sistem posle uvođenja sušača u sistem*

Ponavljanjem postupka merenja vlage u vazduhu, u adaptiranom sistemu, dobijeni su rezultati sa ostvarenom tačkom rose od -23 ^{o}C.

4. CURENJA VAZDUHA

Detekcija curenja ima veoma veliki značaj za ispravno funkcionisanje sistema vazduha pod pritiskom koji se ogleda u smanjenju vremena rada opreme a time i produženje životnog veka opreme, izbegavanje nepotrebnog kapaciteta kompresora, izbegavanje nepotrebnog vremena i broja akcija servisiranja, a samim tim i nepotrebni troškovi održavanja opreme koja pravi veći broj ciklusa i mnogo duže vreme rada itd. Pored ekonomskog, smanjenje gubitaka usled curenja vazduha pod pritiskom značajno je i sa stanovišta ekologije i zaštite životne sredine. Smanjenjem gubitaka vazduha smanjuje se i potreba za njegovom proizvodnjom, a samim tim se smanjuje i emisija CO_2 i drugih štetnih materija u atmosferu.

Detekcija curenja ima veoma veliki značaj za ispravno funkcionisanje sistema vazduha pod pritiskom koji se ogleda u sledećem:

- smanjenje vremena rada opreme a time i produženje životnog veka opreme,
- izbegavanje nepotrebnog kapaciteta kompresora,
- izbegavanje fluktuacija pritiska koje loše utiču na rad alata i efikasnost opreme što može da dovede čak i do zastoja u proizvodnji,
- izbegavanje nepotrebnog vremena i broja akcija servisiranja, a samim tim i nepotrebni troškovi održavanja opreme koja pravi veći broj ciklusa i mnogo duže vreme rada,
- smanjivanje vezanih sredstava u vidu rezervnih delova,
- bolje iskorišćenje radnika u održavanju.

• omogućava da se ne prekorači dozvoljeni nivo buke na radnom mestu.

Ultrazvučni detektor *Ultraprobe 100* precizno detektuje pojavu curenja i mehaničkih oštećenja primenom ultrazvučne tehnologe. Komplet ultrazvučnog detektora Ultraprobe 100 je prikazan na slici 8.



1. kućište pištolja
2. modul za skeniranje
3. kontaktni modul
4. slušalice
5. GTT - 1 Generator treperećeg tona

a) Izgled kompleta *b) Sastavni delovi*

Slika 8. Ultrazvučni detektor curenja Ultraprobe 100

Merenje nivoa curenja vazduha pod pritiskom, izražavaju su preko inteziteta zvuka (dB). Zbog relativno nepreciznog mernog instrumenta (3dB = 1 LED dioda) izmerene vrednosti su orijentacione [2]. One su daju zbog praktičnosti, i treba ih uzeti samo kao smernice. Nakon završenog projekta uvođenja sušača u sistem za proizvodnju i pripremu vazduha pod pritiskom izvršeno je merenje curenja ultrazvučnom detekcijom. Merenja parametara curenja vršeno je, unutar prostorije u kojoj se nalazi ceo sistem za pripremu vazduha, na čeličnim cevima prečnika od 2,5 cm do 5 cm (slika 9).



Slika 9. *Spojnice gde je izmereno curenje vazduha*

Ispitivanjem ultrazvučnim detektorom provereni su svi spojevi cevi, koje su ugrađene posle uvođenja sušača u sistem za pripremu i proizvodnju vazduha pod pritiskom. Zbog nejednakosti u prečniku cevi, koje su prvobitno bile u sistemu i prečnika priključnih cevi sušača, postavljen je veći broj priključnih kolena i spojnica od planiranog. Povećanjem broja spojeva se povećao i broj mesta na kojima je vazduh mogao da curi napolje iz cevi. Na osetljivosti uređaja od 20 dB detektovane su vrednosti curenja spojeva cevi, pri radnom pritisku od 5 bar, koje se mogu videti u tabeli 1.

Tabela 1 . Prikaz vrednosti curenja

Trenutni režim rada komora pri detekciji curenja	**Vrednost curenja [dB] (levi prilključci)**	**Vrednost curenja [dB] (desni prilključci)**
rad leve komore	21	18
rad desne komore	27	27
regeneracija leve komore	15	18
regeneracija desne komore	21	24

Sprečavanje curenja vazduha može se postići razdvajanjem svih spojeva cevi i njihovim boljim zaptivanjem ili postavljanjem dodatnih spojnica preko postojećih.

5. ZAKLJUČAK

Uvođenjem elektronskog indikatora i emitera, pri modernizaciji kolone, ostvarena ja potpuna autonomija u upravljanju propana u toj koloni, a time i efikasnije funkcionisanje ostalih kolona u rafineriji koje su međusobno zavisne.

Automatizacija kolone dovelo je do potrebe za čistijim i suvljim vazduhom pod pritiskom. Uvođenjem sušača, u sistem za njegovu pripremu, došlo se do dobijanja odličnih rezultata što se tiče kvaliteta vazduha, kao i bezbednijeg upravljanja automatizovanom kolonom. Pravci daljeg rada odnose se na:

- smanjenje curenja vazduha iz sistema,
- praćenje funkcionalnosti adaptiranog sistema.

6. LITERATURA

[1] Milosavljević, M.: Vazduh kao prenosnik energije, RS, 2010.

[2] Dudić S., Šešlija D., Ignjatović I., Blagojević V., Stojiljković M.: Leakage quantification of compressed air using ultrasound and infrared thermography, Novi Sad, 2010.

Kratka biografija:



Milan Vasiljev rođen je u Zrenjaninu 1986. godine. Bachelor rad odbranio je na Fakultetu tehničkih nauka 3.3.2011., pod nazivom Sušači vazduha pod pritiskom u pneumatskim sistemima, iz oblasti Mehatronike – Mehatronika, robotika i automatizacija. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Mehatronike – Mehatronika, robotika i automatizacija odbranio je 2011.god.



Dragan Šešlija rođen je 12.09.1955. godine u Kikindi, SAP Vojvodina, SR Srbija. Doktorirao je 1997. na Fakultetu tehničkih nauka iz oblasti robotike, a od 2007. je redovni profesor na istom fakultetu. Oblast interesovanja: sistemi vazduha pod pritiskom, automatizacija procesa rada, robotika, itd.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007.5

SISTEM ZA DALJINSKI NADZOR AUTOMOBILA

SYSTEM FOR REMOTE VEHICLE MONITORING

Predrag Dimković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – *Predmet ovog rada jeste razvoj i praktična realizacija uređaja namenjenog za daljinski nadzor i upravljanje automobilom, dojavu alarma, očitavanje GPS lokacije automobila i kontrolu pristupa putem GSM mreže. Mogućnost kontrolisanja i upravljanja automobilom se vrši pomoću Android Mobile aplikacije, dizajnirane konkretno za ovaj projekat.*

Abstract – *The subject of this ppaper is the development and practical implementation of devices intended for remote monitoring and control, car alarm, car GPS location readings and Controllers to access the GSM network. Ability to control and manage the car is done with Android Mobile applications designed specifically for this project.*

Ključne reči: *Automatika, mikrokontroler, upravljanje, GSM mreža, mobilna aplikacija*

1. UVOD

Pojavom GSM tehnologije došlo je do značajnog povećanja brzine prenosa informacija i revolucije u načinu komunikacije. Niko nije ni pomišljao kakvu popularnost će dostići mobilne komunikacije i mobilni telefoni. Paralelno sa razvojem mobilne telefonije, razvijale su se i ideje o njenoj upotrebi u industrijskim aplikacijama i komunikaciji između čoveka i mašine, kao i između mašine i mašine. U industriji je ovakva vrsta aplikacija poznata kao M2M, odnosno „man to machine, machine to machine".

Pre pojave GSM tehnologije postojala je bežična komunikacija između mašine i mašine, kao i mogućnost bežičnog upravljanja, ali je udaljenost, sa koje to moglo obavljati, bila ograničena. Pojavom GSM tehnologije, udaljenost praktično postaje neograničena. Sada je moguće daljinsko upravljanje i nadzor sa bilo koje tačke na Zemlji.

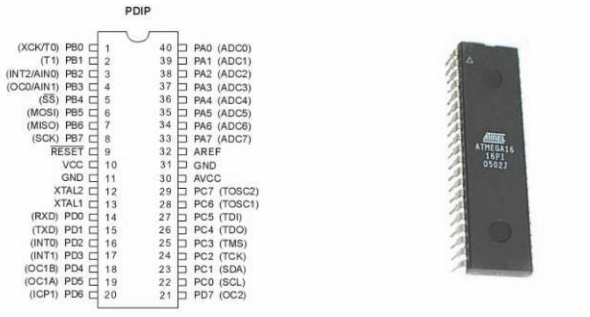
GSM tehnologija donosi još prednosti kada je u pitanju daljinsko upravljanje. Nekada je veliki problem predstavljala upotreba daljinskih upravljača u slučaju kada ima mnogo korisnika. U tom slučaju, svaki korisnik mora da ima namenski daljinski upravljač što ponekad predstavlja veliki trošak. GSM tehnologija omogućava da se lični mobilni telefon koristi kao univerzalni daljinski upravljač, identifikacioni uređaj, sredstvo za plaćanje itd. Danas je skoro nezamislivo da neko nema mobilni telefon.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je dr Stevan Stankovski, red. prof.

Osim bežičnog upravljanja, GSM tehnologija omogućava i daljinski monitoring različitih procesa, senzora, uređaja mašina itd. Na ovaj način moguće je pravovremeno uočiti zastoj ili nepravilnost u radu neke mašine ili pogona, kao i daljinsko otklanjanje problema. Uređaji za monitoring posebno nalaze primenu u aplikacijama gde pravovremena dojava nekog događaja može sprečiti nastanak materijalne štete, npr. provala, krađa, zastoj u radu nekog pogona ili mašine.

2. MIKROKONTROLER ATmega16

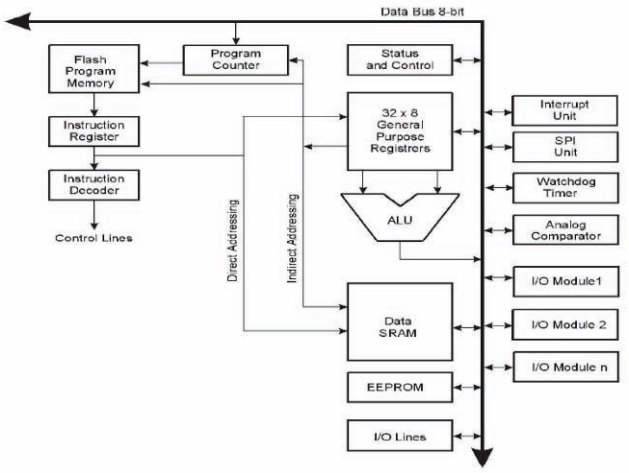
Upravljačka jedinica je modul koji spaja sve komponente u jednu celinu. Pomoću ovog modula se ostavaruje veza sa GSM modemom, tasterima, mobilnom aplikacijom i ostalim ulazima i izlazima. Mozak modula je mikrokontroler ATMega 16. To je osmobitni mikrokontroler proizvođača Atmel. U osnovi se nalazi moderni 8 – bitni RISC mikroprocesor koji sadrži 32 registra opšte namene, 16 K flash memorije koja služi za čuvanje programa koji procesor izvršava, 512 bajta EEPROM memorije čija je namena trajno čuvanje podataka (i bez napajanja) i 1 K interne SRAM memorije namenjene za podatke i promjenjive koje se koriste u programu. Odlikuju ga odlične performanse kao što su velika brzina izvršavanja instrukcija, do 16 MIPS sa oscilatorom od 16 MHz, što je dosta bolje nego kod standardnih 8 – bitnih mikrokontrolera. Memorijske performanse su takođe poboljšane kombinovanim korišćenjem flash i EEPROM memorije. Na slici 1. prikazan je izgled mikrontrolera u PDIP kućištu, kao i prikaz svih izvoda kontrolera.



Slika 1. *ATMega 16 u PDIP kućišti sa prikazom svih pinova*

ATMega familija mikrokontrolera ima jedinstvenu memoriju sa mogućnošću samoprogramiranja, "*sefl – programing memory",* kao i mogućnost čitanja iz memorije tokom upisa, „*Read while write"*. Mikrokontroleri iz ove familije se mogu naći u varijantama sa 8 – 128 K flash memorije, do 4 K EEPROM – a i do 4 K SRAM – a i dolaze u 28 – 64 pinskom pakovanju.

Ovi mikrokontroleri su usmereni na speciojalizovane segmente tržišta i često se koriste kao komponente u prenosivim uređajima, bešičnoj komunikacionoj opremi i sigurnosnim sistemima. Na slici 2. se nalazi na blok šema mikrokontrolera .



Slika 2. *Blok šema mikrokontrolera*

3. GSM modem – Telit GM862-GPS

Telit GM862 – GPS modem namenjen je za korišćenje u bežičnim industrijskim aplikacijama kako za monitoring i prenos komandi, tako i za očitavanje GPS lokacije posmatranog predmeta, objekta.. Malih je dimenzija, male težine i lak za integraciju. Modul je kvalitetnog robusnog dizajna, male potrošnje, proširenog opsega radne temperature i poseduje integrisani držač SIM kartice. Modem poseduje časovnik realnog vremena i alarm sa odgovarajućim izlazom.

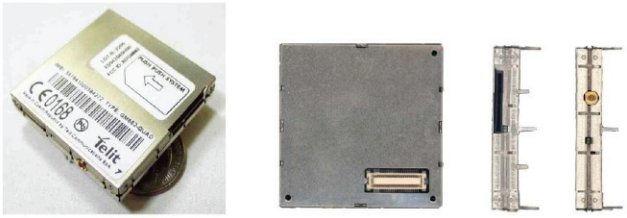
Gabaritne dimenzije modema su:

- dužina: 43,9 mm
- širina: 43,9 mm
- debljina: 6,9 mm
- zapremina: ~ 13 cm^3
- težina: 23 g sa oklopom

Na slici 3. prikazan je fizički izgled modema. Temperaturni opsezi u kojima je garantovan rad GSM modema su:

- funkcionalan rad u temperaturnom opsegu od -10 °C do +55 °C
- funkcionala rad u ekstremnim uslovoma u temperaturnom opsegu od -20 °C do +70 °C
- temperatura skladištenja od -30 °C do +85 °C

Prema specifikaciji GSM modem može da radi na dual-band GSM frekvencijama koje iznose 900 MHz ili 1800 MHz. U zavisnosti za kakvu aplikaciju se koristi modem, prema tome se bira koja GSM antena se povezuje na njega.



Slika 3 *Fizički izgled GSM modema*

Telit GM862-GPS modem dolazi bez napajanja tako da se mora priključiti na eksterno napajanje. Eksterno napajanje se priključuje između pinova VBATT i GND, i mora da ispuni sledeće zahteve:

- nominalni radni napon 3,8 V
- operatibvni opseg napona od 3,4 V do 4,2 V

4. ORGANIZACIJA SOFTVERA

Softver kontrolera je podeljen na četiri osnovna funkcionalna bloka i to:

- Inicijalizacija kontolera
- Inicijalizacija GSM modema
- Prekidne rutine
- Glavni program

5. UPRAVLJANJE UREĐAJEM PUTEM MOBILNE APLIKACIJE

Za upravljanje uređajem kao i nadgledanje automobila putem GPS-a, u kojem je postavljen uređaj, napisana je *Mobile* aplikacija u App Inventoru 2010. Aplikacije pisane u App Inventor, rade pod *Google Labs* operativnim sistemom i koriste *Android* platformu. U ovom slučaju to je "Android 2.2.1" platforma.



Slika 4. *Izgled glavnog menija Android Mobile aplikacije*

Izgled glavnog menija, koje se pojavljuje nakon ispravno unete šifre za pristup prikazan je na slici 4., u čijem se prozoru pojavljuju četiri komandna dugmeta za izbor željenih funkcija.

Controls – potprogram koji vrši upravljanje nad automobilom koje se prikazuju nakon aktiviranja ovog dugmeta, u njegovom "padajućem" meniju

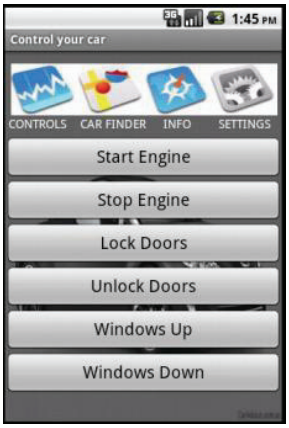
Car finder – potprogram kojim koji služi za upravljnje pronalaženja automobila, izborom jednog od ponuđenih komandi iz padajućeg menija koji se aktivira izborom ove komande

Info – ovo će predstavljati opcioni potproram, u zavisnosti da li automobil u kojem se nalazi uređaj podržava ovakvu opciju, koja će biti objašnjena u daljem tekstu

Settings – potprogram u kojem se može izvršiti izbor jednog od više automobila na kojima se mogu izvršavati sve ponuđene komande u mobilnom aplikaciji

5.1. Izbor kontrole automobila *(Controls)*

U zavisnosti od izabranog komandnog dugmeta iz padajućeg menija, na automobilu će se izvršavati operacije izabrane od strane korisnika. Izgled prozora je prikazan na slici 5.



Slika 5. *Izgled prozora za kontrolu automobila*

Komandna dugmad koja se nalaze u ovom prozoru su sledeća:

Start Engine- predstavlja komandu kojom možemo izvršiti pokretanje motora automobila

Stop Engine- predstavlja komandu kojom možemo da izvršimo zaustavljanje rada motora automobila

Lock Doors- predstavlja komandu kojom možemo da izvršimo zaključavanje svih vrata na automobilu

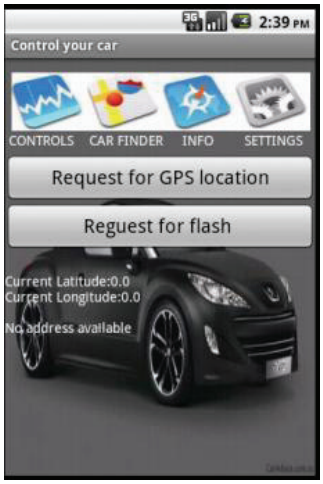
Unlock Doors- predstavlja komandu kojom možemo da izvršimo otključavanje svih vrata na automobilu

Windows Up- predstavlja komandu kojom možemo da izvršimo podizanje svih prozora na vratima automobila

Windows Down- predstavlja komandu kojom možemo da izvršimo na spuštanje svih prozora na vatima automobila

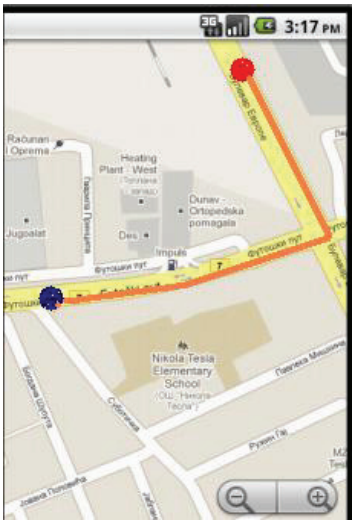
5.2. Izbor za pronalaženje automobila *(Car finder)*

U padajućem meniju vidimo dva komandna dugmeta koja nam pritiskom na njih omogućavaju pronalaženje automobila putem GPS kordinata ili slanjem zahteva za kratkim treptanje farova automobila.. Izgled prozora dat je na slici 6.



Slika 6. *Izgled prozora za pronalaženje automobila*

Request for GPS location- pritiskom na ovo komandno dugme vrši se očitavanje GPS lokacije automobila koji se posmatra. Ukoliko je uređaj pronašao GPS kordinate automobila, aplikacija automatski pokreće Google Maps aplikaciju i na njoj prikazuje lokaciju automobila (slika 7.), kao i lokaciju samog korisnika sa detaljnim informacijama, kao što su geografska dužina i širina i tačna adresa lokacije korisnika.



Slika 7. *Izgled prozora nakon pronalaženja GPS lokacije automobila*

Request for flash- pritiskom na ovo dugme izvršava se komanda koja omogućava treptanje prednjih svetlosnih grupa automobila

5.3. Opcija informacije *(Info)*

Klikom na dugme "Info" u glavnom meniju, otvara se prozor (slika 8.) u kojem imamo labelu u kojoj mogu biti

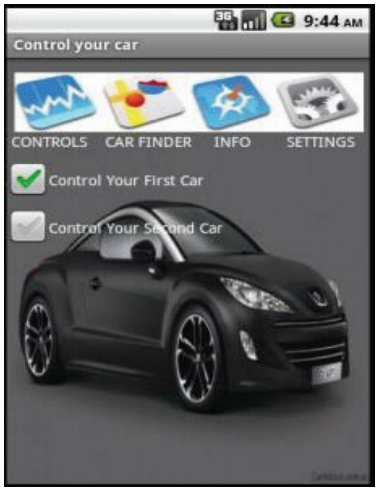
ispisane informacije kao što su temperatura motora, pritisak ulja i ostale bitne informacije vezane za automobil. Da bi ova opcija bila dostupna korisniku, potrebno je da posmatrani automobil ima mogućnost povezivanja uređaja predstavljenog u ovom radu, sa centralnom jedinicom samog automobila.



Slika 8. *Izgled prozora za opciju Info*

5.4. Opcija podešavanje *(Settings)*

U slučaju da je aktivirana ova opcija iz glavnog menija, otvoriće nam se prozor u kojem će se prikazati mogućnost izbora kontrole jednog od više automobila. Izgled prozora dat je na slici 9.



Slika 9 . *Izgled prozora za izbor kontrole jednog od više automobila*

6. ZAKLJUČAK

Sistem za daljinski nadzor automobila, upravljanje i kontrolu pristupa putem GSM mreže se pokazao kao pouzdano i jeftino rešenje. Daljim usavršavanjem moguće je napraviti sistem koji bi predstavljao kvalitetno komercijalno rešenje. Sistem možem biti izrađen od SMD komponenti, što bi ga činilo pogodnim za proizvodnju u većim količinama, jer bi veliki deo proizvodnog procesa mogao biti automatizovan. Automatizacijom proizvodnog procesa bi se u ovom slučaju dobila veća produktivnost i ujednačen kvalitet proizvedenih uređaja.

7. LITERATURA

[1] ATmega16 datasheet, Atmel - www.atmel.com
[2] GM862-GPS hardware user guide - www.telit.com
[3] LM7805 datasheet, ST microelectronics
[4] B9778HFP datasheet, ROHM
[5] Telit AT command reference - www.telit.com
[6] Telit modules softvare user guide – www.telit.com
[7] Six common reasons for power supply instability, Switching Power Magazin, 2006
Comet electronics 2009, katalog elektronskih komponenti - www.comet.co.yu

Kratka biografija



Predrag Dimković je rođen u Sarajevu 1986. god. U Novom Sadu je završio srednju elektrotehničku školu. Fakultet tehničkih nauka u Novom Sadu, odsek za mehatroniku, je upisao 2005. god. Diplomski-master rad je odbranio 2011. godine.

Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.8:007.52

SOFTVER ZA ZAKAZIVANJE KORIŠĆENJA PARKING PROSTORA NA UNIVERZITETU U NOVOM SADU

SOFTWARE FOR SCHEDULING PARKING LOTS AT THE UNIVERSITY OF NOVI SAD

Ferid Sejdović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – *U ovom radu predstavljeno je softversko rešenje problema pristupa i korišćenja parking prostora namenjenog gostima Univerziteta u Novom Sadu. Automatizovani softverski sistem koji se sastoji od klijent i server aplikacija, služi za zakazivanje mesta i omogućavanje pristupa pomoću mobilnog telefona. Opisan je trenutni parking sistem univerziteta sa načinima pristupa istom i modifikovani sistem na kojem se rešenje zasniva. Opisan je GSM modem koji se koristi u realizaciji softvera. Takođe, opisani su principi rada i način komunikacije klijent i server aplikacija softvera koji je isprogramiran u C# programskom jeziku.*

Abstract *– This paper presents a software solution for access and usage of parking spaces intended for guests of the University of Novi Sad. An automated software system which consists of client and server applications, is used for scheduling and providing access to the sites by using a mobile phone. It also describes a current the university parking system with ways of access to it, and a modified system on which the solution is based. In addition, there is described a GSM modem that is used in the implementation of the software, the operating principles and communication between client and server applications in software which is programmed in C # programming language.*

Ključne reči: *Parking sistem Univerziteta u Novom Sadu, klijent-server aplikacije, automatizovani softver, kontrola pristupa, GSM modem, mobilni telefon*

1. UVOD

Život savremenog čoveka ne može da se zamisli bez automobila. Broj automobila, od nastanka prvog davne 1886. godine u Nemačkoj, do danas teži milijardi i samim tim čovek je morao da preduzme nešto u skladu sa rastućim potrebama. Morao je da obezbedi prostor za parkiranje i omogući parkiranje na način koji ne remeti i ne ugrožava redovan tok i bezbednost u saobraćaju. Razvoj novih tehnologija je doneo inovacije i promene na ovom polju. Danas, u 21. veku, savremeni parking prostori imaju novu dimenziju, sa brojnim mogućnostima, od zakazivanja mesta pomoću mobilnog telefona, preko automatskog otvaranja i zatvaranja barijere (rampe) do digitalnih displeja koji pokazuju broj slobodnih i zauzetih mesta.

NAPOMENA:
Ovaj rad proistekao je iz diplomskog-master rada čiji mentor je prof. dr Srevan Stankovski.

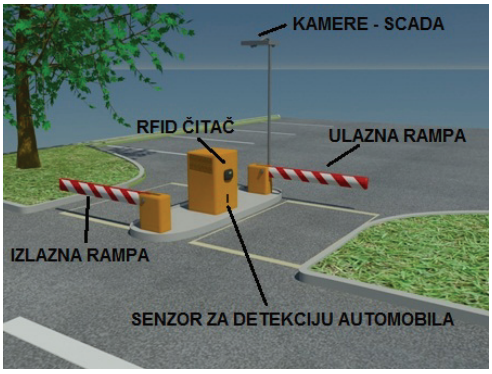
Zbog sve učestalijeg problema nedostatka parking prostora za goste Univerziteta u Novom Sadu, kao posledica je nastala ideja da se obezbede parking prostori koji će biti dostupni za njih ako je prethodno njihov dolazak zakazan. Osnovni cilj ovog rada je razvijanje klijent-server softvera koji će unošenjem podataka gosta u bazu podataka zakazati njegov dolazak i uz pomoć GSM modema davati signal sistemu za nadgledanje i vizuelizaciju (SCADA) da podigne ili spusti barijeru (rampu) gostu, ako je pre toga ispunjen definisani niz preduslova, od kojih su najvažniji da se njegovi podaci i datum dolaska poklapaju sa prethodno unetim podacima u bazi podataka.
Komunikacija između aplikacija mora biti ostvarena elektronskom poštom (eng. e-mail). Pored osnovne funkcije vršenja unosa podataka, to jest zakazivanje gosta (mogućnost zakazivanja za mesec dana unapred), klijent aplikacija mora da poseduje i funkcije: zahtev za izmenu broja i zahtev liste zakazanih gostiju za određeni datum. Osnovne funkcije server aplikacije moraju biti: odobrenje rada klijent aplikacije (to jest svih njenih operacija, ako su ispunjeni svi uslovi), automatska obrada podataka o gostu i unos u bazu podataka, automatsko brisanje starih unosa, mogućnost kreiranja izveštaja primljenih elektronskih pisama, mogućnost brisanja i izmene broja prethodno unetog gosta u bazu podataka i automatskog prikupljanja informacija sa GSM modema.

2. PARKING SISTEM UNIVERZITETA U NOVOM SADU I PRINCIP REŠENJA SISTEMA ZA GOSTE

Osnovna podela parking sistema jeste na otvorene i zatvorene. Otvoreni sistemi ne koriste barijere i omugućeno je da se vozila parkiraju na parking prostor bez ikakve kontrole ulaska i izlaska, dok zatvoreni koriste barijere koje su postavljenje na ulazu i izlazu, kako bi se ograničio pristup parking prostoru. Pristup ovakavom parking prostoru se omogućava pomoću neke vrste identifikacije (Tag sistem, sistem sa karticama, sistemi sa automatskom identifikacijom vozila i dr.).
Na Univerzitetu u Novom Sadu postoje obe vrste parking sistema. Zatvoreni sistem je namenjen samo za zaposleno osoblje Univerziteta koje može da mu pristupi samo uz korišćenje RFID identifikacionih kartica, dok otvorenom parking sistemu mogu da pristupe svi ostali korisnici. Parking sistem namenjen gostima je smešten na parkingu za zaposlene, zbog čega gosti treba da postanu posebna vrsta korisnika kojima treba da se omogući pristup bez korišćenja RFID identifikacionih kartica.

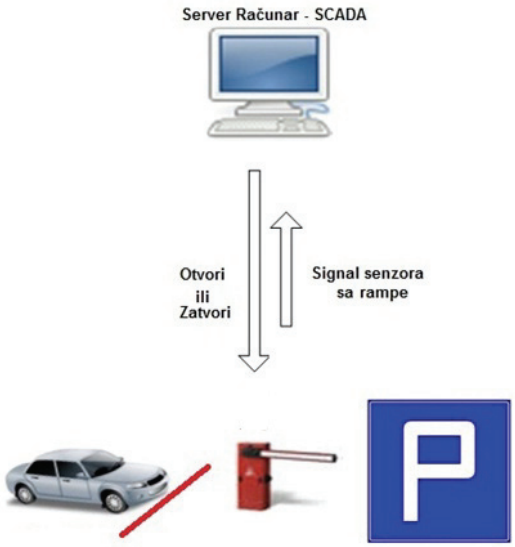
Parking sistem za zaposleno osoblje (slika 1.) se sastoji od:

- barijera (ulazna i izlazna),
- računara, na kojem se nalazi sistem za vizuelizaciju i nadgledanje (SCADA),
- senzora, koji detektuje prisustvo automobila ispred barijere,
- RFID čitača identifikacionog serijskog broja RFID kartica,
- kamera, koje služe za slanje informacija SCADA sistemu.



Slika 1. *3D model ulaza-izlaza parking sistema za zaposlene*

Funkcionalna blok šema sistema prikazana je na slici 2.

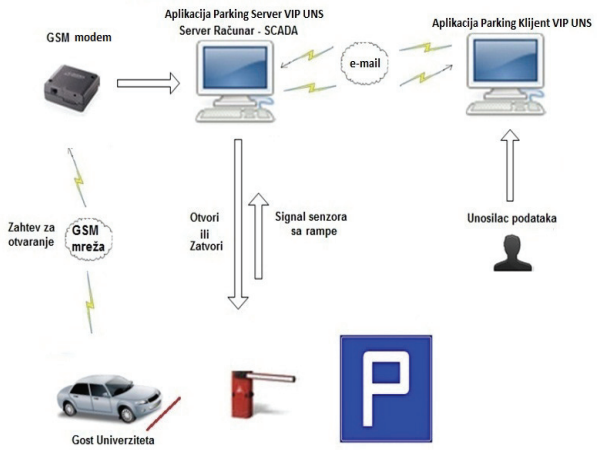


Slika 2. *Funkcionalna blok šema sistema*

Softversko rešenje podrazumeva korišćenje GSM modema kao sredsva sa kojim se umesto RFID kartice može omogućiti pristup i korišćenje parking prostoru. Elementi koje treba dodati osnovnom sistemu su:

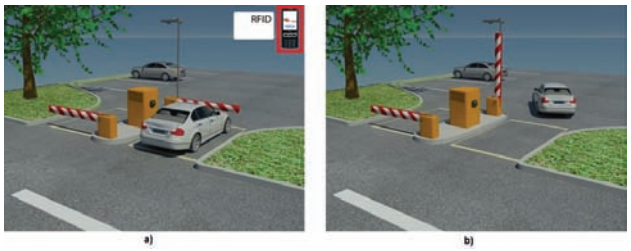
- GSM modem,
- računar na kojem bi se nalazila klijent aplikacija.

Server aplikacija mora biti smeštena na računaru gde je implementiran SCADA sistem.
Takođe, na istom računaru bi se povezao GSM modem. Na slici 3. prikazana je funkcionalna blok šema modifikovanog sistema.



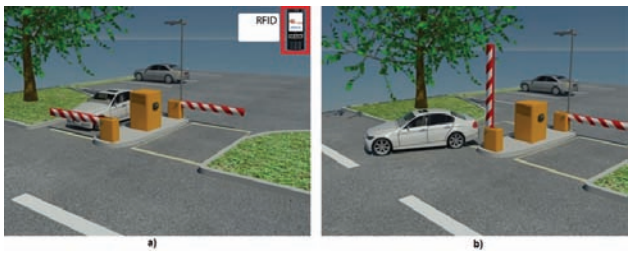
Slika 3. *Funkcionalna blok šema sistema koji koristi GSM modem*

Na slici 4. je prikazan slučaj ulaska gosta na parking prostor. U slučaju zaustavljanja automobila ispred ulazne barijere, senzor detektovanja prisustva automobila šalje signal SCADA sistemu da postoji prisustvo. Nakon toga se čeka sledeći potez korisnika parkinga, to jest da li će koristiti RFID karticu ili će pozvati naznačeni broj telefona (čija se SIM kartica nalazi u GSM modemu).



Slika 4. *Prikaz ulaska gosta na parking prostor*

U slučaju korišćenja mobilnog telefona, to jest poziva naznačenog broja, čeka se provera prethodno unetih podataka zakazanog gosta. Ako se svi kriterijumi zadovolje, barijera će se podići i gost će moći da uđe na parking. Slučaj izlaska sa parking je potpuno analogan ulasku, samo što se odvija u suprotnom smeru gde se podiže i spušta izlazna barijera (slika 5.). U toku dana u kom je zakazan, gostu se omogućava neograničen broj ulazaka i izlazaka.



Slika 5. *Prikaz izlaska gosta sa parking prostora*

3. GSM MODEM Cinterion/Siemens, model MC35i

GSM modem (GSM modul ili GSM terminal), kao i običan mobilni telefon koristi GSM mrežu. Potreba za ovakvom vrstom uređaja je postojala od kada postoje elektromotorni pogoni, ali je tek sa razvojem tehnologije

materijalizovana u obliku GSM modema. Sa njihovom pojavom omogućen je nadzor i mogućnost daljinskog upravljanja sistemima. Njihova velika upotrebna vrednost se pokazala u nadzoru i upravljanju elektromotornim pogonima koji su nepristupačni ili se nalaze u opasnim sredinama (eksplozivne, radioaktivne, fizički nepristupačne, hemijski opasne sredine). U industriji je ovakva vrsta nadzora i upravljanja poznata pod imenom M2M (eng. *machine to machine, machine to mobile, man to machine, machine to man*).



Slika 6. *Fizički izgled GSM modema koji se koristi u radu*

Tehničke karakteristike prethodno pomenutog GSM modema su:

- mogućnost rada na dve frekvencije (eng. *dual band*) 900/1800 MHz,
- niski napon napajanja od 8 do 30 V (mogućnost korišćenja u vozilu ili na terenu),
- male dimenzije, 65 x 74 x 33 mm sa masom od 130g,
- podržana je standardna mini SIM kartica sa PIN-om.

Komunikacija sa GSM modemom se odvija putem serijske RS232 komunikacije. Fabrički parametri serijske komunikacije ovog modema su:

- Brzina prenosa podataka (eng. *Baud rate*) = 9600,
- Broj bitova podataka(eng.*Data bits*) = 8,
- Paritet (eng. *Parity*) = 0,
- Stop bitovi (eng. *Stop Bits*) = 1,

GSM modemom upravljamo preko proširenog seta AT komandi. AT standard je mrežno orijentisani komandni jezik. Inicijalna komanda koja se šalje GSM modemu radi inicijalizacije je *AT*, na šta treba da se dobije odgovor *OK*. Komande koje su korišćene u izradi softvera su:

- AT (inicijalizacija),
- AT+CHUP (prekid poziva),
- AT+CLCC (dobijanje pozivnog broja).

GSM modem predstavlja ključnu figuru u ovom radu, jer se pomoću njega vrši identifikacija gosta i omogućavanje pristupa parking prostoru.

4. SOFTVERSKA REALIZACIJA PARKING SISTEMA ZA GOSTE UNIVERZITETA

Kreiranje softvera je izvedeno u razvojnom okruženju programskog paketa *Microsoft Visual Studio 2008*, dok je kao jezik za programiranje korišćen C# (eng. *C Sharp*). Softver koji čine po jedna klijent i server aplikacija je nazvan *VIP Parking UNS*.

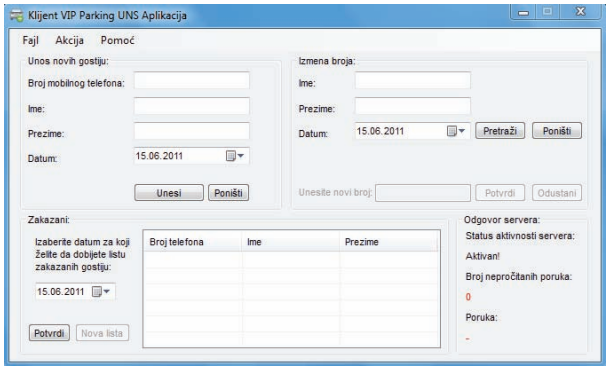
Komunikacija između klijent i server aplikacije ostvarena je putem elektronske pošte. Glavni razlozi zbog kojih je odabran ovaj način komunikacije su nezavisna razmena podataka od lokalne računarske mreže na Univerzitetu i cena obezbeđivanja komunikacije koja je ovim načinom izvođenja u potpunosti besplatna. Dakle, prilikom slanja zahteva za dozvolom izvršavanja neke operacije, klijent aplikacija će slati elktronsko pismo, nakon čijeg „čitanja" će server aplikacija slati odgovor istim putem. Izabrani servis elektronske pošte koji se koristio u izradi softvera je *Gmail*, firme *Google*.

Tom prilikom napravljena su dva korisnička naloga, *klijentparkingvip@gmail.com* i *serverparkingvip@gmail.com*.

Klijent aplikacija je podeljena u pet funkcionalnih delova:

- ***Komunikacija sa serverom*** (Obaveštajni deo aplikacije gde se korisniku aplikacije prikazuju odgovori koje server aplikacija šalje nakon nekog zahteva. Da bi izvršila sve svoje operacije neophodno je da klijent aplikacija pošalje zahtev server aplikaciji. Nakon toga, prelazi se u režim čekanja odgovora.),
- ***Unos podataka novog gosta*** (Ključni deo aplikacije gde se izvršava zakazivanje, to jest unos podataka o novom gostu. Nakon toga, šalje se zahtev serveru radi odobravanja unosa u bazu podataka),
- ***Izmena broja gosta*** (slanje zahteva serveru da izmeni broj telefona već postojećeg gosta u bazi podataka, ako je došlo do pogrešnog unosa ili promene broja pomenutog gosta),
- ***Lista zakazanih gostiju*** (jedini funkcionalni deo koji radi na lokalnom nivou, to jest nije zavistan od servera. Služi da se korisniku klijent aplikacije vizuelno prikaže spisak zakazanih gostiju za određeni datum),
- ***Spisak gostiju*** (Administrativni deo aplikacije, čijim postojanjem je omogućeno olakšano korišćenje lokalnih procesa klijent aplikacije (zahtev za listu zakazanih ili eventualno omugućavanje štampanja podataka).

Na slici 7. i 8. prikazani su korisnički intefejsi klijent i server aplikacija, respektivno.

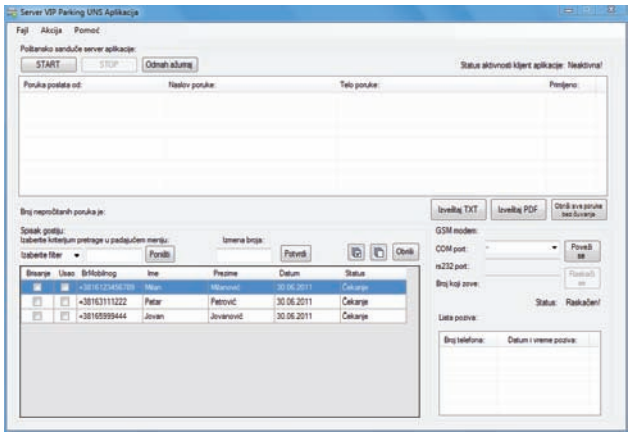


Slika 7. *Interfejs klijent aplikacije*

Server aplikacija je podeljena u četiri funkcionalna dela:

- ***Poštansko sanduče server aplikacije*** (Obaveštajni deo aplikacije gde se prikazuju poruke, to jest zahtevi pristigli od klijent aplikacije. Poruke se detaljno prikazuju sa mogućnošću da se sačuvaju na lokalni tvrdi disk računara. Administrator korisnik aplikacije je dužan da pritisne dugme *START* da bi otpočeo automatski rad. Pored ovog dugmeta postoje i dugmad *STOP* (prekidanje

automatskog rada) i *Odmah ažuriraj* (što predstavlja pregled prispele pošte u jednoj iteraciji)),



Slika 8. *Interfejs server aplikacije*

- ***Prikupljanje podataka sa GSM modema*** (Kao što je već rečeno, GSM modem ima ključnu ulogu u identifikaciji gosta. U trenutku poziva gosta, vrši se automatsko upoređivanje broja koji poziva sa brojem u bazi podataka i ako se podaci podudaraju, šalje se zahtev SCADA sistemu za podizanje ili spuštanje barijere),
- ***Ažuriranje baze podataka brisanjem starih unosa*** (Softverska funkcija koja svojim pokretanjem automatski briše goste nakon isteka njihovog zakazivanja),
- ***Spisak gostiju*** (Identičan funkcionalni deo kao i u klijent aplikaciji. Osnovne funkcije su filtriranje baze podataka, ručno brisanje gostiju i ručna izmena broja).

5. ZAKLJUČAK

Automatizacija ima najvažniju ulogu u modernoj industriji i globalnoj ekonomiji, jer se sistemi čine efikasnijim, autonomnijim, fleksibilnijim i jednostavnijim za upotrebu. Cilj u stvaranju automatizovanih upravljačkih sistema je smanjiti faktor ljudske greške u čemu primenom mehanike, elektronike i računarskih tehnologija možemo stvoriti širok spektar rešenja koji možemo primeniti za rešavanje problema. Pored svojih prednosti, automatizacija i robotika imaju i svoje mane, a to su visoki troškovi implementacije, velika zavisnost od održavanja i unapređivanja već postojećih automatizovanih i robotizovanih sistema.

U okviru ovog rada realizovan je praktičan upravljački softver *VIP Parking UNS* pomoću kojeg gosti Univerziteta u Novom Sadu mogu da zakažu parking mesto koje će im biti dostupno i lako pristupljivo. Testiranjem softvera na sistem parking prostora Univerziteta u Novom Sadu je pokazano da se vreme zadržavanja prilikom ulaska i izlaska svodi na minimum, što je i bio glavni cilj.

Prednosti softvera kao što su visok stepen zaštite parking prostora od moguće zloupotrebe i neregularosti, brzina izvršavanja operacija i automatska obrada podataka bez mogućnosti zloupotrebe daju šansu da se u bliskoj budućnosti prethodno pomenuti softver primeni i dodatno razvije na kompleksnijim parking sistemima.

6. LITERATURA

[1].Ronald W. Stehman, Ed. Fredericksburg, VA: *„Parking 101: A Parking Primer”*, International Parking Insitute, 2001. godina.

[2].Timothy H., Ronald W. Stehman, VA: *„Parking Management – The Next level”*, International Parking Insitute, 2001. godina.

[3].Amit Vyas, Peter O'Grady: *„A Review of Mobile Commerce Technologies”*, Department of Industrial Engineering, University of Iowa, 2001. godina.

[4].Zoran Ćirović, Ivan Dunđerski, „*Tehnike vizuelnog programiranja*”, 2005. godina.

[5].Stankovski dr. Stevan, Raković MSc. Mirko, Materijal sa vežbi i predavanja predmeta *Inteligentni Sistemi*, š.g. 2009-2010.

[6].Ostojić dr. Gordana, Materijal sa vežbi i predavanja predmeta *Računarska Integracija Proizvodnih Sistema*, š.g. 2009-2010.

[7].MicroSoft Developer Network, http://msdn.microsoft.com, 2011. godina.

[8].Internet Engineering Task Force, http://tools.ietf.org, 2011. godina.

[9].Peter Huber, *„POP3 Email Clinent With MIME Support”*, http://www.codeproject.com/KB/IP/Pop3MimeClient.aspx, 2006. godina.

[10].http://itextsharp.com/, 2011. godina.

[11].http://www.codeproject.com/, 2011. godina.

[12]. http://www.gsmworld.com/, 2011. godina.

Kratka biografija



Ferid Sejdović rođen je u 1986. godine u Somboru, gde je 2001. godine upisao Gimnaziju "Veljko Petrović", prirodno-matematički smer. Fakultet tehničkih nauka u Novom Sadu, odsek za Mehatroniku, smer Robotika i Automatizacija je upisao 2005. god. Diplomski-master rad je odbranio 2011.godine.