



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXXII

Број: 1/2017

Нови Сад

Едиција: „Техничке науке – Зборници“

Година: XXXII

Свеска: 1

Издавач: Факултет техничких наука Нови Сад

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

Уредништво:

Проф. др Раде Дорословачки

Проф. др Драгиша Вилотић

Проф. др Срђан Колаковић

Проф. др Владимир Катић

Проф. др Драган Шешилија

Проф. др Миодраг Хаџистевић

Проф. др Растислав Шостаков

Доц. др Мирослав Кљајић

Доц. др Бојан Лалић

Доц. др Дејан Убавин

Проф. др Никола Јорговановић

Доц. др Борис Думнић

Проф. др Дарко Реба

Проф. др Ђорђе Лађиновић

Проф. др Драган Јовановић

Проф. др Мила Стојаковић

Проф. др Драган Спасић

Проф. др Драгољуб Новаковић

Редакција:

Проф. др Владимир Катић, главни
уредник

Проф. др Жељен Трповски, технички
уредник

Проф. др Драган Шешилија

Проф. др Драгољуб Новаковић

Др Иван Пинђер

Бисерка Милетић

Језичка редакција:

Бисерка Милетић, лектор

Софија Рацков, коректор

Марина Катић, преводилац

Издавачки савет:

Савет за библиотечку и издавачку делатност ФТН,
проф. др Радош Радивојевић, председник.

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

CIP-Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

378.9(497.113)(082)
62

ЗБОРНИК радова Факултета техничких наука / главни и одговорни уредник
Раде Дорословачки. – 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 прва 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 веб 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, сада већ мастера, који су радове бранили у периоду од 01.10.2016. до 05.12.2016. год., а који се промовишу 28.06.2017. год. То су оригинални прилози студената са главним резултатима њихових мастер радова.

Известан број кандидата објавили су радове на некој од домаћих научних конференција или у неком од часописа.

Велик број дипломираних инжењера–мастера у овом периоду био је разлог што су радови поводом ове промоције подељени у две свеске.

У овој свесци, са редним бројем 1., објављени су радови из области:

- машинства,
- електротехнике и рачунарства и
- мехатронике.

У свесци са редним бројем 2. објављени су радови из области:

- грађевинарства,
- саобраћаја,
- графичког инжењерства и дизајна,
- архитектуре и
- инжењерства заштите животне средине.

У свесци са редним бројем 3. објављени су радови из области:

- инжењерског менаџмента,
- математике у техници,
- геодезије и геоматике,
- регионалне политике и развоја,
- управљања ризиком од катастрофалних догађаја и пожара и
- инжењерства информационих система.

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: Mašinstvo

1. Igor Tabački, OPTIMIZACIJA PROIZVODNIH I LOGISTIČKIH PROCESA U KOMPANIJI I.G.BAUERHIN PRIMENOM LEAN ALATA I SIMULACIJA	1-4
2. Deana Petrov, PRIMENA SIMULACIJA I LEAN ALATA ZA OPTIMIZACIJU PROIZVODNIH I LOGISTIČKIH PROCESA U KOMPANIJI MK HOLDING-OGRANAK POBEDA	5-8
3. Milan Dugošija, KRATKOROČNO PREDVIĐANJE POTROŠNJE PRIRODNOG GASA	9-12
4. Branko Pavkov, Marin Gostimirović, RAZVOJ APLIKACIJE BAZE PODATAKA ZA MODELIRANJE OBRADNIH PROCESA	13-16

Radovi iz oblasti: Elektrotehnika i računarstvo

1. Srđan Donevski, VEB ALAT ZA NAPREDNU ADMINISTRACIJU SISTEMA ZA ELEKTRONSKU NASTAVU	17-20
2. Filip Savić, RUBY GOOGLE SKETCHUP INTERFEJS IZ ASPEKTA PROGRAMSKIH PARADIGMI	21-24
3. Predrag Radojičić, Dejan Nemec, RAZVOJ OPTIČKIH MREŽA	25-28
4. Nebojša Vujković Cvijin, Željien Trpovski, DIGITALNI VIDEO FORMATI	29-32
5. Nebojša Janković, SERVISI TELEVIZIJE ZASNOVANE NA INTERNET PROTOKOLU	33-36
6. Dragan Janković, Željien Trpovski, INTERAKTIVNA TELEVIZIJA	37-41
7. Sanjin Malešević, Milan Vidaković, INTEGRACIJA SERVERSKE I KLIJENTSKE TEHNOLOGIJE U RESPONSIVE VEB APLIKACIJAMA	42-45

	STRANA
8. Miloš Đukić, POREĐENJE PERFORMANSI REGULATORA KORIŠĆENIH ZA UPRAVLJANJE MEŠNIM VENTILIMA PRI SIMULACIJI RADA TOPLOTNE PODSTANICE	46-49
9. Branislav Potić, SIMULACIJA SLANJA DOGAĐAJA SA PAMETNIH BROJILA	50-53
10. Jovan Zelenković, PRIMENA GA I PSO ALGORITMA ZA PODEŠAVANJE PARAMETARA PID REGULA- TORA KORIŠĆENJEM FUZZY LOGIKE ZA OCENU POTENCIJALNIH REŠENJA	54-56
11. Jelena Jovanović, SISTEM RANOG UPOZORENJA - MODUL BOLNIČKOG INFORMACIONOG SISTEMA	57-59
12. Slobodan Stamenković, MODELOVANJE PODZEMNOG TOPLOTNOG IZMENJIVAČA GEOTERMALNE TOPLOTNE PUMPE POMOĆU HELLSTRÖM-OVOG DVODIMENZIONOG MODELA ..	60-63
13. Darko Vasić, WEB APLIKACIJA ZASNOVANA NA B2B MODELU ELEKTRONSKOG POSLOVANJA ZA POTREBE NABAVKE PROIZVODA I USLUGA	64-67
14. Nenad Milenković, Vladimir Katić, ANALIZA TROFAZNOG DIREKTNOG MATRIČNOG PRETVARAČA	68-71
15. Lazar Tatomirov, VIZUELNA KATEGORIZACIJA UZ POMOĆ KLJUČNIH TAČAKA	72-75
16. Petar Stefanović, Željen Trpovski, ZAKONSKI ASPEKT PLANIRANJA I IZGRADNJE OPTIČKIH KABLOVSKIH MREŽA U REPUBLICI SRBIJI	76-79
17. Mirjana Macura, DINAMIČKO OBJAVLJIVANJE DOKUMENATA POMOĆU DOSTUPNIH SOFTVERSKIH ALATA	80-83
18. Arpad Farkaš, PROŠIRENJE DJANGO WEB APLIKACIJE U OKVIRU KROKI ALATA	84-87
19. Jasmina Ćurčin, POREĐENJE PERFORMANSI NAPONSKI KONTROLISANIH RING OSCILATORA SA JEDNIM IZLAZOM I SA DIFERENCIJALNIM IZLAZIMA U CMOS TEHNOLOGIJI	88-91
20. Milan Milosavljević, Željen Trpovski, RAZVOJ INFRASTRUKTURE U MICROSOFT AZURE	92-95
21. Vuk Šavija, Željen Trpovski, LOGIČKA KRIPTOANALIZA HEŠ FUNKCIJA	96-98
22. Dušan Galić, Vladimir Katić, ISPITIVANJE POSTROJENJA ELEKTROLUČNE PEĆI „R-S SILICON“	99-102
23. Nenad Protić, Željen Trpovski, OSOBINE SAVREMENIH SISTEMA ZA VIDEO NADZOR	103-106
24. Miljan Rogan, Željen Trpovski, PRIMENA OPTIČKIH KOMUNIKACIJA U INDUSTRIJSKIM SISTEMIMA	107-110
25. Mladen Gavrić, SOFTVERSKO REŠENJE ZA UPRAVLJANJE PODACIMA O DISTRIBUTIVNOJ MREŽI ...	111-113
26. Marko Petković, Nikola Vojnović, UPOREDNA ANALIZA PRIMENE NEWTON/RAPHSON-ovog POSTUPKA I KOMPENZACIONE METODE NA SUBPRENOSNIM MREŽAMA	114-117
27. Ivan Lazić, ANALIZA DETEKTORA I DESKRIPTORA ZA KLASIFIKACIJU OBJEKATA U SLICI I PRIMENA U OKVIRU ANDROID PLATFORME	118-121
28. Branko Đukić, AUTOGENERISANJE LISTE RIZIČNIH ISPADA ZA POTREBE ANALIZE SIGURNOSTI ELEKTROENERGETSKOG SISTEMA	122-125

	STRANA
29. Vladimir Rončević, REALIZACIJA PROGRAMSKE PODRŠKE ZA GENERISANJE PROJEKTNE STRUKTURE I PROGRAMSKOG KODA ZA MODUL ESP8266	126-129
30. Vesna Čičković, UPRAVLJANJE RAZMENAMA SNAGE U POVEZANIM ELEKTROENERGETSKIM SISTEMA SA UVAŽAVANJEM DISTRIBUIRANOG BALANSNOG ČVORA	130-133
31. Nikola Stamenković, SOFTVERSKO REŠENJE ZA POSLOVNO IZVEŠTAVANJE U APOTEKARSKOM SISTEMU	134-137
32. Срђан Цвијановић, ДЕТЕКЦИЈА, АНАЛИЗА И КОНВЕРЗИЈА ОПИСА ДОГАЂАЈА У ПАМЕТНИМ ЕЛЕКТРОЕНЕРГЕТСКИМ СИСТЕМИМА	138-141
33. Nikola Vidović, ANALIZA ANOMALIJA U RADU CROSS-PLATFORM SOFTVERSKIH REŠENJA U OKVIRU SMART GRID SISTEMA	142-145
34. Stefan Živković, REKONFIGURACIJA DISTRIBUTIVNIH MREŽA U REALNOM VREMENU	146-149
35. Nikola Žilić, Savo Đukić, ANALIZA POUZDANOSTI DISTRIBUTIVNE MPEŽE PRIMENOM DIJAGRAMA STABLA OTKAZA	150-153

Radovi iz oblasti: Mehatronika

1. Dušan Stanković, UPRAVLJAČKI SISTEM FORMIRAČICE KARTONSKIH KUTIJA	154-157
2. Nikola Bukvić, AUTOMATIZACIJA POSTROJENJA MINI PIVARA	158-161
3. Hana Gostimir, KINESTETIČKO OBUČAVANJE ČETVORONOŽNOG ROBOTA KOJI SE UPRAVLJA NA OSNOVU MOMENATA U ZGLOBOVIMA	162-165
4. Miloš Tanović, RAZVOJ SISTEMA ZA DALJINSKO UPRAVLJANJE PNEUMATSKIM IZVRŠNIM ORGANIMA PREKO INTERNETA	166-169

**OPTIMIZACIJA PROIZVODNIH I LOGISTIČKIH PROCESA U KOMPAJNI
I.G.BAUERHIN PRIMENOM LEAN ALATA I SIMULACIJA****OPTIMIZATION OF PRODUCTION AND LOGISTIC PROCESSES IN THE
I.G.BAUERHIN COMPANY USING LEAN TOOLS AND SIMULATIONS***Igor Tabački, Fakultet tehničkih nauka, Novi Sad***Oblast – MAŠINSTVO**

Kratak sadržaj – U okviru ovog rada opisana je metodologija i primena LEAN filozofije, kao najidealnijeg alata za poboljšanje proizvodnih i logističkih procesa, sa naznakom njegovog uticaja na poboljšanje produktivnosti, povećanje profitabilnosti, kvaliteta ali i smanjenje zaliha. U radu se na primeru nemačke kompanije I.G.Bauerhin sa proizvodnjom u Indiji, pokazao uticaj i poboljšanja dobijena korišćenjem simulacionih modela stanja kao i samo nekih od alata LEAN koncepta poput: 5S, mapiranje toka vrednosti, Kanban i Poka-Yoke.

Abstract – This work illustrates methodology and usage of LEAN philosophy, as the most ideal tool for improvement production and logistic processes, with its influence on making productivity, profitability and quality better and on reduction of inventories. In this work has been showed an influence and improvements from using simulation models and some of the LEAN tools as: 5S, Value Stream Mapping, Kanban and Poka-Yoke, based on production flows of german company I.G.Bauerhin with its production in Indija.

Ključne reči: LEAN filozofija i LEAN alati, simulacije, proizvodnja

1. UVOD

Potrebe za potpuno novim načinom organizacije proizvodnje donelo je moderno doba kao i prevelika konkurencija na globalnom nivou. LEAN filozofija predstavlja koncept reorganizacije proizvodnje i zasniva se na primeni kontinualnih unapređenja od najviših do najnižih hijerarhijskih nivoa preduzeća.

Velika prepreka implementaciji LEAN filozofije, koja može biti neverovatno efektivna ali pogrešno iskorišćena i veoma destruktivna po preduzeće, predstavlja upravo nemogućnost njene implementacije u već postojeći sistem bez ostvarivanja posledica.

Stoga kao idealan pomoćni alat LEAN filozofiji predstavlja primena simulacija. Odgovarajućim simulacionim modelima moguće je u potpunosti eksperimentisati sa proizvodnjom i mogućim poboljšanjima a da tek pri dostizanju trenutno, idealnog poboljšanja se ono i direktno primeni u proizvodnji bez posledica.

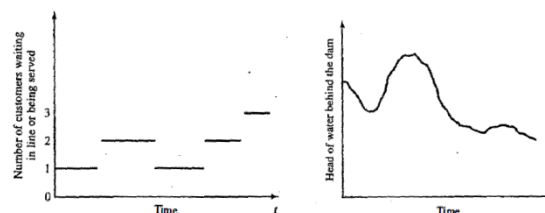
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila doc. dr Sanja Bojić.

Iako relativno stara metoda, LEAN se na globalnom nivou ustoličio prethodnih tridesetak godina i dalje se vrše njegove implementacije širom sveta, a u automobilske industriji, iz koje je u Japanu i proistekao, najviše se primenjuje. U okviru LEAN-a, postoje određeni alati koji neutrališu osam osnovnih gubitaka: greške, prekomernu proizvodnju, čekanje, transport, zalihe, nepotrebno kretanje, dodatnu obradu kao i neiskorišćeni talenat.

2. SIMULACIJE I SIMULACIONI MODELI

Današnje stanje tehnike i tehnologije u velikoj meri ne bi moglo biti zamišljeno bez jednog od najjačih alata pruženih inženjerima, a to su simulacije. Simulacije se ne bi trebale ili nisu preporučljive koristiti u sledećim slučajevima: ako je problem moguće rešiti lako i „zdravo razumski“, problem je moguće rešiti analitički, problem je lakše i isplativije rešiti direktno eksperimentom nego simulacijom, cena simulacije prekoračuje uštede, nedostatak resursa, nedostatak vremena, sistem je toliko kompleksan da ga je nemoguće definisati (npr. ljudski rad je ponekad izuzetno kompleksan za modelovanje). Kod primene simulacija izuzetno je bitno kakav je sistem koji se simulira. Sistemi mogu biti diskretni i kontinualni. Diskretni sistem je onaj kod kog se značajne varijable po stanje sistema menjaju u tačno određenim vremenskim trenucima. Kontinualni sistem je onaj kod kog se značajne varijable po stanje sistema menjaju kontinualno tokom vremena.



Slika 1. Prikaz diskretnog i kontinualnog sistema

U slučaju primera proizvodnje u I.G.Bauerhin-a, korišćen je simulacioni softverski paket Enterprise Dynamics.

3. PRINCIPI I ALATI LEAN KONCEPTA

Lean filozofija u nastojanju postizanja nikad ostvarivog idealnog stanja, definiše osam različitih ali ujedno i povezanih gubitaka: prekomerna proizvodnja, čekanje, dodatna obrada, transport, zalihe, nepotrebno kretanje, nepravilni proizvodi – greške i neiskorišćen talenat.

LEAN koncept zasnovan je na nekoliko principa:

1. vrednost proizvoda definisana je od strane kupca i predstavlja jednog definioca vrednosti,
2. na osnovu želja kupca i definisanih vrednosti od strane njega, u celokupnom proizvodnom toku definišu se sve operacije koje doprinose podizanju vrednosti proizvoda a koje ne, mapiranjem toka vrednosti (eng. *Value Stream Mapping*),
3. implementacija neprekidnog toka materijala, kroz operacije koje doprinose kvalitetu, gde je kretanje bez zastoja i nagomilavanja, čekanja i nepotrebnog transporta,
4. obezbeđivanje *PULL* sistema – proizvodnja tek po nalogu kupca, a ne na osnovu prognoza preduzeća o mogućim potražnjama,
5. konstantno i postepeno unapređenje svih aspekata preduzeća.

Na datom primeru kompanije *I.G.Bauerhin* postignuta su određena unapređenja i sa tim bolje, buduće stanje, upotrebom simulacija i *LEAN* alata kao što su:

1. Mapiranje toka vrednosti,
2. 5S,
3. *Kanban*,
4. *Poka-Yoke*.

4. PRIMENA LEAN ALATA U OKVIRU POBOLJŠANJA PROIZVODNE LINIJE U I.G.BAUERHIN-U

4.1 Mapiranje toka vrednosti

Da bi se prvenstveno ocenilo postojeće stanje u proizvodnom ciklusu jeste izaći na teren (pogon) i što adekvatnije pokupiti dostupne informacije o trajanju obavljanja svih operacija, vremenima između njih kao i o trajanju pripremno-završnih vremena, broju radnika po operaciji itd. i na osnovu tih informacija izraditi takozvano mapiranje toka stvaranja vrednosti ili skraćeno, eng., *VSM*. *VSM* predstavlja grafičko prikazivanje svih operacija, delatnosti koje se vrše unutar pogona.

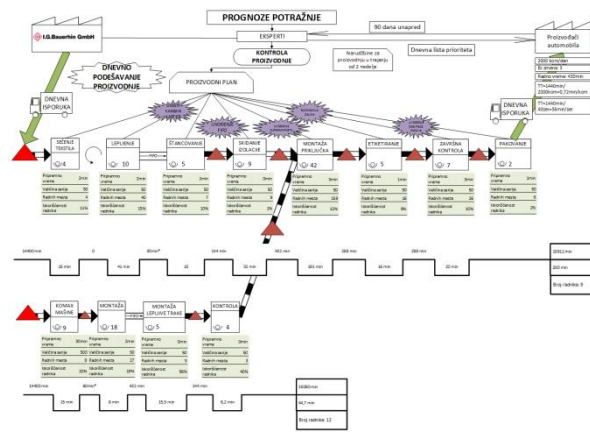
VSM se, u suštini, sastoji od dva dela:

1. mapa trenutnog stanja ,
2. mapa budućeg, optimizovanog, stanja.

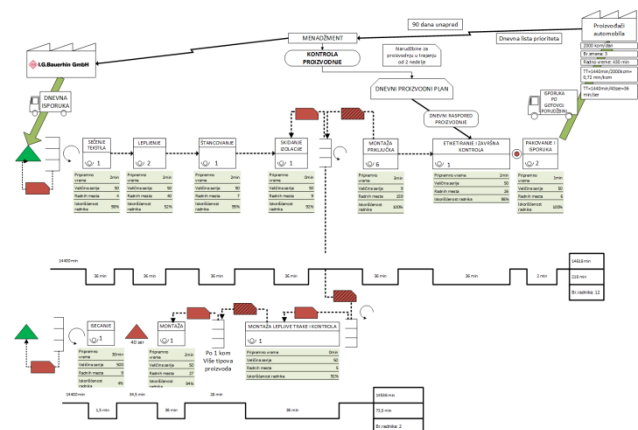
U okviru mape stanja, kako sadašnjeg tako i budućeg stanja, ispod grafičkog prikaza redosleda operacija i toka materijala, hijerarhije i toka informacija, iscrtava se protočno vremenska linija (eng. *Lead Time Bar*), kojom se, za svaki proces i njegovo trajanje, dobija suma svih vremena, vrednosno efektivnih i neefektivnih. Kada se što verodostojnije prikaže celokupan proces, gde je sada moguće izračunati celokupno vreme od ulaska sirovina ili podsklopova, pa sve do trenutka izlaska potpuno gotovog proizvoda kao i vreme koje je suma svih vremena operacija koje donose vrednost proizvodu, moguće je uvideti „rupe“ u proizvodnom procesu tj. razliku između

prva dva vremena i time videti da li je potrebna optimizacija procesa ili ne. Mapa sadašnjeg stanja proizvodnje, treba u što je većoj meri da prikazuje, što realnije sadašnje, zatečeno stanje u proizvodnji, a ne da bude bazirana na dokumentima koji govore o „pretpostavljenim“ vremenima trajanja izvršavanja operacija.

VSM ima cilj, zajedno sa ostalim *Lean* alatima, poboljšanje procesa kao i uspostavljanje *PULL* sistema proizvodnje da bi se izbegla prekomerna proizvodnja pod sloganom „za kupca, ono ŠTA ON želi, ono KADA ON želi i ono GDE ON želi“.



Slika 2. Prikaz mape toka vrednosti sadašnjeg stanja proizvodnje



Slika 3. Prikaz mape toka vrednosti unapređenog stanja proizvodnje

4.2 5S

U okviru svakog preduzeća bitno je da postoji određeni red i urednost radnog mesta iz razloga bezbednosti ali i produktivnosti radnika. Svako radno mesto, prema 5S potrebno je da redom prođe kroz 5 faza:

1. Sortirati pribore
2. Srediti
3. Učiniti ih „Sjajnim“ – čistim
4. Standardizovati
5. Stalno takvim održavati



Slika 4. Prikaz uređenog radnog mesta u I.G.B-u

Potreba za 5S veoma je izražena kako bi se postigla fleksibilnost u zamenama radnika i njihovom lakom i brzom snalaženju na novom radnom mestu.

Na slučaju radnog mesta u okviru održavanja, gde je prvenstveno bilo potrebno dosta vremena za uhodavanje i pronalaženje određenih alata i pribora, sada mesto uređeno prema prethodno navedenim kriterijumima, izgleda potpuno drugačije sa slobodnim radnim prostorom, određenim prostorom u vidu fioka sa čitko napisanim nazivima sitnih elemenata, ofarbanih u različite boje.

Takođe, primenom „osenčene“ table (eng. *Shadow Board*), na kojoj oivičeni i drugom bojom obojeni oblici odgovaraju alatu i priboru koji se na nju stavlja mogućnost greške je smanjena. Ujedno ovim se postiže da radno mesto, za svaku narednu smenu, izgleda potpuno isto kao i prethodnog dana, usled čega je čak i novom radniku, potrebno veoma malo vremena za snalaženje u radnoj okolini. U ovom slučaju radno mesto - radni sto u proizvodnji, je izdignuto nekoliko centimetara od poda čime je obezbeđen lakši pristup pri čišćenju i radno mesto je čistije.

4.3 Kanban

Iz razloga postojanja velikog spektra proizvoda koji se proizvode i potrebama istih ili veoma sličnih proizvoda, sa istih operacija za naredne, a uz to sa velikim amplitudama između operacija, uzrokovano i veličinama serija, potrebno je na određenim mestima uvesti odgovarajući sistem „obaveštenja“. Kako bi pojedine operacije znale koju vrstu proizvoda je kada potrebno proizvoditi.

U okviru operacije za isecanje kablova (prve u proizvodnom toku kablova), potreba za opskrbljivanjem naredne operacije je diskontinualna. Ta potreba nastaje iz razloga što trajanje naredne operacije – prva montaža u okviru proizvodnog toka kablova, zahtevano od strane takta snabdevanja glavne montaže, iznosi 36min, dok je za isecanje kablova (serije veličine 50kom) potrebno svega 1,5min. Odmah se primeti visoka verovatnoća postojanosti velikih količina zaliha na baferu iza operacije sečenja kablova, a kako bi se to izbeglo potrebno je datu operaciju obavestiti „od nazad“, od „kupca“, koji proizvodi su naručeni i koje količine kablova za te proizvode je potrebno iseći.

Kao predlog poboljšanja operacije isecanja kablova, navodi se mogućnost implementacije *Kanban-a*, gde će se određenim informaciono- softverskim sistemom, dostavljati u realnom vremenu informacije za proizvodnju

određenog modela. Informacije bi trebale biti iskazivane na tabletu ili u vidu svetlećih tabli sa naznačenim numeracijama ili nazivima svih modela proizvoda.

Na osnovu dnevnog plana proizvodnje, zahtevanog direktno od strane inostranih kupaca a organizovanog od strane menadžmenta po principima *Heijunka-e*, dostavljale bi se potrebne informacije o redosledu proizvodnje za određeni proizvod, kao i veličine serija koje bi se mogle razlikovati u zavisnosti od modela proizvoda i zahtevanih porudžbina.

4.4 Poka-Yoke

Kao jedan od osnovnih alata *Lean-a* koji obezbeđuje siguran i kontinuiran rad bez mogućnosti prekida i zastoja prouzrokovanih nastankom greške, *Poka – Yoke*, se bazira na obezbeđivanju prvenstveno jednostavnih i ne finansijski prevelikih poboljšanja. Naravno, ako se ne obezbedi takvo adekvatno poboljšanje, pristupa se složenijim analizama kao i finansijski skupljim poboljšanjima. Samo primenom *Poka – Yoke* uspeva se dostići uspešna, samostalna i uzročna inspekcija.

Prema [3] postoje dve vrste *Poka – Yoke* :

- 1) Kontrolna – kada je *Poka – Yoke* aktivirana, mašina ili proces se zaustavlja kako bi problem bio rešen. Kontrolna *Poka – Yoke* je znatno najjači bezbednosni metod iz razloga što ona zaustavlja proces dok se problem ne reši, a isti problem se zatim, ne događa ponovo,
- 2) Upozoravajuća – kada je *Poka – Yoke* aktivirana, zvučni ili svetleći signal se emituje u cilju obaveštavanja radnika na mogući problem, grešku.

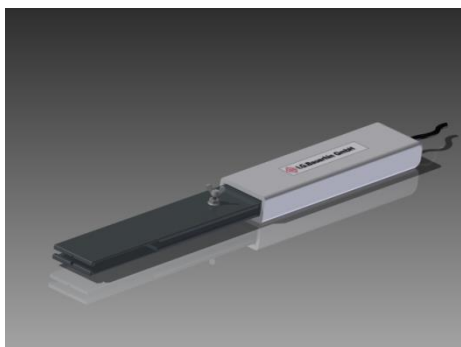
Na radnom mestu, skidanje izolacije sa kabla, kao obavezna operacija iz tehnoloških razloga operacija pre ove, vrši se električnim grejačem u obliku klješta. proizvodi veliki asortiman proizvoda – sedišnih grejača sa različitim temperaturnim i energetskim ospezima, potreban je takođe, veliki asortiman žica različitih prečnika. S toga, po dolasku na ovo radno mesto, radnik iako ima uvid u tip proizvoda a time i u prečnik žice, jedino što mu je omogućeno za adekvatan prelazak sa jednog na drugi tip proizvoda jeste talenat, sigurnost i usresređenost „u rukama“, kako ne bi preterano pritisnuo klješta i time ošteto metalni deo žice.

Kako *Poka – Yoke* nalaže potrebno je obezbediti adekvatan sistem kojim bi se data klješta mogla koristiti sa sigurnošću i bez imalo straha da trenutno neusresređen radnik ili slučajnim dejstvom može oštetiti proizvod, a ujedno i obezbediti da se sada radnik lišen ovog problema može usresrediti na eventualno neki drugi problem i time dodatnim analizama obezbediti narednu eliminaciju problema i time dodatno unapređenje procesa. Upravo sa mogućim usecima na po svakom pokretnom delu klješta kao i sa postavljanjem zavrtnja na jednom od pokretnih delova, (preporučeno gornjem), obezbeđuje se nekoliko stvari:

- radnik po pristizanju serije proizvoda, čiji tip i prečnik žice vidi na transportnoj kutiji, u mogućnosti je da na osnovu prečnika žice

postavi, zavrne ili odvrne, zavrtnaj koliko mu je potrebno,

- takođe, kako je preporučljivo na zavrtnju obeležiti stalnim, neskidajućim bojama nekoliko, bitnih i najčešćih pozicija za postizanje odgovarajućeg prečnika metalnog dela žice, radniku se dodatno olakšava ali i ubrzava rad, smanjivanjem pripremnog vremena i potrebom za veoma preciznim radom,
- izvedenim usecima na „zubima“ pokretnih delova klješta, obezbeđuje se još veća brzina rada, podizanje kvaliteta rada i nema potrebe za doradom.



Slika 5. Prikaz unapređenih klješta za skidanje izolacije

5. ZAKLJUČAK

Zahtevi za napuštanjem tradicionalne, klasične proizvodnje organizovane prema procenama i prognozama proizvođača, sve više su obavezni sa aspekata konkurentnosti preduzeća na tržištu. Iz tog razloga, *Lean* filozofija je, sa svim svojim alatima, pokazala svoju učinkovitost u reorganizaciji bez posledica na preduzeće pri implementaciji predloženih poboljšanja. Kompanija *I.G.Bauerhin* sa proizvodnim pogonom u Indiji, predstavlja klasičan primer preduzeća sa velikim spektrom proizvoda, veoma sličnih, često razlikujući se na osnovu samo jedne operacije.

Značajna prisutnost veoma niskih iskorišćenja radnih mesta i mašina, prisutnost uskih grla i zaliha i njihov direktan uticaj na nepravilnu raspodelu izlaska gotovih proizvoda iz proizvodnje, dokazano simulacionim modelom, predstavljala je osnovu traženja poboljšanja. Uporednom primenom mapiranja toka vrednosti (eng. *Value Stream Mapping*), zahtevanim vremenskim taktom za izlaz gotovog proizvoda, ustanovljene su „rupe“ u vremenu.

Adekvatan i neometan proizvodni tok, bez prisustva zaliha, neiskorišćenosti mašina i ostalog a sa zahtevom postizanja odgovarajućeg *takt time*-a, postignut je sjedinjavanjem odgovarajućih operacija koje to dozvoljavaju sa aspekata prostora, stručnosti i vremena trajanja, smanjivanjem angažovanja radnika koji će biti prvenstveno angažovani na proizvodnji analiziranog proizvoda a u ostatku vremena na drugom proizvodu, kao i upotrebom *Kanban*-a i čuvanja zaliha u obliku "supermarketa".

Najveći problem u celokupnoj proizvodnji predstavlja operacija završne montaže, koja je ocenjena kao „*pacemaker*“ tj. davač takta, kao najduža operacija i potpuno neusklađena sa ostatkom proizvodnje. Postignuto buduće, optimizovano stanje omogućilo je smanjivanje ukupnog vremena trajanja proizvodnje, (eng. *Lead time*), sa prvenstvenih 15912 minuta na 14618 minuta i redukovanje potrebnog broja radnika za ovaj tip proizvoda sa 21 na 14. Kako bi se moglo uopšte i započinjati sa implementacijom *LEAN*-a, potrebno je svako mesto urediti prema principima 5S, što predstavlja da pribore, alate i mašine u okviru radnog mesta treba sortirati, srediti, učiniti „sjajnim“, takvim ga standardizovati i stalno takvim održavati. U okviru isecanja kablova, kao jednog od najkritičnijih po stvaranje zaliha iz razloga što predstavlja veoma brz proces ispred mnogo dužeg procesa, predloženo unapređenje jeste implementacija vizuelnog *Kanban*-a.

U okviru rada na datom primeru pokazano je da *Lean* filozofija ima mogućnosti da se primeni u bilo kojoj industriji, ne ograničavajući se samo na primenu u okviru automobilske industrije. Današnje stanje tržišta dirktira zahteve konkurentnosti i kao takva, preduzeća će biti sve više primorana za okretanjem svojih tradicionalno vođenih organizacija proizvodnje na proizvodnju zasnovanu na principima *Lean*-a, ali samo postepenim i kontinualnim unapređenjima.

6. LITERATURA

- [1] Milosav Georgijević: Tehnička logistika, Zadužbina Andrejević, 2011.
- [2] T.Ohno.: Toyota Production System: Beyond Large Scale-Production (1st edition), Productivity Press, 1988.
- [3] S.Shingo: A Study of the Toyota Production System: From an Industrial Engineering Viewpoint (Produce What Is Needed, When It's Needed), Productivity Press, 1989.
- [4] J.K.Liker: The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer, McGraw-Hill Education, 2004.

Kratka biografija:



Igor Tabački rođen je u Novom Sadu 1992. god. Master rad na Fakultetu tehničkih nauka u Novom Sadu iz oblasti Mašinstva: Projektovanje mašina, transport i logistika odbranio je 2016.god.

PRIMENA SIMULACIJA I LEAN ALATA ZA OPTIMIZACIJU PROIZVODNIH I LOGISTIČKIH PROCESA U KOMPANIJI MK HOLDING-OGRAK POKEDA**APPLICATION OF SIMULATIONS AND LEAN TOOLS FOR THE OPTIMIZATION OF PRODUCTION AND LOGISTIC PROCESSES IN THE COMPANY MK HOLDING-OGRAK POKEDA**

Deana Petrov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Rad obuhvata teorijske postavke logistike proizvodnje, simulacija, tokova materijala i Lean alata, na osnovu čega je, na primeru proizvodnje prikolica, prikazana optimizacija proizvodnih i logističkih procesa u kompaniji MK Holding-Ogranak Pokeda.

Abstract – This paper elaborates the teoretical basics of the production logistics, simulations and LEAN tools. Within the paper, the simulations and LEAN tools are used to optimize production and logistics processes in the production of trailers in the company MK Holding-Ogranak Pokeda.

Кључне речи: LEAN алати, симулације, логистика

1. UVOD

U vreme izrazite konkurencije jedini način za opstanak kompanija na tržištu je optimizacija proizvodnih i logističkih procesa. To prvenstveno znači zadovoljstvo kupca, koje podrazumeva da je željeni proizvod dobio odgovarajućeg kvaliteta u pravo vreme, na pravom mestu po što nižoj ceni. Da bi se to omogućilo potrebno je da tokovi materijala budu neprekidni i fleksibilni, sa minimalnim mogućim zalihama.

Jedna od priznatih metoda optimizacije proizvodnih procesa jeste LEAN, poznatiji kao Tojota Proizvodni Sistem (TPS).

Simulacije predstavljaju efikasan alat za planiranje, organizaciju i optimizaciju proizvodnih i logističkih procesa.

One omogućavaju pravljenje simulacionih modela koji mogu u velikoj meri oponašati realni sistem ili deo sistema i na taj način omogućiti sistemsko posmatranje i uočavanje eventualnih nepravilnosti, uskih grla, problema i mogućnosti za njihovo otklanjanje, kao i vizuelizaciju efekata predloženih izmena.

Korišćenje simulacija predstavlja efikasan način da se pokaže kako neki proces funkcioniše i pritom podstiče razmišljanje kako ga poboljšati, pružajući priliku da se predložena poboljšanja testiraju na simulacionom modelu i tek ukoliko se pokažu efikasnim implementiraju u realan sistem. U ovom radu su simulacije izvršene uz pomoć softvera Enterprise Dynamics.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila doc. dr Sanja Bojić.

2. MK HOLDING-OGRAK POKEDA

Strategija kompanije uključuje usredsređivanje na one poslove u kojima ima dobru tržišnu poziciju, komparativne prednosti i mogućnost liderstva.

Proizvodni planovi u kompaniji usredsređeni su na dva glavna zadatka:

- obavljanje i rast proizvodnje kroz unapređenje procesa rada i razvoj novih proizvoda;
- proširivanje međunarodne saradnje.

Kompanija trenutno uspešno proizvodi traktorsku kiper prikolicu od 15t, podrivače i spojnice.

2.1. PRIKOLICE

Kompanija trenutno razmatra proširivanje proizvodnog programa. Kao potencijalni proizvod se razmatraju prikolice nosivosti 8 i 10t. Cilj je da se kroz simulacije pokaže opravdanost ovakve proizvodnje i postojanje mogućeg "uskog grla". Primena LEAN koncepta i alata treba da omogući optimizaciju procesa i da unapredi planirani sistem.

S obzirom da je uvođenje novih proizvoda uvek rizično traži se dalji razvoj samog proizvoda kao i kontinualno unapređenje proizvodnog procesa. Kompanija kao inicijalni korak razmatra nabavku većine komponenata od lokalnih i inostranih dobavljača dok proizvodnju manjih podsklopova i finalnu montažu planira u sopstvenom pogonu. Za sada je proizvedeno svega nekoliko prikolica u svojstvu prototipa, dok se još uvek razmatra mogućnost proizvodnje većeg obima.

Proizvodnja se sastoji od tri faze. Prva faza se odnosi na proizvodnju šarki. Prikolica se sastoji od četiri stranice, svaku od tih stranica čine dve međusobno povezane polovine. Ta veza između polovina stranice su zapravo šarke koje omogućavaju da se pojedine stranice otvaraju (kompletna stranica ili na sredini) čime je omogućeno pražnjenje prikolice.

Proizvedeni delovi šarki se prvo međusobno zavaruju što je nazvano zavarivanje 1, da bi se nakon toga zavarile na svaku od polovina stranica čineći na taj način jednu celinu. Nakon toga je završena prva faza. Druga faza predstavlja peskiranje i farbanje određenih delova prikolice.

Nakon što je proces sušenja završen pristupa se završnoj tj. finalnoj fazi koja predstavlja montažu svih delova prikolice u jednu funkcionalnu celinu. Nakon toga gotova prikolica napušta pogon i isporučuje se kupcu.

Pre nego što se dublje zađe u problematiku i radi lakšeg razumevanja odmah na samom startu uvešće se tri pojma:

1. Sadašnje stanje – Pod sadašnji stanjem podrazumeva se da u pogonu u kojem će se praviti prikolice ništa nije menjano, raspored mašina je onakav kakav jeste, vremena koja su uzimana su isključivo vremena koja su dobijena na osnovu izrade prototipa prikolice i ostali elementi poput farbare i mesta za peskiranje ostaće na mestu gde se i sad nalaze u preduzeću.
2. Buduće stanje 1 – Pod njim se podrazumeva da su uvedene određene promene. Te promene se odnose na vremena koja su sad za neke procese promenjena jer više odgovaraju realnom stanju, nego u slučaju prototipa (pretpostavljena vremena), zatim umesto peskiranja i farbanja sad se nalaze procesi odmašćivanja, plastifikacije i sušenja u peći. Prostorni raspored mašina je ostao nepromenjen.
3. Buduće stanje 2 – Ovo stanje obuhvata promenjen raspored mašina u pogonu, vremena koja su ovde uzeta za potrebe simulacije i LEAN koncepta su realna vremena i dobijena su ličnim odlaskom u preduzeće i merenjem svakog procesa ponaosob. U odnosu na Buduće stanje 1 zadržani su procesi odmašćivanja, plastifikacije i sušenja u peći, a razlika se ogleda u primeni robotizovanog zavarivanja.

3. SIMULACIONI MODEL

Prilikom primene simulacija i softvera Enterprise Dynamics, od velike važnosti jeste što bolje opisati realan sistem primenjujući pritom odgovarajuće atome i izvršiti njihovo podešavanje na način na koji će što bolje prikazati stvaran sistem. Da bi to bilo moguće neophodno je izvršiti analizu sadašnjeg stanja i svaku mašinu zameniti sa odgovarajućim atomom.

3.1 Sadašnje stanje

Ulaz materijala u pogon predstavljen je preko ArrivalList-a, koji predstavlja svakodnevno donošenja materijala u vidu tabli lima, šipki i cevi kojima se započinje proizvodnja šarki. Svaki dan pre početka rada donose se materijali i smeštaju se u odgovarajuće bafere odnosno, bafer za cevi, lim i šipke konkretno tim redosledom.

Sa plazma sekača material se usmerava na 3 bafera:

- na dva bafera kod zavarivanja 2 za pozicije f i b,
- bafer ispred prese na koju se savija pozicija d, i
- na bafer zavarivanja 1 gde se pozicije d, a i e spajaju sa pozicijama h, g i j koje dolaze sa strugova.

Atom koji se koristio za opisivanje rada strugova je Server. Ono što on uradi jeste da od jedne šipke iseče onoliko koliko je potrebno delova što se kod Server- a lako rešava jednostavnim podešavanjem Batch i Batch rule. Gotove delove sa strugova odnose ljudi do bafera kod zavarivanja 1.

Za prikazivanje rada testere, prese i variočkih stolova kod zavarivanja 1 iskorišćen je ponovo Server jer njihov rad nije previše specifičan ni težak za opisivanje. Zavareni delovi posle toga odnose ljudi u bafere za zavarivanje 2. Da bi se process zavarivanja 2 prikazao što verodostojnije

bilo je neophodno postaviti 5 bafera ispred 3 variočka stola. Prva dva bafera su namenjeni za pozicije f i b koje dolaze sa plazma sekača, a preostala tri su namenjena za šarke a-g, d-h i e-j. Da bi process bio ispravno prikazana bilo je neophodno izvršiti podešavanje tih variočkih stolova kako bi sa svakog bafera uzimali tačno onoliko delova koliko ide na svaku polovinu stranice. Jedini atom koji to opisuje kako treba je Assembler, i on će simulirati rad sva tri variočka stola. Nakon što Assembler završi proces, Transporter (viljuškar) odnosi stranicu po stranicu na peskiranje.

Procesi peskiranja, oba farbanja i sušenja su prikazani preko Server-a, svakom od njih je podešen Cycletime na fiksno 7 sati jer toliko traju procesi osim sušenja jer ono traje 16 sati odnosno do narednog dana. Podešen im je i Batch i Batch rule kako bi se dobilo da isti broj delova uđe i izađe iz procesa.

Nakon poslednjeg sušenja, viljuškar odnosi stranice nazad u pogon gde se vrši montaža. Svaki proces montaže je predstavljen preko Assembler-a, jer je cilj bio da izlaskom iz svakog od njih izlazi jedna celina (dva dela uđu u process, a izlazi jedan) da bi se mogla predstaviti nadgradnja prikolice.

Ovako predstavljen model čini jednu funkcionalnu celinu. Pravilnim odabirom atoma koji će zameniti mašine i procese, kao i njihovim podešavanjem dobijen je simulacioni model koji odgovara realnom sistemu. Dalje je neophodno odrediti dužinu simulacije. Kada se uzme u obzir 365 dana godišnje i kad se odbiju vikendi i neradni dani dobije se okvir od 260 radnih dana, pretvoreno u sate to je 1820 raspoloživih radnih sati u ovom preduzeću.

3.2. Rezultati sadašnjeg stanja

Ono što prvo može da se uoči iz prethodnog simulacionog modela jeste postojanje dva uska grla. Prvo je uzrokovano procesima peskiranja, farbanja i sušenja. Ti procesi uzrokuju veliko čekanje što se može videti po stanju bafera ispred i iza ovih procesa (Bafer 1 i Bafer 2). Razlog za to su predugačka vremena trajanja procesa usled čega trpi cela proizvodnja. Drugi je montaža instalacija, vreme koje je potrebno da se ona postavi je jako dugačko i iz tog razloga ostali procesi imaju mali stepen iskorišćenja. Kao još jedan od problema javlja se i veliko vreme zadržavanja delova u baferima, što predstavlja znak prevelikih zaliha.

4. UNAPREĐENJE SADAŠNJEG STANJA PRIMENOM LEAN ALATA

Posmatrajući prethodni model koji reprezentuje sadašnje stanje pogona uvidelo se da postoje određeni nedostaci. Ti nedostaci su pokušani da se redukuju kroz primenu LEAN alata:

- prvo je primenjeno mapiranje toka vrednosti na osnovu kojeg su uočeni nedostaci koji su se iskazali i kroz simulaciju,
- umesto farbanja i peskiranja uvedena je plastifikacija,
- sledeći korak je bila primena alata Heijunka na osnovu kojeg su se redukovala vremena potrebna za zavarivanje 2,
- primenom 5S-a uvedena su portalna kolica za lakšu manipulaciju sa komponentama prilikom odmašćivanja i plastifikacije,

- promenjeni su dobavljači.

Sledeći korak se ogleda u pravljenju novog modela i mape toka vrednosti (VSM) kako bi se pokazalo da li i u koliko meri uvedene promene utiču na poboljšanje prethodnog modela. U tu svrhu pravi se simulacioni model budućeg stanja 1 i VSM 2.

4.1. Buduće stanje 1

U ovom modelu je posmatran isti raspored mašina u procesu proizvodnje, ali su procesi peskiranja i farbanja zamenjeni procesima odmašćivanja i plastifikacije, uključujući i peć za sušenje gotovih delova.

Ovi procesi su prostorno izmešteni, u prostor u kome je trenutno smešten magacin rezervnih delova, tako da se prostorno nadovezuju na postojeći proces proizvodnje.

Ono što je od velikog značaja jeste da se isprati proces plastifikacije da bi se videli efekti predloženih poboljšanja u krajnjim rezultatima

Sva tri procesa (odmašćivanje, plastifikacija i sušenje u peći) predstavljeni su preko atoma Server. Njihov rad opisan je normalnom raspodelom uz standardno odstupanje od 300 sekundi. Vreme potrebno za odmašćivanje i nanošenje praha ne može biti isto za platformu i za gibnjeve. Iz tog razloga je uzeto odstupanje i normalna raspodela za CycleTime ovih procesa.

Vreme potrebno za sušenje elemenata u peći prema nekim korisnicima je oko 20 minuta. Međutim, preporuka je da se delovi suše sat vremena kako bi se prah što bolje sjedinio na površinama na koje se nanosi i iz tog razloga je upravo to vreme koje je uzeto pri predstavljanju procesa.

Kod odmašćivanja i plastifikacije predviđeno je da se deo po deo procesira što se može predstaviti preko opcija Batch i Batch rule. Što se tiče peći neisplativo je sušenje jednog po jednog dela pa se kao optimalna količina uzima po četiri dela. U praksi postoje različite veličine ovih peći pa je sasvim realno pretpostaviti da veći broj delova može da se suši odjednom

Jednom kada su zamenjeni stvarni procesi sa odgovarajućim atomima i nakon podešavanja procesa, započinje se sa simulacijom za 260 radnih dana, odnosno 1820 radnih sati. Nakon završetka simulacije prelazi se na rezultate.

4.2. Rezultati budućeg stanja 1

Ono što se jasno može videti iz izveštaja je vidno smanjenje zadržavanja elemenata u baferima. Zamenjena peskiranja i farbanja sa plastifikacijom je ubrzala procese i samim tim je rešen jedan problem uskog grla (Baferi 1, 2 i 3). Bafer nazvan "Zalihe" predstavlja magacin u koji se smeštaju nedeljne zalihe.

Problem koji ostaje jeste proces postavljanja instalacija koji je i dalje usko grlo. Proces jeste u potpunosti iskorišćen ali i dalje zbog prevelikog vremena trajanja procesa ostali procesi ostaju neiskorišćeni što se može videti sa dijagrama montaže gibnjeva, točkova i stranica. Plazma sekač se takođe veoma malo koristi što je za jednu tako skupu mašinu neisplativo. Vidno je i smanjenje blokiranja CNC strugova.

Kao promena za sledeći model uvešće se robotizovano zavarivanje i redukovaće se broj strugova, više reči o tome biće u narednoj simulaciji.

4.3. Buduće stanje 2

Ovaj model se sastoji od istih atoma kao i prethodni, osim jednog. Naime umesto zavarivanja 2 kod koga su ranije bila tri variočka stola, ovde imamo dva koja opslužuje robot za zavarivanje. Cilj je bio da se smanjivanjem vremena za obavljanje procesa smanje i zadržavanja u baferima. Takođe se kod zavarivanja 1 umesto dva variočka stola sada nalazi samo jedan za kojim rade dva čoveka na privarivanju manjih delova (šarki). Pošto je ranije pokazano da ne postoji potreba za tolikim brojem strugova u ovoj simulaciji su zadržana samo dva. Na ovaj način će se pokazati da li sistem može da funkcioniše samo sa dva struga kako bi se ostali mogli upotrebljavati za neke druge poslove. Prostorni raspored mašina je promenjen i sad se process odvija sa desna na levo. U prethodnom modelu procesi odmašćivanja, plastifikacije i sušenja su postavljeni na mestu magacina. U ovom modelu nakon razmeštaja ostalo je dovoljno mesta da se ti procesi smeste u istu halu sa proizvodnjom. Na taj način je zadržan magacin u prostoriji iznad da bi se smanjilo nepotrebno kretanje prilikom donošenja materijala iz magacina, mašine su izmeštene tako da se nalaze u njegovoj neposrednoj blizini i smanjuje se prostor potreban za proizvodni proces usled čega se dovodi do prostornih ušteda koje omogućavaju smeštanje kompletne proizvodnje u samo jednu postojeću proizvodnu halu.

Osnovni zadatak ovog modela je bio da se svi procesi smeste u jednu halu, da se dodatno smanje zadržavanja delova u baferima i da se mašine dovoljno približe jedna drugoj kako ne bi imali gubitaka u vidu nepotrebnog kretanja. Kako je ovaj model identičan po atomima prethodnom modelu ovde se neće ponovo razmatrati ti atomi i njihovo podešavanje, već će se preći na analizu rezultata.

4.4. Rezultati budućeg stanja 2

Na osnovu analize rezultata vidi se da su dodatno smanjena zadržavanja u baferima. Bafer 1 se nalazi ispred procesa odmašćivanja i vidi se da u njemu nema zadržavanja. Kako stranica napusti variočke stolove na zavarivanju 2 tako se odmah započinje sa procesom odmašćivanja. Bafer 2 se nalazi između plastifikacije i peći i u njemu postoji malo zadržavanje materijala koje odgovara činjenici da po četiri dela ulaze na sušenje. Bafer 3 se nalazi ispred montaže i u njega ulaze delovi koji dolaze iz peći. Kako se može primetiti u njemu postoji malo zadržavanje koje odgovara vremenu potrebnom da se delovi odnesu na kolicima do montaže. Bafer koji je nazvan "Zalihe" predstavlja magacin u koji se smeštaju nedeljne zalihe, a zadržavanje u njemu odgovara prosečnom vremenu zadržavanja materijala.

5. PREGLED REŠENJA

Da bi proizvodnja iz sadašnjeg stanja mogla da postigne kapacitet koji ima Buduće stanje 2 moralo se stvarati više zaliha. Više materijala se moralo dovoditi mašinama kako bi prvi slučaj mogao proizvoditi 285 prikolica koliko proizvodi i poslednji slučaj bez stvaranja nepotrebnih zaliha. Budućem stanju 1 je takođe bilo neophodno

povećati zalihe da bi mogao da dostigne kapacitet budućeg stanja 2 ali ne u tolikoj meri koliko je trebalo u prvom slučaju. Sve to može da se isprati po zadržavanju delova po baferima. Iz rezultata sva tri stanja se može zaključiti da je buduće stanje 2 dalo najbolje rezultate. Buduće stanje 1 je takođe primenljivo jer ne zahteva prevelike zalihe i daje dobre rezultate. Sadašnje stanje ne treba primenjivati, vremena su prevelika, stvaraju se velika čekanja, postoje dva uska grla i zalihe koje su potrebne su jako velike. Jedno usko grlo nije rešeno ni u jednom slučaju i ono je rešivo jedino ako se dužina procesa smanji, da li kroz promenu tehnologije, organizacije samog procesa ili uvođenjem dodatne radne snage na tom konkretnom procesu. Naredna tabela ilustruje poboljšanja koja su se kontinualno dostizala uvođenjem odgovarajućih promena počevši sa sadašnjim stanjem pa sve do budućeg stanja 2. U tabeli su prikazana prosečna vremena zadržavanja delova u različitim baferima. Primetno je da su se vremena smanjivala polazivši od sadašnjeg stanja pa do budućeg stanja 2.

Tabela 2 Poređenje vremena (u sekundama)

	Sadašnje stanje	Buduće stanje 1	Buduće stanje 2
Bafer za cevi	0	0	0
Bafer za lim	23	23	8
Bafer za šipke	0	0	0
Bafer a, g, e, j	3221	2242	896
Bafer za f i cev	1005595	746610	280
Bafer za b	575667	34223	345
Bafer za a-g	537364	4142	450
Bafer za e-j	582947	3630	458
Bafer za d-h	519627	3095	458
Bafer 1 (pre peskiranja)	1262582	770	0
Bafer 2 (posle peskiranja)	1810987	1437	X
Bafer 3	X	7201	7260

6. ZAKLJUČAK

Savremeni proizvodni sistemi znaju biti izrazito složeni i skupi. Da bi se opravdala ulaganja u nove proizvode neophodno je doneti prave odluke. Donošenje ispravnih odluka je otežano usled nemogućnosti eksperimentisanja sa realnim proizvodnim sistemom, da bi se nakon nekoliko eksperimenata, donela odluka koja će ispuniti očekivanja. U takvim uslovima, kao pomoć pri donošenju odluka, primenjuju se simulacije. Govoreći upšteno proizvodnja i jeste oblast u kojoj je neophodno

primenjivati simulacije kao alat za optimizaciju proizvodnih ali i logističkih procesa. Iz tog razloga se u ovom radu za unapređenje proizvodnog sistema koriste oba pojma (LEAN koncept i simulacije). Na primeru u radu je pokazano da, iako proizvodnja još ne postoji, pomoću simulacija može da se napravi model i izvrši njegovo testiranje još u ranim fazama planiranja proizvodnje kako bi se ustanovili potencijalni problem i isti rešili. Primenom LEAN koncepta se izvršila optimizacija proizvodnje tako da je ona postala ekonomičnija i isplativija. Kombinovanom primenom simulacija i LEAN koncepta ostvaruju se dobri rezultati, a sve u cilju postizanja optimalnih proizvodnih i logističkih procesa.

7. LITERATURA

- [1] Bulatović, M. (2013): LOGISTIKA, Inženjerska komora Crne Gore, Podgorica.
- [2] <http://documents.tips/documents/logistika-proizvodnje-1.html>, pristupljeno 07.10.2016
- [3] Georgijević, M. (2010): Tehnička logistika, Zadužbina Andrejević, Beograd.
- [4] Rother, M., Shook, J. (1999): Learning to see, THE LEAN ENTERPRISE INSTITUTE, Brookline, Massachusetts, USA.
- [5] Georgijević, M. (2001): Logistika i simulacije, Univerzitet u Novom Sadu, Novi Sad.
- [6] Kaurin, T., Bulatović, N., Anuđević, D.: Fakultet za pravne i poslovne studije, Novi Sad.
- [7] TUTORIAL Enterprose Dynamics 9. Inncontrolsim, 2015.
- [8] Čosić, I., Lazarević, M., Sremčev, N., Lazarević, N. (2015): Proizvodne strategije (LEAN, EFPS, CIM), Fakultet tehničkih nauka, Novi Sad.
- [9] Masaki Imai: Kaizen-Ključ japanskog poslovnog uspeha, Mono I Manjana, 2008
- [10] Rich, N., Bateman, N., Esain, A., Massey, L., Samuel, D. (2006): LEAN EVOLUTION, Cambridge University Press, UK.
- [11] Shingo, S. (1989): A Study of the Toyota Production System, Productivity Press, SAD.
- [12] Harris, R., Harris, C., Wilson, E. (2003): Making Materials Flow, The Lean Enterprise Institute, Ink., Brookline, MA, USA.

Kratka biografija:

Deana Petrov rođena je u Novom Sadu 1988. god. Master rad branila je na Fakultetu tehničkih nauka iz oblasti - Projektovanje mašina, transportni sistemi i logistika, 2016.god.



KRATKOROČNO PREDVIĐANJE POTROŠNJE PRIRODNOG GASA SHORT-TERM PREDICTION OF NATURAL GAS CONSUMPTION

Milan Dugošija, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – U ovom radu obrađena je primena neuronskih mreža za kratkoročno predviđanje potrošnje prirodnog gasa u distributivnom sistemu. Kreirana je neuronska mreža koja je iskorišćena za prognoziranje potrebnih količina prirodnog gasa za naredni dan, na osnovu istorijskih podataka o potrošnji dobijenih od distributivnog preduzeća i meteoroloških podataka. Prikazani metod može se smatrati adekvatnim pomoćnim sredstvom koje operateri u distributivnim sistemima mogu koristiti za rezervisanje potrebnih količina prirodnog gasa kod preduzeća koje se bavi transportom gasa.

Abstract – This paper presents the application of neural networks for short-term forecasting of natural gas consumption in the distribution system. A neural network is created and used for forecasting the required quantities of natural gas for the next day based on historical consumption data received from the distribution companies and meteorological data. The present method can be considered adequate adjuvant that operators in distribution systems can use to reserve the necessary amount of natural gas from companies dealing with the transport of gas.

Ključne reči: Prirodni gas, predviđanje potrošnje, neuronske mreže

1. UVOD

Za efikasan rad distributivnog sistema neophodno je unapred predvideti potrebne količine prirodnog gasa za naredni period. U ovom radu su predstavljene metode kojima se vrši predviđanje potrošnje gasa, na osnovu istorijskih podataka o potrošnji i temperaturi vazduha, vremenske prognoze za naredni period, kao i drugih faktora. Akcenat je stavljen na metodu veštačkih neuronskih mreža, za koju su prikazane teorijske osnove i napravljen model koji je primenjen na realnim podacima dobijenih od distributera gasa. Zatim su analizirane performanse modela i doneti zaključci o koracima koje je potrebno sprovesti da bi se dobila što preciznija prognoza potrošnje gasa u budućnosti. Prikazani metod sa greškom manjom od 7% može se smatrati odgovarajućim pomoćnim sredstvom za operatere u distributivnim sistemima, koji se bave nominacijama količina prirodnog gasa koje je potrebno rezervisati kod transportera gasa.

Tržište uslovljava distributivna preduzeća da ponude što konkurentnije cene gasa, zbog čega ona moraju svoje operativne troškove svesti na najmanju moguću meru.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Siniša Bikić.

2. PREGLED STANJA U OBLASTI

Da bi se to postiglo, funkcionisanje sistema mora biti usklađeno sa potrebama potrošača. Kako se potrebe potrošača menjaju tokom vremena distributivni sistem mora biti u stanju da prati te promene i odgovori na njih na najefikasniji mogući način. Optimizacija funkcionisanja sistema se zasniva na predviđanju budućih potreba za energijom. Što je ovo predviđanje preciznije, to su manji skladišni kapaciteti potrebni da se održi stabilnost, čime se redukuju i troškovi [1].

Ako se filtrira popis radova koji se bave prognoziranjem potrošnje prirodnog gasa može se uočiti da je u zadnje tri decenije objavljeno oko 75 % radova iz čega se može zaključiti da je prognoziranje potrošnje prirodnog gasa aktuelni istraživački problem [2].

Za razliku od oblasti električne energije, gde postoje brojni javno dostupni setovi podataka, tehnika i aplikacija vezanih za ovu problematiku, aktivnosti na polju prirodnog gasa su još u početnoj fazi [3]. Zbog visokog nivoa analogije između potrošnje električne energije i prirodnog gasa, brojni autori su koristili reference vezane za predviđanje potrošnje električne energije, što je i ovde primenjeno.

Detaljan pregled radova koji se bave predviđanjem potrošnje prirodnog gasa dat je u radu [4]. Može se videti da se autori bave ovom problematikom još od sredine prošlog veka. Najčešće tehnike koje su autori koristili su regresiona analiza i veštačke neuronske mreže.

Veštačke neuronske mreže korišćene su u radu [1] kako bi se predvidela potrošnja prirodnog gasa u Sloveniji i ekonomski rizici koji potiču od nepouzdanosti predviđanja. Statistički model koji se koristi u istu svrhu u Češkoj prikazan je u radu [5]. Obuhvaćeni su individualni i manji industrijski potrošači, pri čemu su korišćeni i podaci sa zbirnih merenja (merno - regulacione stanice) i sa individualnih brojlara. Prikazani statistički model zasnovan je na principu nelinearne regresije, a odstupanja predviđenih od realnih vrednosti su se kretala od 5 do preko 15 %.

Višeslojna neuronska mreža (MLP - multilayer perceptron) prikazana je u radu [6], sa sigmoidnom i hiperboličnom aktivacionom funkcijom i najmanjom postignutom greškom od 9,36 %. Predviđanje je vršeno za svaki sat, pri čemu su istaknuti kritični sati u kojima se javljaju najveća odstupanja. U radu [7] takođe je prikazana višeslojna veštačka neuronska mreža. Korišćeni su podaci sa jedne merno-regulacione stanice sa koje se u najvećoj meri snabdevaju domaćinstva. Srednja procentualna greška se kretala između 4 i 20 %.

3. TEORIJSKE OSNOVE

Za kratkoročno predviđanje potrošnje prirodnog gasa najčešće se koriste sledeće tehnike:

1. višestruka linearna regresija;
2. metoda vektora podrške i
3. veštačke neuronske mreže.

Metoda višestruke linearne regresije je nastarija metoda i tradicionalno se koristi u ovoj oblasti. Ukoliko se problem koji se posmatra može tretirati kao problem jedne zavisne i više nezavisnih promenljivih, radi se o pogodnoj situaciji za analizu podataka metodom višestruke regresije. Ako je veza između njih linearna, slučaj se svodi na višestruki linearni model.

Metoda vektora podrške je metoda novijeg datuma i predstavlja jedan vid veštačke inteligencije iz familije učenja sa nadzorom za koju se smatra da će se u budućnosti vrlo uspešno koristiti za prepoznavanje oblika i obrazaca, klasifikaciju, regresiju, predikciju i drugo.

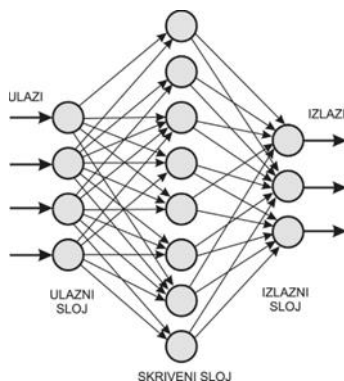
Veštačke neuronske mreže (VNM) svojom strukturom i funkcijom oponašaju funkcionisanje bioloških nervnih sistema (mozga), tako da je omogućeno učenje pojmova, prepoznavanje šablona i donošenje odluka na način koji je sličan čovekovom. Neuronska mreža se ne programira eksplicitno da uči, ona to radi sama, isto kao i mozak. Za razliku od ljudskog mozga, koji se sastoji od 10^{11} neurona pri čemu je svaki neuron povezan u proseku sa deset hiljada susednih, veštačke neuronske mreže sadrže znatno manji broj neurona, pa tako čak i najveće mreže danas imaju do nekoliko hiljada neurona.

Postoji nekoliko strategija učenja koje se koriste pri obučavanju neuronskih mreža:

1. Nadzirano učenje ("supervised learning")
2. Nenadzirano učenje ("unsupervised learning")
3. Podržano učenje ("reinforcement learning")

Razvijeno je oko 50 različitih vrsta VNM, pri čemu se u okviru oblasti kratkoročnog predviđanja potrošnje najčešće koriste VNM sa propagacijom signala unapred (eng. Feed Forward ANN) [8].

Veštačke neuronske mreže su obično organizovane u slojevima, a slojevi su sastavljeni od određenog broja međusobno povezanih neurona koji sadrže aktivacionu funkciju. Podaci se unose u mrežu preko ulaznog sloja, koji komunicira sa jednim ili više skrivenih slojeva, gde se obrada podataka vrši sistemom veza sa određenom numeričkom težinom. Skriveni slojevi se zatim povezuju sa izlaznim slojem, koji daje izlazne podatke kao odgovor (Slika 1).



Slika 1. Slojevita organizacija neuronske mreže

Model neurona prikazan je na slici (Slika 2). Izlazni signal iz neurona se može izraziti u funkciji ulaznih signala [7]:

$$y = f \sum_i^n (w_i x_i - b),$$

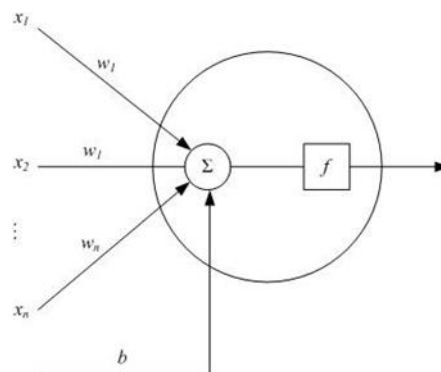
gde su: y - izlaz iz neurona;

x_i - ulazni signali;

w_i - podesivi težinski faktori;

b - podesivi skalarni parameter (bias) i

f - funkcija aktivacije.



Slika 2. Model neurona u veštačkoj neuronskoj mreži

Što je veći "težinski faktor" veštačkog neurona, to će biti veći uticaj ulaznog podatka koji se množi tim težinskim faktorom na izlaz iz neurona. Težinski faktori mogu biti i negativni, tako da se može reći da je signal inhibiran negativnim težinskim faktorom. Podešavanjem težinskih faktora veštačkih neurona mogu se dobiti željeni izlazni podaci za dati ulaz. Međutim, ako se mreža sastoji od nekoliko desetina ili stotina neurona, bilo bi komplikovano ručno pronaći prave vrednosti težinskih faktora, te se za to koriste odgovarajući algoritmi. Ovaj proces se naziva učenje ili treniranje mreže.

Za aktiviranje neurona mogu se koristiti različite aktivacione funkcije: hiperbolična funkcija, sigmoidalna funkcija ili radialno zasnovane funkcije. Svi neuroni istog sloja imaju istu aktivacionu funkciju. Najčešće se koristi sigmoidalna funkcija koja se može predstaviti kao [9]:

$$\sigma = 1 / (1 + e^{-y}).$$

Treniranje neuronske mreže u osnovi podrazumeva podešavanje "težinskih faktora" ponavljanjem dva ključna koraka: propagacije unapred (eng. forward propagation) i propagacije unazad (eng. back propagation).

U propagaciji unapred, set težinskih faktora se primenjuje na podatke sa ulaznog sloja kako bi se sračunali rezultati na izlaznom sloju. Za prvi korak propagacije unapred, uzima se proizvoljan set težinskih faktora. Propagacija unazad koristi princip učenja pod nadzorom: meri se greška na izlazu poređenjem dobijenih i željenih rezultata i podešavaju se težinski faktori kako bi se ona smanjila. Neuronska mreža ponavlja oba ova koraka sve dok se težinski faktori ne podeše tako da mogu da predvide tačnu vrednost izlaza.

Za evaluaciju performansi modela ovom radu korišćena je srednja apsolutna procentualna greška (MAPE), koja predstavlja meru tačnosti predviđenih vrednosti u odnosu na izmerene, izraženu u procentima:

$$MAPE = \frac{100}{n} \sum_{i=1}^n \frac{|y_i - f_i|}{|y_i|},$$

gde su:

- n - broj posmatranih vrednosti;
- y_i - izmerene vrednosti;
- f_i - prognozirane vrednosti.

4. PRIMENA NEURONSKE MREŽE ZA KRATKOROČNO PREDVIĐANJE POTROŠNJE PRIRODNOG GASA

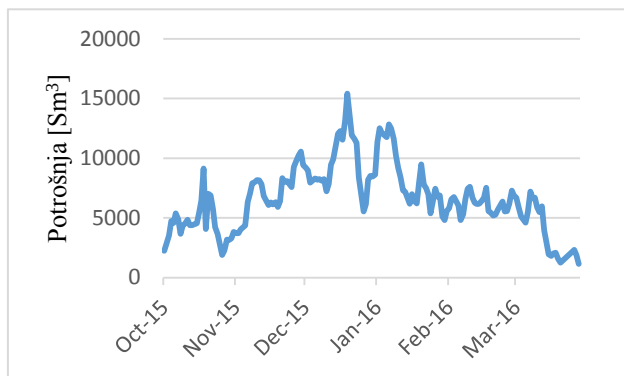
4.1. Ulazni podaci

Podaci o potrošnji prirodnog gasa korišćeni u ovom radu dobijeni su od preduzeća za distribuciju prirodnog gasa DP "Novi Sad - Gas", za merno - regulacionu stanicu Šangaj. Dostupni su podaci o potrošnji u periodu od avgusta 2013. do juna 2016. godine.

Kako je potrošnja u letnjim mesecima (reda veličine 500 Sm³/dan ili svega 5% zimske potrošnje) u datom distributivnom sistemu mnogo manja u odnosu na zimsku potrošnju (reda veličine 10000 Sm³/dan tokom januara), može se zaključiti da se gas u ovom distributivnom području koristi pretežno za zagrevanje objekata (a mnogo manje za kuvanje i pripremu sanitarne tople vode), kao i da ne postoji značajan uticaj industrijskih delatnosti na potrošnju gasa.

Klimatološki podaci za meteorološku stanicu Rimski Šančevi preuzeti su sa sajta Republičkog hidrometeorološkog zavoda Srbije [10]. Dostupni su meteorološki godišnjaci za svaku godinu, a za svaki dan su prikazani vazdušni pritisak, ekstremne, terminske i srednje vrednosti temperature i relativne vlažnosti vazduha.

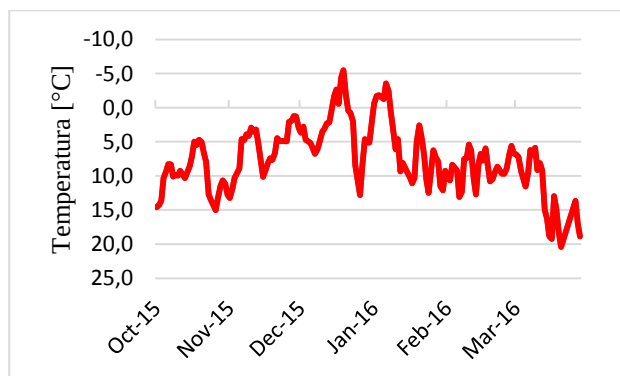
Na sledeća dva grafika prikazane su vrednosti srednje temperature i potrošnje za grejnu sezonu 2015/16. Skala za temperaturu (Grafik 2) je okrenuta kako bi se pokazala sličnost sa grafikom koji prikazuje potrošnju prirodnog gasa (Grafik 1).



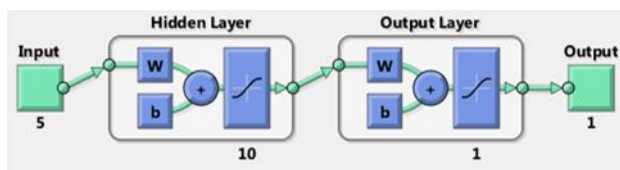
Grafik 1. Potrošnja prirodnog gasa u sezoni 2015/16

4.2. Model neuronske mreže

Neuronska mreža napravljena je u programskom paketu "MATLAB". Korišćen je tip neuronske mreže sa učenjem pod nadzorom. Učenje se vrši metodom propagacije unazad, pri čemu se meri greška dobijena poređenjem dobijenih i željenih rezultata i podešavaju se težinski faktori sa ciljem smanjenja greške. Kreirana neuronska mreža sastojala se (u iteraciji u kojoj je pokazala najbolje performanse) od 5 ulaznih slojeva, jednog skrivenog sloja od 10 neurona i jednog izlaznog sloja (slika 3).



Grafik 2. Spoljašnje temperature u sezoni 2015/16

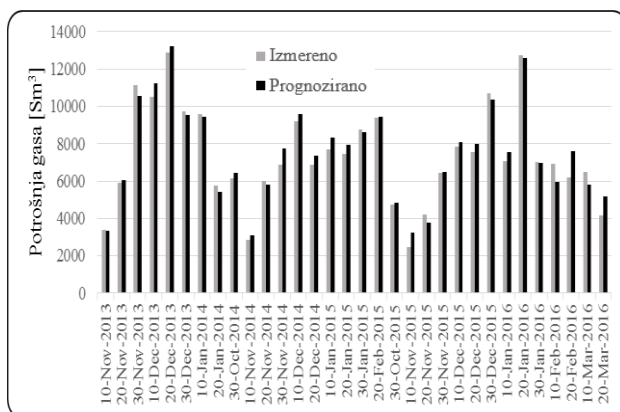


Slika 3. Prikaz kreirane neuronske mreže

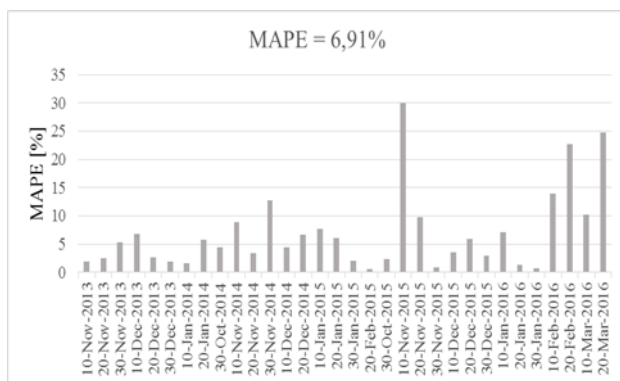
Od oko 400 dana koji su bili dostupni u bazi podataka, većina je korišćena za treniranje mreže, a 25-35 dana je izdvojeno kako bi se testirale performanse modela.

4.3. Rezultati i diskusija

Testirani su modeli neuronskih mreža na više različitih setova ulaznih podataka. Najbolje rezultate pokazao je model čiji su ulazni podaci bili srednja temperatura za posmatrani dan, kao i srednje temperature i potrošnje gasa za dva prethodna dana. Prognoze dobijene na ovaj način prikazane su na Grafiku 3, a performanse modela izražene kroz kriterijum MAPE na Grafiku 4.



Grafik 3. Stvarne i prognozirane vrednosti potrošnje



Grafik 4. Performanse modela za predviđanje potrošnje

MAPE kod najjednostavnijeg modela koji je kao ulaz imao samo temperaturu posmatranog dana iznosila je 21,49 %. Vlažnost vazduha se pokazala kao irelevantan ulazni podatak. Najbolje prognoze postignute su za najhladnije mesece, dok su u prelaznom periodu rezultati nezadovoljavajući.

5. ZAKLJUČAK

U ovom radu prikazane su metode kojima se vrši prognoziranje potrošnje prirodnog gasa u distributivnom sistemu. Obradena je metoda veštačkih neuronskih mreža i prikazane teorijske osnove ove metode. Takođe su prikazane tehnike za određivanje performansi modela kojim se vrši predviđanje potrošnje.

Kreirana je veštačka neuronska mreža u softverskom paketu "MATLAB". U master radi je dato uputstvo kako je moguće kreirati i trenirati veštačku neuronsku mrežu. Prikupljeni su podaci o potrošnji prirodnog gasa od distributivnog preduzeća, kao i meteorološki podaci, prema kojima je izvršeno treniranje neuronske mreže. Podaci o potrošnji prikazani su u prilogu rada i mogu se koristiti u daljim istraživanjima. Nakon nekoliko iteracija, napravljen je model koji ima zadovoljavajuće performanse (MAPE = 6,97 %). Ulazni podaci obuhvatali su srednju temperaturu za dan predviđanja i prethodna dva dana, kao i potrošnju za prethodna dva dana. Vlažnost vazduha proglašena je irelevantnim podatkom.

U budućim istraživanjima potrebno je analizirati uticaj još nekih faktora koji mogu uticati na potrošnju, kao što su: cena gasa, intenzitet solarne radijacije, padavine, smer i brzina vetra, pa čak i raspored godišnjih odmora.

Početak i kraj grejne sezone pokazali su se kao periodi u kojima je teško izvršiti precizno predviđanje potrošnje (greška od nekoliko desetina procenata), dok se u najhladnijim mesecima vrši najpreciznije predviđanje (greška od nekoliko procenata).

Prikazani metod može se smatrati odgovarajućim pomoćnim sredstvom za operatore u distributivnim sistemima, koji se bave nominacijama količina prirodnog gasa koje je potrebno rezervisati kod transportera gasa. Kako bi se obezbedila što veća tačnost neophodno je da model ima dinamički karakter, tj. da se svaki dan ponovo vrši treniranje neuronske mreže i podešavanje parametara u skladu sa novim podacima. Tako bi se izbegli i problemi koji se mogu javiti pojavom novih potrošača u sistemu i gašenjem postojećih.

Iako je u ovom radu analizirana potrošnja prirodnog gasa, prikazani metod je, uz izvesne modifikacije, moguće primeniti i na druga energetska tržišta, a pre svega na tržište električne energije. Takođe, metod nije ograničen veličinom sistema, te se može primeniti i na širem nivou, kao što je regionalni ili nacionalni.

6. LITERATURA

- [1] Potočnik, P., Marko, T., Govekar, E., Grabec, I., Poredoš, A., (2007). "Forecasting risks of natural gas consumption in Slovenia", *Energy Policy*; 35(8), 4271–4282
- [2] Svalina, I., Šimunović, G., Šimunović, K., Šarić, T., Lujčić, R., Pezer, D., Havrlišan, S., (2015). "Usporedba aproksimacija raznih regresijskih modela pri prognoziranju potrošnje prirodnog plina", *Zbornik radova sa 13. skupa o prirodnom plinu, toplini i vodi*, 229-238
- [3] Fagiani, M., Squartini, S., Gabrielli, L., Spinsante, S., Piazza, F., (2015). "A review of data sets and load forecasting techniques for smart natural gas and water grids: Analysis and experiments", *Neuro computing*; 170, 448–465
- [4] Soldo, B., (2012). "Forecasting natural gas consumption", *Applied Energy*; 92, 26-37
- [5] Vondraček, J., Pelikan, E., Konar, O., Čermakova, J., Eben, K., Maly, M., Brabec, M., (2007). "A statistical model for the estimation of natural gas consumption", *Applied Energy*; 85(5), 362–370
- [6] Tonković, Z., Zekić - Sušac, M., Somolani, M., (2009). "Predicting natural gas consumption by neural networks", *Technical Gazette*; 16, 51-61
- [7] Ivezić, D., (2006). "Short - term natural gas consumption forecast", *FME Transactions*; 34 (3), 165-169
- [8] Ilić, S., (2013). "Kratkoročno predviđanje potrošnje električne energije u velikim elektroenergetskim sistemima", *Doktorska disertacija, Fakultet tehničkih nauka, Univerzitet u Novom Sadu*
- [9] Gavran, S., (2016). "Veštačke neuronske mreže u istraživanju podataka: pregled i primena", *Master rad, Matematički fakultet, Univerzitet u Beogradu*
- [10] http://www.hidmet.gov.rs/latin/meteorologija/klimatologija_godisnjaci.php; pristupljeno 20.07.2016.

Kratka biografija:



Milan Dugošija rođen je u Sremskoj Mitrovici 1991. god, gde je završio osnovnu i srednju školu. Diplomirao je 2014. godine na Fakultetu tehničkih nauka, studijski program Mašinstvo, smer Energetika i procesna tehnika, sa prosečnom ocenom 9,79. Zaposlen u kompaniji „Naftna Industrija Srbije“.

RAZVOJ APLIKACIJE BAZE PODATAKA ZA MODELIRANJE OBRADNIH PROCESA

DEVELOPMENT OF DATABASE APPLICATION FOR MACHINING PROCESSES MODELING

Branko Pavkov, Marin Gostimirović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – U radu je prezentovano projektovanje baze podataka i razvoj aplikacije za modeliranje obradnih procesa. Aplikacija omogućuje izbor reznih pločica, držača alata, mašina alatki i merodavnih režima obrade, kao i proučavanje eksperimentalnih ispitivanja. Tehnološka baza podataka je razvijena pomoću Microsoft SQL Server 2008. Aplikacija je razvijena pomoću programskog jezika Visual C#.

Abstract – The aim of this study is to present the design and development of application for machining processes modeling. Application provides selection of inserts, tool holders, machines, cutting data and study of experiments. Technological database is developed using Microsoft SQL Server 2008. GUI application is developed using Microsoft Visual C#.

Ključne reči: Baze podataka, Parametri režima obrade, Modeliranje procesa obrade.

1. UVOD

Dosadašnja istraživanja u savremenim uslovima proizvodnje u industriji prerade metala pokazala su da tehnološka baza podataka predstavlja osnov za unapređivanje tehnološke pripreme proizvodnje i tehnoloških procesa po pitanju ekonomičnosti i proizvodnosti obradnih procesa. Upotrebom tehnoloških baza podataka omogućuje se automatizovano određivanje elemenata režima rezanja kao i optimizacija samog procesa što predstavlja osnovu za razvijanje automatizovanih (CAPP), automatskih (CAM) ili integrisanih (CIM) sistema za projektovanje tehnologije, odnosno za tehnološku pripremu proizvodnje.

Moderni obradni procesi zahtevaju ekonomične parametre obrade i visoku pouzdanost preformansi sistema. Proizvodne operacije su jedan od najvažnijih faktora koji značajno utiču na završne troškove proizvodnje, pa su radi potrebe projektovanja ekonomičnog tehnološkog procesa razvijene mnoge matematičke metode i modeli za izbor i optimizaciju parametara režima obrade.

Cilj ovog rada je da se definiše koncepcija baze podataka za modeliranje procesa obrade struganjem i da se ponudi koncept aplikacije za izbor i unapređivanje postupka izbora merodavnih režima obrade kroz eksperimentalna ispitivanja, upotrebom savremenog sistema za upravljanje bazama podataka Microsoft SQL Server u kombinaciji sa programskim jezikom Visual C#.

NAPOMENA:

Ovaj rad je proistekao iz master rada čiji mentor je bio Marin Gostimirović.

2. STRUKTURA PODATAKA OBRADNOG PROCESA

Analizirajući stanje primene tehnologije obrada materijala u savremenoj proizvodnji, dolazi se do zaključka da danas najšire područje industrijske primene zauzima tehnologija obrade skidanjem materijala. Ova tehnologija obuhvata konvencionalne i nekonvencionalne postupke oblikovanja radnih predmeta odstranjivanjem dela materijala. Kod konvencionalnih postupaka skidanje materijala se ostvaruje obrazovanjem strugotine procesom rezanja sa tvrdim alatom određenog geometrijskog oblika, dok se kod nekonvencionalnih postupaka odnošenje čestica materijala ostvaruje različitim fizičko-hemijskim mehanizmima.

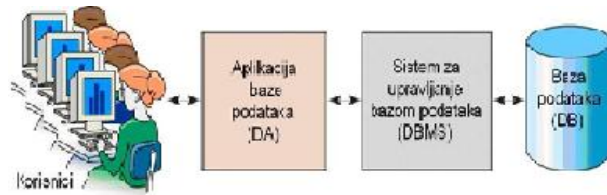
Obradni sistem je elementarna funkcionalna celina proizvodnog sistema koji je neposredni stvaralac proizvoda. Osnovnu strukturu obradnog sistema čine proizvodna sredstva i obradni proces.

3. TEORIJA BAZA PODATAKA

Proces projektovanja i implementacije informacionog sistema se ne može odvojiti od razvoja i izgradnje njemu pripadajuće baze podataka.

Proces modeliranja baze podataka se sastoji od tri osnovne grupe aktivnosti (slika 3.1), koje se praktično izvajavaju paralelno:

- izgradnja strukture i sadržaja baze podataka,
- uvođenje i integracija sistema za upravljanje bazom podataka i
- dizajniranje postupaka obrade podataka u okviru korisnički orijentisanih programa



Slika 3.1 Osnovne aktivnosti procesa modeliranja baze podataka

4. BAZA ELEMENATA REŽIMA OBRADE

Visokoautomatizovana i fleksibilna proizvodnja zasniva se na neophodnosti velike količine pravovremene i pouzdane informacije, pa se u tom smislu teži sistemizaciji svih podataka i znanja u obliku pogodnom za primenu širokog kruga inženjerske populacije. U prvom momentu radi se o sređivanju i arhiviranju prethodno prikupljenih podataka, da bi se potom prešlo na direktno praćenje i prikupljanje tekućih stanja, pojava i procesa obrade.

Centralno mesto tehnološke baze podataka namenjeno je segmentu vezanom za elemente režima obrade.

Permanentno istraživanje, ispitivanje i integracija obradnih procesa omogućuje izgradnju sveobuhvatne baze režima obrade (preporučene, analitičke i optimalne vrednosti), kao osnov pripreme, projektovanja i vođenja procesa proizvodnje.

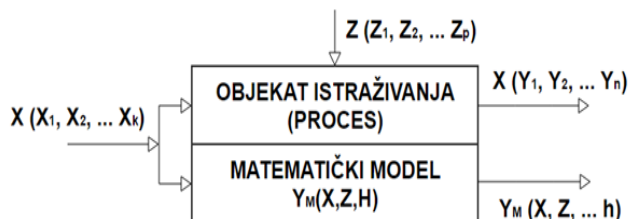
5. MODELIRANJE PROCESA OBRADE PRIMENOM FAKTORNOG PLANA EKSPERIMENTA

Modelima se opisuju i simuliraju određene zakonitosti mehanike, termodinamike i tribologije procesa rezanja. Tačan i sveobuhvatan matematički model omogućuje razvoj i unapređenje sistema:

- Projektovanja tehnoloških postupaka obrade
- Praćenja i upravljanja procesa obrade rezanjem

U cilju uspešnog istraživanja procesa obrade ili pri projektovanju novih procesa ili sistema danas se često primenjuje kibernetički prilaz “crne kutije”, slika 5.1. Kod ovog sistema su poznati:

- Ulazni faktori (X – kontrolisani ili upravljački faktori)
- Izlazni faktori (Y – funkcije stanja ili karakteristike procesa dobijene merenjem)
- Ograničenja ili poremećaji (Z – nekontrolisane veličine koje obuhvataju sistematske ili slučajne poremećajne faktore).



Slika 5.1 Kibernetički prilaz modeliranju pojava, procesa ili sistema

6. SQL (STRUCTURED QUERY LANGUAGE)

SQL je programski paket koji se zasniva na relacionoj algebri i relacionom računu. Skraćenica SQL dolazi iz naziva *Structured Query Language*, što bi u slobodnom prevodu značilo *Strukturni jezik za postavljanje upita*.

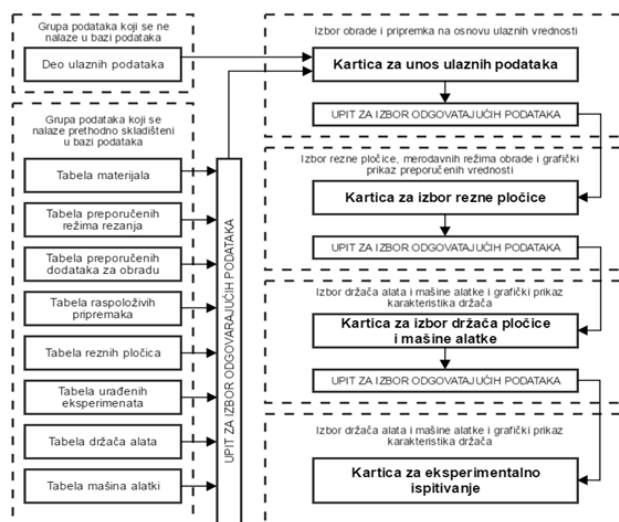
Programski paket SQL omogućava jednostavno :

- kreiranje relacija,
- unos podataka,
- brisanje,
- ažuriranje,
- pretraživanje podataka,
- prezentaciju novih informacija.

7. KONCEPT TEHNOLOŠKE BAZE PODATAKA ZA IZBOR REŽIMA OBRADE PRI STRUGANJU

7.1 Modeliranje globalne logičke strukture

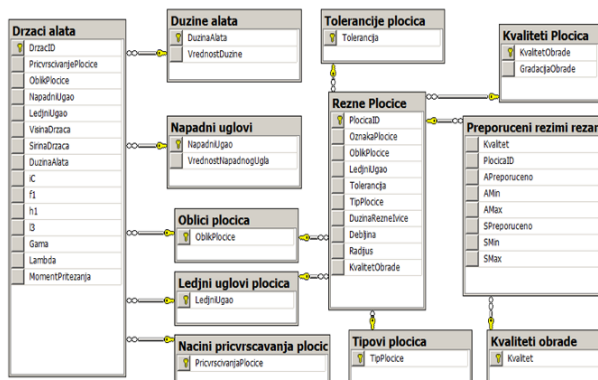
Za koncept aplikacije za izbor merodavnih parametara režima obrade struganjem postavljena je slobodna šema logičke strukture podataka (slika 7.1), koja pretpostavlja da se na kartici za unos ulaznih podataka početni podaci za izbor režima obrade unesu ručno (podaci koji nisu skladišteni u bazi podataka) ili se izaberu sa padajućih listi (oni podaci koji su skladišteni u bazi podataka). Nakon toga sledi kartica za izbor rezne pločice sa preporučenim vrednostima režima obrade i nekim graničnim vrednostima uslovljenim na prethodnoj kartici, gde se nakon izbora pločice i merodavnih režima obrade prelazi na karticu za izbor držača pločice i izbor mašine alatke, posle čega se daje mogućnost izvođenja novog eksperimenta za izabrane režime obrade.



Slika 7.1 Šema globalne logičke strukture aplikacije za izbor merodavnih režima obrade pri struganju

7.2 Projekat fizičke strukture baze podataka

Na osnovu postavljene logičke strukture bira se pogodan sistem za upravljanje bazama podataka i konkretno realizuju svi elementi logičke strukture (slika 7.2).



Slika 7.2 Šema relacionog povezivanja tabela vezanih za reznju pločicu i držač rezne pločice

8. RAZVOJ APLIKACIJE ZA MODELIRANJE OBRADNIH PROCESA

8.1. Unos ulaznih podataka

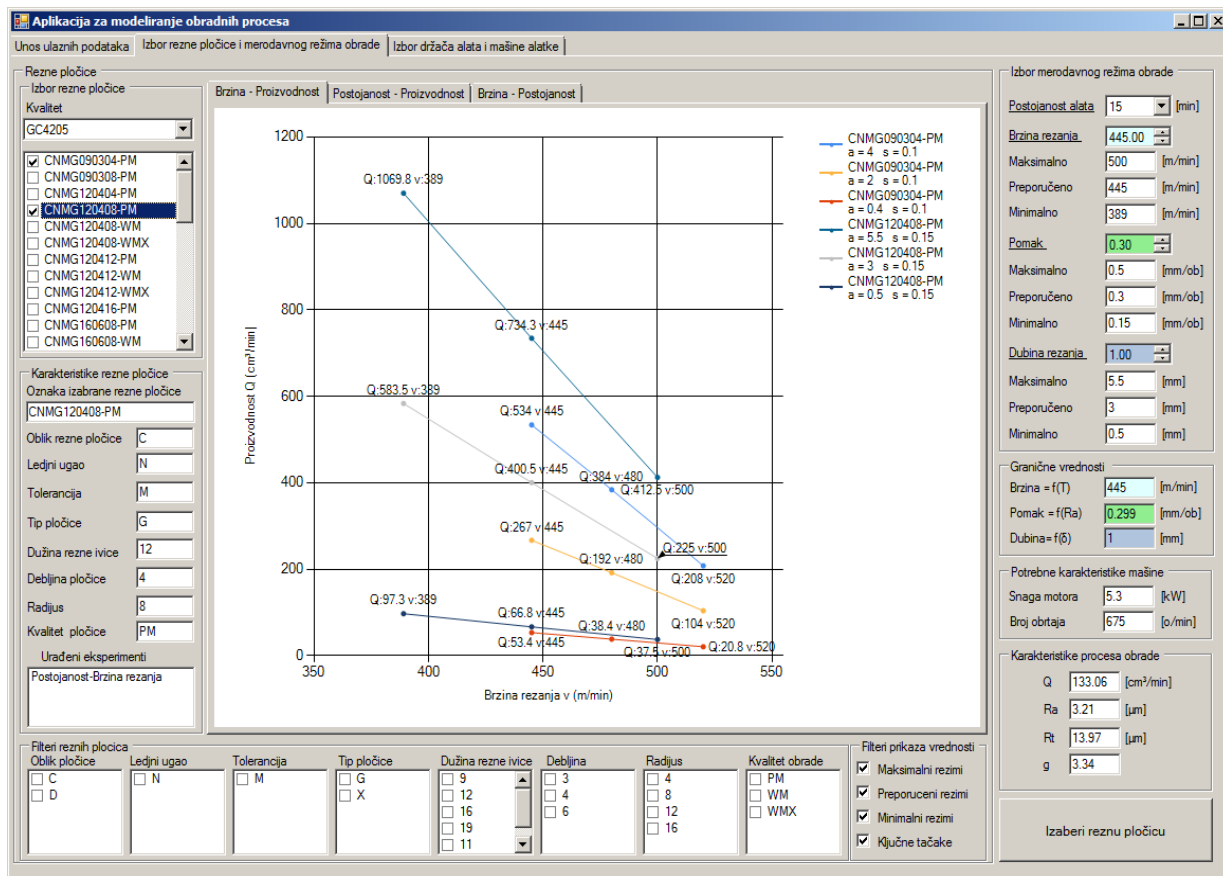
Posle pokretanja programa "Aplikacija za modeliranje obradnih procesa.exe" otvara se forma na kojoj se biraju početni parametri iz baze podataka. Na kartici za unos ulaznih podataka potrebno je izvršiti izbor materijala obratka, izbor obrade i uneti vrednosti karakteristika obratka.

Slika 8.1 Izgled kartice sa unetim ulaznim podacima

Nakon izbora obrade potrebno je uneti karakteristike obratka (dužinu i prečnik) kako bi se na osnovu njih dobile vrednosti potrebnih dodataka za obradu i dimenzije priprema koji zadovoljava tražene uslove. Konačni izgled kartice sa unetim ulaznim podacima dat je na slici 8.1.

8.2. Izbor rezne pločice i merodavnih režima obrade

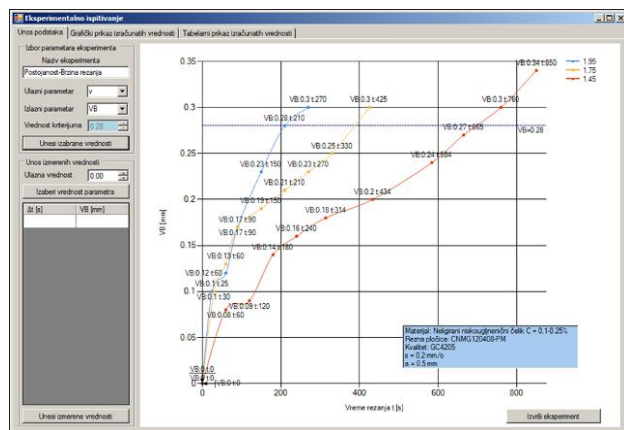
Nakon izvršenog unosa ulaznih podataka prelazi se na karticu za izbor rezne pločice i merodavnih režima obrade, na koju su već unetim podaci o reznim pločicama koje zadovoljavaju uslov granične vrednosti hrapavosti kao i oblika pločice, na osnovu preporuka proizvođača alata. Prvo je potrebno izabrati kvalitet rezne, zatim se može izvršiti izbor oblika, lednog ugla, tolerancije, tipa pločice, dužine rezne ivice, debljine i radijusa vrha rezne pločice. Nakon izbora rezne pločice moguće je videti karakteristike izabrane pločice, preporučene vrednosti režima rezanja za svaku pločicu, kao i grafički prikaz opsega preporučenih vrednosti za odnose brzina rezanja-proizvodnosti, postojanosti alata-proizvodnosti i brzina rezanja-postojanost alata. Pod uslovom da su u bazu podataka uneti podaci o urađenim eksperimentima za izabranu reznju pločicu moguće je videti grafički prikaz ulaznih podataka, grafički prikaz izračunatih vrednosti i tabelarni prikaz izračunatih vrednosti. Konačni izgled kartice za izbor rezne pločice i merodavnog režima obrade i dat je na slici 8.2.



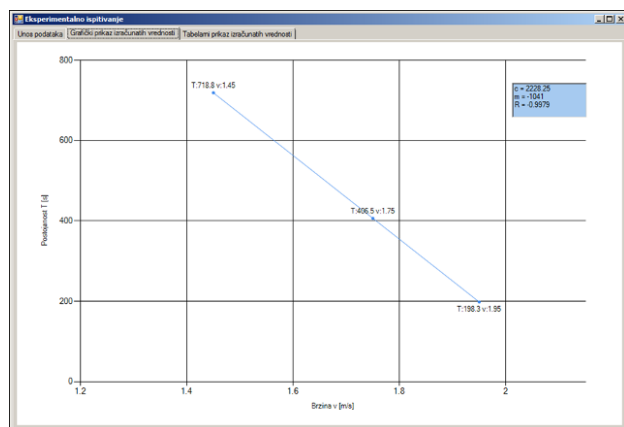
Slika 8.2 Konačni izgled kartice za izbor rezne pločice i merodavnog režima obrade

8.3. Eksperimentalno ispitivanje

Nakon izbora svih potrebnih parametara za proces obrade rezanjem moguće je izvršiti proizvoljno eksperimentalno ispitivanje na osnovu izabranih režima obrade za izabranu rezu pločicu. Na slici 8.3.1 prikazan je izgled kartice za unos podataka, a nakon toga grafički prikaz rezultata eksperimenta na slici 8.3.2 i tabelarni prikaz rezultata eksperimenta na slici 8.3.3.



Slika 8.3.1 Kartica za unos podataka



Slika 8.3.2 Grafički prikaz rezultata eksperimenta

Eksperimentalno ispitivanje

Unos podataka | Grafički prikaz izračunanih vrednosti | Tablični prikaz izračunanih vrednosti

Završi

Računane vrednosti za parabolično zakonito

i	r	T	$\ln r$	$\ln T$	$\ln r^2$	$\ln T^2$	$\ln r \cdot T$	Trac
1	1.95	210	0.68703	0.448	5.34711	28.59159	3.57096	224.06
2	1.75	387	0.55962	0.31717	5.59542	25.50277	3.33445	345.18
3	1.45	726.6	0.37156	0.10806	5.58838	43.40675	2.44769	754.9
m	$\ln(r)$	$\ln(T)$	$\ln(r^2)$	$\ln(T^2)$	$\ln(r \cdot T)$			
c								
R								

m

4.1

c

3483.38

R

0.98938

Računane vrednosti za linearnu zakonito

i	r	T	r^2	T^2	$r \cdot T$	Trac
1	1.95	210	3.8025	44100	409.5	198.3
2	1.75	387	3.0625	149769	677.25	406.5
3	1.45	726.6	2.1025	527947.56	1053.57	718.8
m	$\ln(r)$	$\ln(T)$	$\ln(r^2)$	$\ln(T^2)$	$\ln(r \cdot T)$	
c						
R						

m

1041

c

2228.25

R

0.9979

Izneseni računane vrednosti za linearni i parabolični zakon izračunane su procentualnim greškama meraja

r	T	Trac L	c L	Trac P	c P
1.95	210	198.3	-5.57	224.06	6.7
1.75	387	406.5	5.04	345.18	-9.77
1.45	726.6	718.8	-1.07	754.9	3.89

Slika 8.3.3 Tabelarni prikaz rezultata eksperimenta

9. ZAKLJUČAK

Režimi rezanja direktno utiču na kvalitet i ekonomičnost obrade i zbog toga je unapređenje načina njihovog izbora od velikog značaja za unapređenje proizvodnje.

Osnovni problem pri izboru merodavnih, tj. optimalnih režima rezanja je što ne postoje pouzdani matematički modeli kojima se opisuju osnovne tehnološke veličine procesa rezanja. Zbog toga se vrlo često parametri režima obrade određuju na osnovu "iskustava", referenci postavljenih na osnovu preporuka proizvođača alata ili preporuka iz inženjerskih priručnika.

Proizvođači alata daju dosta široke opsege preporučenih elemenata režima rezanja kako bi se ogradili od eventualnih posledica koje mogu nastati pri lošem izboru režima rezanja, tako da su ponekad preporučene vrednosti daleko od optimalnih.

Na osnovu navedenih činjenica dolazi se do zaključka da je za konkretne uslove potrebno unapređivati način izbora parametara režima obrade postavljanjem tehnološke baze podataka i razvojem postupaka za određivanje merodavnih ili optimalnih režima obrade.

Značajno unapređenje razvijene aplikacije može se ostvariti dodavanjem modula za optimizaciju parametara režima rezanja, dodavanjem modula za modeliranje procesa obrade primenom višefaktornih eksperimentalnih planova i primenom veštačke inteligencije pri izboru i obradi podataka.

10. LITERATURA

- [1] Pavkov B.: RAZVOJ APLIKACIJE BAZE PODATAKA ZA MODELIRANJE OBRADNIH PROCESA, Master rad, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Novi Sad, 2016.
- [2] Gostimirović M.: BAZA PODATAKA OBRADNIH PROCESA, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Novi Sad, 2013.
- [3] Kovač P.: MODELIRANJE PROCESA OBRADE: FAKTORNI PLANOWI EKSPERIMENATA, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Novi Sad, 2006.

Kratka biografija:



Branko Pavkov rođen je u Somboru 1988 godine. Master rad na Fakultetu tehničkih nauka iz oblasti Proizvodnog mašinstva iz predmeta Baze podataka obradnih procesa, odbranio je 2016.godine.



Dr. Marin Gostimirović, red. prof., rođen je 1957. godine. Doktorirao je na Fakultetu tehničkih nauka 1997. god. Oblasti njegovog interesovanja su procesi obrade skidanja materijala i nekonvencionalni postupci obrade.

VEB ALAT ZA NAPREDNU ADMINISTRACIJU SISTEMA ZA ELEKTRONSKU NASTAVU**WEB TOOL FOR ADVANCED ADMINISTRATION OF LEARNING MANAGEMENT SYSTEMS***Srđan Donevski, Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je prikazan veb alat koji obezbeđuje funkcije napredne administracije Canvas sistema za elektronsku nastavu. Alat je realizovan kao eksterna veb aplikacija koja se integriše sa Canvas sistemom putem IMS LTI specifikacije. Podržane funkcije napredne administracije su pretraga studenata po različitim kriterijumima i grupno uključivanje studenata na kurseve. U radu su opisane tehnologije korišćene za izradu alata i dati detalji o implementaciji.

Abstract – The paper presents a web tool for advanced administration of Canvas learning management system. The tool has been implemented as an extern web application which is integrated with Canvas LMS using the IMS LTI specification. The tool provides advanced search of students based on different criteria, as well as group enrollment of students on Canvas courses. The paper gives an overview of technologies used for the tool development and explains important implementation details.

Ključne reči: Veb aplikacija, Canvas LMS, REST API, Learning Tools Interoperability

1. UVOD

Canvas LMS je sve zastupljeniji sistem za upravljanje nastavom [1], kako u ustanovama za osnovno obrazovanje tako i u visokoškolskim ustanovama. Prednost ovog alata u odnosu na druge LMS sisteme se ogleda u jednostavnijem interfejsu, komunikacionim panelima i jednostavnom dodavanju multimedijanog sadržaja u sam Canvas. Iako Canvas kroz svoj bogati interfejs pruža pretragu studenta u okviru kursa po delu imena i/ili ulozi koju korisnik ima u kursu, ili uvid u sve korisnike koji su u vezi sa trenutno ulogovanim korisnikom, neophodne funkcije koje bi nastavnicima olakšale i ubrzale rad su pretraživanje korisnika koji učestvuju u više kurseva kao i jednostavan način dodavanja svih studenata sa jednog kursa na drugi.

U cilju omogućavanja ovih funkcionalnosti bez izmene Canvasovog izvornog koda, kao i mogućnost korišćenja ovih funkcionalnosti na drugim sistemima sa Canvasom kao sistemom za upravljanje nastavom potrebno je realizovati ove funkcionalnosti putem eksternog alata. U cilju integracije eksterne aplikacije u Canvas LMS aplikacija mora pratiti LTI [2] standard definisan od strane IMS Global Learning Consortium organizacije.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Goran Savić.

U radu je prikazan razvoj eksternog alata za naprednu administraciju korisnika koji nastavnicima omogućava pretragu korisnika po imenu ili delu imena, kursevima u kojima učestvuje kao i mogućnost odabira uloge u kojoj se nalaze korisnici. Dodati su i padajući meniji sa logičkim operatorima kojima je moguće kombinovati ove kriterijume pretrage u cilju preciznijeg filtriranja rezultata. Takođe, na strani za pretragu korisnika, dodata je i opcija za čuvanje rezultata u CSV formatu.

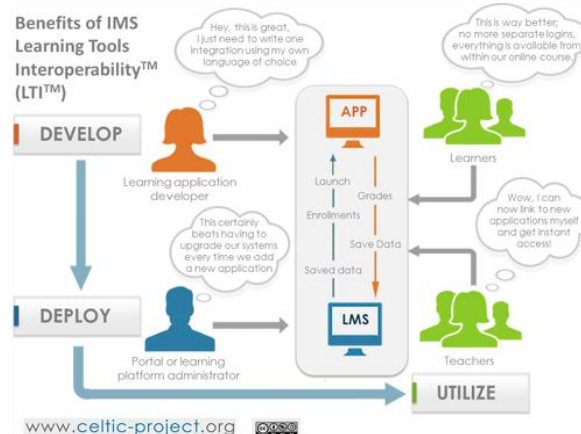
Druga funkcionalnost koju alat obezbeđuje je grupno uključivanje grupe studenata u jedan ili više kurseva. Nakon što nastavnik izabere studente i izabere sve kurseve na koje želi da dodeli studente, automatski se ovi studenti uključuju u izabrane kurseve.

Ostatak rada organizovan je na sledeći način: u narednoj sekciji je opisana arhitektura kroz koju je veb alat realizovan. Treća sekcija sadrži detalji implementacije samog alata. Završna razmatranja kao i pravci budućeg razvoja dati su u poslednjoj sekciji.

2. ARHITEKTURA SISTEMA

U ovom poglavlju predstavljene su tehnologije i standardi koji su korišćene u izradi alata, kao i generalna arhitektura povezivanja eksterne veb aplikacije sa sistemom za elektronsku nastavu.

Canvas LMS je sistem otvorenog koda za upravljanje nastavom napravljen korišćenjem *Ruby On Rails* tehnologije. Canvas je razvijen u skladu sa LTI specifikacijom, čime je omogućeno uključivanje eksternih alata u sistem. Eksterni alati se mogu na jednostavan način, putem URL ili direktnim unosom konfiguracije u XML formatu integrisati u Canvas. Ova arhitektura je prikazana na slici 1.



Slika 1. Integracija eksternog alata u sistem za upravljanje nastavom

Prikazana arhitektura obezbeđuje nezavisnost tehnologije u kojoj se razvija eksterna aplikacija od tehnologije korišćene za sam LMS. Samim tim, dodavanje nove funkcionalnosti ne zahteva instaliranje nove verzije sistema. Sami korisnici sistema ovim imaju mogućnost da prilagode sistem svojim potrebama dodajući željene eksterne alate.

U cilju omogućavanja korisnicima programski pristup podacima koje skladišti, Canvas LMS je izložio svoj REST API na korišćenje [4]. REST (*Representational State Transfer*) [3] je skup pravila za softversku arhitekturu, koja poboljšavaju interakciju komponenti, uvode generalizaciju u interfejs i dozvoljavaju nezavisan razvoj komponenti. Ovo obezbeđuje efikasniju komunikaciju između komponenti sistema, unapređenje bezbednosti i uvođenje standarda u sam sistem.

Canvas REST API rezultate obezbeđuje u JSON formatu. Eksterne aplikacije koje koriste Canvas REST API putem GET i POST HTTP zahteva imaju mogućnost dobavljanja i menjanja podataka samog sistema.

Da bi eksterna aplikacija imala pristup REST API-ju, aplikacija se mora autentifikovati koristeći OAuth2 protokol. Prednost OAuth protokola je da se korisnikovo ime i lozinka ne čuvaju lokalno nego se korisnik redirektuje na sistem čijim resursima aplikacija treba da pristupi.

3. IMPLEMENTACIJA SISTEMA

U ovom poglavlju je prikazan izgled i implementacija eksternog alata kao i postupak integracije alata u Canvas LMS.

3.1. Implementacija eksterne aplikacije

Obzirom da IMS LTI specifikacija omogućuje integraciju eksternih alata nezavisno od tehnologije u kojoj su oni implementirani, za izradu eksternog alata koristio se programski jezik Microsoft Visual C# unutar ASP.NET radnog okvira.

Razvojno okruženje je bilo Visual Studio, a za implementaciju je korišćena tehnologija WebForms. ASP.NET WebForms [5] se baziraju na Microsoft ASP.NET tehnologiji u kojoj se na serverskoj strani generiše strana koja se šalje i prikazuje na klijentu. Ovo je jedan od programskih modela koji se može iskoristiti za kreiranje ASP.NET veb aplikacija.

Kao što je gore navedeno, eksterna aplikacija predstavlja veb sajt koji putem API-ja komunicira sa Canvas LMSom. Za svaku od implementiranih funkcionalnosti bilo je potrebno komunicirati sa odgovarajućim servisima Canvas REST API-ja.

Prilikom implementacije strane najpre je bilo potrebno napraviti dva poziva ka serveru u cilju dobavljanja svih kurseva i uloga za trenutno ulogovanog korisnika. Za dobavljanje uloga, GET zahtev se šalje na sledeći API endpoint.

```
GET /api/v1/accounts/:account_id/roles
```

Dobijeni rezultat u JSON formatu je prikazan u listingu 1.

```
{
  // The label of the role.
  "label": "New Role",
  // The label of the role. (Deprecated alias
  for 'label')

```

```
  "role": "New Role",
  // The role type that is being used as a base
  for this role. For account-level
  // roles, this is 'AccountMembership'. For
  course-level roles, it is an enrollment
  // type.
  "base_role_type": "AccountMembership",
  // JSON representation of the account the role
  is in.
  "account":
  { "id":1019,"name":"CGNU","parent_account_id":73,
    "root_account_id":1,"sis_account_id":"cgnu"},
    // The state of the role: 'active',
    'inactive', or 'built_in'
    "workflow_state": "active",
    // A dictionary of permissions keyed by name
    (see permissions input parameter in
    // the 'Create a role' API).
    "permissions":
    { "read_course_content":{"enabled":true,"locked":
      false,"readonly":false,"explicit":true,"prior_de
      fault":false}, "read_course_list":{"enabled":true
      ,"locked":true,"readonly":true,"explicit":false}
      , "read_question_banks":{"enabled":false,"locked"
      :true,"readonly":false,"explicit":true,"prior_de
      fault":false}, "read_reports":{"enabled":true,"lo
      cked":false,"readonly":false,"explicit":false}}
  }
```

Listing 1. Role object in JSON format

Kako Canvas LMS vraća rezultate u JSON formatu bilo je potrebno to mapirati na klase. Mapiranje JSON objekta na klasu vrši se Newtonsoft.Json.dll bibliotekom. Prilikom deserijalizacije JSON objekta potrebno je samo specificirati povratni tip kod poziva ove metode. Deserijalizacija uloge iz JSON formata u objekat klase RoleDTO vrši se pozivom statičke metode DeserializeObject klase JsonConvert.

Listing 2 prikazuje klasu RoleDTO koja predstavlja objektnu reprezentaciju uloge u Canvas sistemu.

```
public class RoleDTO
{
  [JsonProperty("id")]
  public int RoleId { get; set; }

  [JsonProperty("role")]
  public string RoleType { get; set; }

  [JsonProperty("label")]
  public string Name { get; set; }

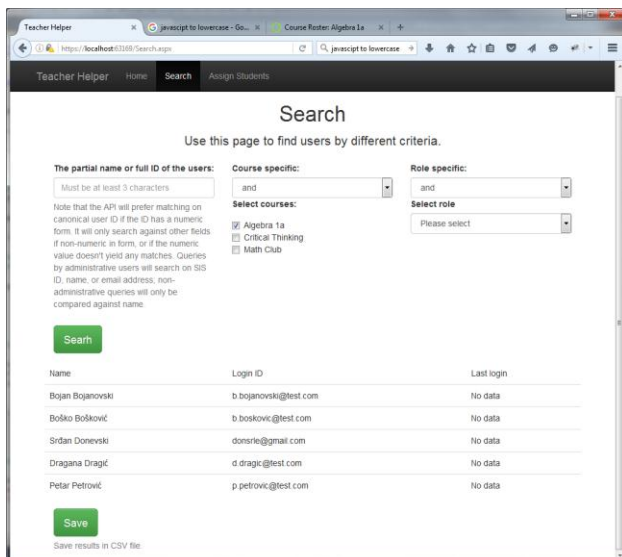
  [JsonProperty("base_role_type")]
  public string BaseRoleType { get; set; }
}
```

Listing 2. Klasa RoleDTO

Kako se iz listinga 2 vidi, svojstva klase anotacijama JsonProperty mapiraju se na odgovarajuće ključ/vrednost parove JSON reprezentacije te se tako npr. Name svojstvo klase RoleDTO mapira na ključ role iz JSON reprezentacije uloge.

Na sličan način se dobija i lista kurseva s kojima je nastavnik u vezi. GET zahtevom koji se šalje na odgovarajući Canvasov API endpoint.

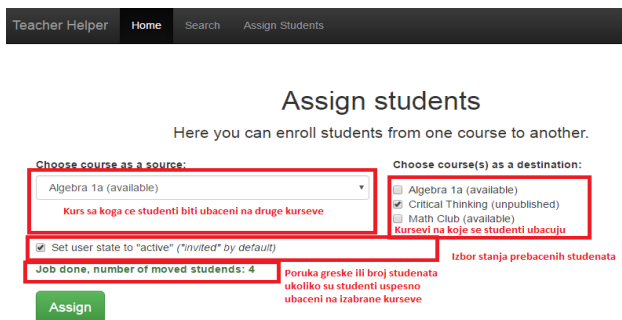
GET /api/v1/users/:user_id/courses U procesu kreiranja veb strane za pretragu studenta korišćene su .NET serverske kontrole. Za stilizovanje i sam izgled stane korišćena je Bootstrap biblioteka. Izgled stranice za pretragu je prikazan na slici 2.



Slika 2. Izgled strane za pretragu korisnika

Nakon pritiska na dugme za pretragu dobija se lista rezultata sa mogućnošću čuvanja rezultata u jedan CSV fajl, klikom na dugme Save.

Druga napredna funkcionalnost implementirana u Canvas sistemu je prebacivanje studenta sa jednog kursa na druge kurseve. Primer korisničkog interfejsa može se videti na slici 3:



Slika 3. Izgled strane za prebacivanje studenata sa jednog kursa na druge kurseve

U implementaciji ovog zadatka bilo je potrebno dobiti listu kurseva. Nakon toga se ista lista kurseva prikazuje u padajućoj listi u levo i desnoj koloni. Korisnik je u mogućnosti da izabere jedan kurs sa koga želi da uzme sve studente i kopira ih u sve one kurseve koje je selektovao u desnoj koloni. Pohađanje kursa od strane studenta predstavljeno je klasom EnrollmentDTO koja je prikazana u listingu 3.

```
public class EnrollmentDTO
{
    [JsonProperty("id")]
    public int EnrollmentId { get; set; }

    [JsonProperty("course_id")]
    public int CourseId { get; set; }

    [JsonProperty("type")]
    public string EnrollmentType
    {
        get; set;
    }

    [JsonProperty("role_id")]
    public int RoleId { get; set; }

    [JsonProperty("role")]

```

```
public string RoleName { get; set; }

[JsonProperty("enrollment_state")]
public string ActiveState { get; set; }

[JsonProperty("user_id")]
public int UserId { get; set; }
}
```

Listing 3. Klasa EnrollmentDTO

Da bi se kompletirao ovaj zadatak potrebno je bilo POST pozivom API metode *Enroll a user* dodati novi *Enrollment* koji bi predstavljao vezu između kursa, korisnika i tipa učestvovanja u kursu. API endpoint je sledeći:

POST /api/v1/courses/:course_id/enrollments

Kao što se vidi na slici 3, pored opcija za izbor kurseva postoji i *checkbox* polje sa labelom „Set user state to *active*“ (*invited* by default) namenjeno za podešavanje stanja pohađanja. Uključeno se to polje izabere *ActiveState* će biti postavljen na vrednost *active* i student će biti u mogućnosti da momentalno učestvuje u kursu. Ukoliko se ovo polje ostavi prazno korisnik će dobiti poziv da učestvuje u kursu ali neće učestvovati u kursu dok ne prihvati pozivnicu. Treba napomenuti da je prilikom dodavanja korisnika na određeni kurs još jedno stanje podržano od strane Canvas API-ja ali podešavanje ovog stanja nije podržano u trenutnoj verziji implementiranog alata. Stanje *inactive* (neaktivan) je takvo da će nastavnik biti u mogućnosti da vidi studente sa stanjem *inactive* među izlistanim studentima za trenutni kurs ali sam student neće biti u mogućnosti da učestvuje u kursu sve dok se ne izvrši aktivacija.

3.2. Integracija eksterne aplikacije u Canvas LMS

Kako je već ranije napomenuto za integraciju aplikacije bilo je potrebno pratiti OAuth2.0 standard. OAuth standard se bazira na razmeni tokena. Preporuka je da se token koji se dobije sa server ubaci u zaglavlje svakog zahteva na sledeći način:

```
curl -H "Authorization: Bearer <ACCESS-TOKEN>"
"https://<CANVAS-LMS-URL>/api/v1/courses"
```

Zahtev za tokenom i redosled izmena poruka obavlja se po proceduri ilustrovanoj na slici 4.

Aplikacija se integriše u Canvas LMS prosledjivanjem konfiguracionog URL-a ili direktnim unosom podešavanja u XML formatu. Primer jednog XML fajla za konfiguraciju je prikazan u listingu 4.

OAuth 2.0 Protocol Flow

1.2. Protocol Flow

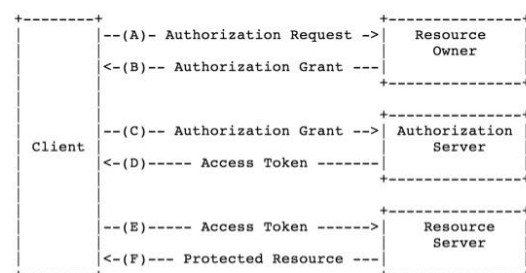


Figure 1: Abstract Protocol Flow

<http://tools.ietf.org/html/rfc6749>

Slika 4. Procedura OAuth protokola


```
<?xml version="1.0" encoding="UTF-8"?>
<cartridge_basiclti_link>
  <blti:launch_url>
    https://localhost:3000/Home.aspx
  </blti:launch_url>
  <blti:title>Teacher Helper</blti:title>
  <blti:description>
    Provides an interactive seating chart and
    attendance tool
  </blti:description>
  <blti:extensions
    platform="canvas.instructure.com">
    <lticm:property name="privacy_level">
      public
    </lticm:property>
    <lticm:property name="domain">
      localhost:3000
    </lticm:property>
    <lticm:options name="account_navigation">
      <lticm:property name="visibility">
        admins
      </lticm:property>
      <lticm:property name="enabled">
        true
      </lticm:property>
    </lticm:options>
  </blti:extensions>
</cartridge_basiclti_link>
```

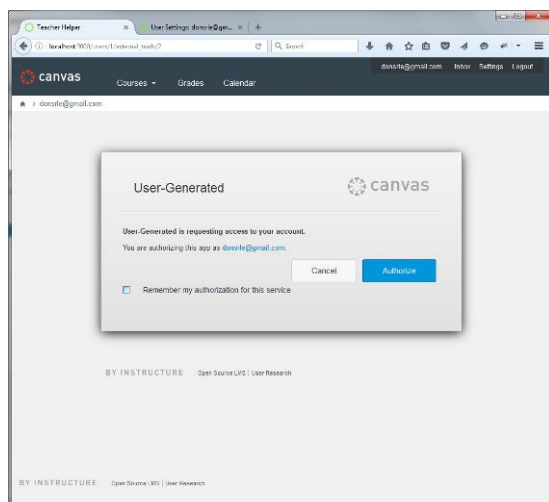
Listing 4. XML fajl za konfiguraciju integracije eksternog alata sa Canvas sistemom

U nastavku će biti opisani neki od ključnih delova ovog fajla. Tag `blti:launch_url` specificira putanju do eksterne aplikacije. Mesto gde će se link pojaviti se definiše `account_navigation` atributom a podešavanjem `visibility` taga na vrednost `admins` se omogućava samo administratorima pristup eksternoj aplikaciji. Ovo je željeno ponašanje jer funkcionalnosti realizovane alatom predstavljaju zadatke koje samo administratori mogu i smeju da izvršavaju.

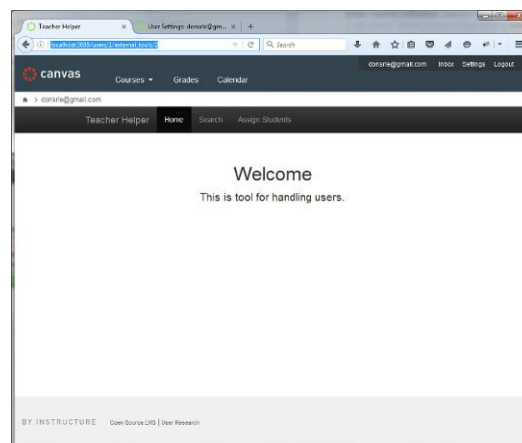
Tokom procesa razmene poruka od klijenta se zahteva da se autentifikuje i da eksternoj aplikaciji dozvoli pristup Canvas LMSu. Dijalog za autentifikaciju je prikazan na slici 5.

Nakon prihvatanja ovih uslova aplikacija dobija token koji se koristiti za pozive Canvasovom API-ju.

Kada se aplikacija integriše u Canvas LMS, nakon pokretanja aplikacije sistem izgleda kao na slici 6. Aplikacija se putem `iFrame` elementa integriše u Canvas i postaje njegov sastavni deo.



Slika 5. Proces autentifikacije aplikacije



Slika 6. Izgled Canvas LMSa sa pokrenutom eksternom aplikacijom

4. ZAKLJUČAK

U ovom radu je prikazan način implementacije i integracije eksterne veb aplikacije u Canvas LMS. Eksterna aplikacija omogućuje naprednu pretragu korisnika, kao i grupno uključivanje studenata u kurs.

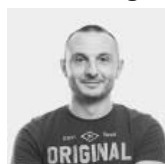
Glavna prednost ovakvog načina realizacije novih funkcionalnosti u sistemu za elektronsku nastavu je što je praćenjem LTI standarda moguće dodati napredne funkcije Canvas sistem u tehnologiji nezavisnoj od samog Canvas sistema. Programer se može posvetiti onoj tehnologiji s kojom ima najviše iskustva prilikom kreiranja eksterne aplikacije.

Dalji pravci razvoja ove aplikacije se odnose na unapređenje postojećih funkcionalnosti, kao i realizaciju potpuno novih funkcionalnosti koje bi sa postojećim činile administratorski portal. Unapređenje postojećih funkcionalnosti se pre svega odnosi na čuvanje tokena u lokalnoj bazi podataka kako se ne bi pravili nepotrebni pozivi ka serveru s ciljem dobijanja novog tokena prilikom pokretanja svake nove sesije. Takođe, plan je da se aplikacija registruje na Canvas Cloud sistemu kako bi bila dostupna širem auditorijumu. Kada je reč o administratorskom portalu, on bi trebao da pruži korisnički interfejs koji objedinjuje sve poslove koje administrator Canvas sistema standardno obavlja. Na ovaj način, administrator bi mogao programskim putem da izvršava zahteve za sve svoje poslove i time automatizuje značajan broj svojih aktivnosti.

5. LITERATURA

- [1] Canvas LMS website. <https://www.canvaslms.com/>
- [2] Learning Tools Interoperability, <https://www.imsglobal.org/activity/learning-tools-interoperability>.
- [3] Roy Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine, 2000. Str 76
- [4] Canvas LMS - REST API and Extensions Documentation, <https://canvas.instructure.com/doc/api/index.html>
- [5] *Introduction to ASP.NET Web Forms*. <http://www.asp.net/web-forms/what-is-web-forms>

Kratka biografija:



Srđan Donevski rođen je u Vršcu 1981. god. Upisao je smer Računarstvo i Automatika na Fakultetu tehničkih nauka Univerziteta u Novom Sadu 1999. godine. Od 2009. godine zaposlen u firmi »Vega IT Sourcing« u Novom Sadu na poziciji *software developer*.

**RUBY GOOGLE SKETCHUP INTERFEJS IZ ASPEKTA PROGRAMSKIH PARADIGMI
GOOGLE SKETCHUP RUBY INTERFACE PROGRAMMING PARADIGMS ASPECTS**

Filip Savić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu objašnjena je primena Ruby programskog jezika u interfejsu Google SketchUp aplikacije za 3D modelovanje iz aspekta programskih paradigmi. U okviru studije slučaja, prikazana je jednostavnost programiranja komponente SketchUp aplikacije gde Ruby, uz potpuno objektno orijentisano programiranje, premešta fokus sa onog što je pogodno za mašinu na ono što je pogodno za čoveka, tj. programera.

Abstract – In this paper the implementation of the Ruby programming language interface in Google SketchUp application for 3D modeling is explained, from the aspect of programming paradigms. In the case study, the easiness of programming SketchUp components is shown, where Ruby, with a fully object-oriented approach, shifts the focus from what is convenient for the machine to what is suitable for programmers.

Ključne reči: Ruby, Google SketchUp, programske paradigme

1. UVOD

Dosadašnje iskustvo sa razvojem informacionih tehnologija ukazuje da je veoma teško predvideti njihov budući razvoj, kao i veštine i alate potrebne za njihov razvoj i eksploataciju.

Kada su programerske veštine u pitanju, pre svega se moraju uzeti u obzir one u oblasti daljeg razvoja interneta i koncepta *Internet of Things*, rada sa sve većim bazama podataka, kao i još masovnijeg i raznovrsnijeg korišćenja tzv. pametnih telefona (*smartphones*) [1].

Kada su alati u pitanju, osnovni programerski alati su svakako programski jezici. Na izbor programskog jezika utiču njegove tehničke karakteristike, kao i namena, ali sve više, bar kada su početnici u pitanju, i lakoća i brzina učenja tog jezika, dostupna literatura, podrška šire programerske zajednice, biblioteke primera korišćenja, itd.

Cilj ovog rada je da se kroz kreiranje komponente softverske aplikacije *SketchUp* u programskom jeziku Ruby, ilustruje jedan pristup u savremenom programiranju, koji zagovara brzo i lako učenje programskog jezika, odnosno programiranje prilagođeno čoveku, a ne računaru.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Srđan Popov, docent.

2. PARADIGME PROGRAMIRANJA**2.1. Pojam paradigme programiranja**

Razvojem programskih jezika, vremenom su se uočavali različiti pristupi programiranju, da bi se od 1978. godine pojavio koncept "paradigma programiranja" [2].

Paradigma programiranja bi mogla da se definiše i kao obrazac za određenu školu mišljenja, odnosno skup ideja vezanih za računarsko programiranje. Treba je razlikovati od tehnike programiranja, koja se odnosi na algoritamsku ideju za rešavanje određene klase problema, kao i od stila programiranja, koji se odnosi na način izražavanja svakog programera, unutar svog programa [3].

Ponekad, "paradigma programiranja" predstavlja pokušaj klasifikacije programskih jezika, na osnovu njihovih najdominantnijih karakteristika. Te karakteristike mogu da budu način organizacije ili način izvršavanja programskog koda, ili sintaksa određenog programskog jezika. Od oko 30 poznatih paradigmi, ako se posmatraju sama ideja i osnovna disciplina, mogu da se izdvoje u literaturi najčešće pominjane: imperativna, funkcionalna, deklarativna, objektno orijentisana, proceduralna, logička i simbolička paradigma [4].

Svaka paradigma podržava određene koncepte koji je čine najpogodnijom za rešavanje određene vrste problema. Tako je objektno orijentisana paradigma najpogodnija za probleme sa velikim brojem povezanih i hijerarhijski organizovanih podataka, a logičko programiranje je najpogodnije za transformaciju složenih simboličkih struktura u skladu sa logičkim pravilima.

Razvoj različitih paradigmi programiranja ne treba uvek posmatrati kao progres, odnosno kao prelazak na viši nivo, ili bolju paradigmu. Izuzetno je teško precizno poređenje efikasnosti postojećih paradigmi, naročito zbog korišćenja različite terminologije za slične pojave i procese, kao i zbog brojnih razlika u njihovoj implementaciji u raznim jezicima.

Neki istraživači iz oblasti teorije programskih jezika kritikuju korišćenje pojma paradigme u cilju klasifikacije programskih jezika, jer veliki broj programskih jezika sadrži karakteristike više različitih paradigmi [5].

Posebno su interesantni jezici koji podržavaju više paradigmi programiranja, dozvoljavajući programeru da koristi najpogodniji način programiranja za određeni problem, jer ni jedna paradigma sama ne rešava sve probleme na najjednostavniji i najefikasniji način.

2.2. Statički i dinamički jezici

Neophodno je pomenuti još jednu podelu programskih jezika, koja je povezana sa temom ovog rada. To je podela na statičke i dinamičke jezike. I u ovoj podeli jezika postoji određena konfuzija, čemu doprinosi i

činjenica da mnogi programski jezici imaju neke osobine koje su karakteristične i za dinamičke i za statičke jezike. Iako ne postoji potpuno precizna definicija, može se reći da su dinamički programski jezici oni viši programski jezici koji dozvoljavaju da se određene aktivnosti u programu (provera ispravnosti svakog koraka u programu, dodavanje novog koda, kreiranje nove klase objekata ili deklarisanje varijabli) izvode tokom izvršenja tog programa, dok se kod statičkih jezika ove aktivnosti izvode tokom kompajliranja.

Često se upravo određivanje i provera tipa varijabli uzima kao kriterijum za podelu na statičke i dinamičke jezike. Tako, kod statičkih jezika, svaki naziv varijable je tokom kompajliranja povezan sa objektom i (deklarativno) sa tipom tog objekta, i ne može da se povezuje sa objektima drugačijeg tipa. Kod dinamičkih jezika, svaka varijabla je povezana samo sa objektom, pa je tokom izvršenja programa moguće povezivanje naziva varijable sa različitim tipovima objekata.

Aplikacije nastale korišćenjem statičkih programskih jezika se smatraju stabilnijim, skalabilnijim i lakšim za održavanje. Statičko definisanje tipova varijabli, kao i provera tokom kompajliranja, čine ih i pouzdanijim u odnosu na aplikacije kreirane dinamičkim jezicima. Generalno, dinamički jezici se zbog svoje fleksibilnosti smatraju lakšim za učenje. To su jezici visokog nivoa, što omogućuje da se sa manje linija koda, bez strogih pravila pisanja i bez upuštanja u teoriju programiranja, brzo dođe do prvih rezultata – funkcionalne aplikacije.

U ovu grupu jezika spadaju takozvani skript jezici *JavaScript* i *PHP*, kao i *Python* (najpopularniji početni programski jezik na univerzitetima u SAD) i *Ruby*, koji će biti detaljnije opisan u narednim poglavljima.

3. RUBY – OSNOVNI KONCEPTI

Ruby je dinamički, objektno orijentisani programski jezik opšte namene, koji dozvoljava funkcionalno i imperativno programiranje. Kreiran je sa dobrim korisničkim interfejsom, sa ciljem da unapredi produktivnost programera, ali i da mu programiranje učini lakšim i zabavnijim [6].

Zbog svoje sintakse, *Ruby* spada u kategoriju najlakših jezika za učenje i najpogodnijih za početnike. *Ruby* je potpuno objektno orijentisan jezik. Sve vrednosti su predstavljene kao objekti, uključujući i brojeve, konstante i promenljive, a svaki deo koda poseduje svoje karakteristike (instance) i odgovarajuće aktivnosti (metode), uključujući i klase i tzv. primitivne promenljive. Varijable su uvek povezane sa objektima. Svaka funkcija je i metoda, a metode se uvek pozivaju na objekat. Metode najvišeg nivoa su članovi klase tog objekta (*Basic Object*) i kao "najstarije" mogu da se pozivaju i od drugih objekata i da služe kao globalne procedure.

U *Ruby* jeziku, objekat sadrži samo podatke potrebne za modelovanje neke stvari – fizičkog objekta ili apstraktnog pojma. Dva objekta istog tipa moraju da imaju iste karakteristike, ali ne i iste vrednosti.

Priroda podataka i operacija nad tim podacima je definisana klasom objekta. Klasa definiše tip, odnosno strukturu objekta. Pošto je u *Ruby* jeziku sve sa čim se radi objekat, sve ima i klasu. Procedure ili operacije nad podacima, definisane u svakoj klasi, nazivaju se metode.

Mnoge metode zahtevaju dodatne podatke (parametre ili argumente), koji se u *Ruby* jeziku prikazuju u zagradi, odvojeni zarezima. Imena metoda se završavaju znacima `?` ili `=`. U prvom slučaju, rezultat metode je "tačno" ili "pogrešno", a ukoliko se metoda završava znakom jednakosti, objekat se ažurira podacima iz parametara. *Ruby* metode mogu da se lančano povezuju, tako što jedan metod može da radi sa vrednostima koje su rezultat druge metode.

Neke metode se, umesto nad objektom, izvršavaju nad samim klasama. To su metode klase, za razliku od metoda koje se izvršavaju nad objektima i nazivaju se metode instance jer je objekat instanca klase. Jedna od važnih metoda klase, sadržana u svakoj *Ruby* klasi, je metoda `new`, koja kreira novi objekat od klase.

Različite klase mogu da sadrže neke slične karakteristike, tako da iste metode mogu da se primene nad tim klasama. Da se ne bi iste metode više puta programirale, efikasniji način je da se one smeste u posebnu klasu i time učine raspoloživim objektima iz klase sa sličnim karakteristikama. Ovakva relacija među klasama se naziva nasleđivanje, a nova klasa sa zajedničkim metodama se naziva superklasa. Ako klasa B nasleđuje iz klase A, onda će sve metode iz klase A biti na raspolaganju klasi B. To znači da će i bilo koji objekat klase B moći da pristupi istim metodama kao i objekti klase A.

Sve klase u *Ruby* programskom jeziku mogu da budu otvorene i da im se dodaju ili redefinišu metode.

Svaka funkcija vraća implicitno rezultat poslednjeg izraza u njoj i svaki iskaz (poput `if-else` konstrukcije) takođe može imati povratnu vrednost. Pošto *Ruby* poseduje sopstveni interpreter, programski kod se izvršava automatski, u izvornom kodu i prilikom izmene koda nije potrebno vršiti kompajliranje, već se promene automatski primenjuju u sistemu. U *Ruby* jeziku svaka promenljiva mora da bude određenog tipa, ali taj tip vremenom može da se promeni. Posebna prednost kod dinamičke tipizacije primećuje se prilikom definisanja funkcija – argumentima funkcija ne dodeljuje se tip eksplicitno, već se tip argumenta određuje nakon prosleđivanja objekta kao argumenta definisane funkcije. Ukoliko se nad argumentom u funkciji pozivaju neke druge funkcije, očekuje se da klasa čijeg je tipa argument implementira pozvane funkcije.

Fleksibilnost *Ruby* jezika se ogleda i u mogućnosti da korisnik slobodno menja neke njegove delove, redefiniše ih prema svojim potrebama, izbacuje neke delove ili dodaje nove.

4. GOOGLE SKETCHUP APLIKACIJA

U mnoštvu na tržištu raspoloživih aplikacija za 3D modeliranje, *Google SketchUp* se izdvaja svojom jednostavnošću, lakoćom učenja i širokom primenom. Prihvaćen je od strane arhitekata i inženjera, ali i od grafičkih umetnika i animatora, matematičara i sl. Iako zadovoljava ovako široki krug korisnika, njegova funkcionalnost se može značajno poboljšati pomoću programskih dodataka (*plugins*). Na primer, mogu da se dodaju novi prozori za dijalog, novi alati i nove stavke u postojećim menijima, itd. Ovi programski dodaci se realizuju u formi skript datoteka, pisanih u *Ruby* jeziku. Prednost smeštanja *SketchUp* naredbi u skript datoteku je

u tome što se one unose samo jednom, a mogu više puta da se koriste i što se izvršavaju automatski. To je naročito pogodno kod crtanja nekih složenijih figura [7].

Zahvaljujući mogućnostima koje donosi *Ruby* (objektna orijentisanost, kontrolne strukture, iterativne petlje, uslovne naredbe, velika biblioteka kodova, itd), *SketchUp* skripte mogu, pored crtanja geometrijskih objekata, da izvršavaju sve zadatke koji se očekuju od modernog softvera.

Većina *Ruby* skripti poziva jednu od metoda u *SketchUp* modulu. *SketchUp* modul je skup metoda koje se odnose na kompletnu *SketchUp* aplikaciju (verzija, jezik i sl.). Najvažnija metoda u *SketchUp* modulu je `active_model`, koja daje `Model` objekat koji odgovara trenutno otvorenom *SketchUp* projektu.

Nasuprot *SketchUp* modulu, koji predstavlja celu *SketchUp* aplikaciju, `Model` objekat predstavlja informacije o aktuelnom projektu, smeštene u jedan *SketchUp* fajl (*.skp). Otvaranjem novog fajla u *SketchUp* aplikaciji, karakteristike *SketchUp* modula ostaju iste, ali podaci u aktivnom `Model` objektu postaju drugačiji. Svaki `Model` objekat može da sadrži grupisane različite objekte. Metode u `Model` klasi pružaju informacije o trenutno aktivnom projektu.

Svaka geometrijska figura, tekst, slika, grupa ili komponenta, u *SketchUp* projektu je predstavljena `Entitet` objektom ili njegovom podklasom. Klasa `Entitet` je superklasa za sve oblike koji se crtaju u *SketchUp* aplikaciji, a primarna podklasa klase `Entitet` je element koji se crta, a koji je opet superklasa za određene oblike, ivce, površine, tekstove, itd.

Metode klase crtanih elemenata kontrolišu način prikazivanja tih elemenata u *SketchUp* projektu (vidljivost elementa, senčenje, material, itd.).

5. STUDIJA SLUČAJA

Principi objektno orijentisanog programiranja u *Ruby* jeziku su prikazani u sledećem primeru 3D modela čekića. Sledeći kod kreira klasu `Cekic`:

```
class Cekic
  # Postavka dužine drške
  def drska_d=(d)
    @drska_d = d
  end
  # Postavka prečnika drške
  def drska_p=(p)
    @drska_p = p
  end
  # Postavka dužine glave
  def glava_x=(x)
    @glava_x = x
  end
  # Postavka širine glave
  def glava_y=(y)
    @glava_y = y
  end
  # Postavka visine glave
  def glava_z=(z)
    @glava_z = z
  end
  # Pristup dužini drške
  def drska_d
    @drska_d
  end
  # Pristup prečniku drške
  def drska_p
```

```
    @drska_p
  end
  # Pristup dužini glave
  def glava_x
    @glava_x
  end
  # Pristup širini glave
  def glava_y
    @glava_y
  end
  # Pristup visini glave
  def glava_z
    @glava_z
  end
end
```

Kreiranje objekta pod nazivom `c`, koji je klase `Cekic` vrši se na sledeći način:

```
c = Cekic.new
```

Jedna od prednosti *Ruby* programskog jezika je u tome što je za promenu načina kreiranja objekata dovoljno kodirati metod koji se naziva `initialize`. Sledeća skripta pokazuje kako ovo funkcioniše. Metod `initialize` prima 5 argumenata i koristi ih da postavi vrednosti instance promenljivih:

```
class Cekic
  # Kreiranje instanci promenljivih i
  # njihovih pristupnih metoda
  attr_accessor :drska_d, :drska_p,
    :glava_x, :glava_y, :glava_z
  # Dozvola objektima za kreiranje sa
  # postavljenim podacima
  def initialize(drska_d, drska_p, glava_x,
    glava_y, glava_z)
    @drska_d = drska_d
    @drska_p = drska_p
    @glava_x = glava_x
    @glava_y = glava_y
    @glava_z = glava_z
  end
  # Iscrtavanje čekića
  def draw
    ...
  end
end
```

Metod `initialize` omogućuje korisniku da postavi inicijalne vrednosti za promenljive u isto vreme kada se objekat `Cekic` kreira.

Nasledjivanje je jedno od osnovnih prednosti korišćenja klase. Jednom kada je klasa kreirana, može lako da se kreira podklasa koja sadrži sve metode i konstante originalne klase.

Na primeru klase `Cekic` može da se kreira jedna podklasa `Mali_Cekic`. Ona će imati iste osobine kao `Cekic` klasa, ali će vrednosti 4 instanci promenljivih (`@drska_p`, `@glava_x`, `@glava_y` i `@glava_z`) biti ograničene redom na vrednosti 1, 4, 3 i 3. Korisniku će ostati mogućnost da odredi samo dužinu drške, tj. da dodeli vrednost instanci promenljive `@drska_d`.

Kreiranje podklase vrši se isto kao i kreiranje klase, s tim da se pri deklarisanju klase, posle njenog naziva stavlja znak `<` i naziv superklase:

```
require "C:/scripts/cekic.rb"
class Mali_Cekic < Cekic
  # Premošćavanje initialize metode u Cekic
  klasi
  def initialize(drska_d)
```

```

        @drska_d = drska_d
        @drska_p = 1
        @glava_x = 4
        @glava_y = 3
        @glava_z = 3
    end
    # Premošćavanje drska_p=, glava_x=...
    # metoda u Cekic klasi
    def drska_p=(drska_p)
        @drska_p = 1
    end
    def glava_x=(glava_x)
        @glava_x = 4
    end
    def glava_y=(glava_y)
        @glava_y = 3
    end
    def glava_z=(glava_z)
        @glava_z = 3
    end
    # Novi metod za Mali_Cekic
    def print_name
        "Ovo je mali čekić!"
    end
end

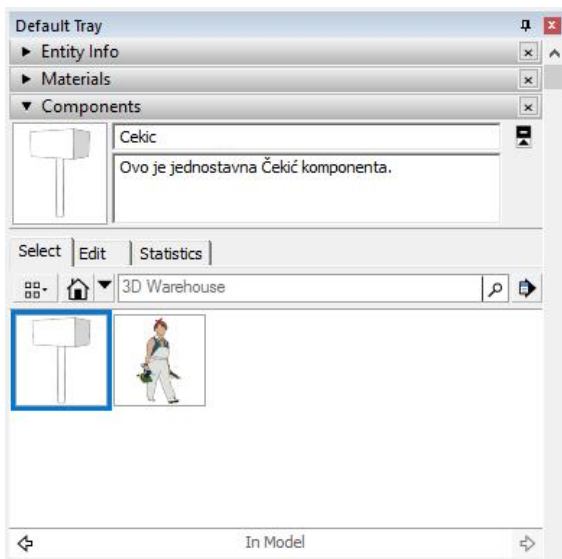
```

U sledećem primeru prikazano je kako se kreira nova komponenta, i kako ta komponenta izgleda u SketchUp interfejsu (Slika 1).

```

# Kreiranje definicije komponente
list = Sketchup.active_model.definitions
comp_def = list.add "Cekic"
comp_def.description = "Ovo je
jednostavna Čekić komponenta."
# Dodavanje entiteta definiciji
ents = comp_def.entities
# Kreiranje glave
glava_face = ents.add_face [3,2,0], [3,-
2,0], [-3,-2,0], [-3,2,0]
glava_face.reverse!
glava_face.pushpull 3, true
# Kreiranje drske
drska_curve = ents.add_circle [0, 0, -8],
[0, 0, 1], 0.5
drska_face = ents.add_face drska_curve
drska_face.pushpull 11
# Čuvanje definicije komponente
save_path = Sketchup.find_support_file
"Components", ""
comp_def.save_as(save_path +
"/cekic.skp")
# Ispisivanje kompletne putanje fajla
puts "Definicija je sačuvana u: " +
comp_def.path

```



Sl.1: Prikaz kreirane komponente u SketchUp aplikaciji

6. ZAKLJUČAK

Kroz prikaz kreiranja komponenti softverske aplikacije *SketchUp* u programskom jeziku *Ruby*, ilustrovan je objektno orijentisani pristup u savremenom programiranju, koji u skladu sa savremenim tendencijama u razvoju softverskih alata, istovremeno omogućuje brže i lakše učenje programskog jezika.

Kao ilustracija multiparadigmatskog pristupa, u radu je opisan programski jezik *Ruby* i njegova primena u programiranju skripti – programskih dodataka za aplikaciju *Google SketchUp*.

U studiji slučaja, kroz kreiranje komponente *Google SketchUp* aplikacije, prikazane su objektno orijentisane karakteristike programskog jezika *Ruby*, uz istovremenu ilustraciju pojednostavljene sintakse i korisničkog interfejsa ovog jezika.

7. LITERATURA

- [1] P. Wayner, "9 predictions for the future of programming", InfoWorld, <http://www.infoworld.com/article/3022874/application-development/9-predictions-for-the-future-of-programming.html>, (datum pristupa 15.06.2016.)
- [2] R. W. Floyd, "The paradigms of programming", Communications of the ACM, Vol 22, No 8, pp.. 455, 1979.
- [3] K. Nørmark, "Functional Programming in Scheme", <http://people.cs.aau.dk/~normark/prog3-03/html/notes/theme-index.html>, 2014. (datum pristupa 16.06.2015.)
- [4] K. Nørmark, "Overview of the four main programming paradigms", http://people.cs.aau.dk/~normark/prog3-03/html/notes/paradigms_themes-paradigm-overview-section.html, (datum pristupa 16.06.2016.)
- [5] S. Krishnamurthi, "Teaching programming languages in a post-linnaean age", ACM SIGPLAN Notices, Vol 43, No 11, pp.. 81-83, 2008.
- [6] <https://www.ruby-lang.org/en/about/> (datum pristupa 15.06.2016.)
- [7] M. Scarpino, "Automatic SketchUp - Creating 3-D Models in Ruby", Eclipse Engineering LLC, Hanover, MD, USA, 2010.

Kratka biografija:



Filip Savić rođen je u Novom Sadu 1983. god. U novom Sadu završio je osnovnu i srednju školu. Osnovne studije završio je na Fakultetu tehničkih nauka na smeru za Energetiku, elektroniku i telekomunikacije.

RAZVOJ OPTIČKIH MREŽA**THE DEVELOPMENT OF OPTICAL NETWORKS**Predrag Radojičić, Dejan Nemec, *Fakultet tehničkih nauka, Novi Sad***Oblast - ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu dat je pregled osnovnih komponenti telekomunikacionih optičkih sistema, dizajn i performanse optoelektronskih sistema i tehnologije optoelektronskih sistema prema generacijama nastanka. Takođe, prezentovane su osnovne karakteristike pasivnih optičkih mreža, njihova arhitektura, moduli koje koristi, načini primene, kao i njihove varijante tokom razvoja optičkih mreža.

Abstract – This graduate master's work provides an overview of the basic components of optical telecommunication systems, design and performance of optoelectronic systems and technologies of optoelectronic systems according to the generations of occurrence. Also, the basic characteristics of passive optical networks are presented, their architecture and modules, methods of application, as well as their variations during the development of optical networks.

Ključne reči: Optički komunikacioni sistemi, SONET/SDH, Carrier Ethernet, WDM sistemi, PON optičke mreže.

1. UVOD

Optičke mreže predstavljaju veoma značajnu oblast savremenih komunikacija. One su se intenzivno počele razvijati u drugoj polovini 20. veka i od tada njihov razvoj neprestano traje. Ranije su se koristili različiti postupci za prenos podataka koji su korisnicima bili vidljivi kao što su signalizacija, svetionici, zastavice i slično.

Problemi koji su se rešavali na početku razvoja optičkih telekomunikacija su bili: slabljenje na vezama između kablova i slabljenje usled propagacije. U komercijalnu upotrebu 1980. godine puštena je prva generacija optičkih komunikacionih sistema koja je radila na talasnoj dužini $\lambda=0.8 \mu\text{m}$ sa protokom od 45 Mb/s i dometom 10 km. Druga generacija, koja je puštena u komercijalnu upotrebu 1987. godine, radila je na talasnoj dužini $\lambda=1.3 \mu\text{m}$ pri protoku od 1.7 Gb/s i dometom 50 km. Gubici na 1.3 μm su bili 0.5 dB/km. Treća generacija je radila na 1.55 μm , i podržavala je protok 2.5 Gb/s i domet oko 100 km, ali je imala nedostatak jer se signal obnavljao periodično, na svakih 60, 70 km posredstvom regeneratora, koji su konvertovali optički signal u električni i obrnuto i time ograničavali brzinu prenosa i povećali cenu celog sistema.

Četvrta generacija optičkih sistema je koristila optičke pojačavače i tehniku multipleksiranja po talasnim dužinama WDM (Wavelength Division Multiplexing) čime se protok značajno povećao. Podržana je bila mogućnost prenosa podataka preko 21000 km brzinom od 2.5 Gb/s, odnosno 14300 km brzinom od 5 Gb/s.

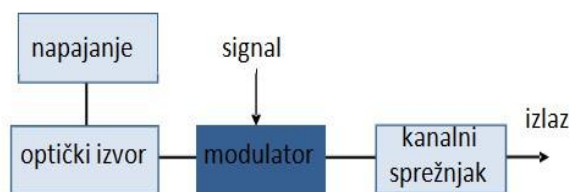
Ovim rezultatom došlo se na ideju interkontinentalne komunikacije. Peta generacija optičkih komunikacionih sistema se najviše bavila problemom disperzije u vlaknu. Trenutno, novi koncept predstavljaju optički solitoni, tj. optički impulsi koji zadržavaju svoju formu pri propagaciji tako što se kompenzuje uticaj disperzije kroz nelinearne efekte vlakna.

2. OPTIČKI KOMUNIKACIONI SISTEMI

Optički komunikacioni sistemi se sastoje od predajnika, komunikacionog kanala i prijemnika. Optički signal se iz optičkog predajnika šalje preko komunikacionog kanala koji je sastavljen od optičkih vlakana i optičkih pojačavača sve do optičkog prijemnika.

Većina optičkih komunikacionih sistema koriste optička vlakna kao kanal čime se omogućava prenos sa vrlo malim gubicima snage korisnog signala. Pored gubitaka, važna odlika vlakna jeste disperzija.

Uloga optičkih predajnika je da izvrši konverziju električnog signala u optički signal (tj. svetlost) i da izvrši transmisiju optičkog signala u optičko vlakno kako je prikazano na slici 1.



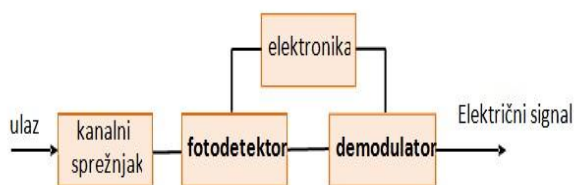
Slika 1. Optički predajnik

Iz optičkog vlakna na ulaz u optički prijemnik pristiže optički signal, što je predstavljeno na slici 2. Glavni zadatak prijemnika jeste da izvrši konverziju signala iz optičkog u električni domen.

Komunikacioni sistemi prenose informacije s jednog mesta na drugo, bez obzira da li je rastojanje nekoliko kilometara ili transokeansko rastojanje. Nosilac informacija u savremenim telekomunikacionim sistemima su elektromagnetski talasi.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Trpovski, vanr. prof.



Slika 2. Optički prijemnik

3. DIZAJN I PERFORMANSE OPTOELEKTRONSKIH SISTEMA

Optoelektronski sistemi služe za komunikaciju između uređaja, koji tu komunikaciju traže. Ove uređaje nazivamo stanicama, i to mogu biti kompjuteri, telefoni, terminali ili druga komunikaciona oprema. Ove stanice se u stručnoj terminologiji nazivaju oprema za krajnje podatke, DTE (data terminal equipment).

Mreže možemo podeliti prema veličini prostora koji obuhvataju, naravno u geografskom smislu. Mreže unutar zgrada zovemo LAN (local area network), dok se mreže koje obuhvataju površinu grada nazivaju MAN (metropolitan area network).

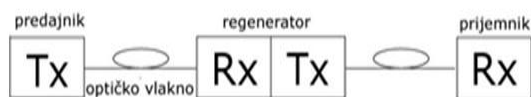
Mreže koje povezuju vrlo veliki prostor nazivamo WAN (wide area network).

Drugu podjelu vršimo prema nameni. Optički komunikacioni sistemi dele se, prema arhitekturi, u tri široke kategorije: PP veze (point-to-point links), Emisione i distributivne mreže i LAN mreže.

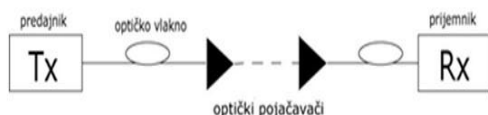
3.1 Point – to – point linkovi

Ovi sistemi su najprostiji optoelektronski sistemi. Namena im je prenos informacija, u formi digitalnog niza bitova, od predajnika do prijemnika duž optičkog vlakna sa što većom tačnošću. Zbog toga propusni opseg i gubici ovde nisu najznačajnija prednost optoelektronskog sistema.

Njihova glavna prednost jeste imunost na elektromagnetne smetnje. U zavisnosti od potrebe dužina veze izvršena je podjela na short – houl i long – houl. Short – houl predstavlja veze čija dužina ne prelazi 1 km. Ovakve veze su osnova LAN i svih pristupnih mreža. Long – houl su veze čija dužina prelazi 20 km. Koriste se kod globalnih mreža koje predstavljaju MAN i WAN. Zbog slabljenja, da signal ne bi postao neupotrebljiv moramo koristiti optičke pojačavače ili regeneratore. Na slici 3 prikazan je optoelektronski sistem sa regeneratorima, a na slici 4 sa pojačavačima.



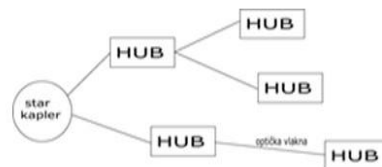
Slika 3. Optoelektronski sistem sa regeneratorima



Slika 4. Optoelektronski sistem sa optičkim pojačavačima

3.2 Emisione distributivne mreže

Emisione i distribucione mreže služe za distribuciju telefonskih servisa, višestrukih video kanala preko kablovske televizije – CATV “common-antenna T”, za integraciju audio i video servisa preko širokopojasnog ISDN, prenos telefonskog faksimila, kompjuterskih podataka kao i kod video prenosa na malim rastojanjima. Postoje dve vrste topologija ove mreže, a to su HUB gde se raspodela kanala odvija na centralnoj lokaciji prikazana na slici 5 i BUS topologija prikazana na slici 6 gde svako pojedinačno vlakno prenosi multipleksni višekanalni optički signal kroz oblast koju opslužuje.



Slika 5. Topologija HUB mreže



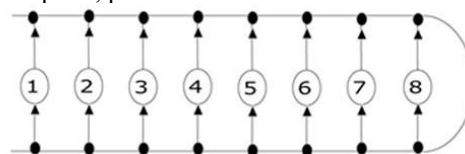
Slika 6. Topologija BUS mreže

3.3 LAN – Local area networks

Mreža u kojoj je veliki broj korisnika povezan na takav način da svaki korisnik može da pristupi mreži proizvoljno i da pošalje podatke drugom korisniku u nekoj lokalnoj oblasti, naziva se LAN.

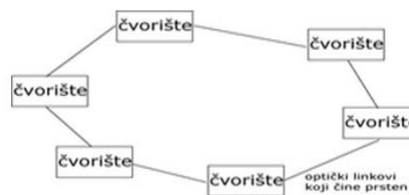
Postoje tri vrste topologija ove mreže, a to su BUS, RING i STAR.

Kod BUS topologije čvorovi razmenjuju informacije preko optičkih kaplera, prikazano na slici 7.



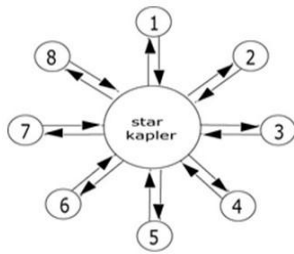
Slika 7. Topologija LAN BUS mreže

Kod RING topologije svi čvorovi su povezani point – to – point linkom, tako da je formiran prsten kao na slici 8.



Slika 8. Topologija RING mreže

Kod STAR topologije su svi čvorovi povezani sa centralnim star optičkim kaplerom kako je prikazano na slici 9.



Slika 9. Topologija STAR mreže

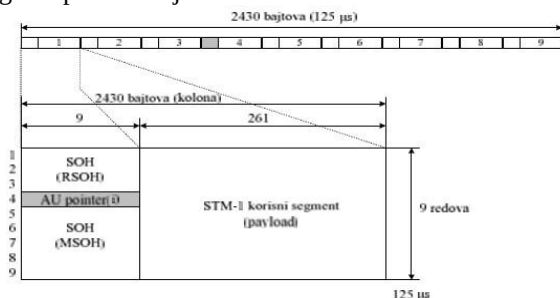
4. PREGLED TEHNOLOGIJA OPTOELEKTRONSKIH SISTEMA PREMA GENERACIJAMA NASTANKA

Prema generacijama nastanka optoelektronski sistemi imaju dve generacije. Prvu generaciju predstavljaju sistemi kod kojih je transmisioni medij optički, ali se distribucija, procesiranje i rutiranje obavljaju u elektronskom domenu. Druga generacija predstavlja pokušaj da se pojedini delovi obrade, npr. distribucija i rutiranje, obave u optičkom domenu. U okviru ovih generacija u ovom radu razmotrene su osnovne osobine SONET, Ethernet Carrier i WDM mreže.

4.1 SDH/SONET

Sinhrona digitalna mreža SONET (Synchronous Optical Networking – po američkom ANSI standardu) ili SDH (Synchronous Digital Hierarchy – po evropskom ETSI standardu) često koriste različite termine za opisivanje istih funkcija i procesa. Prema ITU-T (International Telecommunication Union) preporuci G-702 definisane su evropska i američka hijerarhija vremenskog multipleksiranja.

Osnovni nivo u SDH predstavlja signal protoka 155.520 kbit/s i naziva se STM-1. Osnovna struktura STM-1 signala prikazana je na sici 10.



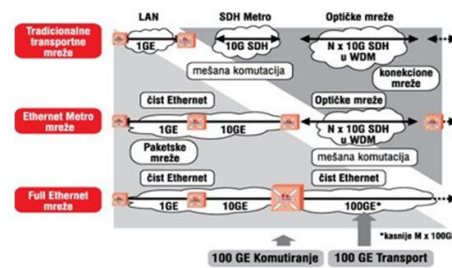
Slika 10. Struktura STM-1 signala

Pod elementima SDH mreže podrazumevamo sve multipleksne uređaje bazirane na SDH.

4.2 Carrier Ethernet

Ethernet tehnologija prenosa pruža veliki propusni opseg po relativno niskoj ceni, jednostavna je za korišćenje i podržava multipoint konfiguracije mreža. Carrier Ethernet koristi Ethernet kao osnovnu tehnologiju, a zatim joj dodaje određene osobine i funkcije kako bi dobio rezultatne proizvode koje bi pozicionirao u transportne

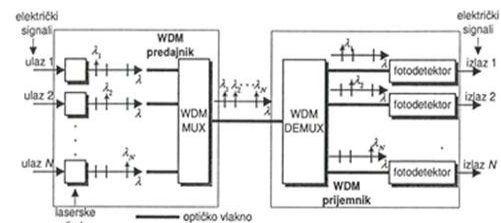
mreže. Na slici 11 simbolično je prikazan prelazak Ethernet-a iz LAN-a na transportni nivo.



Slika 11. Evolucija Ethernet-a iz LAN-a u Metro i transportnu mrežu

4.3 Optičke mreže sa xWDM multipleksiranjem

WDM danas omogućava prenos od 16 do 160 informacionih signala različitih protokola i različitih bitskih brzina na različitim talasnim dužinama kroz jedno optičko vlakno. Princip rada xWDM sistema prikazan na slici 12 zasniva se na istovremenom prostiranju više svetlosnih talasa različitih boja kroz jedno optičko vlakno.



Slika 12. Princip funkcionisanja xWDM sistema

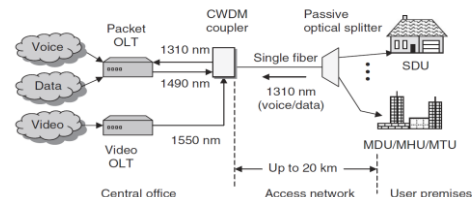
Postoje dva tipa optičkog WDM multipleksiranja, i to: CWDM (Coarse WDM) i DWDM (Dense WDM).

5. RAZVOJ OPTIČKIH MREŽA

Za razliku od konvencionalnih mreža, pasivne optičke mreže (Passive Optical Network – PON), nemaju aktivnih komponenta između glavnog komunikacionog centra i korisničkih prostorija. Ključna prednost PON sistema, zasnovanih na potpuno pasivnim komponentama, jeste u tome što ne zahtevaju napajanje.

5.1 Osnovne PON arhitekture

Pasivne optičke mreže (PON) ne sadrže nikakve aktivne optičke elemente na svojim mrežnim trasama. Slika 13 ilustruje osnovnu arhitekturu tipične PON mreže.



Slika 13. Osnovna arhitektura tipične PON mreže

5.2 Aktivni PON moduli

U aktivne PON module spadaju: OLT (Optical Line Terminal) koji se nalazi u glavnom komutacionom centru i on je zadužen za kontrolu bidirekcionih protoka informacija preko optičke distributivne mreže, ONT (Optical Network Terminal) je lociran direktno u korisničkim prostorijama, a njegova osnovna funkcija na tom mestu jeste da pruži optičku vezu sa PON mrežom, omogućujući tok u *upstream* smeru ka mreži i ostvari električnu vezu sa korisnikovom opremom na drugoj strani i ONU (Optical Network Unit) koji je obično smešten u posebnim kućistima za spoljnu opremu otpornim na vodu, jake vetrove i ojačanim protiv eventualnih vandalskih napada.

5.3 Prenosni tokovi

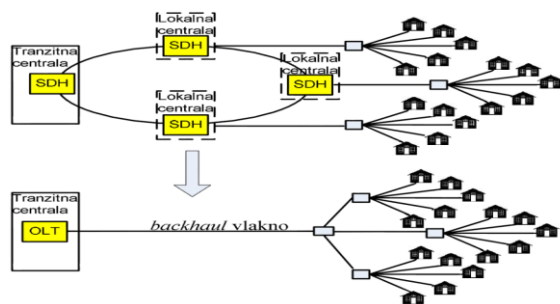
Dve ključne mrežne funkcije koje treba da obavlja OLT jesu kontrola korisničkog saobraćaja i dinamično dodeljivanje propusnog opsega ONT modulima. Kontrola saobraćaja obavlja se primenom različitih oblika sinhronizacija, a najjednostavniji način je vremenski multipleks, odnosno TDMA.

5.4 Aplikacije pasivnih komponenti

Tri glavne fotonske komponente kod pasivnih optičkih mreža su: različiti tipovi optičkih kablova korišćenih u mreži, spliteri optičke snage u okviru ODN-a, i kapleri koji se koriste u primopredajnicima.

5.5 PON varijacije

Postoje nekoliko različitih PON implementacionih šema. Tri najčešće korišćene su Broadband PON (BPON), Ethernet PON (EPON) i Gigabit PON (GPON). Aktuelna istraživanja u oblasti TDM PON sistema usmerena su ka implementaciji pristupnih mreža za velika rastojanja, koje imaju za cilj da skupe i nefleksibilne metro mreže zasnovane na prstenastim SDH/ATM arhitekturama, "premoste" i time omoguće da krajnji korisnici, putem *backhaul* linkova, budu direktno povezani na čvorove *core* mreže kao na slici 14.



Slika 14. Kombinacija metro mreže i pristupne mreže

6. ZAKLJUČAK

Nove usluge kao što su video na zahtev (VoD – *Video on Demand*), interaktivna IPTV, HDTV (*HighDefinition Television*), 3D TV, video konferencija više učesnika,

telemedicina i ostale aplikacije koje zahtevaju značajniji kapacitet, zasigurno će biti glavni pokretač nove generacije pristupnih mreža koje će moći zadovoljiti sve veće zahteve u pogledu brzine prenosa.

U narednih nekoliko godina telekomunikacionu mrežnu infrastrukturu i operatore očekuju značajne transformacije, koje će se realizovati kroz dva simultana procesa. Kao prvo, očekuje se unapređenje informacionog kapaciteta međunarodne i nacionalne mrežne okosnice. To podrazumeva povećanje bitskog protoka u transportnom delu mreže za barem jedan red veličine, što odgovara vrednosti od 10-15 Tb/s po jednom vlaknu. Drugi veliki događaj odnosi se na definitivni oproštaj od bakarne telefonske infrastrukture u pristupnom delu mreže i uvođenje FTTx tehnologije. U ovom trenutku među ponuđenim FTTx tehnologijama dominira TDMA-PON, ali se po svoj prilici, kao dugotrajnije rešenje nameće WDMA-PON. Optičke komunikacije ipak imaju izvanredne osobine i sve veću primenu, a time i dosta svetlu budućnost, a ostaje da se vidi kada će se izvršiti potpuna implementacija ovih mreža i u našoj zemlji, jer početne troškove za zamenu pristupnih mreža treba posmatrati kao neku vrstu investicije, jer se kasnije mogu ostvariti dosta veći prihodi iz tih mreža pošto će provajderi biti u mogućnosti da korisnicima pruže nove, modernije i zahtevnije servise i usluge.

7. LITERATURA

- [1] Dejan M.Gvozdić, "Trendovi razvoja optičkih telekomunikacionih sistema", Novembar 2008
- [2] Petar Matavulj, "Optoelektronika", Beograd 2007
- [3] Gerd Keiser, "FTTX Concepts and Applications", Jun 2008
- [4] Josep Prat, "Next-Generation FTTH Passive Optical Networks", Novembar 2008
- [5] Lazar Aščerić, "Kriptografski aspekti zaštite optičkih telekomunikacionih mreža", Beograd 2011
- [6] Davor Randelović, "Optički pristup do krajnjeg korisnika (FTTP)", master rad, FTN Novi Sad, Septembar 2012
- [7] Leonid G. Kazovsky, "Broadband Optical Access Networks", Oktobar 2011.

Kratka biografija:



Predrag Radojičić rođen je u Kraljevu 1984. god. Diplomski-master rad odbranio je 2016. godine na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Mikroelektronika.



Dejan Nemec rođen je 1972. god. Diplomirao, specijalizirao i magistrirao je na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva. Oblast interesovanja su telekomunikacije i obrada signala.

DIGITALNI VIDEO FORMATI**DIGITAL VIDEO FORMATS**Nebojša Vujković Cvijin, Željko Trpovski, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu će se sagledati kako su se video formati razvijali kroz istoriju i koliko se kvalitet digitalnog videa približio filmu na filmskoj traci.

Za razumevanje nastanka video formata, bilo je neophodno objasniti šta je to film, šta je filmska traka i koje vrste postoje, kao i šta je televizija. U radu je pokazan istorijski razvoj analognih video formata kao i postupak digitalizacije analognog video signala. Tehnologija koja se takođe objašnjava u radu je video kompresija sa pregledom vrsta kompresije kao i prednostima odnosno nedostacima pojedinih vrsta kompresije. U radu su opisani i neki značajniji digitalni video formati.

Abstract – This paper will examine how the video formats did evolve throughout history and how the quality of digital video did get close to the film on the film strip.

In order to understand the emergence of video formats, it is necessary to explain what the film is, what the film strip is and which types of film strip exist, and also what television is. This paper presents an historical development of analog video formats as well as the process of digitizing analog video signals. The technology that is also explained in this paper is the video compression, with overview of types of compression as well as the advantages and disadvantages of certain types of compression. The paper describes some important digital video formats.

Ključne reči: Film, televizija, digitalizacija, kompresija kodek

1. UVOD

Autor je tokom deset godina bio deo tima filmskog festivala Uvati sa mnom ovaj dan gde je bio zadužen za tehničku obradu filmova koji se prikazuju na festivalu. Paralelno sa tim, bavio se poslovima video montaže gde se susretao sa problematikom video formata i produkcijskim standardima. Odatle se razvila ideja da se u ovom radu pozabavi tom tematikom.

U ovom radu će se sagledati kako su se video formati razvijali kroz istoriju i koliko se kvalitet digitalnog videa približio filmu na filmskoj traci.

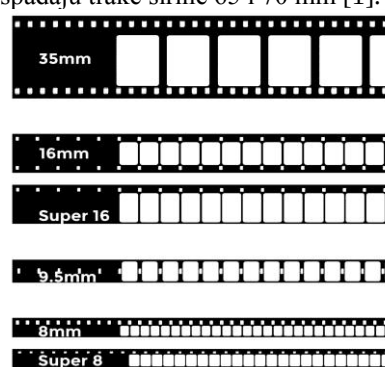
2. FILM I TELEVIZIJA

Prva javna projekcija pokretnih slika održala se 28. decembra 1895. u Parizu. Ovaj datum se zvanično smatra rođendanom filma, Francuska se smatra kolevkom, a „ocem filma“, fotograf, proizvođač fotoaparata i konstruktor sinematografa Lujis Limijer.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Trpovski, vanr.prof.

Filmskom kamerom snimaju se na filmsku traku jedan za drugim odvojeni filmski snimci. Pri normalnom snimanju u jednoj sekundi snimi se 24 (ili za televiziju 25) snimaka. Danas su se zadržali samo pojedini formati, prikazani na slici 1. Normalni ili standardni format – traka je širine 35 mm. Uzani format, substandardni – širina trake je manja od 35 mm. Među uzane formate ubrajamo 2x8, Super 8, 16 mm. Široki format, superformat – traka je šira od 35 mm. Tu spadaju trake širine 65 i 70 mm [1].



Slika 1. Razni formati video trake

Televizija je jedan od najatraktivnijih telekomunikacionih sistema današnjice. Godine 1929. ruski emigrant Zvoričin demonstrirao je u Americi prvi sistem koji je omogućavao prenos slike na daljinu. Prvi javni TV sistem počeo je sa radom u Londonu 1939. godine. 1954. godine u Americi su postavljene osnove sistema za prenos slike u boji, kompatibilnog sa klasičnim crno-belim sistemom. Kod nas je emitovanje programa počelo ranih šezdesetih godina, u crno beloj tehnici. Sredinom sedamdesetih godina i kod nas je počelo emitovanje programa u boji [2].

U zavisnosti od načina na koji se iz televizijske slike u boji dobija crno-bela razvijeno je više međusobno nekompatibilnih standarda, kao što su NTSC, PAL i SECAM. NTSC - (National Television System Committee) je razvijen 1950. godine kao prvi televizijski standard. Zasnovan je na 525 linija i frekvenciji osvežavanja slike od 60 Hz. Bio je u upotrebi u SAD, Kanadi i Japanu. PAL - (Phase Alternating Line) je 15 godina mlađi od NTSC-a. Njegove odlike su 625 linija i frekvencija osvežavanja od 50 Hz. Nešto je detaljniji od NTSC standarda jer ima 100 linija više, podložniji treptanju slike zbog niže frekvencije osvežavanja. U upotrebi je u najvećem delu Evrope. SECAM - (Sequence Couleur a Mémoire) je vršnjak PAL standarda. Eliminisanje grešaka u boji koje postoje kod NTSC standarda izvedeno je na drugačiji način od PAL-a, ali u svakom drugom pogledu s njim deli prednosti i nedostatke. U upotrebi je u Francuskoj, Grčkoj, Rusiji, nekim zemljama istočne Evrope i u Africi.

3. ANALOGNI VIDEO FORMATI

Q-format je praktično prvi upotrebljiv format za snimanje videa je, poznat i kao quadruplex, 2“ quad.

Nastao je iz potrebe odloženog emitovanja, naredne dve decenije je bio jedini način snimanja video sadržaja. Jedini VTR format koji koristi traku širine 2 inča i snima poprečne (transverzalne) tragove sa četiri magnetne glave pod međusobnim uglom od 90 stepeni. Sami uređaji su bili veliki, teški nekoliko stotina kilograma i u prvim verzijama sa elektronskim cevima zauzimali još i tri ormana elektronskih sklopova. Tek 1966. godine elektronske cevi su zamenili tranzistori kada je suženjem traga zapisa, omogućeno snimanje kolor signala. Naredne godine se pojavila i prenosna verzija, a 1974. i prvi Cart uređaj koji je omogućavao automatsko emitovanje.

Do **C-formata** je došlo kada su se posle dužeg samostalnog razvoja Ampex (A-format) i Sony usaglasili oko zajedničke specifikacije. Na traku širine 1 inč, pod malim uglom beleženi su veoma dugi (oko 40 cm) tragovi, opisujući helikoidalnu krivu. Sa užom trakom, koja se uz to kretala manjom brzinom, bilo je lakše manipulirati, a kako je u jednom tragu beležena cela poluslika, bez problema se mogao videti snimak i kada se traka nije kretala standardnom (normalnom) brzinom. Kvalitet tako dobijene slike bio je sasvim dobar za pretraživanje i pri velikim brzinama i traženju pravog mesta za rez pri malim brzinama. Za proizvođače uređaja predstavljao je izazov da se ostvari usporena (Slow Motion) kao i zamrznuta slika (Stop kadar).

U matik je sedamdesetih godina unapređen za potrebe ENG i rastuće TV produkcije vesti. Umesto snimanja PAL (ili NTSC) kompozitnog signala, U-matic koristi tehniku Color Under, gde se kvadraturno modulirane kolor komponente konvertuju u donji deo frekvencijskog spektra, a luminentna komponenta se za toliko pomeri naviše. Na taj način sačuvan je gotovo u potpunosti kolor spektar, i niske i srednje učestanosti luminentnog spektra, koji nosi najveću energiju video signala. Uvođenje kasete, automatsko šniranje trake i bolja zaštita trake tokom upotrebe i čuvanja, značajne su novine koje su omogućile prvi portabl snimač razumnih dimenzija i težine i značajno doprinele ogromnom uspehu U-matic formata.

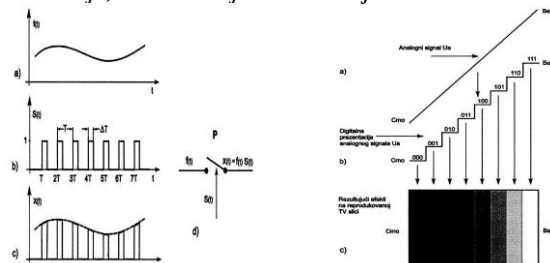
Prvo, kao neprofesionalni format namenjen edukaciji i industrijskom filmu, a kasnije profesionalnoj primeni U-matic je omogućio video snimanje na terenu sa troškovima kao sa filmom sa prihvatljivim kvalitetom. Tako je krajem sedamdesetih omogućio ENG i izbacio složeni proces filmske produkcije vesti i sportskih reportaža iz gotovo svih televizija. Kvalitet snimljene slike i zvuka bio je niži nego kod studijskih, Q i C formata, ali je doneo mobilnost i jednostavnost rukovanja.

VHS i Video 2000 su nastali u želji da zadovolje potrebu za amaterskim videom, ali još više za kućnim snimanjem rastućeg broja TV kanala. SONY i Panasonic su se upustili u jednu od najvećih industrijskih trka ikada. su. Oba rešenja koriste istu tehniku Color Under kao i odgovarajuću kasetu. Formati su u osnovi veoma slični, ali su međusobno nekompatibilni mehanički na nivou kasete ali i na nivou signala. U želji da snimak traje što duže, koristi se beleženje trag do traga bez zaštitnog prostora između tragova obaveznog kod profesionalnih formata. Brzina trake je veoma mala, nekoliko puta niža od U-matica tako da detalja u slici ima veoma malo, kao i visokih audio učestanosti. To je bila cena koja se morala platiti da bi se na video kasetu veličine manje knjige snimilo nekoliko sati programa.

BETACAM ili je uspešan komponentni format koji koristi Betamax kasetu sa magnetnom trakom. Na susedne tragove, naizmenično se beleže luminentna komponenta i dve hrominentne komponente prethodno vremenski komprimovane i multipleksirane. Na taj način hrominentne komponente su tokom celog procesa snimanja i reprodukcije odvojene međusobno i od luminentne komponente čime su prevaziđeni svi problemi preslušavanja koji su degradirali kvalitet kod kompozitnih i Color Under formata. Pored toga zahvaljujući maloj kaseti SONY je spajanjem kamere i snimača lansirao i prvi „kombo“ kamkorder, omogućavajući napokon jednočlanu ENG ekipu sličnu odgovarajućoj filmskoj [3].

4. DIGITALIZACIJA VIDEO SIGNALA

Pod digitalizacijom se podrazumeva konvertovanje kontinualnih analognih signala u niz diskretnih digita ili cifara. Radi žičnog ili bežičnog prenosa, niz digita se predstavlja nizom impulsa, koji su električni ekvivalent digita. Amplitude diskretnih impulsa odgovaraju trenutnim naponskim vrednostima koje analogni signal ima u pojedinim vremenskim intervalima. Da bi se realizovala ovakva digitalizacija, potrebno je obaviti tri osnovne operacije: odmeravanje, kvantovanje i kodovanje.



Slika 2. Model odmeravanja

Slika 4. Model kvantizera

Odmeravanje se može ilustrovati na modelu elektronskog prekidača, slika 3. Na ulaz prekidača dovodi se analogni signal $f(t)$, koji se menja kontinualno u vremenu. Povorka uskih pravougaonih impulsa jedinične amplitude naziva se funkcijom odmeravanja $s(t)$. Period ponavljanja ovih impulsa obeležen je sa T , a recipročna vrednost od T predstavlja učestanost odmeravanja $f_s = 1/T$. Prekidač P se elektronski zatvara za vreme trajanja jednog impulsa odmeravanja. U pauzi između ovih impulsa prekidač je otvoren. Jasno je da će se na izlazu prekidača pojaviti odmereni signal $x(t)$, koji se matematički može izraziti kao proizvod analognog signala $f(t)$ i funkcije odmeravanja $S(t)$, ili $x(t) = f(t)S(t)$.

Dalji postupak A/D konverzije sastoji se u kvantovanju trenutne naponske amplitude svakog odmerka u odgovarajućem vremenskom trenutku tako što se njegova vrednost izražava brojem usvojenih mernih jedinica i ona se naziva kvant amplitude, prikazano na slici 4. Ovaj postupak se obavlja u kvantizeru. Završna operacija digitalizacije ulaznog analognog signala obavlja se pretvaranjem tih kvantova svakog odmerka u odgovarajući binarni broj postupkom kodovanja pomoću nekog binarnog koda, što se obavlja u koderu [4].

5. KOMPRESIJA

Video kompresija je tehnologija kojom se omogućuje da se video zapiše na takav način da zauzme što manje memorijskog prostora a da se taj video zapis pri puštanju (čitanju, reprodukciji) što manje razlikuje od originala. Codec je skraćenica od CODER-DECODER ili drugim rečima: enkoder-dekoder. Enkoder znači ono što obavlja

stvaranje video zapisa u određenom formatu. To je softverska komponenta koja je obično ugrađena u neki program za enkodiranje ili se posebno instalira i povezuje sa programima. Dekoder obavlja "čitanje" video-zapisa.

Lossless video kompresija je vrsta video kompresije kod koje nema nikakvog gubitka kvaliteta. Reprodukciom lossless kompresovanog video zapisa se postiže prikaz videa identičan originalu..

Lossy video kompresija je svaka kompresija kod koje je nemoguće zadržati 100% kvalitet slike od originala. Kod ovakve kompresije uvek postoji barem mala razlika u odnosu na original tj. bar mali gubitak kvaliteta. Današnje tehnologije omogućavaju da taj gubitak kvaliteta bude neprimetan. Ogromna prednost Lossy kompresije je što se dobijaju mnogo manji fajlovi. Postoje dve osnovne tehnike kompresije. Intraframe (unutar frejma) ili spatial kompresija obuhvata tehnike koje su uobičajene kod formata za zapis slika (fotografija). Interframe (međufrejmovska) ili temporal kompresijom videa postiže se smanjenje niza sličnih frejmova. U praksi, razlika između frejmova koji su u dodiru je veoma mala. Interframe tehnike kompresije koriste tu sličnost kao osnovu za uštedu. Umesto da se svaki frame zapiše u celosti, zapisuje se samo opis u čemu je taj frejm različit u odnosu na prethodni (ili sledeći) frame.

Video zapisi se najčešće sastoje od tri vrste frejmova. **I-Frames (Intra-frames) ili ključni frejmovi (key-frames)** imaju veoma važnu ulogu i često se nazivaju ključni frejmovi (key-frames). Naziv intra-frames potiče od toga što su ovi frejmovi zapisani isključivo intraframe kompresijom. Oni sadrže kompletan zapis slike i potpuno su nezavisni od drugih tipova frejmova. **P-frames (Predicted-frames)** su složeni frejmovi koji se sastoje od delova koji se predviđaju na osnovu prethodnih frejmova. Oni koriste međufrejmovsku (interframe) kompresiju. Sadrže samo opis koji dekeru govori u čemu je razlika u odnosu na prethodni frame i zbog toga zauzimaju znatno manje memorijskog prostora nego I-frejmovi. **B-frames (Bi-directional-frames) i Bi-directional encoding** u bukvalnom prevodu znači dvosmerno enkodiranje. Pri korišćenju Bi-directional metoda enkodiranja se osim I i P frejmova koriste i B-frejmovi. B-frames su slični kao P-frames samo što za razliku od njih mogu da se preračunavaju i od prethodnog i od sledećeg frejma i tako postižu najmanje moguće zauzeće memorijskog prostora. Svaki niz frejmova koji počinje I-frejmom i traje sve do sledećeg I-frejma se naziva GOP (group of pictures).

Veći bitrate daje veći kvalitet ali i veće fajlove. Prednost većeg bitratea je najviše u tome što olakšava posao codecima pa se može koristiti i neki od jednostavnijih i bržih metoda enkodiranja. Manji bitrate daje manji kvalitet ali i manje fajlove [5].

6. DIGITALNI VIDEO FORMATI

U svetu videa postoje tri glavne vrste fajlova: Containeri, Audio streamovi i Video streamovi. Container je fajl koji sadrži video i audio zapise i prateće informacije neophodne za sinhronizovanu reprodukciju audio i videa. Container se još naziva stream ili AV stream. Video zapis (video stream) i audio zapisi (audio stream) koji su elementi containera se još nazivaju elementary streams.

MPEG-2 je savremeniji standard koji se prvenstveno koristi za video zapis na DVD video diskovima. MPEG-2

se takođe koristi kao standardan format za digitalnu televiziju (zemaljsku, kablovsku i satelitsku). To su uglavnom industrijske primene a osim navedenih retko se koristi. Najbitnija osobina MPEG-2 videa je da on vrlo malo opterećuje procesor i pri enkodiranju i pri reprodukciji. Zbog te osobine je najpogodniji lossy format za snimanje TV programa na slabijim računarima.

MPEG-4 part 2 ("MPEG-4 ASP") je savremen standard koji se izuzetno mnogo koristi kod videa na PC računarima a u novije vreme i svi kućni plejeri ga podržavaju. Najvažniji deo ovog standarda čini profil ASP (Advanced Simple Profile) koji je postao veoma popularan zahvaljujući poznatom DivX codecima. Video u ovom formatu je širim masama poznat kao DivX. Naravno DivX nije jedini koji koristi ovaj standard ali je svakako najpoznatiji. MPEG-4 ASP postiže 50-70% snažniju video kompresiju od MPEG-2 tehnologije.

H.264 je najnaprednija tehnologija video kompresije. Omogućava do 30% snažniju video kompresiju od MPEG-4 ASP tehnologije. H.264 je osnovni standard za video na budućim Blu-ray diskovima i televiziji visoke rezolucije (HDTV). Nedostatak ove tehnologije je u tome što zahteva dosta snažan procesor pri dekodiranju. Ova video tehnologija je definisana sa dva tehnički identična standarda "H.264" i "MPEG-4 part 10 (Advanced Video Coding)". Zbog toga se osim naziva H.264 ponekad koristi naziv MPEG-4 AVC.

HEVC (High Efficiency Video Coding, drugi naziv standarda je i H.265) nastoji da postigne evolutivni napredak u odnosu na H.264 kompresiju. Trebalo bi da postigne oko 50% smanjenje bitratea (u odnosu na H.264) uz povećanje kompleksnosti (hardverske zahtevnosti pri enkodiranju i dekodiranju) ili manju uštedu bitratea ali uz smanjenje kompleksnosti. Takođe donosi mnogo veću univerzalnost, fleksibilnost i otpornost na greške i zastoje tokom video streaminga. Podržava sve visoke rezolucije (čak do UHD TV 7680x4320 i 8192x4320), visokokvalitetne boje (čak do 12-bit 4:4:4), visok framerate (čak do 300fps), 3D (stereoskopski) video, progressive scan (svaki frejm je Progressive a Interlaced je omogućen kroz meta-podatake za reprodukciju), podrška za višezegrene procesore, skalabilnost bitratea tokom video streaminga i slične zahteve.

MJPEG 2000 video zapis tretira kao seriju slika u JPEG 2000 formatu. Može se koristiti i za Lossy i Lossless kompresiju. Zbog superiornog kvaliteta slike i mogućnosti obrade, veoma je zanimljiv filmskoj industriji. DCI (Digital Cinema Initiatives), organizacija najvećih holivudskih studija, je usvojila Motion JPEG 2000 kao format filmske industrije.

AVI je najpopularniji format container fajlova na PC računarima. Sadrže video zapis u MPEG-4 ASP formatu i audio zapis u MP3 formatu.

DIVX je Format veoma sličan AVI fajlovima. Napravljen je tako da zadrži maksimalnu kompatibilnost sa AVI fajlovima ali i da pruži neke naprednije mogućnosti kao što su meniji, više video snimaka, subtitlovi, poglavlja (chapters) itd. Praktično DivX fajlovi su specifični po tome što mogu da sadrže kompletnu DVD funkcionalnost u samo jednom fajlu.

MKV (Matroska) je napredni format containera koji ima mnogo prednosti u odnosu na AVI i MP4. Najvažnije je to što je univerzalan. Može da sadrži neograničen broj

video i audio zapisa u bilo kom formatu. Iako je tehnički gledano najbolja i ima mnogo prednosti, veliki nedostatak je u tome što Matroska nije industrijski standard pa nije podržana u programima za obradu videa.

MP4 je format container fajlova koji je definisan standardom MPEG-4 part 14. Iako je standardan retko se koristi osim na mobilnim telefonima.

3GP je standardni container namenjen za video na mobilnim telefonima. U suštini 3GP je pojednostavljena verzija MP4 formata fajlova koja je posebno prilagođena za mobilne telefone i druge male uređaje.

MPEG Program Stream (PS ili MPEG-PS) je standardni container za MPEG-2 video zapis koji je prvenstveno namenjen za pouzdane medije kao što su optički diskovi, kompjuterski fajlovi i slično.

MPEG Transport Stream (MPEG-TS) je standardni container koji je namenjen za digitalni televizijski signal i druge oblike emitovanja (broadcasting) audio-video signala.

VOB je posebna varijanta MPEG Program Stream containera koji se koristi na DVD diskovima. VOB fajlovi su komplikovane strukture. U sebi sadrže veći broj zapisa videa, zvuka, menije, subtitlove itd. Mogu biti zaštićeni od kopiranja enkripcijom. Kreiraju se samo u specijalizovanim programima za DVD authoring (izradu DVD-a).

M2TS (BD-RE MPEG-2 Transport Stream) je posebna varijanta MPEG Transport Stream containera koja je namenjena isključivo za korišćenje na Blu-ray Disc medijima. Ovaj format nije ograničen na MPEG-2 video zapise već može i H.264 (MPEG-4 AVC).

DivX Plus izdvajamo kao poseban format fajlova jer se radi o najboljoj kombinaciji popularnih savremenih tehnologija. DivX Plus su fajlovi u kojima su kombinovana sledeća tri formata: container fajl .MKV (Matroska), video zapis H.264 (preciznije H.264 High Profile Level 4.0) i audio zapis AAC (preciznije LC AAC), DivX Plus plejeri takođe podržavaju i AC-3 kao format audio zapisa u DivX Plus fajlovima

AVCHD je familija formata za savremene digitalne kamere (kamkordere). Razvile su ga zajednički kompanije Sony i Panasonic. Veoma dobar i veoma popularan. Prisutan je kao osnovni format snimanja na skoro svim amaterskim i poluprofesionalnim kamerama. Koristi isključivo H.264 video kompresiju

iFrame je format snimanja razvijen od kompanije Apple i pre svega namenjen za male i jeftine digitalne kamere.

ASF ili WMV je Microsoftov format audio-video fajla. Često se koristi za video klipove na internet

Apple-ov format **MOV** je po strukturi sličan MP4 fajlovima ali je jako slabo podržan u programima za rad sa videom. Najčešće ga srećemo kod videa ugrađenog u web stranice.

Real Media je poseban format kompanije Real koji je prvenstveno namenjen za upotrebu na Web stranicama i gledanje pomoću Real Player-a. Real video se kreira i obrađuje isključivo oficijelnim softverom i zbog toga je poprilično nepopularan.

Flash Video je format video fajlova koje koristi Adobe Flash Player. Namenjen je za korišćenje na web stranicama. Pušta se direktno iz Flash aplikacija ugrađenih u web stranice. Retko koji program može da barata sa ovim formatom [6].

7. ZAKLJUČAK

Filmska traka je od svojih početaka do danas, bila i ostala najkvalitetniji medij za snimanje „pokretnih slika”. U novije vreme ima veliku konkurenciju u digitalnim sistemima zapisa slike. Kroz svoje postojanje, filmski medij se uz frekvenciju od 24 sl./sek. raširio kao standard za snimanje i projekciju u celom svetu. S druge strane u elektronskom mediju izmenilo se mnogo formata zapisa i reprodukcije.

Dodatna poteškoća za standardni elektronski mediji su, različiti svetski standardi TV signala. Ne čudi stoga da se u produkcijama namenjenim za svetsku televizijsku i video distribuciju iz kvalitativnih, kao i praktičnih razloga bira filmska traka kao originalni medij snimanja, a gotov proizvod se zatim bez poteškoća pretvara u elektronski format. Filmska traka još nije izašla iz upotrebe u modernoj filmskoj produkciji, ali ono što se uveliko promenilo su načini celokupne postprodukcije, kao i načini sinhronizacije slike i zvuka [1].

Razvoj digitalnih video formata je uslovljen razvojem procesora koji ih kodiraju i dekodiraju. Bilo je potrebno mnogo vremena da mogu da cenom i kvalitetom pariraju analognim video formatima.

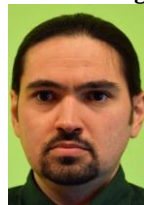
H.264 format (MPEG4 verzija 10) je postao standard u digitalnoj televiziji zamenivši MPEG-2 format jer pri manjem protoku obezbeđuje isti kvalitet slike.

Motion JPEG 2000 je postao deo standarda za digitalne bioskopske projekcije, istisnuvši tako filmsku traku kao medij za distribuciju.

8. LITERATURA

- [1] Skripta (2012), Osnovi tehnologije filmske i TV montaže, FDU - dostupno na internetu http://www.akademijaumetnosti.edu.rs/images/Vesti_2013_1/SKRIPTA_Osnovi_tehnologije_filmske_i_tv_montaze.pdf
- [2] Trpovski, Ž. (2004), Osnovi telekomunikacija, Delta Press, Sremska Kamenica
- [3] Isakov, S. (2009), Video snimanje : (prakčan vodič kroz popularne formate za snimanje videa), Medi art service international, Novi Sad
- [4] Zdravković, S (2012), Osnovi video-tehnike, MST Gajić, Beograd
- [5] <http://www.videomajstor.com/kompresija-osnove> (datum pristupa 16.07.2016.)
- [6] <http://www.videomajstor.com/standardi> (datum pristupa 28.07.2016.)

Kratka biografija:



Nebojša Vujković Cvijin rođen je u Subotici 1982. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Osnovi telekomunikacija odbranio je 2016. god.



Željen Trpovski rođen je u Rijeci 1957. godine. Doktorirao je na Fakultetu tehničkih nauka 1998. god. Oblast interesovanja su telekomunikacije i obrada signala.

SERVISI TELEVIZIJE ZASNOVANE NA INTERNET PROTOKOLU**SERVICES OF INTERNET PROTOCOL BASED TV**Nebojša Janković, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj - U ovom radu je predstavljena arhitektura IPTV sistema i dat je opis njegovih glavnih delova. Pažnja je data kompresiji, kao osnovnom uslovu za prenos slike i zvuka kroz prenosni sistem. Opisan je Headend, kao centar za prijem i pripremu sadržaja za slanje, zatim IP/MPLS mreža kao visoko adaptibilna mreža za različite tipove prenosnih medija i protokola i middleware kao posrednički element koji se proteže od Headenda do terminala.

Pomenuta je i pristupna mreža i opisane su najbitnije tehnologije vezane za nju. Na kraju je nešto više rečeno i o servisima koji predstavljaju osnovnu prednost IPTV-a u, odnosu na druge tipove televizija.

Abstract - In this document the IPTV system architecture is represented. A description of its main parts is given. The attention is on the compression, as basic requirement for the transmission of video and sound through the transmission system. Headend, a center for reception and preparation of a sending content is described. IP/MPLS network is shown as highly adaptable network for different types of transmission media and protocols and middleware as mediatory element which has influence from Headend to terminal. In addition, the access network and its most important technologies are described. Services which represent the main advantage of IPTV, relative to other types of television are described.

Ključne reči: IPTV, MPEG, Headend, Middleware, Multicast, VoD.

1. UVOD

U drugoj polovini dvadesetog veka televizija je, samom svojom pojavom, u velikoj meri uticala, ne samo na tehnički, već i na društveni razvoj. U početku se, zbog tadašnjih tehničkih mogućnosti, zasnivala na zemaljskoj radiodifuziji analognog video signala. Neprestanim razvojem došla je do hibridnog nivoa, kada se početkom sedamdesetih godina pojavio Teletext. Prvi korak ka digitalizaciji televizije napravljen je pre petnaestak godina, tačnije 1994. godine, kada je uz pomoć „CU-SeeMe video konferencija” softvera emitovan prvi televizijski program putem Interneta. 1995. godine stvara se program uz pomoć koga se uspostavljaju pravila i formira se internetski video proizvod pod nazivom IPTV. IPTV jeste televizijski sistem sposoban za primanje i prikazivanje video stream-a iz različitih izvora, kodovanih na bazi IP paketa.

NAPOMENA:

Ovaj rad je proistekao iz master rada čiji je mentor bio dr Željko Trpovski, vanr. prof.

Takođe, sposoban je da pruži Triple Play televizijske usluge, kao što su video, audio i prenos podataka preko IP (Internet Protocol) standarda. Ključna i najvažnija prednost koju korisniku pruža IPTV jeste njegov prelazak u digitalnu i interaktivnu sferu televizije. Nova korisnička iskustva nastala pojavom IPTV jesu VoD (Video on Demand), PPV (Pay-per-View), nPVR (Network Personal Video Recorder), interaktivno učestvovanje, dvosmerna komunikacija (novi korisnički doživljaj), pristup Internet stranicama preko TV uređaja, pristup elektronskoj i govornoj pošti preko TV uređaja, online glasanje u nekim televizijskim emisijama, kreiranje programskih paketa u zavisnosti od ciljne grupe, ciljani marketing, izbor različitih uglova prikazivanja ako je program tako snimljen i druga. Sva ona pridonose rastućoj popularnosti ovog vida komunikacija. IPTV je podržan od strane velikih telekomunikacionih kompanija koje pružaju ovaj vid usluge svojim klijentima.

2. Kompresija u IPTV sistemu

Kompresija je ključna u razvoju multimedije, ne samo u vidu nosača sadržaja, već posebnu važnost ima u prenosu multimedije na daljinu. Nekompresovan sadržaj je praktično nemoguće koristiti za prenos pa kao jedina opcija ostaje kompresovanje sadržaja, i to u visokom stepenu.

Postoje tehnike kojima se služimo da bismo smanjili veličinu video sadržaja, a to su:

Intra-frame kompresija - tu se frejmovi posmatraju kao zasebne celine i vrši se kompresija samog frejma. Ova tehnika se primenjuje kod nepokretnih slika, a kako je video skup nepokretnih slika to je ova tehnika primenjiva i na njega. Radi se o prostornoj kompresiji.

Inter-frame kompresija - razlika između frejmova koji su susedni veoma je mala. Velika sličnost tih frejmova koristi se kao osnova za uštedu jer umesto da zapišemo svaki frejm u celosti mi beležimo samo razlike između aktuelnog frejma i prethodnog ili sledećeg. Ova kompresija je efikasna i mnoge tehnologije je koriste. Razliku predstavlja način na koji pojedine tehnologije beleže promene među frejmovima. Spada u vremensku kompresiju.

Postoje tri vrste frejmova koji se koriste kod video zapisa:

I-Frames, imaju osnovnu ulogu pa ih često nazivaju *key frames* (ključni frejmovi). Služe kao osnova od koje se preračunavaju ostali frejmovi. Sadrže kompletan zapis slike i potpuno su nezavisni od drugih tipova frejmova.

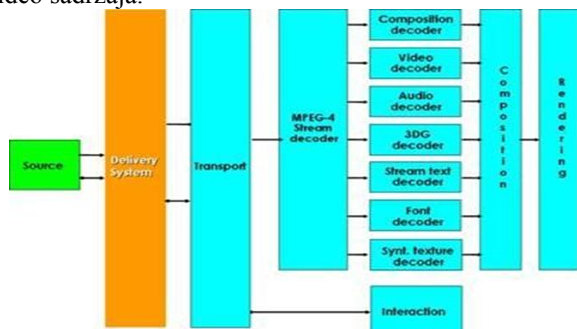
P-Frames, ovo su složeniji frejmovi jer predviđaju i preračunavaju osnovu prethodnih frejmova. Dekoder u reprodukciji video sadržaja tumači zapis P-frejma i

prethodnog frejma, a zatim proračunava sliku koja se prikazuje na osnovu tih podataka.

B-Frames, kod upotrebe B-frejma dekoderi dekoduju u oba smera. Postiže visok nivo sažimanja sadržaja. Spada u naprednu tehniku video kompresije jer omogućuje dobar kvalitet slike na slabije propusnim mrežama. Postoji više formata za slanje video podataka putem Interneta, a najzastupljeniji jeste MPEG format.

MPEG

Prvo se razvio MPEG1 sa stepenom kompresije 40:1 i brzinom prenosa 1,2 Mbit/s. Zatim je usledio MPEG2 sa kompresijom koja obezbeđuje brzine od 2-20 Mbit/s te je postao pogodan za distribuciju HDTV televizije. Nakon MPEG2 sledi MPEG4, koji kodira video u objektnom obliku i time se omogućuje mešanje prirodnih i sintetičkih slika. U drugom izdanju MPEG4, poznatom po kodeku H.264, podržavaju se: tekst, bitmapi (grafički prikaz načinjen od malih tačaka, tj. piksela), 2D i 3D vektorski grafički objekti, sintetički govor i muzika (*Text-To-Speech, MIDI*), kao i jezici koji služe za kompoziciju svih tih objekata, kreiranje scena i interakcije objekata sa korisnikom [1]. H.264/AVC p.10 ima prednosti kao što su: male dimenzije blokova, promenljiva veličina blokova za transformaciono kodovanje, unapređeni filter debloking, unapređena kompenzacija poretka itd. H.264 predstavlja jedan robusan kodek za prenos videa, koji odgovara današnjim potrebama za prenos videa visoke definicije. Na slici 1. vidimo primer procesa dekodovanja video sadržaja.



Slika 1. MPEG-4 proces dekodovanja [2]

H.265 HEVC, kod njega je glavni zahtev bio da se za isti kvalitet slike kompresija duplo unapredi što bi za posledicu imalo upola manji protok. Glavne promene HEVC-a u odnosu na H.264 se odnose na proširenje obrazaca po kojima se upoređuju delovi, kao i proširenje rezolucija kodiranja od 16x16 do 64x64 piksela uz poboljšanu varijabilnu segmentaciju blokova.

3. IPTV sistem

IPTV postaje zajednički imenitelj za sisteme kod kojih se televizija i/ili video signali distribuišu do rezidencijalnih pretplatnika preko širokopojasne mreže. IPTV podrazumeva funkcionisanje u zatvorenom mrežnom okruženju, koje osigurava propusnost neophodnu za nesmetan protok videa i audia visokog kvaliteta.

3.1 Arhitektura IPTV platforme

HeadEnd

Head-End predstavlja glavni deo u kome se vrši prihvatanje signala (putem satelita, kabla, zemaljskim putem ili

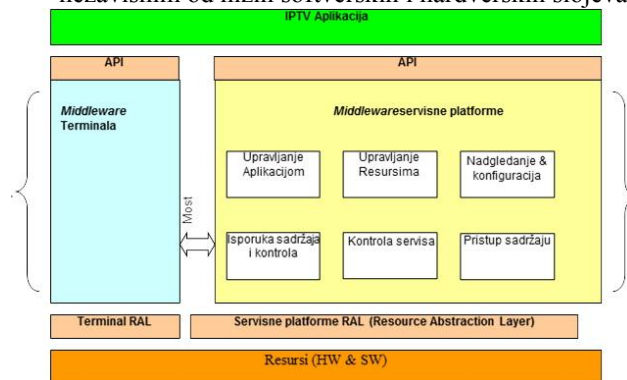
drugog IPTV-a), njegovo procesuiranje (vrše se određene radnje da bi signal bio spreman za slanje) i distribucija ka mreži. Uloga *Headend-a* jeste da signal prilagodi mreži kroz koju se prenosi da bi se postigao optimalan nivo iskorišćenja. *Headend* može da opsluži VoD servere, *middleware* (posredničke-servere srednjeg sloja) servere, EPG servere i može da podrži operativne sisteme provajdera servisa. Najčešće se sistem sastoji od jednog centralnog *Headend-a* (SHE - *Super Headend*) i više lokalnih (VHO - *Video Headend Office*) *Headend-ova*. Na centralnom *Headend-u* se nalaze kontinualni (uživo) sadržaji video servisa i *real-time* koderi, koji se koriste za obradu video signala, kao i systemske komponente za usluge „na zahtev”. VHO je mesto gde se nalazi kompleks video servera, koji se takođe koriste za servise „na zahtev”, kao i *realtime* dekoderi za stanice lokalnih televizija.

Middleware

Middleware je deo IPTV sistema od koga najviše zavise performanse i korisničko iskustvo. Predstavlja vezu između fizičkog uređaja (korisnik) i srednjeg sloja (pod-sistem). Integriše različite delove IPTV platforme u celinu. Slika 2. prikazuje osnovnu arhitekturu *middleware*.

Osnovnu strukturu čine nekoliko blokova:

- **IPTV aplikacioni sloj** – to je sloj gde operatori i „treće strane” obezbeđuju servise i aplikacije.
- **API sloj** - set interfejsa za provajdere servisa ili proizvođače, kako bi napravili određene aplikacije i kako bi ih prikazali sa detaljnim osnovama za različite namene.
- **IPTV middleware** - poziva niže slojeve resursa kako bi ih kontrolisao i kako bi obezbedio API za više slojeve. Podeljen je na *middleware* servisne platforme i *middleware* terminala. *Middleware* terminala IPTV-a smatramo centrom za hardversku kontrolu. Slojevi resursa za kontrolu (RAL) postoje za svaki određeni hardver i operativni sistem. *Middleware* servisne platforme vrši kontrolu servisa i sadržaja, upravlja aplikacijom i resursom, nadgleda sistem i konfiguriše.
- **Slojevi izdvojenih resursa** - Resource Abstraction Layer (RAL) služi da napravi *middleware* nezavisnim od nižih softverskih i hardverskih slojeva



Slika 2. IPTV arhitektura middleware-a

MPLS mreža

Kod tradicionalnih IP mreža rutiranje između dve tačke se zasniva na najkraćem putu. Kod MPLS mreža se uvodi složeniji model rutiranja, pod nazivom CR (*Constraint-*

based Routing), čiji se protokoli proširuju i time se uspostavljaju konkretne rute kroz celu IP/MPLS mrežu. Pomoću MPLS protokola obezbeđuje se prosleđivanje unicast (*unicast*) IP paketa, zahvaljujući posebnoj oznaci unutar paketa, nalepnici (*label*). Nalepnice (labele) se, uglavnom, dodeljuju na osnovu destinacije IP adrese paketa. Jezgro i spoljni prsten mreže čine strukturu MPLS-a. Jezgro se sastoji od uređaja koji obavljaju komutaciju MPLS paketa - LSR (*Label Switch Router*) uređaji, dok spoljni prsten čine uređaji, koji sa jedne strane primaju standardne IP pakete („labeliraju“) i šalju ih u jezgro MPLS mreže na komutaciju sa LER ili ELSR, a sa druge strane na izlazu iz MPLS mreže skidaju labele i usmeravaju IP pakete ka odredištu. Prednosti MPLS mreža jesu: Veća pouzdanost i bolje performanse u odnosu na tradicionalnu mrežu, lako kreiranje virtuelnih privatnih mreža, mogućnost postojanja različitih klasa saobraćaja itd.

IP Multikast

IP multikast predstavlja tehnologiju kojom se obezbeđuje ušteda u potrebnom opsegu za prenos podataka na način da se isti tok podataka istovremeno šalje ka većem broju korisnika. Korisnici zainteresovani za prijem multikast paketa, koji čine određenu multikast grupu, šalju ruterima, koji su im nadređeni, zahteve za prijem multikast saobraćaja upotrebom protokola za upravljanje grupama na Internetu - IGMP (*Internet Group Management Protocol*), nakon čega ruteri preuzimaju na sebe obavezu prosleđivanja saobraćaja između izvora i multikast grupe.

Pristupna mreža

Pristupne mreže su poslednji nivo distribucije do krajnjih korisnika. Pristupna mreža (*access network*) se sastoji od niza lokalnih petlji (*local loops*) i mrežne opreme koja se na njih povezuje. Fiksne pristupne mreže obuhvataju: POTS sisteme, N-ISDN, E1/T1, i xDSL tehnologije. Danas su najaktuelnije i najvažnije xDSL tehnologije. Kod njih se na korisničkoj strani nalazi razdelnik, koji u dolaznom smeru odvaja POTS saobraćaj od DSL saobraćaja. Na drugom kraju arhitekture DSL-a, u lokalnoj centrali, imamo DSLAM (*Digital Subscriber Line Access Multiplexer*) u kojem su ugrađene modemske DSL kartice. ADSL je bila prva xDSL tehnologija i razvojem je stigla do ADSL2 plus verzije kod koje je *upstream* brzine do 1 MBit/s, a *downstream* do 24 MBit/s. Druga bitna xDSL tehnologija je VDSL. 2005. godine je uveden VDSL druge generacije koji podržava brzine saobraćaja do 100 Mbit/s, u oba smera. Osmišljen je kako bi podržao prenos višekanalnog HDTV-a, prenos govora preko IP-a te videa na zahtev i videokonferencija. 2009. godine konzorcijum na čelu sa Alcatel-Lucent-om je osmislio sistem nazvan *vectoring*. On je smanjio preslušavanje do 70 procenata i time omogućio znatno veće brzine krajnjem korisniku, u dolasku do 300 Mbit/s, a u odlasku do 100 Mbit/s. Nova nadogradnja VDSL2 je *bonding* (usnoplajvanje parica, korišćenjem više parica za prenos istovremeno), koji je povećao domet signala i obezbedio znatno veće protoke.

Korisnička oprema

IPTV korisničku opremu najčešće čini STB (*Set Top Box*) uređaj, koji je mrežnim kablom povezan za ADSL ili VDSL modem. STB predstavlja softversku i hardversku

infrastrukturu sa strane korisnika, kojom se zahtevaju ili izvršavaju nalozi IPTV *middleware*.

Razlika između Internet televizije i IPTV-a

Internet TV ili TV putem Interneta je, za razliku od IPTV-a (*Internet Protocol TV*), sistem isporuke video sadržaja putem javne i otvorene Internet mreže. Taj video sadržaj je ograničenog kvaliteta. IPTV funkcioniše u zatvorenom mrežnom okruženju, koje osigurava propusnost neophodnu za nesmetan protok videa visokog kvaliteta.

4. PRENOS U IPTV SISTEMU

Protokoli su važan deo sistema za prenos video sadržaja. Nastali su kao poseban skup pravila prilikom povezivanja dve krajnje tačke ili čvora. Na Internet mreži se koriste dva glavna protokola. Jedan je UDP (*User Datagram Protocol*), koji radi bez uspostavljanja direktnih veza i drugi je TCP (*Transmission Control Protocol*), koji radi sa uspostavljanjem direktne veze.

UDP protokol, u zaglavlju se nalaze prenosni segmenti kapaciteta 8 bajta. Brojevi koji se nalaze u zaglavlju govore o izvršnom i odredišnom čvorištu, sadrže tridesetdvobitne Ipv4 adrese izvršnog i odredišnog računara. U drugom delu protokola se nalaze podaci, odnosno paketi AMD (*Application Message Data*).

RTP protokol, velike potrebe za prenosom podataka u realnom vremenu uticale su na uvođenje opšteg transportnog protokola koji bi prenosio takve podatke. Osnovna funkcija RTP protokola jeste da vrši multipleksiranje više tokova podataka, u realnom vremenu, u jedinstveni UDP paket. Sam format RTP-a je konfigurisan tako da ima više karakteristika kojima primaoci obrađuju multimedijalne podatke. Svakom paketu, koji ide u RTP tok, se dodeljuje redni broj (*sequence number*), koji je za jedan veći od prethodnog paketa. Ta numeracija omogućava da se na odredištu utvrdi da li nedostaje neki paket.

RTCP protokol predstavlja protokol za upravljanje prenosom u realnom vremenu. Definisan je zajedno sa RTP protokolom, a posao mu je da pruža povratne informacije i vrši sinhronizaciju.

IGMP protokol

Protokol koji je ključan za rad *multicast*-a naziva se IGMP (*Internet Group Management Protocol*). Jedna te ista poruka se može slati grupi korisnika više puta zasebno (višestruka isporuka sa ponovljenim slanjem), međutim, ako koristimo protokol kao što je IGMP onda možemo vršiti istovremeno slanje jedne poruke višestrukim korisnicima. Na ovaj način, imamo značajnu uštedu transportnih resursa. Trenutno je aktuelna treća verzija IGMP protokola, a jedna od njegovih primena je i zamena TV kanala kod multikast strima.

5. IPTV SERVISI

Kvalitet servisa (QoS)

ITU-T E.800 je definisala kvalitet usluge kao „ukupni učinak delovanja usluga koje određuju nivo zadovoljstva korisnika usluga“ [3]. QoS se može pratiti na tri nivoa:

- Aplikacija- Korisnik aplikacije je čovek i mora se pružiti aplikacija koja zadovoljava kvalitativne zahteve samog korisnika
- Sistem- Kvalitet na nivou sistema. Korisnik je aplikacija. To su kvantitativni parametri (propusnost, sistem posluživanja i raspoređivanja, vreme odziva).
- Mreža- Korisnik je sistem. Kvalitet same mreže. Mogu se porediti putem kvantitativnih i kvalitativnih parametara usluge. Najčešće je to propusnost, kašnjenje, kolebanje kašnjenja, raspoloživost, gubici i blokiranje.

Kvalitet korisničkog iskustva (QoE)

Kvalitet korisničkog iskustva *Quality of Experience* (QoE) je korisnikova percepcija o tome koliko je dobar posao uradio provajder isporučujući svoj servis. Ovo je veoma subjektivno i uzima u obzir različite faktore koje su van kvaliteta samog servisa, kao što je cena servisa, sredina gledanja, nivo stresa itd.

Elektronski programski vodič – EPG

Electronic Program Guide (EPG) je interaktivni vodič za planiranje i informisanje o televizijskim programima. Omogućava navigaciju, detaljne informacije o sadržaju, vremenu i kanalima, kao i podešavanje podsetnika putem menija daljinskog upravljača. EPG koristi grafički prikaz liste kanala u vidu tabele, kao i IP adresu za višestruko emitovanje na koju se šalju kanali unutar IP mreže.

Video na zahtev –VoD

Usluga koju korisnik dobija, omogućava pristup širokom spektru video sadržaja visokog kvaliteta i punu kontrolu nad njim. Ovoj usluzi se može pristupiti u svakom trenutku. Najčešći primer jesu filmovi i serije. VoD podrazumeva interaktivnost od strane korisnika što postavlja dodatne zahteve u smislu pouzdanosti, smanjenja kašnjenja i povećanja odlazne brzine od korisnika ka mreži.

Network personal video recorder- nPVR

Network Based Personal Video Recorder, korisnički servis koji snima uživo program i smešta ga na server, odakle korisnik može da pristupi sadržaju kad god poželi.

Bliski video na zahtev- NVoD

Near Video on Demand nudi isti program na različitim kanalima, sa različitim vremenom početka. NVoD koristi *multicast* format prosleđivanja video saobraćaja.

GoD (Game On Demand)

TV igre postaju neizostavan deo IPTV usluga, kako unutar određenog bloka korisnika (iste *multicast* grupe), tako i *on-line* igranjem putem Interneta. Velika je primena u velikim hotelskim lancima.

Plati i gledaj – PPV

Pay Per View ili *Pay Tv* - pretplatnik plaća sadržaj koji će gledati u rezervisanom terminu.

Broadcast TV

Ovu uslugu možemo opisati kao gledanje „klasične“ televizije. Dakle, usporediva je s ponudom koju možemo dobiti putem zemaljskih TV kanala i ponudom operatora kablovske TV.

Audio usluge

Osnovna audio usluga jeste radio putem IPTV-a. Krajnim korisnicima se nude različite muzičke liste sa odgovarajućim muzičkim žanrovima, koristeći vizuelne mogućnosti svojih TV uređaja (muzika na zahtev).

Slika u slici PiP

Picture in picture omogućava se da se unutar velike slike dobije manja slika u kojoj može biti, npr. EPG, drugi kanal, video nadzor itd.

Triple Play servisi

Triple Play, kao tehničko rešenje koje je zasnovano na širokopojasnim pristupnim mrežama i IP tehnologijama, je implementiran sa IPTV uslugom, a sve u cilju da se krajnjem korisniku obezbede tri bazna servisa: brzi Internet, IP telefonija (VoIP) i IP video servis.

6. ZAKLJUČAK

Vredi se zapitati zašto je IPTV postao toliko zanimljiv telekomunikacionim kompanijama. Odgovor je jednostavan. Zbog toga što alternativni operatori, nudeći triple play usluge, žele da postanu jedini dobavljači komunikacijskih usluga, a IPTV je temelj te strategije. Ono zbog čega je IPTV interesantan sve većem broju pretplatnika jesu usluge, kao VoD (*Video on Demand*), nPVR (*Network Personal Video Recorder*) i TS (*Time Shifting*), odnosno mogućnost iznajmljivanja filmova, mogućnost snimanja programa bez potrebe za video rekorderom, ali i mogućnost gledanja TV programa unazad. Po prvi put gledalac postaje urednik sopstvenog programa. Drugačije rečeno, IPTV predstavlja nov način distribucije televizijskog programa.

Dalji napredak ide ka kompletnom usklađivanju postojećih komunikacionih sistema, što će, na kraju, konvergirati u jedan IPTV sistem. Takav sistem omogućice isporuku interaktivnog TV sadržaj svuda i na svakom mestu, pri čemu će prelaz sa mobilnog na fiksni prijem, i obratno, biti bez primetnog prekida. Potrebno je doraditi sve segmente mreže, kako bi sama usluga i njeno korišćenje bili što jednostavniji. Zbog toga za očekivati je da će u predstojećem periodu širenje visoko propusnih mreža biti od velikog značaja.

7. LITERATURA

- [1]- Izvor PC Press, računarski časopis, Beograd 2006.
 [2]-Design and implement advanced digital media: <http://leonardo.chiariglione.org>, (Pristup u Septembru 2016).
 [3]-Wikipedia, the free encyclopedia: www.wikipedia.rs (Pristupljeno u septembru 2016).

Kratka biografija



Nebojša Janković rođen je u Prištini 1973. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva odbranio je 2016.god.

INTERAKTIVNA TELEVIZIJA**INTERACTIVE TELEVISION**Dragan Janković, Željen Trpovski, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj - U ovom radu je dat prikaz celovite strukture HbbTV sistema sa njegovim elementima. Ispraćen je razvoj hibridne televizije i dat je uvid u standarde zasnovane na različitim middlewareima. Posebno su obrađeni najbitniji standardi i tehnologije koji čine glavni deo specifikacija HbbTV-a. Fokus je bio na kanalima za prijem podataka i aplikacija, Broadcast i Broadband, kao i odgovarajućim prijemnikom, koji je u centru ovakvog sistema. Opisane su i vrste aplikacija, koje predstavljaju suštinu i razlog razvoja jednog ovakvog sistema, kao i njihova primena.

Abstract - In this document the whole structure of HbbTV with its elements is shown. We followed the development of hybrid television and we provided insight in standards founded on different middleware. In particular, the most important standards and technologies are processed which are the main part of HbbTV specifications. The focus was on the channels for receiving data and applications, Broadcast and Broadband, as well as corresponding receiver which is in the center of this system. Certain types of applications which represent the essence and the reason for development of such a system, as well as their use are described.

Ključne reči: iTV, HbbTV, Broadcast, Broadband, DVB, OIPF, AIT.

1. UVOD

Nekada je TV signal bio prilično jednostavan i podrazumevao je samo prenos slike i zvuka. Ubrzan tehnološki razvoj je doveo do toga da se sa analogne televizije pređe na digitalni koncept. Vremenom su se u TV signal počeli utiskivati razni servisi pa je on postao složeniji.

Ovakav progres stvari je doveo do toga da danas imamo pojavu interaktivne televizije, koja predstavlja evolutivni razvoj početnih servisa.

Postoji više koncepata, ali možda najbolji predstavnik takve televizije jeste HbbTV - *Hybrid Broadcast Broadband TV* (Hibridna Ematersko Širokopojasna TV mreža).

HbbTV je evropska inicijativa i industrijski standard čiji je osnovni cilj pružanje zabave krajnjem korisniku usluga, kao i pružanje raznih informacija od interesa u kombinaciji sa snimljenim ili uživo emitovanim TV programom, korišćenjem širokopojasnog pristupa Internetu upotrebom hibridnih STB uređaja.

NAPOMENA:

Ovaj rad je proistekao iz master rada čiji je mentor bio dr Željen Trpovski, vanr. prof.

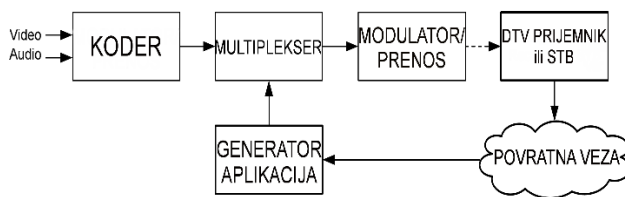
Hibridni STB uređaji imaju sposobnost da istovremeno mogu biti konektovani na dve mreže: ematersku prenosnu mrežu (*Broadcast network*) i širokopojasnu - Internet mrežu (*Broadband network*).

HbbTV predstavlja tehnologiju kojom realizujemo hibridne interaktivne DTV aplikacije, bilo da one u radu koriste emitovani prenosni tok ili pak vezu sa Internetom. Zasniva se na otvorenoj platformi, gde su članovi i osnivači HbbTV konzorcijuma uspešno razvili specifikacije kako bi stvorili globalni standard za uslugu hibridne zabave. Specifikacije HbbTV-a se baziraju na postojećim standardima i Web tehnologijama, a tu uključujemo OIPF (open IPTV Forum), CEA, DVB, EBU-TT i W3C.

Ovaj rad je imao za cilj da predstavi interaktivnu TV, a HbbTV posebno, kao budućnost u televizijskoj industriji, koji sa sobom nosi novi pogled na televiziju kao medij dajući joj jednu potpuno novu dimenziju. Rad se nije bavio nekim konkretnim detaljima, već celokupnim prikazom HbbTV sistema želeći da ga što sveobuhvatnije predstavi.

2. iTV (interaktivna televizija)

Interaktivnost podrazumeva da osoba koja prati neki TV program ne bude samo pasivni posmatrač dešavanja na ekranu, nego da može zahtevati dodatne informacije o onome što se prikazuje, kao i da u vezi sa tim preduzme određene akcije. Ovo predpostavlja da postoji nekakva povratna veza preko koje će gledalac komunicirati sa ponuđačem određene usluge.



Slika 1 - Dijagram TV ematersko-prijemnog koncepta [1]

Slika 1. daje opšti prikaz strukture iTV-a gde vidimo vezu između pojedinih blokova. Najpre se A/V podaci komprimuju po MPEG algoritmu i vode se do multipleksera gde se mešaju zajedno sa iTV aplikacijama, koje su prethodno generisane u generatoru servisa. Izlaz se vodi do modulatora koji proizvodi RF signal spreman za prenos na jedan od poznatih načina (satelitski, zemaljski, kablovski).

DTV prijemnik ili STB prihvata taj signal i vrši demodulaciju i reprodukciju A/V sadržaja i drugih servisa. Povratna veza, dalje, povezuje STB korisnika sa dava-

ocem TV usluga. Na taj način se ostvaruje interakcija korisnika i provajdera, a najbolje je koristiti običnu Internet vezu.

Middleware

Middleware označava posrednički softver i u tehnologiji DTV-a i STB-a predstavlja softverski sloj između OS-a i aplikacija. Developerima olakšava pisanje kompleksnih aplikacija jer oni ne moraju da brinu o problemima niskog nivoa STB-a. Pre svega se misli na drajvere i OS. Omogućava prenosivost između različitih hardvera i OS-a:

- Aplikacije će raditi na bilo kojoj platformi na kojoj je middleware portovan (prenesen).
- Programeri ne moraju da se brinu o odnosu između OS-a i hardvera svakog STB-a na koji će se aplikacija pokretati.

Danas imamo nekoliko različitih koncepata iTV-a (različiti *middleware*), koji se po tipu *middleware* dele na [2]:

- Vlasničke *middleware*, kompanijsko vlasništvo
 - Otvorene *middleware*, može ih svako implementirati
- Danas su najzastupljeniji MHEG-5, MHP, Ginga i HbbTV.

MHEG - Ovaj *middleware* je definisan kao otvoren standard od strane IMPALA i ISO. Osnovni pristup pri razvoju MHEG bio je kreiranje otvorenog standarda gde bi se pravile aplikacije koje bi mogle raditi na bilo kom OS-u, dok god su usklađene sa standardom. U toku razvoja se prošlo kroz nekoliko faza dok se nije došlo do verzije 5.

MHEG-5 se zasniva na prenosu podataka u obliku cikličnih objekata (*object carousel*). Dizajnirana aplikacija se koduje u ASN.1 fajlove. Podaci, kao što su slike ili tekst i kodovani fajlovi, su hijerarhijski ukombinovani i tako se prenose. Prijemnik čita *carousel* fajlove i propušta izlaz ka monitoru gde interaktivnost dolazi do izražaja. Jedna od glavnih karakteristika koju MHEG-5 podržava jeste povrata veza ili interakcioni kanal preko koga se dodatni podaci mogu dobijati i kombinovati sa aplikacijom.

MHP - Razvijen je od strane DVB Project. MHP je Java baziran *middleware* sistem. Osnovni principi rada su vrlo slični MHEG-u, a glavna razlika je na prijemnoj strani budući da dekoderski *engine* podržava Javu. Samim tim potencijali u programiranju su veliki, gotovo neograničeni [1]. Osnovni problem ovog sistema jeste njegova cena jer da bi se aplikacije izvršavale potrebna je Java virtualna mašina, koja zahteva jači, a time i skuplji hardver.

Važan detalj jeste da MHP u svojoj sistemskoj informacionoj tabeli ima dodatnu sekciju, a to je AIT (*Application Information Table* – Informaciona tabela o aplikaciji). Ona je potrebna da bi dekodek upirao da u prenosnom toku podataka (*Transport Stream*), na prijemu, postoji MHP aplikacija.

Ginga - *middleware* je razvijen od strane Brazilskih vlasti. Napravljen je od skupa standardizovanih tehnologija i nekih sopstvenih inovacija. Podeljen je na dva glavna podsistema. Jedan se naziva Ginga-J i zasniva se na Java aplikacijama, a drugi je Ginga-NCL i zasniva se na NCL aplikacijama. NCL je XML aplikacijski jezik koji obezbeđuje podršku za specifičnu vremensko-prostornu

sinhronizaciju između medijskih objekata i sadržaja, kao i proizvodnju uživo interaktivnih nelinearnih programa.

3. HbbTV

Pojam i sistem HbbTV nastali su u Februaru 2009. godine udruživanjem, u HbbTV Asocijaciju, francuskog projekta H4TV i nemačkog projekta German HTML Profil. Juna 2014. godine pridružuje im se Open IPTV Forum.

Može se reći da Teletext predstavlja neku vrstu preteče današnjih složenih TV servisa. Vremenom je izgubio bitku sa složenim tehnologijama baziranim na Internetu, ali sama filozofija isporučivanja dodatnog sadržaja uz TV program nije gubila na značaju. HbbTV se može opisati kao fuzija TV programa i savremenih tehnologija za prenos i prikazivanje podataka. Ideja je da uz video i audio sadržaj emiter isporučuje dodatne podatke koji mogu biti od značaja za gledaoca.

Dodatni podaci se isporučuju slanjem TCP/IP paketa zajedno sa A/V sadržajem, ali za razliku od teletexta gde smo dobijali tekst, ovde dobijamo sadržaj u vidu HTML stranica ili preciznije rečeno CE-HTML, koji se koristi kao osnova za prikazivanje sadržaja na ekranu. Sam naziv "hibridni" ukazuje na to da se u ovom sistemu terminal može povezati sa različitim tipovima mreža. Istovremeno se može povezati na dve mreže, s jedne strane se preko emiterkog kanala povezuje na DVB (DVB-T, DVB-C, DVB-S) mrežu, dok se sa druge strane preko svog širokopojasnog sučelja povezuje na Internet mrežu.

Slika 2 prikazuje HbbTV sistem sa oba svoja prenosna kanala (*Broadcast* i *Broadband*) i prijemnikom u sredini. *Broadcast*-om terminal može primiti standardni A/V sadržaj (TV kanali uživo - linearni A/V sadržaj), aplikacijske podatke i informacije o signaliziranju aplikacije. Od verzije 2.0 se može prenositi i nelinearni A/V sadržaj (sadržaj koji se ne emituje u realnom vremenu).

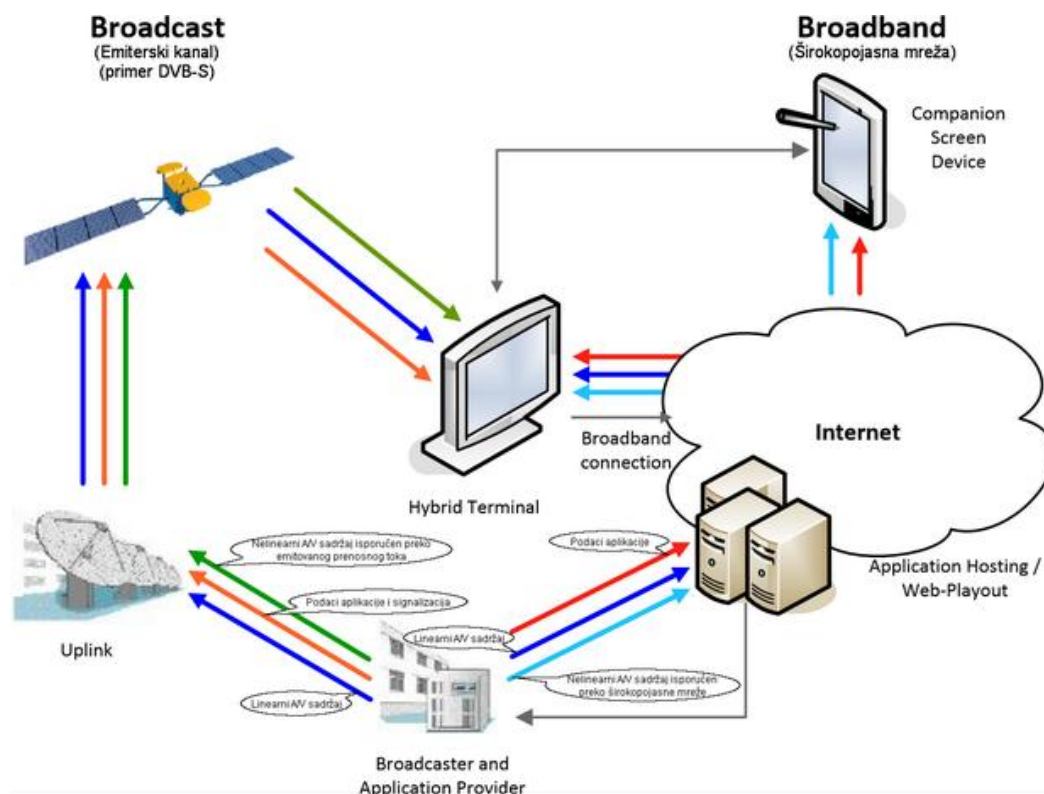
Preko širokopojasne mreže moguće je ostvariti dvosmernu vezu i komunikaciju sa davaocem usluga. Ovom vezom je moguće primiti aplikacijske podatke, nelinearni A/V sadržaj (A/V sadržaj na zahtev) i nelinearni A/V sadržaj koji se prvo dovede pa se gleda (*push-VoD*). Od verzije 2.0 moguće je primiti i linearni A/V sadržaj.

Broadcast predstavlja emitovanje A/V sadržaja ili sadržaja podataka do korisnika putem komunikacionog medija. HbbTV je preuzeo specifikacije organizacije DVB vezane za emitovanje prenosnog toka podataka, tako da se načini prenosa sadržaja i signalizacija aplikacija vrše po DVB standardu.

To znači da signal preko kodovanja u MPEG formatu i multipleksovanja nekoliko MPEG tokova podataka u prenosni tok podataka-TS, dolazi do modulatora koji ga moduliše i na kraju do *upconvertera* koji ga diže na potrebnu učestanost. Svaki TS se sastoji od elementarnih tokova podataka (*elementary stream* - ES).

Ti ES-ovi predstavljaju MPEG kodovane sadržaje (video, audio ili učeurene podatke). Radi lakšeg multipleksovanja i određivanja grešaka postoji 2 nivoa pakovanja gde se na kraju dobiju prenosni paketi (TP), čija je dužina 188 bajta i oni čine konačni TS. Svaki TP ima svoj PID koji predstavlja jedinstveni broj za određeni TP unutar TS-a.

Da bi različite sadržaje na prijemu tačno rekonstruisali u TS se ubacuju Servisne informacije (SI), koje opisuju strukturu TS-a, servise unutar TS-a i još neke informacije koje prijemnik koristi da bi sve mogao pravilno izdvojiti.



Slika 2 – Prikaz HbbTV sistema [3].

Broadband je drugi kanal hibridnog terminala i kao svoj medij koristi Internet. U pitanju je dvosmerna IP konekcija sa dovoljno širokim propusnim opsegom da omogući dovlačenje i strimovanje A/V sadržaja. **Broadband** sučelje se može iskoristiti i za povezivanje sa *Companion Screen* uređajima ili drugim HbbTV terminalima.

Za prenos A/V sadržaja **Broadband**-om definisani su sledeći formati [4]:

- H.264/AVC i MPEG-2 u HD i SD varijanti, za video.
- AAC, ACC, unapređeni AC-3, MPEG-1 Sloj II i III, Wave, DTS i MPEG Surround za audio.
- Sistemski podržani formati su TS i MP4.

Protokoli koji se koriste za media tok podatka su HTTP. Glavne koristi koje imamo od širokopojasne mreže su: prenos sadržaja na zahtev, otkrivanje aplikacija nezavisnih od prenosnog toka, prenos aplikacija zavisnih i nezavisnih od prenosnog toka, kao i pridruživanje podataka prenosu i razmena informacija između aplikacija i servera aplikacija.

Hibridni terminal se nalazi na korisničkoj strani i služi za prihvatanje A/V sadržaja. Na njemu se izvršavaju aplikacije koje provajder aplikacija može ponuditi korisnicima. Na **Broadcast Interface** stižu podaci AIT tabele, linearni i nelinearni A/V sadržaj, podaci aplikacija i *stream events*. **Runtime Environment** predstavlja komponentu terminala gde se interaktivne aplikacije izvršavaju, a čine ga tri komponente: menadžer aplikacija, pregledač i sučelje CS. **Broadcast Processing** komponenta obrađuje A/V koji se emituje u realnom vremenu - linearni A/V sadržaj. **Media Player** skalira medijski sadržaj za prikazivanje. **Broadband** sučelje ima vezu sa Internetom i preko njega terminal, može prihvatiti aplikacijske podatke sa servera davaoca. Preko ovog sučelja se može primiti i linearni i nelinearni A/V sadržaj. Uz pomoć *Internet Protocol*

Processing komponente se aplikacijski podaci šalju do *Runtime Environment*-a, a A/V sadržaj se prosleđuje do *Media Player*-a. Sinhronizacijski upravljač sinhronizuje A/V sadržaj dobijen preko emitterskog dela sa A/V sadržajem dobijenim preko širokopojasnog dela i to kombinuje sa *Media Player*-om. CS sučelje je komponenta koja pomaže terminalu da otkrije *Companion Screen* uređaje, kao i druge hibridne terminale. Putem CI+ sučelja terminal odgovara na zahtev aplikacijskih podataka nekog drugog *File System*-a kojeg nudi CI+ CAM.

3.1 Aplikacije

Standardom definišemo podelu HbbTV aplikacija na: aplikacije koje zavise od emitovanog prenosnog toka (*broadcast-related*) i aplikacije koje ne zavise od emitovanog prenosnog toka (*broadcast-independent*). Aplikacije predstavljaju skup dokumenata koji se sastoje od multimedijalnih fajlova, JavaScript-a, HTML-a, XML-a i CSS-a, ali i raznih unapređenih interaktivnih servisa.

Aplikacije koje zavise od emitovanog prenosnog toka su povezane sa servisima prenosnog toka ili sa događajima prenosnog toka u sklopu servisa i mogu se pokretati automatski ili na zahtev korisnika. Definišu se preko signalnih informacija u emitovanom prenosnom toku podataka. Signalne informacije sadrže lokacije samih aplikacija, ali i podatke koji nam govore koja od aplikacija treba automatski da bude pokrenuta. Ovaj tip aplikacija se dostavlja emitterskim kanalom, širokopojasnom mrežom ili preko CI+ CAM-a. Tu spadaju: *Red button* aplikacije (reklame, kvizovi, klađenja), EPG i zamene za Teletext.

Aplikacije koje ne zavise od emitovanog prenosnog toka nisu povezane ni sa jednim *broadcast* servisom, a dovlače se isključivo širokopojasnom mrežom i pristupaju

potrebnim podacima preko širokopojasne mreže. Tu spadaju: VoD, Catch-up TV, igrice, društvene mreže.

Portal aplikacije - predstavljaju složenije aplikacije koje su u stanju da realizuju grupu funkcija. To može biti neki složeni servis ili prodavnica aplikacija (*app.store*).

AIT tabele su deo signalizacije, multipleksuju se u TS i nose informacije o dostupnim aplikacijama i njihovim lokacijama. Ako se aplikacija prenosi Internetom iz AIT tabele se iščitava početna Internet stranica aplikacije i sa te lokacije se podaci dobijaju HTTP protokolom [5]. Ukoliko se šalje preko emitterskog kanala onda se koristi slanje podataka DSM-CC tehnikom u obliku cikličnih objekata (*carousel*). AIT sadrži podatke o kontrolnom kodu aplikacije, koji služi za njeno pokretanje. HbbTV standard definiše životni ciklus aplikacije, a to predstavlja vreme od stvaranja do uništenja aplikacije. Signalizacija, putem AIT tabela, koristi mehanizam kojim se započinje i zaustavlja život aplikacije.

3.2 Standardi

DVB – ETSI je povezao prenos interaktivnih aplikacija sa prenosom DTV signala. Za prenos koristi DSM-CC *object carousel*. Informacija o aplikacijama ide preko AIT tabele.

OIPF - od njega je HbbTV preuzeo neke specifikacije i servise koji rukuju različitim delovima sistema, kao što su: IMS, DLNA RUI, *Content Download*, *CoD metadata*, *Media Playback*, *Application Manager/Widgets*, *Content Service Protection*, *Remote Management*, *Scheduled Recording*, *Capabilities*, *Scheduled Recording*, *Notification*. DAE specifikacija kod HbbTV-a služi za prezentaciju korisničkog interfejsa i uključuje skripting podršku za interakciju sa aplikacijama koje se nalaze na serverskoj strani aplikacija. Obezbeđuje funkcionalnost kao što su personalizacija, glatko iskustvo između uređaja i funkcionalnost *Red button*.

CEA - definiše osnovnu funkcionalnost pregledača, koja je potrebna da bi se pravilno prikazale HbbTV aplikacije. Predstavlja *framework* kojim se prenose korisnička sučelja i preko Interneta i preko kućne mreže. Ovim standardom je omogućeno da svaki uređaj u UPnP kućnoj mreži svoje sučelje predstavi kao *Web* stranicu koju može prikazati bilo koji drugi uređaj kućne mreže.

HTML5 – HTML je jezik kojim se prezentuje *Web* stranica tj. prikazuje se izgled i sadržaj same stranice. HTML-om se obrađuju slike i tekst, formatiraju stranice, odnosno oblikuje se sadržaj i stvaraju hiperveze hipertekstualnih dokumenata. Osnovni zadatak HTML-a jeste da pokaže *Web* pregledaču kako da prikaže hipertekstualni dokument i on služi samo za njihov opis. Verzija HTML5 donosi unapređenja koja se tiču izrade *Web* stranica. Jedna od bitnijih je mogućnost reprodukcije videa na stranicama bez korišćenja raznih dodataka. Dodati su elementi za podršku audia, *canvas*-a, skalabilni vektorski grafički sadržaj, *drag and drop* podršku. Bolja je i obrada grešaka, smanjenjeno je korišćenje raznih dodataka, bazira se na DOM-u, *Javascript*-u, CSS-u. Doneo je i komponentu *WebSocket*, a to je asinhroni protokol koji obezbeđuje potpunu dupleksnu komunikaciju preko jedne TCP konekcije.

CI+ - *Common Interface* (CI) je tehnologija DVB-a koja omogućava da se funkcionalnost uslovnog pristupa izdvoji iz DTV prijemnika u prenosivi CAM. CI+ je specifikacija koja proširuje originalni DVB *Common Interface* standard. Glavni dodatak nove specifikacije je zaštita od kopiranja sadržaja između CI+ CAM i TV prijemnika.

4. Tehnologije

Sinhronizacija medija – sposobnost terminala da preko širokopojasne i emitterske mreže prima višestruke tokove podataka koje će dekodovati, renderovati i uskladiti. Ovo kontroliše HbbTV aplikacija preko *MediaSynchroniser* objekta. Kao primer primene sinhronizacije višestrukog toka podataka možemo navesti širokopojasnu isporuku audia koji se sinhronizuje sa *broadcast* videom.

Companion Screen (CS) - je uređaj koji može pokretati aplikaciju koja se povezuje sa HbbTV terminalom ili HbbTV aplikacijom. To može biti neki tablet ili mobilni telefon. Aplikaciju koju pokreće CS obezbeđuje ili davalac usluga da bi radila u sprezi sa nekom drugom HbbTV aplikacijom koja radi na terminalu ili sam proizvođač terminala kako bi se povezala i radila sa terminalom. Tri su modela upotrebe CS-a: pokretanje CS aplikacije, daljinsko pokretanje HbbTV aplikacija i komunikacija između aplikacija terminala i CS-a.

MPEG DASH - je tehnika za isporuku medijskog sadržaja kod HbbTV standarda. Baziran je na HTTP aplikacijskom protokolu. Glavni alat koji omogućuje ovu adaptibilnu funkcionalnost jeste MDP fajl. Ovaj fajl je zasnovan na XML jeziku i obezbeđuje HTTP klijenta sa informacijama koje su potrebne da bi se izabrao fajl ili tok podataka, koji je najodgovarajući u odnosu na korisnikove mogućnosti [6]. Prednosti MPEG DASH-a su u uslovima nestabilnog i varirajućeg protoka Interneta, a takođe nema problema sa *Firewall* i NAT-om. Koristi TCP kao prenosni protokol.

DSM-CC - je tehnika koja se koristi za prenos fajlova preko prenosnog toka podataka. Osnovni koncept u ovoj tehnici je način na koji se ova tehnika ophodi prema podacima poslatim ka prijemniku budući da je između prijemnika i predajnika komunikacija jednosmerna. Procedura kojom se podaci šalju se naziva *object carousel*. *Object carousel* funkcioniše po principu stabla podeljenog na module. Svaki modul može nositi direktorijume ili fajlove. Moduli se nakon kompletiranja emituju jedan po jedan, nekoliko puta. Ovo pomaže u postizanju boljih *download* (dovlačenje podataka) rezultata kod STB-a.

DRM - mehanizam služi kao podrška za zaštićeni sadržaj koji se preuzima preko Interneta. DRM predstavlja brojne tehnologije kontrole pristupa, koje određuju šta korisnik može uraditi sa svojim digitalnim sadržajem. Kod HbbTV-a je usvojeno da, ukoliko se zahteva interakcija između korisnika i CAM-a, autori aplikacija strogo preporučuju korišćenje *oipfDrmAgent* API-ja, koji će davati dozvolu za komunikaciju između CAM-a i HbbTV aplikacije. Koristi se kod *unicast* strima (primer, VoD).

PUSH VOD - uz pomoć ovog koncepta, HbbTV aplikacija zakazuje dovođenje raznih multimedijalnih sadržaja putem Interneta, pre njihovog prikazivanja.. Kada se sadržaj skine, može se pustiti i na taj način se dobija VoD iskustvo, oslobođeno eventualnih grešaka u prenosu.

Sadržaj je trenutno dostupan jer je smešten lokalno i terminal mu može lako pristupiti. Sve te funkcije aplikacija implementira korišćenjem naprednog API-ja za dovođenje sadržaja, *Content Download API*-ja.

Terminal mora da podržava "download feature" funkciju.

5. ZAKLJUČAK

Za HbbTV se može reći da je postao standard u Evropi. Glavni razlog njegovog proboja leži u činjenici da je prihvatio najbolja rešenja dotadašnjih koncepata, kao i u činjenici da je uzeo standarde i tehnologije koje su bile široko poznate.

Od samog starta je bio usmeren ka definisanju specifikacija baziranih na HTML-u i na ovaj način je HbbTV, već u startu imao velik potencijal u razvoju aplikacija. Za sada je 20 država u svetu HbbTV prihvatilo kao standard, razvijeno je oko 250 različitih aplikacija, a postoji preko 30 miliona uređaja u svetu povezanih na razne usluge HbbTV-a.

Razvoj DTV-a je otvorio vrata upotrebi mnogih servisa i time ubrzao razvoj HbbTV-a. Posebno je tu važan IPTV, jer je HbbTV uzeo koncept servisa IPTV-a i pokušao je da ih zaokruži i uobliči u celovito rešenje, prenosivo na sve platforme DVB-a. Zahtevi korisnika za različitim uslugama i bogatijim sadržajima su, takođe, odigrali važnu ulogu u razvoju HbbTV-a.

HbbTV danas korisniku obezbeđuje izuzetno iskustvo nudeći brojne zanimljive servise, kao što su: VoD, *Catch-Up*, razni informacioni servisi (vesti, sport, vremenska prognoza), administrativne e-usluge, EPG, igrice, edukacija, e-kupovina, reklame, kvizovi, interaktivne emisije i drugo.

Veliki potencijal HbbTV ima u živom kontekstu gde gledalac može da učestvuje u glasanju u nekoj emisiji ili takmičenju ili se može kladiti uživo, što je ljudima privlačno jer vole da budu uključeni u društvene medije i zabave raznog tipa. Takođe, mogućnost da se gleda neki film ili serija bez prekidanja (VoD) predstavlja veliku prednost, tako da se čini da je budućnost HbbTV-a prilično svetla.

6. LITERATURA

- [1] Panaousis Marioss, *Service Provision In The Digital TV Environment*, Jul 2011.
- [2] Steve Morris, *What is digital TV middleware?*, 2005
- [3] HbbTV Association, "*HbbTV 2.0.1 Specification*", april 2016.
- [4] Padam Chantyal, *Personalized Interactive EPG based on HbbTV*, januar 2013.
- [5] RTRK, *Digitalna Televizija, HbbTV*
- [6] Lourdes Beloqui, *Next generation HBBTV services and applications through multimedia synchronisation*, Septembar 2015

Kratka biografija



Dragan Janković rođen je u Skoplju 1976. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva odbranio je 2016.god.



Željko Trpovski rođen je u Rijeci 1957. godine. Doktorirao je na Fakultetu tehničkih nauka 1998. god. Oblast interesovanja su telekomunikacije i obrada signala.

**INTEGRACIJA SERVERSKE I KLIJENTSKE TEHNOLOGIJE
U RESPONSIVE VEB APLIKACIJAMA****INTEGRATION OF SERVER-SIDE AND CLIENT-SIDE TECHNOLOGIES
IN RESPONSIVE WEB APPLICATION**

Sanjin Malešević, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj: U radu je opisana implementacija programskog rešenja Kanban tabla kao i analiza tehnologija pogodnih za implementaciju pomenutog rešenja. Rešenje je implementirano kao veb aplikacija primjenom Angular 2 framework-a koji je iskorišten za implementaciju klijentske aplikacije dok je serverska strana implementirana primjenom Spring 4 framework-a.

Abstract: This paper describes development of Kanban board web application and investigation of adequate technologies for the implementation of the mentioned application. The application is implemented as web application using Angular 2 framework for the client side and the server side is implemented using Spring 4 framework.

Ključne reči: Klijentske tehnologije, klijentske aplikacije, Angular 2

1. UVOD

Zahtjevi korisnika informacionih sistema su se vremenom mjenjali, od prvih zahtjeva kao što su bile jednostavne veb prezentacije pa sve do današnjih zahtjeva kao što su jako kompleksni poslovni sistemi. Kako se povećavala kompleksnost korisničkih zahtjeva jednostavno je došlo do potrebe za razvojem i unapređenjem postojećih alata za razvoj i metodologija razvoja programskih rešenja.

Današnje kompanije na sve moguće načine pokušavaju da izađu u susret svojim klijentima. Koriste se nova programska okruženja i biblioteke kao i agilne metode razvoja programskih rešenja.

Svrha programskih okruženja i biblioteka jeste da poveća efikasnost prilikom razvoja programskih rešenja. Oni mogu pozitivno uticati na programera odnosno mogu povećati njegovu produktivnost, kvalitet koda, pouzdanost i robusnost proizvoda.

Klasične metode projektovanja programskih rešenja kao što je vodopad zahtjeva jako opširnu dokumentaciju koja je neophodna za razvoj. Činjenica da su ove metode vremenski zahtjevnije dovodi do zaključka da nisu dovoljno efikasne da se suprostave brzim promjenama koje se dešavaju. Danas veći fokus imaju agilne metode razvoja. Agilne metode se zasnivaju na iterativnom i inkrementalnom razvoju informacionih rešenja. Obično se projektuje u vremenski ograničenim iteracijama.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Milan Vidaković.

Takođe zasniva se na stalnim povratnim informacijama od strane razvojnog tima a pruža brz i fleksibilan odgovor na česte promjene.

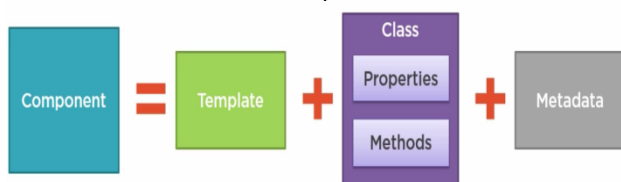
2. Tehnologije

Za razvoj Kanban table ovaj rad integriše sledeće tehnologije:

- **Java 8**, za razvoj serverske aplikacije u ovom radu korišten je Java programski jezik. Osmo verzija Java platforme donosi mnoge novitete, a ovde ćemo spomenuti nekoliko značajnih i većih: Lambda izrazi, Interface ima nove mogućnosti kao što je default method, novi Date i Time API.
- **Tomcat 8**, zbog velikog broja servera koji su danas dostupni izbor odgovarajućeg servera može biti prilično izazovan zadatak. Modularnost Tomcat aplikacionog server i činjenica da ne sadrži dodatne stvari koje su vezane za JEE kao i činjenica da za razvoj Kanban table se koristi Spring okruženje dovodi do zaključka da Tomcat predstavlja pravi izbor aplikacionog servera u ovom slučaju.
- **Spring 4**, Spring predstavlja vjerovatno najpopularniji framework za razvoj informacionih sistema zasnovanih na Java platformi. Jednostavnost korišćenja Spring framework-a je možda i najveći razlog zbog čega je toliko popularan. On pruža sredstvo za razvoj dok programeru ostaje da se fokusira na rješavanje domenskog problema informacionog sistema kog razvija. Takođe velika prednost Spring framework-a jeste njegova modularnost. Velika je šansa da Spring sadrži sve što je potrebno za razvoj jedne moderne aplikacije ali ako i dođe do potrebe za dodatnim bibliotekama Spring ima mogućnost vrlo lake integracije sa externim bibliotekama. Spring sa svojom verzijom 4 donosi podršku za rad sa Java 8 platformom, kao i da od verzije 4 Spring u potpunosti pruža podršku za Java konfiguraciju odnosno konfiguracija Spring aplikacije je moguća i kroz Java klase a ne samo putem XML datoteka kao što je to bio slučaj u predhodnim verzijama.
- **Web sockets**, cijela veb komunikacija je bila zasnovana na ideji da je klijent taj koji zahtjeva podatke od servera a server je taj koji te zahtjeve ispunjava. Ova paradigma nije narušena dugo godina, ali pojavom AJAX poziva mnogi su

počeli da istražuju mogućnosti dvosmjerne komunikacije između klijenta i servera. Razvijene su razne strategije koje omogućuju server da pošalje podatke klijentu a najpopularnije je *long-polling* i zasniva se na tome da *HTTP* veza bude otvorena dok server ne pošalje podatke klijentu. Vremenom količina podataka koja se razmjenjuje između klijenta i servera je postajala sve veća. WebSocket nam pružaju upravno što nam je i bilo potrebno, stalna veza između klijenta i servera i bilo ko od njih može slati podatke u bilo kom trenutku. Klijent je taj koji uspostavlja vezu kroz proces zvan WebSocket handshake (uspostavljanje parametara komunikacije). Proces počinje tako što klijent pošalje normalan HTTP zahtjev na server sa Upgrade zaglavljem koji informiše server da klijent želi da uspostavi WebSocket vezu.

- **MongoDB**, odabir odgovarajuće baze prilikom razvoja informacionih sistema je jako komplikovan zadatak i jako je bitno da se odabere baza koja nabolje reprezentuje domen informacionog sistema koji se razvija. MongoDB pripada grupi *NoSQL* baza i predstavlja najpopularniju bazu koja služi za čuvanje podataka u obliku dokumenata. MongoDB čuva podatke u *BSON* (Binary *JSON*) formatu koji je jako sličan *JSON* format.
- **Angular 2**, Angular predstavlja jedan od popularnijih JavaScript framework-a za razvoj klijenskih aplikacija koristeći HTML, CSS i JavaScript. Angular ima jako veliku zajednicu što predstavlja veliku prednost prilikom izbora JavaScript framework-a. Angular 2 predstavlja u potpunosti novi *JavaScript framework* čiji se rad zasniva na komponentama kao što je to slučaj kod većine modernih *JavaScript framework-a*. Svaka Angular 2 aplikacije se sastoji od skupa komponenti i servisa koje pružaju svoje funkcionalnosti komponentama, slika 1.

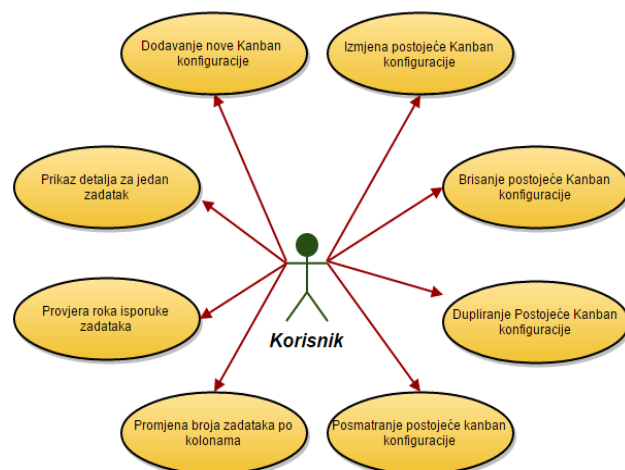


Slika 1. Sadržaj Angular 2 komponente

3. Specifikacija zahtjeva i model podataka

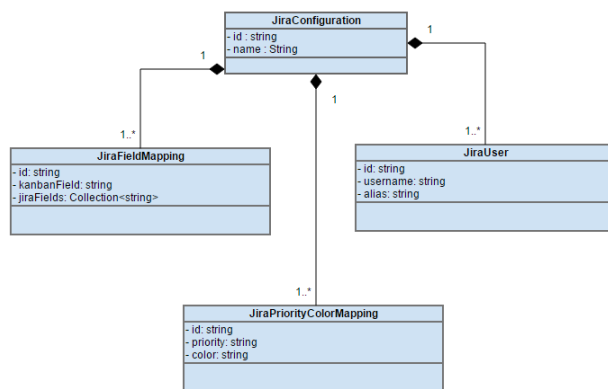
Specifikacija zahtjeva sistema je podijeljena na specifikaciju funkcionalnih i nefunkcionalnih zahtjeva. Slika 2. prikazuje slučajeve korišćena Kanban aplikacije i to su: dodavanje nove Kanban konfiguracije, izmjena postojeće Kanban konfiguracije, brisanje postojeće Kanban konfiguracije, dupliranje Kanban konfiguracije, posmatranje Kanban konfiguracije, promjena broja zadataka po kolonama, provjera roka isporuke zadataka i prikaz detalja jednog zadatka. Korisnički interfejs treba da prikazuje najnovije informacije iz Jira aplikacije u skladu sa definisanim pravilima za konkretnu Kanban konfiguraciju.

Prikazivanje podataka na korisničkom interfejsu se periodično osvježava i taj period ne smije biti duži od 60 sekundi što predstavlja jedini nefunkcionalni zahtjev.



Slika 2. Slučajevi korišćenja

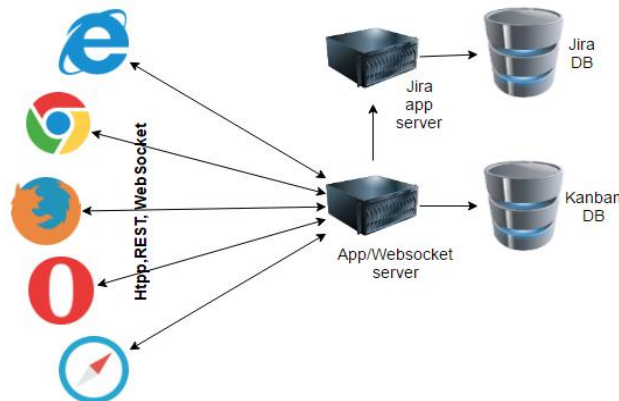
Kao što je prikazano na slici 3. model podataka je prilično jednostavan i modeluje način na koji podaci treba da se prikažu na korisničkom interfejsu odnosno kako Kanban tabla treba da izgleda.



Slika 3. Model podataka

4. Arhitektura sistema

Sistem koji se opisuje u ovom radu se zasniva na klijent-server arhitekturi koja se zasniva na *HTTP* (*Hypertext transfer protocol*) komunikaciji odnosno na *REST* (*Representational State Transfer*) protokolu. Arhitektura sistema je prikazana na slici 4. U ovom slučaju klijent predstavlja jedan od mnogobrojnih pretraživača koji iniciraju komunikaciju i šalju poruke u *JSON* formatu.



Slika 4. Arhitektura sistema

Server u ovom slučaju predstavlja Tomcat aplikativni server na kom se nalazi Kanban aplikacija. Ovaj server takođe omogućava komunikaciju klijenta i servera putem WebSocket protokola odnosno omogućava serveru da inicira komunikaciju sa klijentom u bilo kom trenutku.

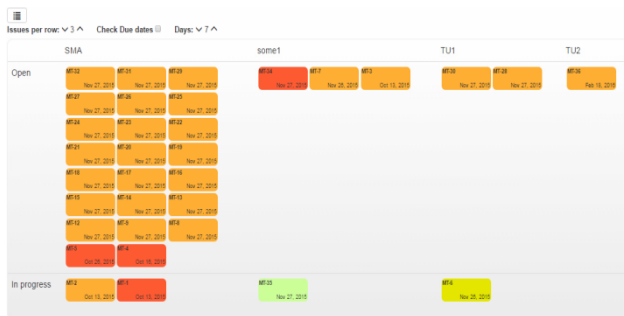
Kanban aplikacija koristi *WebSocket* protokol samo u određenim situacijama kada se razmjenjuje velika količina podataka između klijenta i servera.

U svim ostalim slučajevima za komunikaciju se koristi standardna klijent-server arhitektura putem *HTTP/REST* protokola. Server podatke pohranjuje u svoju bazu podataka koju u ovom slučaju predstavlja MongoDB bazu podataka. Server periodično inicira komunikaciju ka još jednom aplikativnom serveru na kom se nalazi Jira aplikacija i koja u ovom slučaju predstavlja izvor podataka koji se na osnovu logike Kanban aplikacije prikazuje na korisničkom interfejsu. Jira aplikacija svoje podatke čuva u relacionoj bazi odnosno SQL bazi podataka.

Kanban aplikacija komunicira sa Jira aplikacijom putem Jira API-a čija se komunikacija zasniva na *REST* protokolu.

5. Implementacija

Funkcionalnost koju ćemo opisati jeste posmatranje kanban konfiguracije. Da bi posmatrali promjene koje se dešavaju vezane za jednu od Kanban konfiguracija prvo trebamo sa prikaza svih Kanban konfiguracija u sistemu izabrati opciju za posmatranje. Nakon što korisnik izabere koju konfiguraciju želi da posmatra na korisničkom interfejsu sistem prikazuje Kanban tablu kao što je to prikazano na slici 5.



Slika 5. Prikaz Kanban table

Spring framework dolazi u paketu sa modulom za rad sa WebSocket protokolom tako da korišćenje WebSocket protokola u Spring framework-u je prilično jednostavno kao što je to prikazano na listingu 1. Klasa *JiraKanbanWebSocketConfig* je anotirana sa *@Configuration* anotacijom što ukazuje Spring framework-u da se radi o konfiguracionoj klasi. Takođe klasa je anotirana sa *@EnableWebSocketMessageBroker* anotacijom i na taj način omogućavamo rukovanje sa WebSocket porukama putem MessageBroker-a koji je i definisan u metodi *configureMessageBroker*.

Pored *MessageBroker*-a potrebno je još registrovati *endpoint*. *Endpoint* je registrovan u metodi *registerStompEndpoints* i u ovom radu je definisan kao *"/kanbanWs"* što znači da SockJS klijent treba da se konektuje preko *"/JiraKanban/kanbanWs"* URL-a.

```
@Configuration
@EnableWebSocketMessageBroker
public class JiraKanbanWebSocketConfig
extends
AbstractWebSocketMessageBrokerConfigurer {
    @Override
    public void
configureMessageBroker(MessageBrokerRegistry config) {
        config.enableSimpleBroker("/topic");
        config.
            setApplicationDestinationPrefixes(
                "/app");
    }
    @Override
    public void
registerStompEndpoints(StompEndpointRegistry registry) {
        registry.addEndpoint("/kanbanWs").
            setAllowedOrigins("*").withSockJS();
    }
}
```

Listing 1. WebSocket konfiguracija

Da bi se korisniku prikazale informacije na interfejsu prvo je potrebno te informacije prikupiti iz Jira aplikacije i onda primjeniti logiku vezanu za konkretnu Kanban konfiguraciju. Komunikacija sa Jira aplikacijom se ostvaruje preko Jira REST API-a. Jira API-u se prosljeđuje *JQL* (*Jira Query Language*).

Da bi mogli napisati *JQL* upit za Jira aplikaciju moramo imati potrebne podatke vezane za Kanban konfiguraciju koja se čuva u MongoDB bazi.

Osvježavanje korisničkog interfejsa odnosno komunikacija servera sa klijentom putem WebSocket protokola se poziva periodično iz *Scheduled* zadatka. Da bi se *Scheduled* zadaci omogućili u Spring-u dovoljno je napraviti konfiguracionu klasu i anotirati je sa *@EnableScheduling* anotacijom.

Kao što vidimo sa listinga 2. zadatak se aktivira svakih 5 sekundi što znači da će izmjene iz Jira aplikacije biti vidljive na korisničkom interfejsu u roku od 5 sekundi.

```
@Scheduled(initialDelay = 10000,
fixedDelay = 5000)
public void refreshData() {
    Iterable<JiraConfiguration> jcs =
jiraConfigurationRepository.findAll();
    for (JiraConfiguration jc : jcs) {
        Response response =
getIssuesForConfiguration(jc.getId());

        template.convertAndSend("/topic/
greetings/" + jc.getId(), response);
    }
}
```

Listing 2. Periodično slanje poruka klijentu

Klijentska aplikacija kao što je to već pomenuto je pisana u Angular 2 framework-u koji predstavlja *JavaScript framework*.

Za razvoj serverske aplikacije korišten je Eclipse IDE koji nije baš pogodan za rad sa modernim *JavaScript* okruženjima za razvoj. Iz tog razloga za razvoj klijentske aplikacije u ovom radu je korišten WebStorm. Angular 2 *framework* podržava kako ES5 (*ECMAScript* 5) tako i ES6 standard. Kako ES6 donosi novitete poput klasa što dosta olakšava razvoj klijentskih aplikacija ovaj rad je pisan upravo u šestoj verziji ES standarda.

Pored ES6 standarda u ovom radu je dodatno iskorišten *TypeScript* koji nam uvodi tipove u *JavaScript*-u. Treba voditi računa da ES6 nije u potpunosti podržan od svih pretraživača tako da treba biti jako oprezan kada se koristi neka funkcionalnost iz ES6 standarda. Takođe sav *TypeScript* kod mora biti kompajliran u čist *JavaScript*. Angular 2 se bazira na komponentama tako da je funkcionalnost posmatranje Kanban konfiguracije u klijentskoj aplikaciji implementirana kao zasebna komponenta koja po potrebe koristi druge komponente kao što je komponenta za Jira zadatak.

Svaka Angular 2 komponenta ima životni ciklus. Metode koje su u ovoj komponenti od velikog značaja su `onInit()` i `onDestroy()`. Kao što im sam naziv govori `onInit()` metoda se poziva prilikom inicijalizacije komponente što smo iskoristili za uspostavljanje WebSocket konekcije sa serverom. `onInit()` metoda poziva metodu `connect()` koja prikazana na listingu 3. i ona sadrži logiku za uspostavljanje websocket veze. Metoda `onDestroy()` se poziva prilikom uništavanja komponente što smo iskoristili da zatvorimo WebSocket konekciju.

```
connect() {
    let socket = new
SockJS('http://localhost:8090/JiraKanban/kanbanWs');
    var stompClient = Stomp.over(socket);
    this._stompClient = stompClient;
    let configId = this._configurationId;
    let mapdata = this.mapIncomingDataContext;
    let context = this;
    stompClient.connect({}, function(frame) {
        console.log('Connected: ' + frame);
        var url = '/topic/greetings/' + configId;
        stompClient.subscribe(url, function(data){
            console.log("new data arrived!");
            mapdata(JSON.parse(data.body), context);
        });
    });
}
```

Listing 3. Uspostavljanje WebSocket konekcije

4. ZAKLJUČAK

Zahvaljujući velikom broju tehnologija i alata koji se danas nalaze na tržištu ispunjavanje kompleksnih korisničkih zahtjeva je dosta olakšano. Upravo zbog velikog broja tehnologija koje su danas dostupne treba biti jako oprezan prikom izbora tehnologija za razvoj programskog rešenja. U slučaju pogrešnog izbora obično se traže alternative a sam prelazak sa jedne tehnologije na drugu ponekad može biti prilično komplikovano.

Za implementaciju ovog rada izabrane su tehnologije koje su prilično rasprostranjene i koje bi trebalo da su u mogućnosti da pokriju i najkomplikovanije zahtjeve korisnika. Za razvoj klijentske aplikacije u ovom radi je korišten Angular 2 *framework* koji je u trenutku razvoja Kanban table bio u alfa verziji. Programerska zajednica ima velika očekivanja od Angular 2 *framework*-a i po svemu sudeći trebao bi biti veliki konkurent trenutno aktivnim JavaScript alatima za razvoj klijentskih aplikacija kao što je ReactJS.

Dalji razvoj sistema obuhvata optimizaciju obrade podataka kako na serverskoj tako i na klijentskoj strani da bi se mogao podržati veći broj Kanban konfiguracija. Što se tiče tehnologija klijentska aplikacija bi se trebala prebaciti na zvaničnu verziju Angular 2 *framework*-a.

5. LITERATURA

- [1] Henry H. Liu, Spring 4 for Developing Enterprise Applications: An End-to-End Approach, CreateSpace Independent Publishing Platform, 2012, ISBN 978-1480284708
- [2] Clarence Ho; Rob Harrop; Chris Schaefer, Pro Spring, 4th edition, Apress, 2014, ISBN 978-1430261513
- [3] Kyle Banker; Peter Bakum; Shaun Verch; Doug Garrett; Tim Hawkins, MongoDB in Action, 2nd edition, Manning Publications, 2016, ISBN 978-1617291609
- [4] Jira, <https://www.atlassian.com/software/jira>
- [5] Angular 2, <https://angular.io/>
- [6] Spring, <https://spring.io/>
- [7] Raoul-Gabriel Urma; Mario Fusco; Alan Mycroft, Java 8 in Action, 1st edition, Manning Publications, 2014, ISBN 978-1617291999

Kratka biografija:

Sanjin Malešević, rođen 24.09.1990. godine u Derventi, Republika Srpska (BiH). Osnovnu školu „19 April“ završio je 2005. godine nakon čega je u okviru Gimnazije sa tehničkim školama u Derventi 2009. godine stekao zvanje tehničar računarstva. Iste godine upisao se na Fakultet Tehničkih Nauka, odsek Računarstvo i automatika. Školske 2012/2013. godine upisao se na smer Računarske nauke i informatika. Položio je sve ispite predviđene planom i programom.

Milan Vidaković je rođen u Novom Sadu 1971. godine. Na Fakultetu tehničkih nauka u Novom Sadu završio je doktorske studije 2003. godine. Na istom fakultetu je 2014. godine izabran za redovnog profesora iz oblasti *Primenjene računarske nauke i informatika*.

POREĐENJE PERFORMANSI REGULATORA KORIŠĆENIH ZA UPRAVLJANJE MEŠNIM VENTILIMA PRI SIMULACIJI RADA TOPLOTNE PODSTANICE

PERFORMANCE COPMARISON OF DIFFERENT REGULATORS USED FOR MIXING VALVES CONTROL IN SIMULATION OF THE HEATING SUBSTATION

Miloš Đukić, *Fakultet tehničkih nauka, Novi Sad*

ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je simuliran rad toplotne podstanice u Matlab-u, obezbeđeno je upravljanje mešnim ventilima pomoću PI, PID i PI-like fuzzy regulatora, a na kraju je sistem implementiran u programskom paketu Labview.

Abstract – The paper simulated operation of heating substations in Matlab, Mixing valves control was provided by PI, PID, PI-like fuzzy controller. The system was implemented in Labview software package.

Ključne reči: Toplotna podstanica, razmenjivač toplote, mešni ventil, PI, PID i PI-like fuzzy regulator, Matlab, Labview

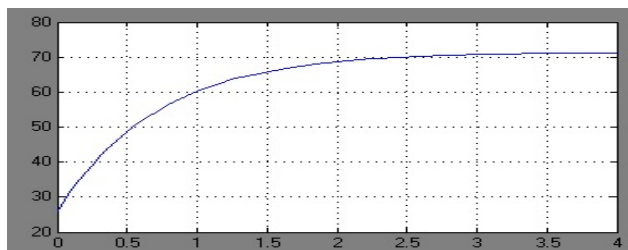
1. UVOD

U radu su korišćena dva softverska paketa (Matlab i Labview) za simulaciju modela, kao i za prikaz dobijenih rezultata. U Matlab-u je model raden u simulinku, dok se u Labview softverskom paketu, pored modela može videti i izgled Scade. Za početak je bilo potrebno odrediti i prikazati temperature na izlazu primara i sekundara, ako su nam poznate vrednosti temperatura na ulazu istih.

2. STUDIJA SLUČAJA

Za izračunavanje temperature na izlazu sekundara (T_2'') iskoristili smo opšte poznatu formulu prenosa toplote. Temperatura u narednom trenutku bila bi jednaka temperaturi u prethodnom trenutku plus proizvod razlike temperatura na ulazu primara i sekundara, i koeficijenta prenosa toplote.

$$T_2'' = T_2'' + (T_1' - T_1'') * k$$



Slika 1. Prikaz izlazne temperature na sekundaru

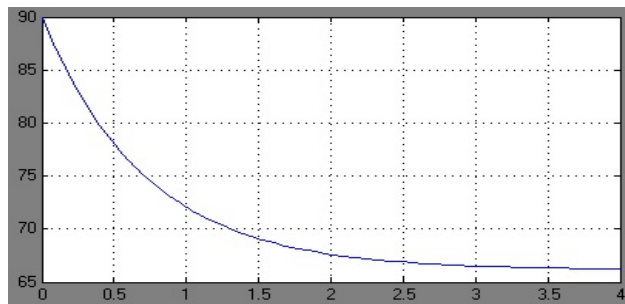
Ova formula bila bi važeća u idealnom slučaju da nismo imali gubitke energije. Uzevši u obzir gubitke energije, za poznate vrednosti temperatura na ulazu primara i sekundara (T_1' , T_1''), kao i koeficijenta prenosa toplote (k) i temperature spoljašnjosti (T_{spolj}) dobili smo temperaturu na izlazu iz sekundara (T_2'') i prikazali je na slici 1.

Na grafiku se može videti da za poznate vrednosti $T_1' = 90^\circ\text{C}$, $T_1'' = 26^\circ\text{C}$, $T_{spolj} = 25^\circ\text{C}$, temperatura na izlazu sekundara (T_2'') se posle određenog vremena ustali na 72°C , što je maksimum našeg izmenjivača toplote.

Dalje, u radu smo, za potrebe izračunavanja temperature na izlazu iz primara, iskoristili opšte poznatu formulu održavanja energije. Ukupna potrošena količina energije na primaru jednaka je ukupnoj potrošenoj količini energije na sekundaru, odnosno, proizvod ukupnog protoka i razlike temperatura na primaru jednak je proizvodu ukupnog protoka i razlike temperatura na sekundaru.

$$m_1 * (T_1' - T_2') = m_2 * (T_2'' - T_1'')$$

Dakle, za poznate vrednosti temperatura na ulazu primara i sekundara i dobijene temperature na izlazu sekundara, kao i za poznate vrednosti protoka, izračunali smo temperaturu na izlazu primara i prikazali je na grafiku:



Slika 2. Prikaz izlazne temperature na primaru

Na grafiku se može videti da za poznate vrednosti $T_1' = 90^\circ\text{C}$, $T_1'' = 26^\circ\text{C}$, $T_{spolj} = 25^\circ\text{C}$, kao i poznate vrednosti protoka na primaru $m_1 = 42 \text{ kg/s}$, i ukupnog protoka na sekundaru $m_2 = 22 \text{ kg/s}$ ($4 + 8 + 10$, zbir protoka za sva tri tipa grejanja), dobijamo temperaturu na izlazu iz primara koja se posle određenog vremena ustali na $T_2' = 66,5^\circ\text{C}$.

U nastavku smo ceo ovaj Matlab model implementirali u LabView-u.

Na Scadi se može videti celokupan sistem, sa svim temperaturama na primaru i sekundaru, ukupna potrošnja

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Velimir Čongradac.

energije u sistemu, kao i otvorenost mešnih ventila za svaki pojedinačni tip grejanja (radijatorsko, fancoilno i podno). Upravljanje mešnim ventilima obezbeđeno je pomoću PI, PID i PI-like fuzzy regulatora o čemu će biti reči u nastavku rada.

2.1 Upravljanje mešnim ventilima pomoću PI, PID i PI-like fuzzy regulatora

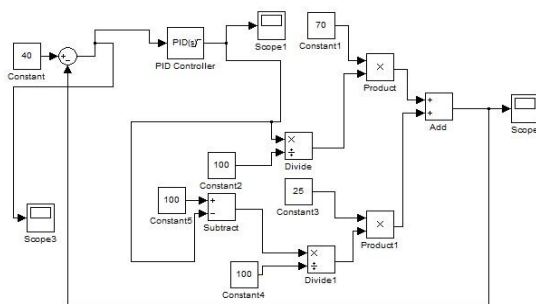
Postoje tri mešna ventila u sistemu, svaki za pojedinačni tip grejanja (radijatorsko, ventilokonvektorsko, podno), gde sa jedne strane imamo temperaturu vode sa izlaza sekundara, a sa druge strane imamo temperaturu vode na povratu iz zgrade. Regulacija se vrši pomoću PI, PID i PI-like fuzzy regulatora. Obezbeđen je odabir regulatora sa Scade.

Željena temperatura za svaki pojedinačni tip grejanja se dobija na osnovu sledećeg zakona:

$$\text{Željena temperatura} = T2'' * \text{OtvorenostVentila}/100 + T1'' * (100 - \text{OtvorenostVentila})/100$$

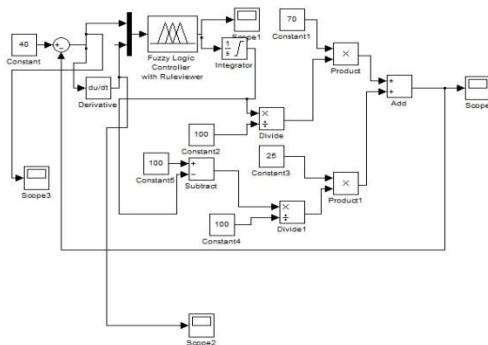
Za PI, kao i za PID regulator, iskoristili smo već postojeće funkcije Matlab-a i Labview-a.

Bilo je samo potrebno odrediti parametre. Ovo je izgled Matlab modela:



Slika 3. Izgled Matlab modela (PI, PID)

PI-like fuzzy regulator funkcioniše na MISO principu (multiple input single output) gde ulazne parametre čine Greška i Izvod greške, a izlazni parametar je du/dt . Za početak smo ovaj model simulirali u Matlab-u što se može videti na sledećoj slici:

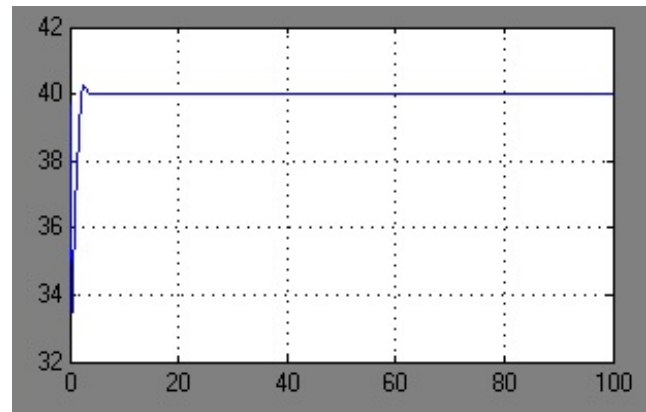


Slika 4. Izgled Matlab modela (PI-like fuzzy)

U ovom simulink modelu uzeli smo konstantne vrednosti za ulaznu i izlaznu temperaturu na sekundaru (25°C i 70°C), kao i za željenu vrednost (40°C).

2.2 Problemi pri rešavanju

Kod PI i PID regulatora, osnovni i jedini problem bio je određivanje parametara. Koristili smo ručnu metodu. Podešavali smo proporcijalno, diferencijalno i integralno dejstvo, respektivno i posmatrao ponašanje sistema. Prikaz rezultata za PI i PID regulator iz Matlab modela se može videti na sledećim slikama:

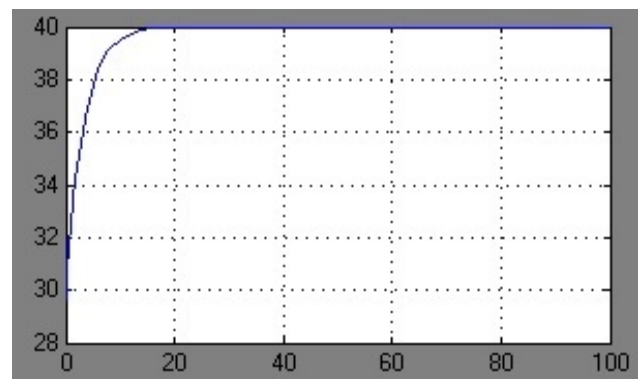


Slika 5. Prikaz rezultata PI regulatora

Vreme uspona, $Tr = 0.9$ s

Greška u ustaljenom stanju, $Ess = 0$

Minimalan preskok



Slika 6. Prikaz rezultata PID regulatora

Vreme uspona, $Tr = 3$ s

Greška u ustaljenom stanju, $Ess = 0$

Nema preskoka

Prikazi dobijenih rezultata u Labview-u, za različite parametre PI i PID regulatora će se videti u nastavku rada.

Kod PI-like fuzzy regulatora prvi problem na koji smo naišli jeste taj da smo pokušali da napravimo regulator samo sa jednim ulazom (greškom), bez izvoda greške, što nije davalo dobre rezultate.

Kada smo uveli izvod greške kao drugi ulaz u fuzzy regulator, naišli smo na druge probleme. Pogrešno smo zadavali granične vrednosti funkcija pripadnosti. Uzimali smo samo pozitivne vrednosti, što nam je na izlazu regulatora davalo konstantnu vrednost. Pokušavali smo da menjamo granice, ali smo uvek dobijali isto, dok nismo proširili granice i na levu poluravan.

Uzimali smo i negativne i pozitivne granične vrednosti za oba ulaza i za izlaz regulatora, što je i logično.

Za početak smo uzeli po tri funkcije pripadnosti i za izlaz i za oba ulaza i to trougaonog oblika:

Za ulaz Greške (negativna, pozitivna i dobra),
Za ulaz Izvoda greške (negativan, pozitivan i oko_nule),
Za izlaz (otvori, zatvori i održavaj).

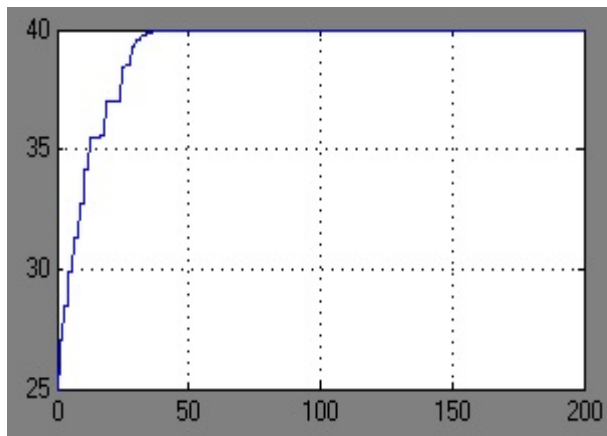
A zatim smo uveli 9 pravila. U slučajevima kada bi greška bila negativna ili pozitivna, bez obzira kakav je izvod greške, ventil bi se zatvarao ili otvarao, a samo u slučaju dobre greške, ventil bi zadržavao svoje stanje. Primetili smo da bi bilo dobro imati međunivo kao funkcije pripadnosti, u vidu brzog i sporog otvaranja i zatvaranja ventila, pa smo uveli 5 funkcija pripadnosti za izlaz i to: brzo_zatvori, sporo_zatvori, održavaj, sporo_otvori, i brzo otvori.

U slučajevima kada je greška pozitivna, a izvod greške pozitivan ili oko nule, i u slučajevima kada je greška negativna, a izvod greške negativan ili oko nule, koristili smo isključivo brzo_otvori i brzo_zatvori kao izlaze, respektivno.

Dok u slučaju kada je greška pozitivna, a izvod negativan, sa tendencijom pada, i u slučaju kada je greška negativna, a izvod pozitivan, sa tendencijom rasta, koristili smo sporo_otvori i sporo_zatvori izlaze, respektivno, što se ispostavilo kao dobra odluka.

Ipak, još uvek nismo dobijali potpuno dobre rezultate. Imali smo malu grešku u ustaljenom stanju. Onda smo menjali opsege funkcija pripadnosti dok nismo dobili konačne zadovoljavajuće rezultate.

Za ovako podešene parametre PI-like fuzzy regulatora, i za željenu vrednost temperature (40°C) dobijamo sledeći rezultat na izlazu:



Slika 7. Prikaz rezultata PI-like fuzzy regulatora

Vreme uspona, $T_r = 10$ s

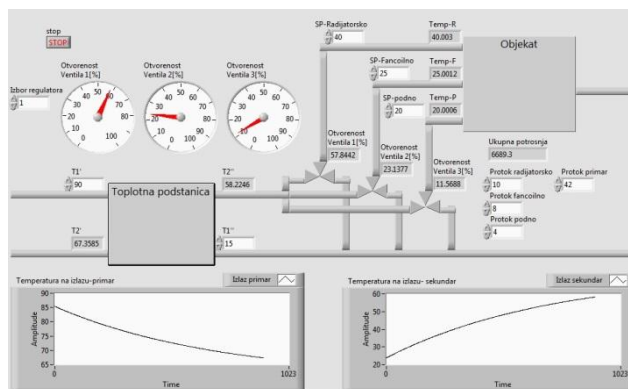
Greška u ustaljenom stanju, $E_{ss} = 0$

Nema preskoka.

Kasnije je model prebačen u Labview, i kada je ustanovljeno željeno ponašanje modela, onda je PI-like fuzzy regulator integrisan u celokupni sistem.

Na kraju, za poznate vrednosti temperatura na ulazu primara i sekundara ($T_{I'}=90^{\circ}\text{C}$, $T_{I''}=15^{\circ}\text{C}$), poznate vrednosti masenih protoka na primaru i sekundaru

($m_1=42\text{kg/s}$, $m_2=10$ (radijatorsko) + 8 (ventilokonvektorsko) + 4 (podno) = 22kg/s), za zadate vrednosti željenih temperatura za svaki pojedinačni tip grejanja ($T_r=40^{\circ}\text{C}$, $T_f=25^{\circ}\text{C}$ i $T_p=20^{\circ}\text{C}$) dobijamo sledeći prikaz na Scadi:



Slika 8. Prikaz Scada ekrana

2.3 Dobijeni rezultati i analiza

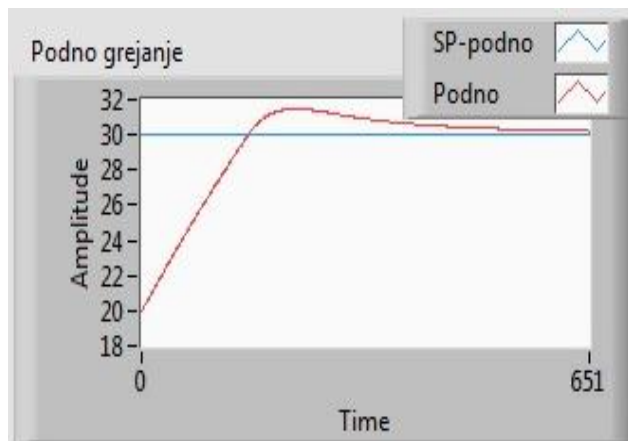
Za početak je odabran PI regulator. Sistem je pušten u rad, korišćeni su različiti parametri za svaki pojedinačni tip grejanja:

Za radijatorsko grejanje ($k_p=1$, $k_i=0.01$),

Za ventilokonvektorsko ($k_p=1$, $k_i=0.02$),

Za podno ($k_p=1$, $k_i=0.05$).

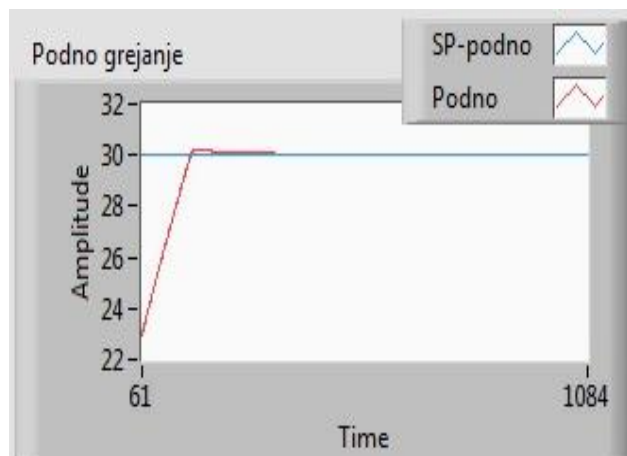
Na sledećim graficima će se videti prikaz rezultata svih regulatora za podno grejanje:



Slika 9. Prikaz rezultata PI regulatora za podno grejanje

Kod podnog grejanja, preskok je znatno uočljiv. Vreme uspona je veliko, jer je u pitanju spor proces, a i regulatori su pušteni u rad simultano s početkom rada sistema, pa nije bilo moguće dostići željenu vrednost temperature dok temperatura na izlazu sekundara nije dostigla određenu vrednost. Iz ovih razloga je bilo dobro izvršiti simulaciju u Matlab-u za poznate vrednosti temperatura na primaru i sekundaru.

Za PID regulator su korišćeni sledeći parametri:
 Za radijatorsko grejanje ($k_p=0.2$, $k_i=0.1$, $k_d=0.01$),
 Za ventilokonvektorsko ($k_p=0.5$, $k_i=0.1$, $k_d=0.01$),
 Za podno ($k_p=0.1$, $k_i=0.05$, $k_d=0.01$)

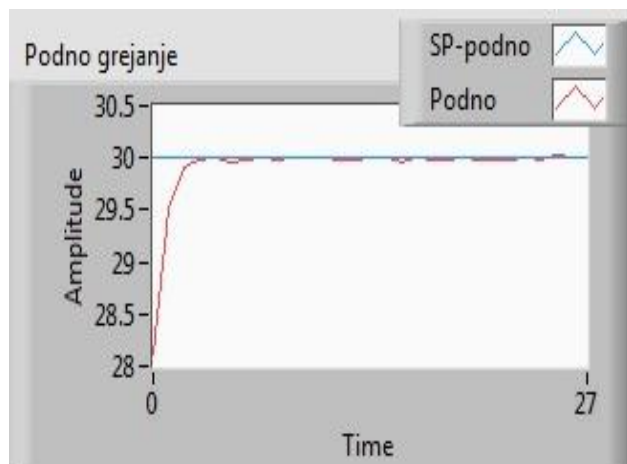


Slika 10. Prikaz rezultata PID regulatora za podno grejanje

Sa dobro podešenim parametrima PID-a preskoka nema, kao ni greške u ustaljenom stanju, a brzina odziva, odnosno vreme uspona je zadovoljavajuće.

Za PI-like fuzzy regulaciju, iskorišćen je potpuno isti model kao i u Matlab-u (iste funkcije pripadnosti, i ista pravila).

Dobijen je sledeći rezultat:



Slika 11. Prikaz rezultata PI-like fuzzy regulatora za podno grejanje

Iz priloženih rezultata može se uočiti da upravljanje pomoću PI-like fuzzy regulatora daje odlične rezultate. Nema greške u ustaljenom stanju, nema preskoka, a i vreme uspona je malo (regulator pušten u rad, tek nakon dostizanja maksimalne temperature na izlazu sekundara).

3. ZAKLJUČAK

Na osnovu dobijenih rezultata moglo se utvrditi sledeće:

-PI regulatorom se eliminiše greška u ustaljenom stanju, gotovo uvek postoji preskok, ali je pogodan za procese koji ne zahtevaju brzo reagovanje, baš kao što je ovaj temperaturni proces.

-Dobro podešenim parametrima PID-a može se eliminisati preskok i dobiti optimalno ponašanje sistema.

-Dobro definisanim pravilima i dobro definisanim funkcijama pripadnosti, PI-like fuzzy regulator može davati odlične rezultate. Nema preskoka, nema greške u ustaljenom stanju, a vreme uspona i smirenja je veoma zadovoljavajuće.

Iako su PID regulatori i dalje neprevaziđeni i danas se koriste u većini industrijskih procesa, česta inženjerska praksa je da se za sisteme prvog reda kao što je ovaj koriste PI regulatori. Fuzzy regulator se pokazao kao idealan izbor u procesima sa velikim brojem parametara.

Temperaturni procesi, odnosno, sistemi grejanja su spori procesi i ne zahtevaju brzo reagovanje, tako da se za potrebe ovog sistema može iskoristiti bilo koji od ova tri regulatora.

4. LITERATURA

- [1] Yung C. Shin, Chengying Xu —Intelligent Systems:Modeling, Optimization, and Control CRC Press 2009.
- [2] Fried land B., —Advanced Control System Design,Prentice Hall, New Jersey, 1996.
- [3] Petar Kalušić 1991. Mehanika i toplota, Školska knjiga Zagreb;
- [4] Advanced Heat and Mass Transfer ,Amir Faghri, Yuwen Zhang, John Howell, 2010
- [5] Dimiter Driankov, Hans Hellendoorn, Michael Reinfrank, Fuzzy control

Kratka biografija:



Miloš Đukić je rođen 16.11.1992. godine u Vrbasu. Osnovnu školu „Petar Petrović Njegoš" je završio u Vrbasu 2007. godine, a gimnaziju „Žarko Zrenjanin" - prirodno matematički smer 2011.godine. Iste godine je upisao Fakultet Tehničkih Nauka u Novom Sadu, odsek elektrotehnika i računarstvo, smer računarstvo i automatika, usmerenje automatika i upravljanje sistemima i završio osnovne akademske studije 2015.godine.

SIMULACIJA SLANJA DOGAĐAJA SA PAMETNIH BROJILA
SIMULATION OF SENDING EVENTS FROM SMART METERSBranislav Potić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Pametna mreža poboljšava funkcionalnosti tradicionalne električne mreže, omogućujući dodatne funkcije. Napredna merna infrastruktura se najčešće koristi u implementaciji pametne mreže. Pametna mreža obuhvata i pametna brojila koja šalju vrednosti sa polja, ali mogu primati komande od strane elektrodistribucije. Jedna od vrednosti koje stižu sa brojila su i događaji, koji se dešavaju prilikom nekih nepravilnosti na delu mreže. U ovom radu je prikazan servis koji upravlja događajima, ali glavni akcenat rada je na aplikaciji koja generiše i simulira slanje događaja sa polja.

Abstract – Smart grid improves functionalities of traditional electric grid by enabling additional functions. Advanced metering infrastructure is mostly used in implementation of smart grid. Smart grid includes smart meters which send values from field, but can also receive commands from power company's system. One of the values coming from smart meters are events, which happened during abnormality in some part of grid. This document is about service that handles events, but the main focus is on application which generates events and simulates event sending from field.

Ključne reči: Pametna mreža, napredna merna infrastruktura, pametna brojila, događaji

1. UVOD

Pred električnu mrežu postavljeni su veliki zahtevi krajem 20. i početkom 21. veka. Sve je više rastao broj korisnika usluga koje pružaju elektrodistribucije i zahtevi korisnika za električnom energijom. U tradicionalnim elektromrežama, manjak nadzora i upravljanja, bili su razlozi koji su nedostajali kako bi mreža održala korak sa vremenom. Pametna mreža donosi nove funkcionalnosti koje se nadograđuju na stare i tako omogućuju veću upravljivost, smanjenje troškova, kao i veću efikasnost sistema.

Zadatak ovog rada jeste opisivanje glavnih principa, komponenata, kao i funkcionalnosti električne mreže. U radu su opisane i pogodnosti koje pametna mreža donosi elektrodistribuciji i potrošačima.

Objašnjen je i način funkcionisanja AMI (*Advanced Metering Infrastructure*) infrastrukture, a opisana su i pametna brojila kao najvažnije komponente ove infrastrukture.

U okviru rada opisani su i načini komunikacije između komponenata u sistemu. Takođe, zadatak ovog rada jeste

da predstavi događaje koji se dešavaju u okviru mreže, da prikaže funkcionisanje servisa za prihvatanje istih, a glavni deo rada predstavlja primer implementacije aplikacije za generisanje i slanje događaja kroz simulaciju.

2. PAMETNA MREŽA

Električna mreža predstavlja skup strujnih generatora, dalekovoda, transformatora i relejnih sistema i ima za cilj snabdevanje potrošača električnom energijom. Električna mreža sastoji se iz 4 oblasti: proizvodnja, prenos, distribucija i potrošnja. Prve električne mreže datiraju još sa kraja 19. veka i ciljevi tadašnjih mreža bili su isporuka električne energije potrošačima i naplata potrošene električne energije na određeni vremenski period. Generisanje električne energije vršilo se u blizini velikih gradova, gde je i bio skoncentrisan najveći broj potrošača, a sve u cilju smanjenja gubitaka prilikom prenosa. Kako je rastao broj potrošača i zahtevi za električnom energijom, jednosmerna distribucija energije teško se nosila sa zahtevima za električnom energijom. Prema definiciji sekretarijata za energiju SAD, pametna mreža je električna mreža koja koristi informacione i komunikacione tehnologije da prikupi i automatski deluje na osnovu informacija o proizvodnji i potrošnji, u cilju unapređivanja efikasnosti, pouzdanosti ekonomičnosti i održivosti proizvodnje i distribucije električne energije [1]. Prednosti koje pametna mreža donosi elektrodistribuciji su: veća pouzdanost mreže, bolja efikasnost sistema i smanjenje troškova. Potrošači imaju veću kontrolu nad potrošnjom električne energije, koju je moguće pratiti, a time i smanjiti potrošnju energije. Potrošači imaju mogućnost da koriste energiju kada joj je cena najniža i time smanje svoje račune. Pored velikih distributera energije, postoje i manji izvori električne energije koji se nazivaju distribuirani električni resursi (DER). Pametna mreža daje mogućnost potrošačima koji proizvode sopstvenu energiju da prodaju višak električne energije elektrodistribuciji. Sve je veći broj manjih distribuiranih energetske resursa koji se integrišu u sistem pametne mreže [2].

3. NAPREDNA MERNA INFRASTRUKTURA

Glavni zadatak elektrodistribucija pored distribucije električne energije, jeste i naplata potrošene električne energije potrošačima. Najpre je elektrodistribucija slala svoje posade, kako bi očitali vrednosti potrošene energije sa svakog brojila. To je zahtevalo mnogo ljudskih kao i materijalnih resursa. Kasnije je upravo iz tih razloga električna mreža izmenjena, tako da nisu bili potrebni izlasci članova posade na teren, već su električna brojila integrisana u sistem, u kojem su se sa brojila mogle očitavati vrednosti potrošene električne energije. U okviru

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Srđan Vukmirović, red.prof.

ovakvog sistema bila je moguća samo jednosmerna komunikacija, gde se sa brojila potrošača, automatski očitavaju vrednosti i šalju se do komponente za procesiranje podataka, odakle se šalju do elektrodistribucije, tačnije do sistema za naplatu i nadzor mreže. Napredna merna infrastruktura (AMI) uvodi koncept bidirekcionane komunikacije što podrazumeva da se podaci i informacije šalju u oba smera. U okviru takve infrastrukture, osim što se očitavanja potrošnje automatski šalju do distributera električne energije, informacije se šalju i od distribucije ka potrošačima. Takođe, elektrodistribucija ima mogućnost da šalje komande ka pametnim brojlama i time efikasnije upravlja sistemom. Pored prikupljanja sirovih podataka, AMI vrši analizu dobijenih podataka. Najčešća arhitektura AMI sistema je takva da se podaci očitavaju iz jedne grupe pametnih brojila u lokalne koncentratore podataka, a onda se preko *backhaul* kanala distribuiraju do centralne komande. Tamo se podaci skladište i procesiraju, vrše se upravljanja i naplate potrošene električne energije [3]. AMI infrastruktura je primenjiva na različite sisteme: električne, sisteme za snabdevanje gasom, vodom i druge. Jednu od najvažnijih komponenata napredne merne infrastrukture, pored same komunikacije, predstavljaju pametna brojila.

4. PAMETNA BROJILA

Pametna brojila su uređaji koji sakupljaju podatke i merenja u određenim vremenskim periodima. Podaci i merenja koji se sakupljaju sa pametnih brojila su vremenski označeni. Kao što je već rečeno, pametna brojila komuniciraju sa udaljenim centrima za prikupljanje podataka i šalju podatke elektrodistribuciji. Obzirom na to da je komunikacija u AMI infrastrukturi bidirekciona, pametni uređaji, osim slanja podataka i merenja, mogu da primaju informacije kao i komandne signale i da deluju u skladu sa njima. In-Home Displays (IHD) predstavljaju podatke koje dolaze sa merila potrošačima i tako im stavljaju do znanja kolika je njihova potrošnja i omogućuju im da spram toga lakše upravljaju potrošnjom. Ukoliko se desi neki ispad u mreži, pametno brojilo i AMI sistem obezbeđuju trenutno informisanje servisa na strani elektrodistribucije, omogućujući brzo lociranje ispada i brzo delovanje. Osnovne karakteristike pametnih brojila su sledeće: vremenski bazirano određivanje cene, uvid u potrošnju, informacije o kvarovima i ispadima, primanje komandnih signala od strane sistema za upravljanje, nadzor kvaliteta ispostave električne energije, otkrivanje krađe električne energije, komunikacija i integracija sa drugim pametnim uređajima, kao i smanjivanje štetne emisije i zagađivanja životne sredine [4].

5. DOGAĐAJI U OKVIRU PAMETNE MREŽE

Pametna brojila šalju vrednosti potrošene električne energije ka elektrodistribuciji i tako omogućuju nadzor i elemente za aktivno upravljanje nad mrežom. Osim tih podataka, postoji još jedna vrsta podataka koja se automatski šalje sa pametnih brojila. To su događaji (*Events*). Oni se generišu u slučaju da se desi neka neželjena promena u okviru mreže ili dela mreže (npr. pad napona ispod neke vrednosti, skok napona iznad neke vrednosti, nestanak dotoka električne energije), ili u

slučaju da se neka od tih vrednosti vrati u normalno stanje nakon nekog poremećaja u mreži. U okviru DMS (*Distribution Management System*) sistema postoji 6 tipova događaja [5]:

- **Power down** – predstavlja nestanak napajanja električnom energijom na delu električne mreže
- **Power up** – predstavlja povratak napajanja električnom energijom na delu mreže
- **Undervoltage** – predstavlja pad vrednosti napona na delu mreže ispod određene granice
- **Undervoltage return to normal** – predstavlja povratak vrednosti napona na delu mreže iznad određene donje granice
- **Overvoltage** – predstavlja povećanje vrednosti napona na delu mreže iznad određene granice
- **Overvoltage return to normal** – predstavlja povratak vrednosti napona na delu mreže ispod određene gornje granice

Svaki od tipova događaja ima svoj suprotni događaj (npr. događaj *Undervoltage*, koji predstavlja pad napona na nekom delu mreže, ima svoj suprotan tip događaja, *Undervoltage return to normal* koji predstavlja povratak vrednosti napona u normalu, tj. iznad postavljene donje granice). Česti su slučajevi da do elektrodistribucije stigne jedan događaj (npr. *Undervoltage*), a nedugo posle njega stigne njegov suprotan događaj (*Undervoltage return to normal*), tako da je nakon toga stanje u tom delu mreže normalno, stoga elektrodistribucija ne mora da preduzima nikakve mere.

5.1. Smart Meter Management Service

Implementacija simulatora događaja koja je glavni zadatak ovog rada, oslanja se na servis DMS sistema. Servis koji upravlja događajima sa polja naziva se SMMS servis (*Smart Meter Management Service*). Servis je implementiran kao WCF (*Windows Communication Foundation*) servis i izlaže metodu za primanje liste događaja sa polja. U okviru servisa, postoji profil (XML fajl), iz kojeg se pri podizanju servisa učitavaju vrednosti parametra. Takođe na servisu su izložene i metode za dobavljanje i promenu vrednosti parametara profila. Servis čuva podatke u internim strukturama. To su *Element cache* i *Item cache*. Prilikom primanja događaja na servisu, vrše se validacije (*Message*, *Age*, *Profile*, *FaultedSM*). Ukoliko događaj ne prođe validacije, ne prolazi u dalju obradu. U okviru *Message* validacije proverava se da li događaj poseduje sve obavezne atribute. *Age* validacijom se odbacuju događaji koji su zastareli, tj. kod kojih je razlika između trenutnog vremena i vremenske oznake događaja veća od vrednosti parametra konfiguracije (*SMMS Config Max Age*). U okviru profilne validacije postavljaju se vrednosti atributa događaja *Stewed* i *Retained*. *FaultedSM* validacijom se proverava da li se brojilo sa kojeg je stigao događaj nalazi u listi „neispravnih“ brojila, te se događaji sa tog brojila odbacuju. Ispravan pristigli događaj se upisuje u jednu od dve strukture (*Element* ili *Item cache*). U okviru *Element cache*-a nalaze se dve tabele: *Stewing* i *Retention*. Atributi *Stewed* i *Retained* postavljaju se na osnovu vrednosti parametara iz profila, *StewTime* i *Retained*. *StewTime* je vreme tokom kojeg, ukoliko na servis stigne događaj sa istog brojila, suprotan događaju koji se nalazi u *Stewing*

tabeli, postojeći događaj se briše iz tabele, dok se novi događaj odbacuje. Događaji koji završe u *Retention* tabeli, su „zadržani“ i ne propagiraju se na dalju obradu. Na servisu postoji nit zadužena za proveravanje događaja koji se nalaze u *Stewing* tabeli i vršenje određenih akcija nad njima u zavisnosti od parametara događaja i konfiguracije. Nit se pobuđuje na svakih 20 sekundi.

6. SIMULATOR DOGAĐAJA

Aplikacija testira izložene metode servisa, odnosno omogućuje ručnu simulaciju slanja događaja na servis, kao i automatsku simulaciju (automatsko slanje) događaja sa polja i dodaje još neke funkcionalnosti poput promene konfiguracije profila u vremenu izvršavanja, čuvanje događaja u bazi podataka, pregled svih događaja koji se čuvaju na servisu, kao i pregled i filtriranje događaja koji se čuvaju u bazi podataka. Aplikacija je implementirana u C# programskom jeziku, koristeći WPF (*Windows Presentation Foundation*) tehnologiju. Kao razvojni framework, korišćen je .NET framework 4.5.1. Aplikacija je napisana po MVVM (*Model View View Model*) šablonu. Korišćen je i *Command* šablon za pozivanje funkcija dostupnih putem korisničkog interfejsa. Za arhiviranje događaja poslatih ka *SMMS* servisu, koristi se *SQL Server* baza podataka.

6.1. Komponente aplikacije

Aplikacija se sastoji iz tri prozora, glavnog i dva pomoćna. Na glavnom prozoru se nalaze sve funkcionalnosti, dok jedan pomoćni prozor služi za prikaz vrednosti atributa profila, a drugi prozor za pregled svih arhiviranih događaja pristiglih sa polja. *MainWindow* predstavlja glavni prozor aplikacije i sastoji se iz 6 celina. Svaka od ovih 6 celina je implementirana kao korisnička kontrola (*User Control*) i sve zajedno čine korisnički interfejs glavnog prozora aplikacije. Svaka od kontrola ima određeni kontekst, tj. svakoj korisničkoj kontroli odgovara jedan *ViewModel*. Od 6 celina, 4 celine imaju sopstveni kontekst. Dodatni prozor za pregledanje događaja takođe poseduje sopstveni *ViewModel*. Dodatni prozor za prikaz parametara nema sopstveni *ViewModel*, već koristi *ViewModel* dela zaduženog za konfiguraciju. U klasama koje predstavljaju *ViewModel*-e, nalaze se metode koje se koriste u odgovarajućim modulima glavnog prozora. *Stewing* i *Retention* deo služe da predstave događaje sa servisa u dve različite tabele. Osim korisničkih kontrola i odgovarajućih *View Model*-a, u okviru rešenja nalaze se i klase za dodavanje događaja u bazu podataka, kao i pomoćne klase za popunjavanje atributa događaja ili za slanje GDA (*Generic Data Access*) upita. Sve akcije koje korisnik može da preduzme dostupne su preko komponenti (dugmadi) na korisničkom interfejsu.

6.2. Funkcionalnosti za ručnu simulaciju događaja

U okviru modula *Sending random events*, korisnik je u mogućnosti da simulira slanje slučajnih događaja sa polja. Slučajan događaj podrazumeva slučajno odabrano brojlilo sa kojeg se simulira događaj, kao i slučajno odabran tip događaja. Tom prilikom je potrebno da korisnik unese broj događaja koji želi da simulira (pošalje sa polja), kao i broj događaja koji će se generisati i poslati ka servisu, ali neće proći neku od 4 validacije. Odabir validacije koju određeni broj događaja neće zadovoljiti, takođe bira sam

korisnik. Taj broj događaja biće tako generisan da ne zadovolji neki od zahteva koje postavlja odabrana validacija. Osim slanja događaja na servisnu stranu, ti događaji se skladište u bazi podataka. Događaji se čuvaju u jednoj tabeli instance *SQL* servera. Ta baza podataka može se smatrati jednom vrstom arhive. Na slici 1 nalazi se *Sending random events* modul.

Sending random events:

Number of events:

Number of events to fail:

☐ Age Validation

☐ Message Validation

☐ Profile Validation

☐ FaultedSM Validation

Slika 1. *Sending random events* modul

U okviru korisničke kontrole *Sending random events*, nalazi se još i dugme *Archive*. Pritiskom na ovo dugme otvara se novi prozor, u kome je moguće pregledati i filtrirati događaje koji su zabeleženi u bazi podataka (arhivi). Događaji se mogu filtrirati po jednom od 6 tipova događaja, kao i po datumu nastanka događaja. Korisnik bira vremenski raspon (u danima) tako što bira početni i krajnji datum. U okviru modula *Sending events on the same meter* moguće je slati događaje sa istih brojlila koja već postoje u okviru nekog događaja u *Stewing* tabeli. Opcijom *Send opposite events* šalje se suprotan tip događaja na istom brojlilu za slučajno odabran događaj iz *Stewing* tabele. Korisnik je obavezan da unese broj događaja koje želi da pošalje. Sledeća opcija ovog modula je *Send same events*. Ova funkcionalnost obavlja isto što i prethodna, jedina razlika je u tome što se kao tip događaja za slanje postavlja isti tip kao slučajno odabran događaj iz *Stewing* tabele. Na slici 2 prikazan je modul *Sending events on the same meter*.

Sending events on the same meter:

Number of opposite:

Number of same:

Number of events:

Retained property:

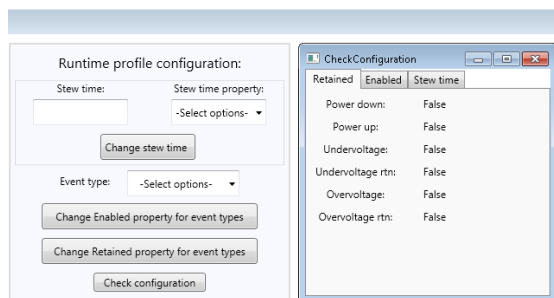
Slika 2. *Sending events on the same meter*

Poslednja opcija koja je na raspolaganju korisniku je *Send to retained*. Ova komanda uzima slučajno odabran događaj iz *Retained* tabele i dodaje suprotan tip događaja nad istim brojlilom u *Stewing* tabelu. Nakon toga, generiše

i šalje događaj istog tipa kao prethodni dodat u *Stewing* tabelu.

6.3. Profilna konfiguracija

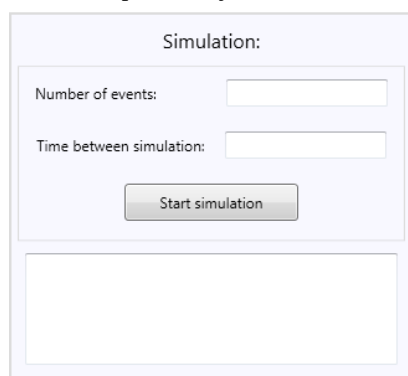
Da bi se mogli simulirati različiti scenariji koji se dešavaju na servisu, potrebno je da korisnik bude u mogućnosti da menja parametre *StewTime*, *Retained*, *Enabled*. To je omogućeno u okviru modula *Runtime profile configuration*. Početne vrednosti parametara se učitavaju iz konfiguracionog fajla. U ovom fajlu moguće je definisati više instanci profila. Takođe, u okviru ovog fajla definiše se koji tipovi validacije postoje. Osim ova tri parametra, u konfiguraciji se postavljaju još neki parametri bitni za funkcionisanje sistema. Parametar *StewTime* može imati bilo koju celobrojnju vrednost, dok parametri *Enabled* i *Retained* mogu imati vrednosti *true* ili *false*. Na slici 3 je prikazan *Runtime profile configuration* modul i prozor za prikaz parametara.



Slika 3. *Runtime profile configuration* modul i prikaz parametara profila

6.4. Automatska simulacija

U okviru ovog modula, korisnik je obavezan da unese broj događaja za simulaciju. Automatska simulacija podrazumeva slanje različitih tipova događaja sa slučajno odabranih brojila. Šalju se događaji sa raznovrsnim vrednostima parametara, kako bi obuhvatili sve scenarije koji se odvijaju na serverskoj strani. Prilikom svakog ciklusa više puta se menjaju parametri profila, radi izvršavanja pomenutih scenarija. Automatska simulacija se izvodi u posebnoj niti. Vremenski interval između dva ciklusa slanja je takođe konfigurabilan od strane korisnika. Na slici 4 prikazan je *Simulation* modul.



Slika 4. Modul za automatsku simulaciju slanja događaja

6.6. Testiranje

Testiranje je vezano za generisanje slučajnih događaja i njihovo slanje na servisnu stranu. Ovo testiranje vršeno je u modulu *Send events using validation*. Osim slanja događaja na servisnu stranu, u okviru aplikacije, vrši se i upisivanje događaja u bazu podataka, koje takođe ulazi u

vreme testiranja. Testirani su slučajevi slanja 10, 50, 100, 500 i 1000 događaja. Testiranja su vršena za slanje 10, 50 i 100 događaja po 10 puta, dok su za slanje 500 i 1000 događaja testiranja vršena po 5 puta. Prilikom merenja, dešavala su se manja odstupanja, pa su kao konačan rezultat uzimane srednje vrednosti izmerenih vremena. Prilikom merenja, uočen je skoro linearan porast vremena potrebnog za slanje događaja od njihovog broja. Vreme za generisanje, slanje i skladištenje jednog događaja variralo je od 103 ms do 119 ms. U tabeli 1 prikazana je zavisnost vremena slanja od broja događaja.

Br. dog.	10	50	100	500	1000
Vreme[s]	1,14	5,15	11,98	53,99	111,4

Tabela 1. Zavisnost vremena slanja od broja događaja

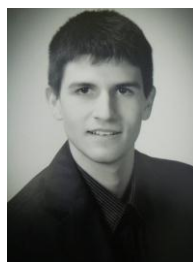
7. ZAKLJUČAK

Današnji život ne možemo da zamislimo bez električne energije. Korišćenje AMI infrastrukture, kao i upotreba pametnih brojila na strani potrošača, poboljšava delovanje elektromreže, a takođe nudi različite funkcionalnosti koje mogu biti vrlo korisne elektrodistribucijama. Upotreba pametnih brojila donosi mogućnost dvosmerne komunikacije između elektrodistribucije i pametnih brojila, što u velikoj meri povećava kontrolu koju elektrodistribucija ima nad mrežom. Događaji igraju značajnu ulogu jer obaveštavaju elektrodistribuciju o neželjenim promenama u mreži. Rezultati testiranja daju lineran porast vremena potrebnog za slanje od broja događaja, što je i bio očekivan rezultat. Jedan od pravaca daljeg razvoja bi bila registracija (*subscribe*) aplikacije na promene koje se dešavaju u internim strukturama servisa, što bi rezultovalo smanjenjem opterećenja resursa prilikom konstantnog vršenja osvežavanja tabela koje u simulatoru prikazuju događaje sa servisa.

8. LITERATURA

- [1] Softver sa kritičnim odzivom u elektroenergetskim sistemima, Novi Sad, 2014. Branislav Atlagić
- [2] <http://smartgrid.ieee.org/newsletters/november-2015/distributed-energy-resources-integration-in-distribution>
- [3] Advanced Metering Infrastructure, 2008, NETL Modern Grid Strategy Powering out 21st-Century Economy
- [4] A survey on Advanced Metering Infrastructure, 2014, Ramyar Rashed Mohassel, Alan Fung, Farah Mohammadi, Kaamran Raahemifar
- [5] ADMS Smart Meter Infrastructure Designs, 2015, Schneider Electric DMS NS

Kratka biografija:



Branislav Potić rođen je 1993. godine u Novom Sadu gde i dalje živi. Školske 2011/2012. upisuje Fakultet tehničkih nauka, smer Računarstvo i automatika. Diplomski rad je odbranio 2015. godine kada je i upisao master studije, smer Primenjeno softversko inženjerstvo.

PRIMENA GA I PSO ALGORITMA ZA PODEŠAVANJE PARAMETARA PID REGULATORA KORIŠĆENJEM FUZZY LOGIKE ZA OCENU POTENCIJALNIH REŠENJA**COMPARISON OF GA AND PSO ALGORITHMS FOR AUTO TUNING OF PID REGULATOR PARAMETERS, USING FUZZY LOGIC FOR GRADING POTENTIAL SOLUTION**

Jovan Zelenković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**2. OPIS RADA**

Kratak sadržaj – U ovom radu opisana je primena GA i PSO algoritma kao alata za automatsko podešavanje parametara PID regulatora. Takođe izvršeno je poređenje rezultata datih algoritama u cilju procene primenljivosti ova dva algoritma na problemu podešavanja parametara PID regulatora.

Ključne reči: PID regulator, Fuzzy regulator, Genetski algoritam, PSO algoritam

Abstract – This paper describes the application of GA and PSO algorithms as a tools for automatic tuning of PID controller parameters. Also, there is performed a comparison of results given algorithms in order to assess the applicability of these two algorithms on the problem of automatic tuning of PID controller parameters.

Key words: PID regulator, Fuzzy regulator, Genetic algorithm, PSO algorithm

1. UVOD

PID regulacija spada među najrasprostranjenije tipove regulacije. PID (Proporcionalno Integralno Diferencijalne) kontrolere srećemo u velikom broju industrijskih procesa. Od svoga nastanka prešli su veliki put od pneumatike, vakuumskih cevi, tranzistora pa sve do danas kada se koriste i u mikročipovima. PID regulator dobijamo kombinovanjem tri osnovna zakona upravljanja: proporcionalnog, integralnog i diferencijalnog. Ova tri zakona upravljanja takodje možemo kombinovati u nekoliko različitih kombinacija pa tako možemo imati: P, PI, PD i PID regulator. Uz sve prednosti koje PID regulator donosi sa sobom, on takodje ima i jednu manu. Veoma je teško odrediti parametre ovog regulatora, pogotovo za sisteme čija funkcija prenosa nam nije poznata. I tako, vrlo često se pribegava metodama za automatsko podešavanje parametara PID regulatora. Ove metode određenim postupcima teže da izračunaju što bolje vrednosti ovih parametara. Loša strana ovih metoda je ta da vrlo često da bi došle do zadovoljavajućih vrednosti parametara, moraju veliki broj puta pokretati sistem i ocenjivati odziv. Parametri PID regulatora su konstanta proporcionalnog, integralnog i diferencijalnog dejstva (k_p , k_i i k_d). Postoji širok spektar metoda i algoritama koji se primenjuju za automatsko određivanje vrednosti ovih parametara. Između ostalog u te metode spadaju i genetski (GA) i PSO algoritam. Primena ova dva algoritma je opisana u ovom radu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Velimir Čongradac.

2.1. Komponente i početni uslovi

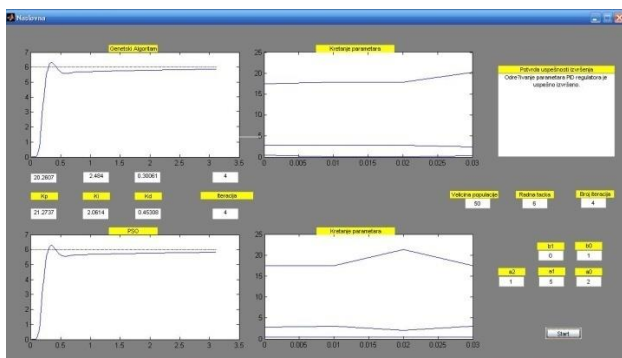
Osnovni uslov za rešavanje problema automatskog podešavanja parametara PID regulatora pomoću GA i PSO algoritma, i poređenja njihovih rezultata, je taj da je potrebno obezbediti da oba algoritma izvršavaju pretragu pod istim uslovima. Pre samog pokretanja GA ili PSO algoritma, potrebno je kreirati početnu populaciju jedinki. I GA i PSO algoritam svoj proces pretrage započinju sa istom početnom populacijom. Takođe kriterijum zaustavljanja pretrage je i kod jednog i kod drugog dostignut zadati broj iteracija. Jedinka u populaciji je formirana u obliku strukture tri broja. Ti brojevi predstavljaju vrednosti parametara k_p , k_i i k_d . Fuzzy logika u ovom slučaju ima ulogu alata za ocenjivanje potencijalnih rešenja.

Sam teorijski rad je materijalizovan u vidu softverskog rešenja pomoću programskog paketa Matlab.

2.2. Opis softverskog rešenja

Samo softversko rešenje je podeljeno na zasebne celine, baš kao i teorijski kocept rešenja zadatog problema. Svi parametri koji zavise od korisnika, a neophodni su za uspesno izvršavanje softverskog rešenja, unose se preko grafičkog interfejsa. U te parametre spadaju: koeficijenti funkcije prenosa sistema za koji želimo da vršimo automatsko podešavanje parametara PID regulatora, zadata vrednost (*setpoint*), veličina populacije i zadati broj iteracija. Takođe na grafičkom interfejsu se prikazuju rezultati nakon izvršavanja GA i PSO algoritama. Rezultati koji se prikazuju su: najbolje vrednosti parametara k_p , k_i i k_d koje je uspeo genetski algoritam da pronađe, najbolje vrednosti parametara k_p , k_i i k_d koje je PSO algoritam uspeo da pronađe, redni broj iteracije (generacije) u kojoj je genetski algoritam uspeo da pronađe najbolje vrednosti parametara k_p , k_i i k_d kao i redni broj iteracije u kojoj je PSO algoritam uspeo da pronađe najbolje vrednosti parametara k_p , k_i i k_d . Osim toga na grafičkom interfejsu se iscrtavaju i četiri grafika. Na prvom grafiku se iscrtava odziv sistema sa PID regulatorom na odskočni signal čija je vrednost jednaka zadatoj vrednosti (*setpoint*), sa parametrima koje je odredio genetski algoritam. Na drugom grafiku se iscrtava odziv sistema sa PID regulatorom na odskočni signal čija je vrednost jednaka zadatom *setpoint*-u sa parametrima koje je odredio PSO algoritam. Na trećem grafiku se iscrtava kretanje vrednosti parametara k_p , k_i i k_d najbolje jedinke iz svake generacije/iteracije kod pretrage pomoću genetskog algoritma. I na posletku, na četvrtom grafiku se iscrtava kretanje vrednosti parametara k_p , k_i i k_d najbolje jedinke iz svake generacije/iteracije kod pretrage pomoću PSO

algoritma. Osim svega ovoga, na grafičkom interfejsu se nalazi još i jedan prozor u kome se po završetku izvršavanja softverskog rešenja ispisuje poruka o uspešnom ili neuspešnom izvršavanju. Ukoliko je u pitanju neuspešno izvršavanje ispisuje se poruka koja ukazuje na uzrok neuspešnog izvršavanja. Na Slici 1. prikazan je izgled grafičkog interfejsa.

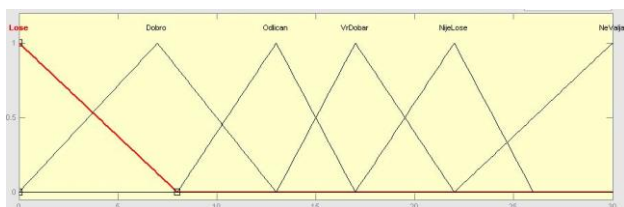


Slika 1. – Izgled grafičkog interfejsa

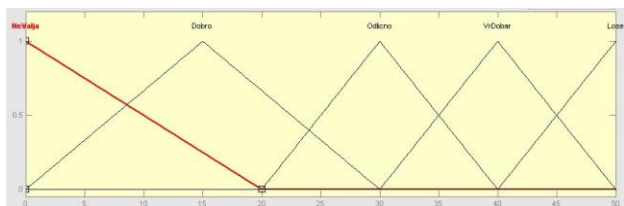
Grafički interfejs preko funkcije „main()“ upravlja kompletnim izvršavanjem softverskog rešenja. U principu funkcija „main()“ ima pet osnovnih koraka, a to su :

1. Provera da li su na grafičkom interfejsu pravilno unete vrednosti koeficijenata funkcije prenosa sistema za koji podešavamo parametre PID regulatora (ne može se desiti da je brojilac funkcije prenosa zadatog sistema većeg stepena od imenioca funkcije prenosa tog sistema)
2. Kreira se Fuzzy regulator koji će vršiti ocenjivanje potencijalnih rešenja
3. Kreira se početna populacija
4. Izvršava se podešavanje parametara PID regulatora pomoću genetskog algoritma
5. Izvršava se podešavanje parametara PID regulatora pomoću PSO algoritma
6. Dobijeni rezultati se ažuriraju na grafičkom interfejsu

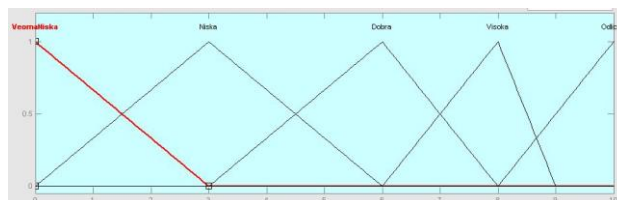
Fuzzy regulator se kreira kroz zasebnu skriptu koja se poziva iz „main()“ funkcije. Ulazi u fuzzy regulator su preskok sistema za zadati *setpoint* i vreme smirenja sistema. Izlaz fuzzy regulatora je ocena potencijalnog rešenja za zadate ulazne parametre. Na Slici 2, Slici 3 i Slici 4 su prikazane funkcije pripadnosti ulaznih i izlaznog parametra fuzzy regulatora.



Slika 2. – Funkcije pripadnosti parametra „preskok“

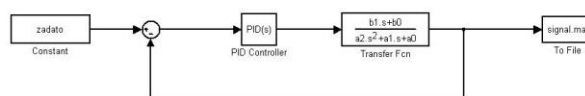


Slika 3. – Funkcije pripadnosti parametra „smirenje“



Slika 4. – Funkcije pripadnosti parametra „ocena“

Da bi se izračunale vrednosti parametara „preskok“ i „smirenje“ za svako potencijalno rešenje, potreban je odziv sistema sa PID regulatorom na zadati *setpoint*. On se dobija pokretanjem Simulink modela koji nakon izvršavanja dobijeni odziv snima u fajl pod nazivom „odziv.mat“. Ovaj fajl se učitava i pomoću funkcije „ocena()“ izračunavaju se neophodni parametri i dobija se ocena potencijalnog rešenja. Izgled Simulink modela je prikazan na Slici 5.



Slika 5. – Izgled Simulink modela

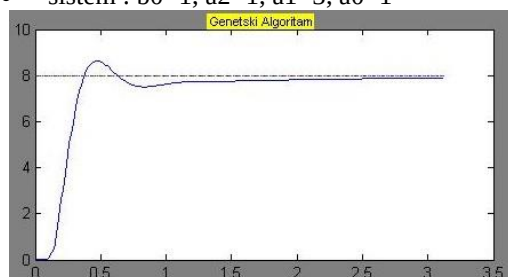
Najvažnije osobine genetskog algoritma primenjenog u ovom softverskom rešenju su te da kroz elitizam u narednu generaciju prelaze četiri najbolje jedinke iz trenutne generacije i to da je ukrštanje jedinki turnirsko. Pod turnirskim ukrštanjem u ovom slučaju se podrazumeva to da se nasumično bira pet jedinki, a zatim od tih pet jedinki se uzimaju dve najbolje od kojih će se kreirati nove dve jedinke naredne generacije. Što se tiče PSO algoritma tu je najvažnije pomenuti da je veličina koraka za koji će se jedinka (potencijalno rešenje) pomeriti u prostoru rešenja obrnuto proporcionalna vrednosti ocene te jedinke. To znači da što je bolje neka jedinka (potencijalno rešenje) ocenjena, to će se manje pomeriti za sledeću iteraciju. Jedinke sa visokim ocenama su najbliže pronalaženju zadovoljavajućih rešenja i one nemaju potrebu da se značajno pomeraju iz tačke u kojoj se trenutno nalaze.

2.3. Prikaz rezultata

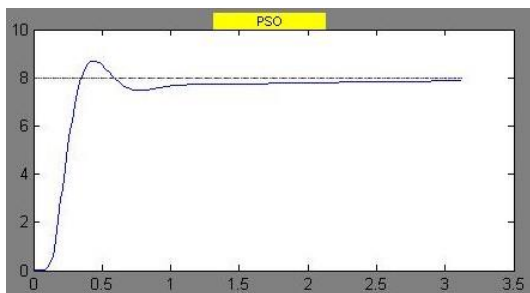
Na sledećih nekoliko slika prikazani su primeri dobijenih rezultata automatskog podešavanja parametara PID regulatora pomoću opisanog softverskog rešenja, sa određenim zadatim parametrima.

2.3.1 Primer 1:

- veličina populacije : 50
- broj iteracija : 4
- *setpoint* : 8
- sistem : $b_0=1, a_2=1, a_1=3, a_0=1$



Slika 6. – Odziv sistema sa PID regulatorom sa parametrima koje je pronašao GA – Primer 1

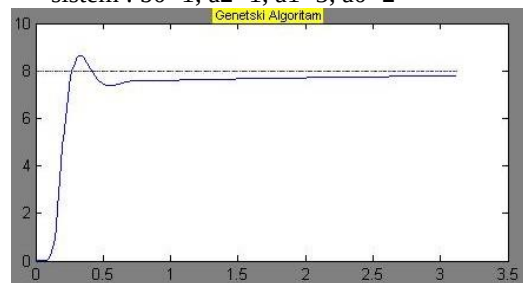


Slika 7. – Odziv sistema sa PID regulatorom sa parametrima koje je pronašao PSO – Primer 1

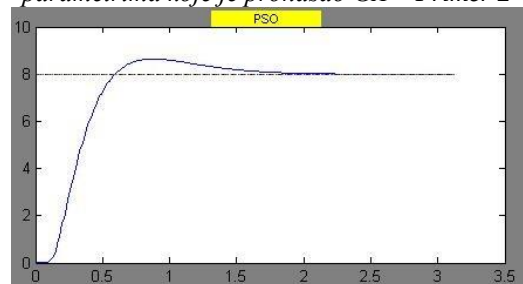
PSO algoritam je najbolje parametre pronašao u 2. iteraciji. GA algoritam je najbolje parametre pronašao u 4. generaciji.

2.3.2 Primer 2:

- veličina populacije : 200
- broj iteracija : 400
- setpoint : 8
- sistem : $b_0=1$, $a_2=1$, $a_1=5$, $a_0=2$



Slika 8. – Odziv sistema sa PID regulatorom sa parametrima koje je pronašao GA – Primer 2



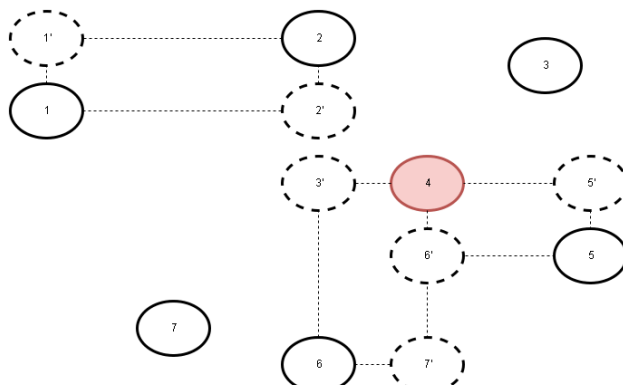
Slika 7. – Odziv sistema sa PID regulatorom sa parametrima koje je pronašao PSO – Primer 2

PSO algoritam je najbolje parametre pronašao u 14. iteraciji. GA algoritam je najbolje parametre pronašao u 53. generaciji.

3. ZAKLJUČAK

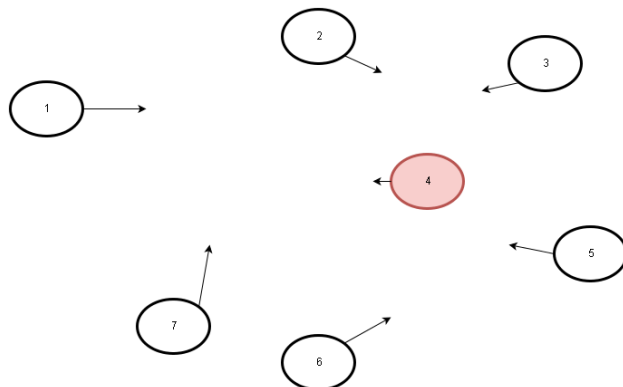
Metode za automatsko podešavanje parametara PID regulatora su izuzetno koristan alat koji operaterima i inženjerima omogućava da imaju dobro podešen regulator, a samim tim i dobru kontrolu procesa. Što se tiče GA i PSO algoritma, oni imaju izuzetan potencijal za primenu u ovoj oblasti automatike i upravljanja. I jedan i drugi algoritam garantuju pronalazjenje kvalitetnih vrednosti parametara kp, ki i kd za određeni broj iteracija/generacija. U svakom slučaju, nakon više desetina pokretanja programa za automatsko podešavanje parametara PID regulatora, opisanog u ovom radu, nametnuo se zaljučak da je PSO algoritam dosta efikasniji u izvršavanju datog problema. U više od 90% slučajeva PSO algoritam je imao bolji krajnji rezultat nego GA, a u više od 80% slučajeva je bio brži od

GA. Uzrok tome je sama činjenica koja proističe iz osnovnih karakteristika i jednog i drugog algoritma. Prednost PSO algoritma nad GA leži u tome da jedinke PSO algoritma imaju dosta fleksibilniju slobodu kretanja. Na Slici 9 prikazan je princip kretanja jedinki u prostoru rešenja kod Genetskog algoritma.



Slika 9. – Princip kretanja jedinki kod GA

Na Slici 10 prikazan je princip kretanja jedinki u prostoru rešenja kod PSO algoritma.



Slika 10. – Princip kretanja jedinki kod PSO algoritma

4. LITERATURA

- [1] K.Aström, T.Hägglund - „PID controllers: Theory, Design and Tuning“, 2nd Edition, Instrument Society of America, ISBN 1-55617-516-7
- [2] „Auto Tuning of PID Controller Using Swarm Intelligence“, M. H. T. Omar, W. M. Ali, M. Z. Mostafa
- [3] „PID controller“, Wikipedia
- [4] Bandyopadhyay, U.K.Chakraborty, D.Patranabis - „Autotuning a PID controller: A fuzzy-genetic approach“, Journal of Systems Architecture

Kratka biografija:



Jovan Zelenković je rođen 12.07.1990. godine u Sremskoj Mitrovici. Odbranio je Master rad na Fakultetu tehničkih nauka 2016. god.

email: zjovanftn@gmail.com

linkedin: Jovan Zelenković

SISTEM RANOG UPOZORENJA - MODUL BOLNIČKOG INFORMACIONOG SISTEMA
EARLY WARNING SYSTEM - A SEGMENT OF A HOSPITAL INFORMATION SYSTEM

Jelena Jovanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Predmet rada je opis ideje i realizacije sistema ranog upozorenja kao modula bolničkog informacionog sistema Instituta za kardiovaskularne bolesti Vojvodine (IKVBV). Rad obuhvata programsko rešenje rada (kod) i opis rada sistema ranog upozorenja.

Abstract – This paper presents the idea and implementation of a early warning system as part of a hospital information system of the Institute for cardiovascular diseases of Vojvodina (ICVDV). Moreover, the source code for the system is presented, as are the algorithms and examples of how the system works.

Ključne reči: *Java programski jezik, SQL, informacijski sistemi, bolnički informacijski sistem*

1. UVOD

Savremen bolnički informacijski sistem sadrži informacije o svim aspektima poslovanja bolnice. Osnovna ideja bolničkih informacijskih sistema (u daljem tekstu BIS) je dostupnost podataka o pacijentima svim učesnicima u procesu zdravstvene nege (lekari i medicinsko-tehničko osoblje) sa ciljem poboljšanja bezbednosti pacijenta i efikasnosti lečenja. BIS predstavlja sistem čija je primarna uloga čuvanje i prikaz podataka, tako da ga po složenosti najčešće možemo svrstati u grupu sistema za obradu transakcija [1]. BIS pruža masu podataka o pacijentima, koji zahtevaju određeno vreme za analizu. Primećeno je da u određenim situacijama jedan lekar ima zadatak da prođe kroz razne, njemu nepotrebne podatke da bi došao do onih koji su mu u tom, neretko kritičnom trenutku za lečenje pacijenta, od velikog značaja. Osnovni cilj sistema ranog upozorenja je da se omogućí da lekari imaju brz uvid u stanje pacijenta, koji im olakšava vizite i signalizira pogoršavanje stanja pacijenta, čime se povećava bezbednost pacijenata i smanjuje mogućnost previda i greške u lečenju.

2. REALIZACIJA SISTEMA RANOG UPOZORENJA

Sistem ranog upozorenja je nadogradnja postojećeg BIS-a i realizovan je kroz izmene programskog koda i baze podataka.

2.1. Baza podataka

Baza podataka BIS-a (PostgreSQL 9.4) proširena je dodavanjem četiri tabele za potrebe realizacije sistema ranog upozorenja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji je mentor bio dr Velimir Čongradac, vanr. prof.

Tabele sadrže 62 stanja koja se prate (tabela *tip_stanja_ews*), vrednost svakog stanja za određenog pacijenta (tabela *stanje_ews*), istorijat stanja po pacijentu (tabela *stanje_ews_istorijat*), i tabelu parametara koji utiču na stanja sa njihovim graničnim vrednostima (tabela *ts_parametar_ews*).

Parametri koji se prate su logički (da/ne), tekstualni, ili numerički (vrednosti laboratorijske analize).

2.2. Grafički interfejs sistema ranog upozorenja

Grafički interfejs je realizovan klasama iz paketa *javax.swing* i *java.awt* [2], kao i klasama koje nasleđuju klase koje su već realizovane za BIS (Java programski jezik, NetBeans okruženje [3]).

Sistem se oslanja na četiri klase:

- *pnIEWSBrowse.java* - početni ekran sistema ranog upozorenja (slika 1),
- *frmEWSDetalj.java* - prikaz stanja pacijenta (slika 2) i
- prikaz podstanja jednog sistema organa (slika 3), *frmEWSGrafik.java* i
- *frmEWSProtokol.java*, koji obezbeđuju prikaz detalja podstanja u vidu liste pregleda (sl. 4) ili grafika vrednosti laboratorijske analize (sl. 5).

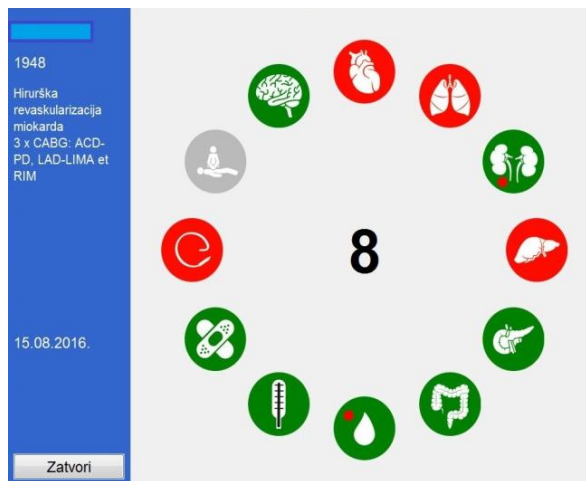
[illegible]

Slika 1. Početni ekran sistema ranog upozorenja-detali

Početni ekran prikazuje listu pacijenata koji trenutno leže na IKVBV i imali su kardiohiruršku operaciju. Prikazane kolone su ime i prezime pacijenta (link na njegov elektronski zdravstveni karton), njihovi identifikacioni broj u BIS-u, lokaciju gde se nalaze, broj postoperativnih dana, pol, starost i vrednost stanja (sistema organa). Dvostrukim klikom na bilo koje polje osim imena pacijenta otvara se detaljniji prikaz stanja za odabranog pacijenta (slika 2).

2.3. Određivanje stanja

Na slici 2 se vidi da su vrednosti stanja prikazana grafički, gde crvena boja obeležava "loše" stanje, zelena boja "dobro" stanje, zelena boja sa crvenim kružićem "poboljšanje", dok siva boja označava nedostatak podataka o stanju.



Slika 2. Prikaz stanja jednog pacijenta

Svako stanje (njih 12) definisano je vrednostima podstanja, čiji broj je promenljiv (stanje kardiovaskularnog sistema ima 6 podstanja, stanje respiratornog sistema 7, itd). Prikaz podstanja je dat na slici 3, i dobija se dvostrukim klikom na ikonicu stanja.

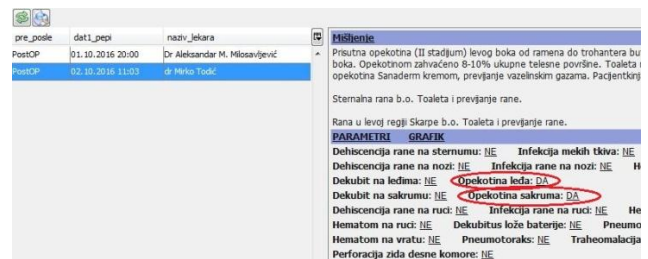


Slika 3. Prikaz podstanja kardiovaskularnog sistema

Vrednost stanja se određuje proverom parametara iz baze podataka, što rade funkcije pisane u PL/pgSQL proceduralnom jeziku [4]. Svako stanje i podstanje koje se nalazi u tabeli *tip_stanja_ews* u određenoj koloni sadrži naziv SQL funkcije koja određuje vrednost stanja. Na jedno podstanje može uticati više parametara, i ako je vrednost jednog od tih parametara van graničnih, podstanje je crveno. Ako je makar jedno podstanje crveno, i nadređeno stanje postaje crveno (slika 3). Ako je vrednost stanja bilo crveno u prošlosti, a sad je zeleno, vrednost postaje poboljšana - zeleno sa crvenim kružićem. Nadređena PL/pgSQL funkcija poziva sve funkcije koje vraćaju vrednosti stanja. Ta nadređena funkcija se poziva cron naredbom svakih 10 minuta [5].

2.4. Prikaz vrednosti podstanja

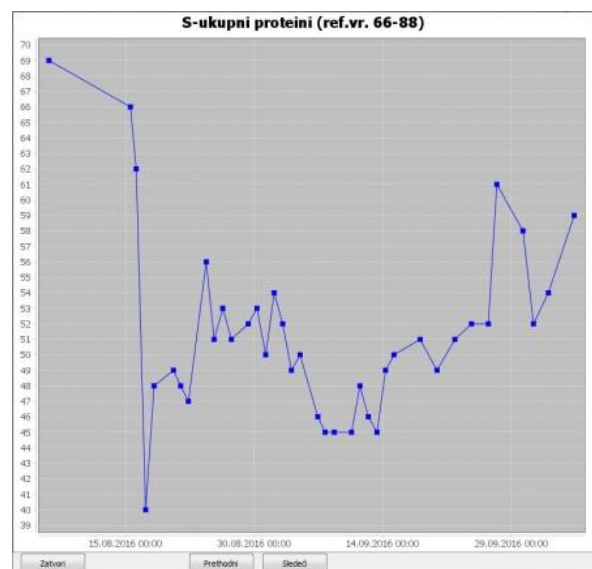
U sistemu ranog upozorenja su realizovana dva načina prikazivanja vrednosti podstanja, jedna je prikaz pregleda iz BIS koji sadrže parametre od interesa (slika 4), a drugi je prikaz grafika laboratorijske analize koja utiče na podstanje (slika 5).



Slika 4. Prikaz pregleda sa parametrima od interesa- detalj

Svako podstanje, koje nije vrednost laboratorijske analize, ima definisane tipove pregleda čiji parametri utiču na njegovu vrednost, i dvostrukim klikom na podstanje oni se prikazuju u formi liste, sa kolonama koje označavaju da li je pregled vršen pre ili posle operacije, tačan datum pregleda i ime lekara koji je uradio pregled.

Klikom na jedan pregled u listi, na desnoj strani se prikazuju detalji pregleda: mišljenje, parametri, dijagnoze itd.



Slika 5. Grafik laboratorijske analize

Grafik je realizovan uz pomoć klase *org.jfree.chart.ChartPanel*, i sadrži dve ose relativno određene u odnosu na maksimalnu i minimalnu vrednost analize, kao i datume analize.

Naslov se sastoji iz imena analize i referentnih numeričkih vrednosti iz tabele *ts_parametar_ews*. Dugmad (JButton) *prethodni* i *sledeći* daju grafike pojedinačnih analiza, u slučaju da postoji više analiza čije vrednosti utiču na podstanje.

3. ZAKLJUČAK

Sistem ranog upozorenja je inovacija koja je usmerena na poboljšanje ishoda lečenja operisanih pacijenata. Realizovana je ideja je da se stanje pacijenta prikaže grafički, tako da lekaru bude intuitivno jasno.

Pušten je u rad pre nekoliko meseci (proleće 2016. godine), prihvaćen je od strane lekara, ali je vremenski period od puštanja u rad do pisanja ovog rada kratak da bi

se prikupio reprezentativni uzorak operisanih pacijenata, gde bismo uporedili parametre lečenja (komplikacije, mortalitet) pacijenata pre i posle uvođenja ovog sistema.

4. LITERATURA

- [1] Rainer, R. Kelly Jr and Casey G. Cegielski.
Introduction to Information System: Support and
Transforming Business Fourth Edition. New Jersey:
John Wiley and Sons, Inc., 2012. Print.
- [2] Java javax.swing i java.awt paketi :
[https://docs.oracle.com/javase/7/docs/api/javax/swing/
package-summary.html](https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html)
[https://docs.oracle.com/javase/7/docs/api/java/awt/package-
summary.html](https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html)
- [3] NetBeans programsko okruženje
<https://netbeans.org/features/index.html>
- [4] PL/pgSQL proceduralni jezik, adresa:
[https://www.postgresql.org/docs/8.3/static/plpgsql-
structure.html](https://www.postgresql.org/docs/8.3/static/plpgsql-structure.html)
- [5] CRON/Crontab, adresa:
<https://help.ubuntu.com/community/CronHowto>

Kratka biografija:



Jelena Jovanović rođena je u Novom Sadu 1989. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Automatizacija nadzorno-upravljačkog sistema ventilacije i osvetljenja tunela Mišeluk odbranila je 2013. god.

MODELOVANJE PODZEMNOG TOPLOTNOG IZMENJIVAČA GEOTERMALNE TOPLOTNE PUMPE POMOĆU HELLSTRÖM-OVOG DVODIMENZIONOG MODELA**MODELING A UNDERGROUND HEAT EXCHANGER OF A GROUND COUPLED HEAT PUMP USING HELLSTRÖM'S TWO DIMENSIONAL MODEL**

Slobodan Stamenković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Toplotne pumpe mogu da obezbede toplotnu energiju efikasno i ekonomično uz veoma malu emisiju štetnih gasova. Koncept toplotnih pumpi poznat je još od 19 veka, a komercijalna primena postoji skoro 60 godina. Toplotne pumpe prenose termalnu energiju sa nižeg na viši temperaturni medijum, slično frižiderima. Osnovni delovi jednog sistema geotermalne toplotne pumpe (toplotne pumpe sa izvorom zemlje) su: geotermalna toplotna pumpa, podzemni toplotni izmenjivač, unutrašnji sistem za distribuciju toplotne energije. Cilj ovog rada je da se napravi model podzemnog toplotnog izmenjivača koji oslikava njegovo stvarno ponašanje i pokaže njegovu efikasnost. U tu svrhu je iskorištena teorija Hellström-ovog dvodimenzionog modela kao osnov za ovaj proces, koja je potom implementirana u Matlab Simulink okruženje radi simuliranja i tumačenja dobijenih rezultata.

Gljučne reči: geotermalna toplotna pumpa, podzemni izmenjivač, Hellström-ov model, Matlab, Simulink

Abstract – Heat pumps can provide energy efficiently and economically with little emission of harmful gases. The concept of heat pumps is known since the 19th century, while commercial use exists for almost 60 years. Heat pumps transfer thermal energy from lower to higher temperature mediums, similar to refrigerators. The main parts of a geothermal heat pump system (ground source heat pump) are: geothermal heat pump, underground heat exchanger, internal system for distribution of thermal energy. The aim of this paper is to model a ground heat exchanger that represents his behaviour in a real environment, and show his efficiency. In that manner, Hellström's two dimensional model was used as a theoretical element for the modeling process, which was then implemented in Matlab Simulink in order to simulate and interpret the obtained results.

Key words: geothermal heat pump, ground heat exchanger, Hellström's model, Matlab, Simulink

1. UVOD

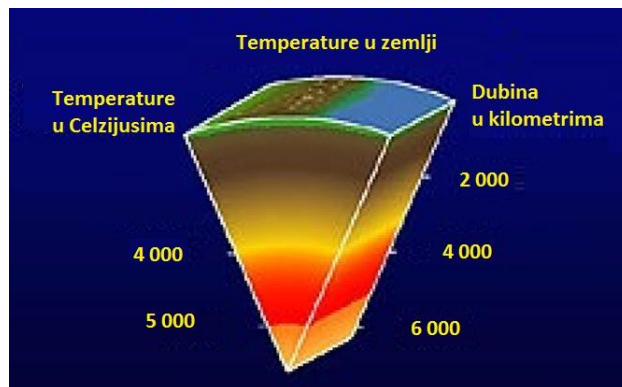
Veliki udeo globalnih energetske izvora koristi se za proizvodnju električne energije i prostorno grejanje, pri čemu većinu ove energije čine fosilna goriva. Fosilna goriva su ograničeni izvori i njihovo sagorevanje je štetno po životnu sredinu, putem emisije gasova koji stvaraju

efekat „staklene bašte“, koji doprinose klimatskim promenama i drugim zagađenjima.

Potreba za energetske izvorima raste i buduće nestašice fosilnih goriva mogu da se nagoveste. Klimatske promene predstavljaju realnu opasnost za našu budućnost, a jedan od glavnih razloga je upotreba fosilnih goriva koje snabdevaju energijom, kako industriju, tako i domaćinstva. Obnovljivi izvori energije, kombinovani sa energetske efikasnošću, pružaju održivo i potentno rešenje u borbi protiv uticaja globalnog zagrevanja.

Obnovljivi oblici energije kao što su solarna energija, energija vetra, energija biomase, hidroenergija i energija zemlje emituju minimalan ili nikakav efekat „staklene bašte“ u okruženje. Sistemi geotermalnog grejanja i hlađenja (koji se takođe nazivaju i earth – energy sistemi (EES)) predstavljaju toplotne pumpe koje sakupljaju i prenose toplotnu energiju iz zemlje putem niza tečnošću ispunjenim cevima povezanim sa objektom, pri čemu se ta toplotna energija koristi unutar objekta. Geotermalne toplotne pumpe ne stvaraju toplotnu energiju sagorevanjem – one samo pomeraju toplotu sa jednog mesta na drugo.

Geotermalne toplotne pumpe predstavljaju visoko efikasne uređaje u tehnologiji obnovljive energije za prostorno grejanje i hlađenje. Ova tehnologija se oslanja na činjenicu da, na određenim dubinama, zemlja ima relativno konstantnu temperaturu, topliju od vazduha zimi i hladniju od vazduha leti. Geotermalna toplotna pumpa prenosi toplotu koja se nalazi u zemlji u objekat u toku zime, i prenosi tu istu toplotu iz objekta u periodu leta. Ovi tipovi geotermalnih toplotnih pumpi mogu da se primene praktično svuda, pri čemu posebni geološki uslovi, kao što su gejziri ili vrela, nisu potrebni za uspešnu primenu.



Slika 1: Toplotna energija unutar zemljine kore

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Velimir Čongradac, red.prof.

2. GEOTERMALNE TOPLOTNE PUMPE

Geotermalne toplotne pumpe, koje se nazivaju i toplotne pumpe sa izvorom zemlje (ground source heat pumps), zemljani energetski sistemi (earth energy systems), georazmenjivačke toplotne pumpe (geoexchange heat pumps), toplotne pumpe spregnute sa zemljom (earth-coupled heat pumps) i sistemi sa izvorom zemlje (ground-source systems), sastoje se od tri glavna podsistema:

- Geotermalna toplotna pumpa: prenosi toplotnu energiju između objekta i zemlje i modifikuje (prilagođava) njenu temperaturu

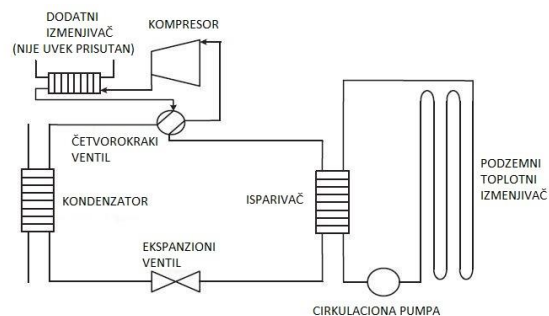
- Podzemni izmenjivač: olakšava ekstrakciju toplotne energije iz zemlje preko sistema toplotnih izmenjivača za potrebe toplotne pumpe

- Unutrašnji sistem za distribuciju toplotne energije: Uslovljava i distribuira toplotnu energiju širom prostorije (objekta).

Toplotne pumpe funkcionišu korištenjem električne energije koja pokreće kompresore koji obezbeđuju neophodni rad za koncentraciju i transport termalne energije. Osnovne toplotne pumpe funkcionišu po rashladnom ciklusu isparivanja-kompresije. Radni fluid unutar toplotne pumpe je najčešće neki rashladni, koji se bira tako da se ostvare odgovarajuće karakteristike i zahtevi sistema geotermalne toplotne pumpe. Geotermalna toplotna pumpa prenosi termalnu energiju između zemlje i objekta kontrolišući pritisak i temperaturu putem kompresije i ekspanzije. Pet glavnih komponenti su međusobno usaglašene u toplotnoj pumpi (slika 2): kompresor, ekspanzioni ventil, četvorokraki ventil i dva toplotna izmenjivača. Tu su takođe i različite manje komponente i dodaci kao što su ventilatori, cevi i upravljački uređaji koji pomažu pri radu.

Geotermalna toplotna pumpa u režimu grejanja funkcioniše na sledeći način:

1. Termalna energija se ekstrahuje iz zemlje i prenosi do isparivača.
2. Unutar jedinice toplotne pumpe rashladno sredstvo, koje se nalazi u dominantno tečnom agregatnom stanju, ulazi u isparivač. Toplotna energija se prenosi iz zemlje (sistem veze sa zemljom) na rashladni fluid i prouzrokuje njegovo zagrevanje. Usled zagrevanja, rashladni fluid menja agregatno stanje u paru, i kao rezultat dobija se para niskog pritiska i povećane temperature.
3. Para zatim ulazi u električno napajan kompresor, gde se pritisak povećava, što rezultuje u povećanju temperature. U ovoj fazi se dobija para visokog pritiska i visoke temperature.
4. Para visoke temperature zatim odlazi u kondenzator. Rashladni medijum je na višoj temperaturi u odnosu na objekat, podstičajući toplotni prenos sa medijuma na objekat. Rashladni medijum se postepeno hladi i kondenzuje, i postaje fluid visokog pritiska i visoke temperature.
5. Vreo fluid prolazi kroz ekspanzioni ventil koji mu smanjuje pritisak, što prouzrokuje i smanjenje temperature. Rashladno sredstvo ulazi u isparivač gde se započinje novi ciklus.



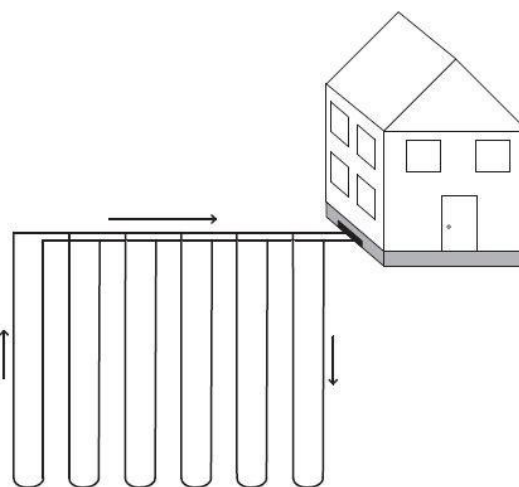
Sl. 2: Osnovni izgled sistema geotermalne toplotne pumpe

2.1. Podzemni toplotni izmenjivač

Nasuprot toplotnim pumpama sa izvorom vazduha koje koriste spoljašnji vazduh kao izvor toplote, geotermalne toplotne pumpe koriste okolnu zemlju. Spoljašnji vazduh prezentuje veliku varijaciju u temperaturi tokom godine čak i na dnevnoj bazi, u poređenju sa zemljom. Na dnevnoj bazi, temperatura zemlje varira na dubinama manjim od 0.8m; na većim dubinama ta varijacija opada značajno. Varijacije su izraženije više na sezonskoj nego na dnevnoj bazi. Veza sa zemljom ili zemljana petlja toplotnih izmenjivača je sastavljena od kolekcije cevi koje prenose fluid između jedinice toplotne pumpe i zemlje. Dve glavne zemljane petlje postoje: konfiguracija dvostruke i jednostruke petlje.

Konfiguracija dvostruke petlje je najčešće primenjen sistem i zahteva vezu sa zemljom odvojenom od toplotne pumpe. Toplotna energija se prenosi na rashladno sredstvo putem toplotnog izmenjivača koji se sastoji od vode ili kombinacije vode i antifrizu, koja cirkuliše kroz cevi jedinice toplotne pumpe u zemlji. Standardne cevi trenutno u upotrebi sačinjene su od polietilena ili polipropilena prečnika od oko 19mm (3/4 in.) za male i srednje aplikacije. Dva tipa konfiguracije sa dvostrukom petljom trenutno postoje: otvorena i zatvorena.

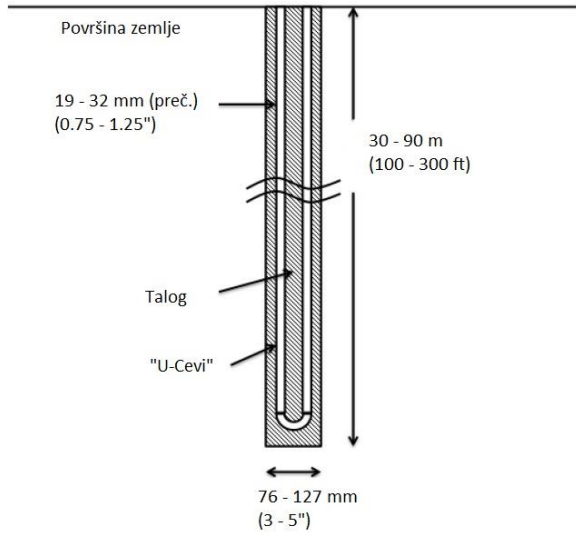
U sistemima zatvorene petlje, koji se često primenjuju, toplotni prenosni fluid je priložen u kružnu petlju i nema direktan kontakt sa zemljom; prenos toplotne energije sa zemljom odvija se preko cevnog materijala. Postoje četiri klase sistema izmenjivača toplotne energije u zatvorenoj petlji: vertikalni, horizontalni, spiralni, i jezerski (potopljeni).



Slika 3: Primer vertikalne zatvorene petlje

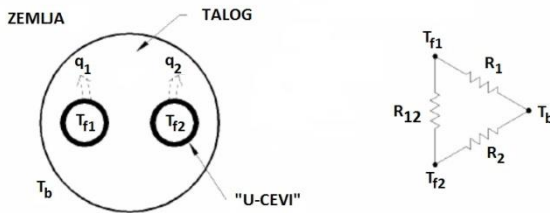
3. MODELOVANJE PODZEMNOG IZMENJIVAČA

Modelovanje podzemnog izmenjivača sa vertikalnim bušotinama može da se podeli u dve oblasti interesovanja: oblast koja se bavi sagledavanjem dešavanja unutar bušotine, kao i oblast koja sagledava dešavanja van bušotine. Razmena energije van okvira bušotine, odnosno odsijavanje energije u okolnu zemlju tipično se posmatra kao problem provođenja energije sa linijskim ili cilindričnim izvorom u beskonačnom ili skoro beskonačnom medijumu. Prenos energije unutar bušotine je kombinacija konvekcije između vode i cevi, kao i provodljivosti između dva kraja cevi, taloga, i zemlje koja okružuje bušotinu. Slika 4 predstavlja izgled jedne bušotine sa podzemnim izmenjivačem.



Slika 4: Izgled podzemnog izmenjivača sa vertikalnom bušotinom

Hellström je razvio analitičko rešenje za prenos energije između dva bitna segmenta, postavljenih cevi i zemlje unutar bušotine. Model stvara otporničku mrežu koja se sastoji od čvorova za svaku temperaturu fluida, kao i uzimanja u obzir temperature unutar obima bušotine. Temperatura unutar obima bušotine se smatra uniformnom. Slika 5 prikazuje temperature čvorova i rezultujuću otporničku mrežu Hellström-ovog modela [1].



Slika 5: Dijagram prenosa energije unutar bušotine

$$\begin{aligned} T_{f1} - T_b &= R_1 q_1 + R_{12} q_2 \\ T_{f2} - T_b &= R_{12} q_1 + R_2 q_2 \end{aligned} \quad (1)$$

Hellström-ov dvo-dimenzioni model je proširen od strane Zeng-a u cilju stvaranja kvazi tro-dimenzionog modela koji uzima u obzir energetske kapacitete fluida, koji se koristi kao prenosni medijum u cevima, u jednačinu 1,

definišući je preko ukupne dužine (dubine), H , bušotine. Rezultujuće jednačine prikazane su jednačinama 2, sa dva ograničavajuća uslova definisanim jednačinama 3. R_1^* , R_2^* , i ekvivalentna otpornost R_{12}^* su transformisane vrednosti otpornosti, M je masa fluida, c je energetska kapacitet fluida, dok je z osa duž dužine (dubine) bušotine [2].

$$\begin{aligned} -Mc \frac{dT_{f1}}{dz} &= \frac{(T_{f1} - T_b)}{R_1^*} + \frac{(T_{f1} - T_{f2})}{R_{12}^*} \\ -Mc \frac{dT_{f2}}{dz} &= \frac{(T_{f2} - T_b)}{R_2^*} + \frac{(T_{f2} - T_{f1})}{R_{12}^*} \end{aligned} \quad (2)$$

$$\begin{aligned} T_{f1}(0) &= T_{in} \\ T_{f1}(H) &= T_{f2}(H) \end{aligned} \quad (3)$$

Prikazane jednačine Hellström-ovog dvo-dimenzionog modela, jednačine 2 i 3, iskoristiće se u daljem radu u cilju projektovanja modela podzemnog izmenjivača geotermalne toplotne pumpe. Radi lakšeg procesa modelovanja Hellström-ove jednačine dvo-dimenzionog modela biće prilagođene i malo izmenjene, u smislu da se neće uzeti u obzir prenos toplote između ulazne i izlazne cevi podzemnog izmenjivača. Posledica ovoga je da dve diferencijalne jednačine Hellström-ovog dvodimenzionog modela postaju nezavisne.

$$m'c \frac{dT_{in}(H)}{dH} = \frac{T_{z1}(H) - T_{in}(H)}{R_e} \quad (4)$$

$$m'c \frac{dT_{out}(H)}{dH} = \frac{T_{z2}(H) - T_{out}(H)}{R_e} \quad (5)$$

Diferencijalne jednačine su konfigurisane sledećim promenljivama: m' – koja predstavlja protok mase vode kroz cevi podzemnog izmenjivača, c – koja predstavlja toplotni kapacitet vode (medijuma) koja kruži kroz podzemni izmenjivač, R_e – koja predstavlja toplotnu otpornost zemlje unutar bušotine, H – koja predstavlja posmatranu dubinu bušotine, $T_{in}(H)$ – koja predstavlja temperaturu vode u ulaznoj cevi na dubini H , $T_{out}(H)$ – koja predstavlja temperaturu vode na izlaznoj cevi na dubini H , $T_{z1}(H)$ – koja predstavlja temperaturu zemlje na datoj dubini.

U jednačinama 6 i 7 pojavljuje se nova promenljiva koja karakteriše temperaturu zemlje na datoj dubini, $T_{z1}(H)$, odnosno $T_{z2}(H)$. Pošto temperatura zemlje linearno zavisi od dubine, ona se može posmatrati na sledeći način [3]:

$$T_{z1}(H) = 0.0706 * H + 8.8912 \quad (6)$$

$$T_{z2}(H) = -0.0706(H_{max} - H) + 16 \quad (7)$$

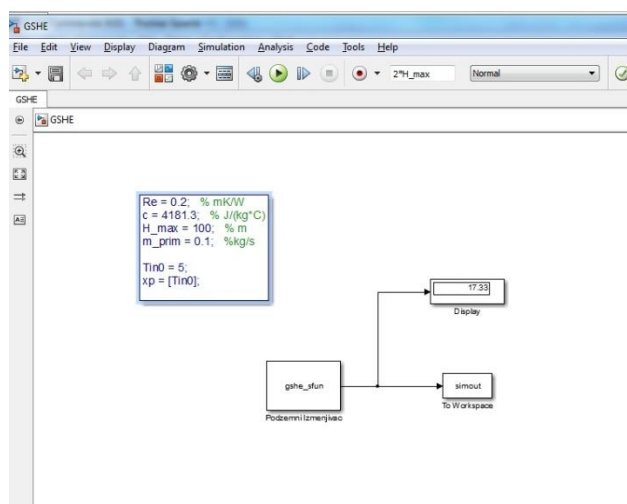
Kako bi se model upotpunio, odnosno dobio malo veći smisao, iskoristiće se jednačina za proračun geotermalnih resursa. Toplotna energija raspoloživih hidrogeotermalnih resursa iz pojedinih bušotina se može izračunati uz pomoć sledeće formule [4]:

$$Q = m' * (T_{out} - T_{in}) * 4181.3 * t \quad (8)$$

Prethodne navede jednačine će biti ključne u realizaciji modela podzemnog izmenjivača geotermalne toplotne pumpe. Cilj našeg modela biće da izračuna temperaturu vode iz izlazne cevi podzemnog izmenjivača, T_{out} , i da se na osnovu tog podatka, kao i neophodnih pratećih podataka, izračuna kolika se ukupna energija dobija od podzemnog izmenjivača.

4. SIMULACIJA I TUMAČENJE REZULTATA

U cilju simulacije iskorišten je komercijalni softverski paket **Matlab** 2013a, sa svojim ugrađenim podsistemom za izvršenje simulacija koji nosi naziv **Simulink**. Simulink je okruženje za modelovanje dizajnirano za simuliranje dinamičkih sistema u višestrukim oblastima. Simulink obuhvata grafički korisnički interfejs koji omogućuje korisniku da vrši modifikaciju modela. Model se predstavlja u formatu blok dijagrama, pri čemu blokovi izvršavaju matematičke funkcije, manipulaciju podacima, ili uslovnim izrazima. Samo Simulink okruženje ima izgled kao na slici 6.



Slika 6: Izgled Simulink okruženja

Pozivanjem „callback“ funkcije Simulink modela parametri se smeštaju u radnu površinu Matlab-a i koriste se za dobijanje rezultata Tout. U konkretnom slučaju vrednosti parametara su sledeći: Re=0.2, c=4181.3, H_max=100, m_prim=0.1, Tin=5, xp=[Tin0].

Izvršavanjem simulacije dobijamo vrednost 17.33, što predstavlja vrednost Tout, odnosno izlaznu temperaturu vode iz podzemnog izmenjivača, što je i cilj ovog modela. Potom analitički izračunavamo ukupnu energiju toplotnog izmenjivača prema jednačini 8, pri čemu uzimamo da je vreme rada geotermalne toplotne pumpe 1h, odnosno 3600s. Izraz ima sledeći oblik:

$$Q = 0.1 \frac{m}{s} * 4181.3 \frac{J}{kgC} * (17.33C - 5C) * 3600s = 18559954.44 J = 5.16 kWh \quad (9)$$

U nastavku je tabelaran prikaz dobijenih vrednosti za različite vrednosti parametara simulacije.

Tabela 1: Vrednosti simulacije za različite parametre

	Re [mK/W]	c [J/kg°C]	H_max [m]	m_prim [kg/s]	Tin0 [C]	Tout [C]	Q [kWh]
1.	0.5	4181.3	98	0.15	5	10.41	3.39
2.	0.8	4200	80	0.2	8	9.56	1.31
3.	1	4500	67	0.5	7.5	7.89	0.87
4.	1.2	4000	50	1	6	6.13	0.52
5.	0.1	3700	20	0.7	6.2	6.79	1.52
6.	0.01	3950	45	0.65	10	13.52	9.03
7.	0.45	4100	30	0.9	10.5	10.52	0.07
8.	0.75	4150	52	0.87	12	12.03	0.11
9.	1.35	4400	75	1.5	11.5	11.56	0.39
10.	1.85	4600	87	1.35	8.7	8.80	0.62

Posmatrajući tabelu 1 u kojoj se nalaze različite vrednosti parametara kao i rezultati modela na te vrednosti, možemo doći do određenih zaključaka. Menjanjem parametara Re direktno utičemo na rezultat modela, i to tako što njegovim povećavanjem dobijamo manju izlaznu temperaturu fluida, Tout, a samim tim i ukupne izlazne energije podzemnog izmenjivača, dok je njegovim smanjenjem obrnuta situacija, povećavaju se Tout i Q. Takođe, iz iste tabele možemo videti da H_max takođe ima značajan uticaj na rezultate modela.

Što je veća dubina bušotine izlazna temperatura je veća, dok je izlazna temperatura manja ukoliko je i sama dubina bušotine manja. Ostali parametri, c i m_prim, imaju uticaj na rezultat samog modela kao i na ukupnu energiju podzemnog izmenjivača, ali uticaj na ukupnu energiju je veći.

5. ZAKLJUČAK

Sistem sa geotermalnom toplotnom pumpom koristi zemlju, podzemne vode, ili površinske vode kao toplotni izvor/odvod pružajući grejanje i hlađenje. Geotermalna toplotna pumpa se prepoznaje kao jedna od izvanrednih tehnologija za grejanje i hlađenje stambenih i poslovnih objekata, jer obezbeđuje visok koeficijent performansi. U okviru rada priložena je teorija korištena za potrebe simulacije podzemnog izmenjivača geotermalne toplotne pumpe.

Dobijeni rezultati opravdavaju samu teoriju i prikazuju realno ponašanje izmenjivača. Jedna od njih je svakako visoka inicijalna investicija, kao i duži vremenski period otplate. Posmatrajući to tako, vremenski period otplate iznosi od 10-15 godina, u zavisnosti od tipa izvedbe. Pojavom novih tehnologija poboljšala bi se situacija smanjenjem samih troškova, što bi izazvalo sve veću primenu geotermalnih toplotnih pumpi.

Sa većim tržištem došlo bi do dodatnog smanjenja troškova, većeg izbora ponude koja uključuje i kvalitetniju opremu čime bi se neposredno uticalo na kvalitetniji odnos prema životnoj sredini, a ujedno i na kvalitetniji život ljudi.

6. LITERATURA

- [1] G Hellström. Ground heat storage: Thermal analysis of duct storage systems. Sweden: Department of Mathematical Physics University of Lund. 1991.
- [2] H.Y.Zeng, N R Diao, and Z H Fang. Efficiency of vertical geothermal heat exchangers in ground source heat pump systems. Thermal Science, 12(1):77-87, 2003.
- [3] Méhes Szabolcs. Optimisation of compressor driven heat pumps for building services, 2011.
- [4] M. Matinović, M. Zlokolica Mandić, Z. Vukićević. Geotermalni altas Vojvodine, 2010.

Podaci za kontakt:

Slobodan Stamenković,
slobodan.stamenkovic88@gmail.com

**WEB APLIKACIJA ZASNOVANA NA B2B MODELU ELEKTRONSKOG POSLOVANJA
ZA POTREBE NABAVKE PROIZVODA I USLUGA****WEB APPLICATION BASED ON B2B MODEL OF ELECTRONIC COMMERCE FOR
PROCUREMENT OF GOODS AND SERVICES**Darko Vasić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je opisan primer web aplikacije zasnovan na B2B modelu elektronskog poslovanja za potrebe nabavke proizvoda i usluga. Dijagrami klase su implementirani korišćenjem Sybase Power Designer alata a aplikacija je razvijena u JAVA programskom jeziku korišćenjem Eclipse SDK razvojno okruženje.

Abstract – This work describes an example of web applications based on the B2B model of e-business for the purpose of procurement of goods and services. Class diagrams are implemented by using Sybase Power Designer tools and applications developed in JAVA programming language and by using the Eclipse SDK development environment..

Ključne reči: *Purchase Order, Request for Quotation, Vendor Management, Spring, Maven, Hibernate*

1. UVOD

Tradicionalno poslovanje zasnovano je na korišćenju strukturiranih papirnih dokumenata u prethodno definisanoj i opšteprihvaćenoj komunikaciji između učesnika u procesu poslovanja. Ovakav vid dokumenata čuvao se najčešće u papirnom obliku.

Problem takvih dokumenata predstavljaju skladištenje, potraga za određenim informacijama, vreme potrebno za pronalazak informacija, troškovi, gubljenje dokumenta, greške u prekućavanju teksta, oštećenja u transportu i sl. Pojavom interneta i njegove sve masovnije upotrebe dolazi do pojave novih tehnologija i novih oblika poslovanja.

Za razliku od klasičnog (tradicionalnog) poslovanja, elektronsko poslovanje predstavlja razmenu standardizovanih elektronskih poruka u obavljanju raznih poslova u kompanijama, bankama upravi i u svim drugim poslovnim transakcijama.

Elektronsko poslovanje (engl. Electronic Business) ili e-Poslovanje (e-Business) jeste vođenje poslova na internetu, što ne podrazumeva samo kupovinu ili prodaju, već organizaciju poslovanja firme u mrežnom okruženju, organizovanje poslovne komunikacije prema klijentima, brigu o klijentima, udruživanje kompanija (integracija) itd.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Gordana Milosavljević.

Izraz elektronsko poslovanje prvi put je upotrebljeno od strane IBM američke kompanije da bi se označilo poslovanje uz značajnu primenu savremenih, elektronskih tehnologija. E-poslovanje ima vrlo široko područje uticaja i primene, pa bi osnovni oblici bili:

- e-trgovina – predstavlja poslovnu komunikaciju i prenos dobara i usluga (kupovina i prodaja)
- e-bankarstvo – predstavlja poslovanje na relaciji banka – klijent
- e-uprava – predstavlja elektronsko poslovanje u upravi ili administraciji (države, preduzeća, itd.)

U ovom radu stavićemo akcenat na e-trgovinu, a prevažno na njen model (oblik) poslovanja B2B (engl. Business-to-business). B2B elektronsko poslovanje odnosi se na korišćenje interneta i web tehnologija za kupovinu, prodaju, jeftiniju, bržu i bolju saradnju poslovnih subjekata. Ono predstavlja poslovanje između dve ili više kompanija, odnosno, razmenu proizvoda, usluga ili informacija koju jedna kompanija obavlja sa jednom ili više drugih kompanija. U ovom modelu uglavnom učesnici predstavljaju dobavljače i distributere.

Zadatak rada predstavlja implementacija web baziranog softverskog rešenja koji se zasniva na B2B modelu elektronskog poslovanja za potrebe nabavki proizvoda i usluga između poslovnih subjekata. Softver treba da realizuje proces elektronske nabavke kroz sledeće module:

- Upravljanje zahtevima za ponudu (Request for quotation)
- Upravljanje nalogima za kupovinu (Purchase order)
- Upravljanje poslovnom saradnjom sa prodavcima (Vendor management)
- Upravljanje katalogima proizvoda ili usluga (Item/Service catalog management)

2. IDENTIFIKACIJA I OPIS PROBLEMA

B2B model elektronskog poslovanja se može podeliti na sledeće celine:

E – prodaja podrazumeva korišćenje web baziranih tehnologija u cilju automatizacije procesa poručivanja.

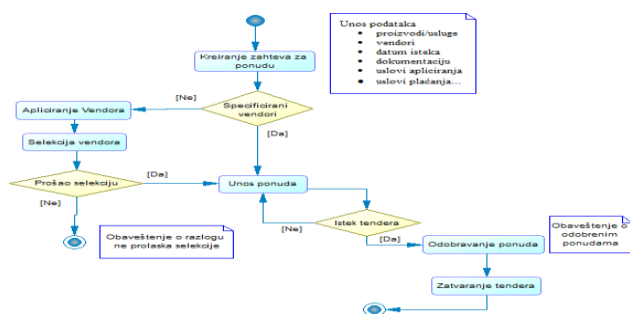
E – kupovina se odnosi na korišćenje interneta kao kanala za prodaju proizvoda ili usluga.

Mehanizmi za podršku – odnose se na korišćenje interneta za saradnju sa poslovnim partnerima u smislu širem od kupovine i prodaje.

Elektronska nabavka ili e-Nabavka predstavlja B2B model kupovine i prodaje materijala, rada ili usluga putem interneta, kao i drugih informacija i umrežavanje sistema kao što su elektronske razmene podataka i planiranje resursa.

2.1 Zahtev za ponudu – Request For Quotation (RFQ)

Zahtev za ponudu opisuje standardni poslovni proces koji za cilj ima da pozove prodavce u procesu nadmetanja na tenderu za specifične proizvode i/ili usluge. Pored cena za određene proizvode/usluge uključuje i informacije poput dodatnih uslova, uslova plaćanja, taksu, datum do kada prodavci moraju da pošalju svoje ponude, prateću dokumentaciju. Poslovni proces zahteva za ponudu prikazan je na slici 1.

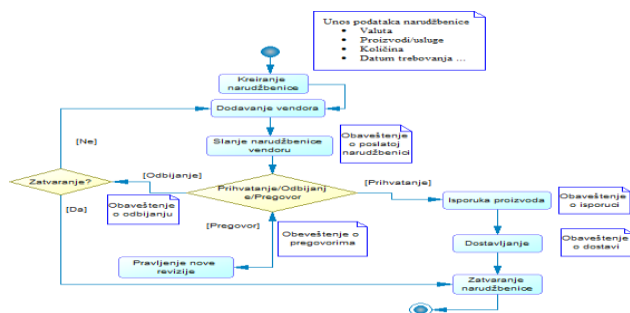


Slika 1. Poslovni tok procesa zahteva za ponudu

2.2. Nalog za kupovinu – Purchase Order (PO)

Nalog za kupovinu je komercijalni dokument izdat od strane kupca prodavcu, koji ukazuje na vrstu, količinu i cenu proizvoda ili usluga koje će prodavac pružiti kupcu. Slanje narudžbine dobavljaču predstavlja zvaničnu ponudu za kupovinu proizvoda ili usluga. Prihvatanje narudžbine od strane prodavca obično stvara jednokratni ugovor između kupca i prodavca. Postoji nekoliko razloga zašto kompanije koriste narudžbine.

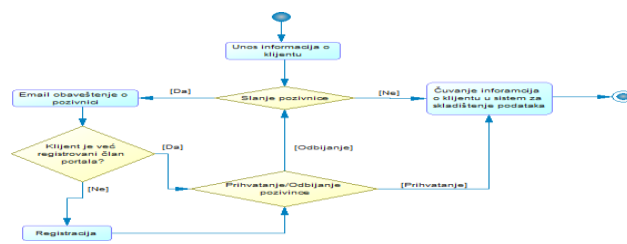
One omogućuju kupcima da jasno izraze svoje namere ka prodavcu, a prodavci su zaštićeni u slučaju odbijanja kupca da plati za robu ili usluge. One takođe pomažu pri organizovanju nabavke za privredno društvo, unapređuju proces kupovine tako što ga uvide u standardnu proceduru. Dobijanje finansijske pomoći od komercijalnih kreditora ili finansijskih institucija je još jedan razlog za korišćenje narudžbine. Na slici 2. prikazan je tok procesa narudžbine.



Slika 2. Poslovni tok procesa narudžbine

2.3 Upravljanje poslovnim saradnjom sa prodavcima – Vendor management

Ova funkcionalnost omogućava da se svaka interakcija sa prodavcima evidentira tako da se efikasnost komunikacije i saradnja sa klijentom povećava, a unutar organizacije kupca obezbeđena je mogućnost praćenja događaja sa prodavcima pojedinačno. Zbog strukture sistema koja podrazumeva da svi korisnici koji koriste aplikaciju moraju biti registrovani tj. da moraju posedovati svoj profil, a zarad olakšanog održavanja kontakata i procesa poslovanja, treba uvesti pozivnice. Pozivnice predstavljaju digitalni format tradicionalnih pozivnica koje za cilj imaju da određenu kompaniju pozovu na saradnju sa kompanijom koja pozivnicu pravi što ilustruje slika 3. Pozivnica će u sebi sadržati linkove koje vode ka registraciji na portalu ili linkove za proces prihvatanja poslovne saradnje sa kompanijom koja je pozivnicu poslala.



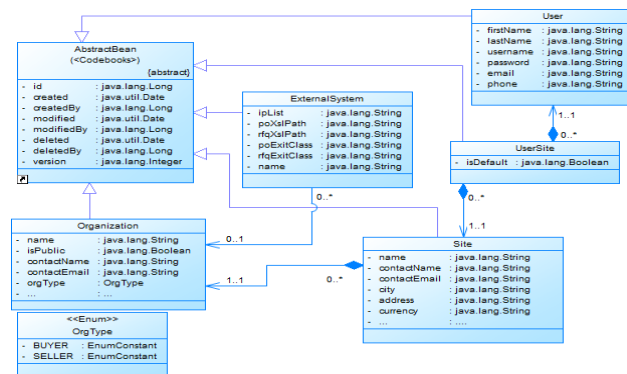
Slika 3. Tok stvaranja poslovne saradnje između prodavca i kupca

3. ARHITEKTURA APLIKACIJE

U ovom poglavlju detaljnije ćemo se baviti arhitekturom sistema, strukturom i međusobnom zavisnošću komponenta sistema. Koristeći alat za kreiranje UML dijagrama Sybase Power Designer biće predstavljeni modeli klasa i njihove međusobne zavisnosti.

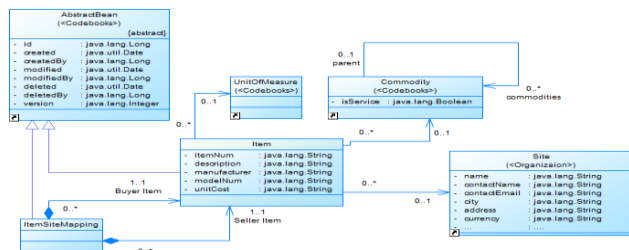
3.1 Organizaciona struktura kompanija

Velike kompanije imaju svoju hijerarhijsku strukturu, počev od samog vrha organizacije, preko svojih ispostava (filijala) do zaposlenih. Ovo rešenje se oslanja na model organizacione strukture prema teritoriji, koja definiše organizaciju kompanija koje su stacionirane na različitim lokacijama širom sveta. To podrazumeva da imamo jednu glavnu kompaniju koja vrši poslovanje na različitim lokacijama, kako unutar zemlje gde je osnovana, tako i van nje.



Slika 4. Dijagram klasa organizacione strukture kompanije

3.2 Proizvodi i usluge



Slika 5. Dijagram klasa proizvoda i usluga

Registrovane kompanije mogu da definišu proizvode ili usluge koje potražuju (ako su u pitanju kupci) ili koje pružaju (ako su u pitanju prodavci). Glavni akcenat ima klasa **Item** koja opisuje proizvod ili uslugu (Slika 5). Pored svojih atributa sadrži i jedinicu mere, artikal/usluga i kompaniju za koju je vezana.

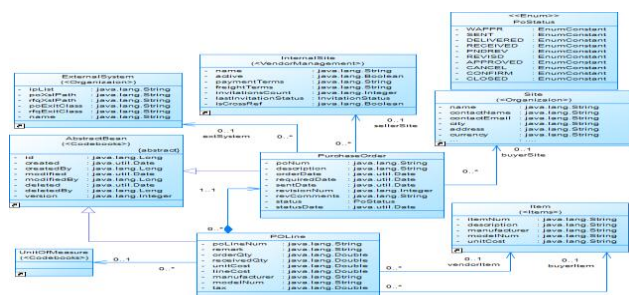
Svaka organizacija održava svoj katalog proizvoda ili usluga. Isti proizvod ili usluga mogu postojati za obe strane, ali tako da ih prodavac zavodi pod jednim nazivom a kupac pod drugim. Radi lakšeg sporazumevanja, omogućeno im je da povezuju svoje proizvode.

Klasa **ItemSiteMapping** upravo to realizuje. Povezivanjem instance klase **Item**, kupac i prodavac specificiraju da se radi o potpuno istom proizvodu samo što su drugačije sačuvani i obeleženi u sistemu.

3.3 Nalog za kupovinu - Purchase Order

Sa slike 6. može se uočiti klasa **PurchaseOrder** (u daljem tekstu narudžba) opisuje osnovne podatke o narudžbini. Sadrži podatke o kompaniji kupca (veza `buyerSite`), podatke o kompaniji prodavca (veza `sellerSite`), podatke o eksternom sistemu (veza `extSystem`), proizvoljan broj linija (POLine).

Linija opisuje potraživanja za jedan proizvod ili uslugu. Sadrži jedinicu mere (**UnitOfMeasure**), proizvod ili uslugu (**Item**) koju je specificirao kupac i isti takav **Item** samo od strane prodavca ukoliko je kod njega drugačije zaveden. Enumeracija **PoStatus** opisuje statuse kroz koje PO prolazi.



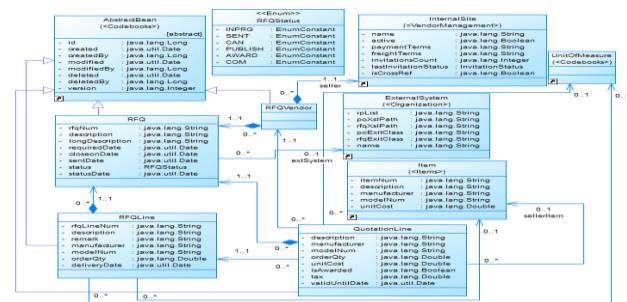
Slika 6. Dijagram klasa za obradu narudžbe (purchase order)

3.4 Zahtev za ponudu - Request For Quotation

Na slici 7 je prikazan dijagram klasa zahteva za ponudu (u daljem tekstu RFQ). RFQ se sastoji od linija/stavki (RFQLine) koje, slično kao kod narudžbe, predstavljaju ono što kupac zahteva. To su proizvodi ili usluge, vendore koje kupac specificira ili dopušta da se zahtev radi po

principu javnog tendera ne unoseći nijednog vendora. Time ostavlja mogućnost vendorima da licitiraju ukoliko su zainteresovani.

Ponude (**QuotationLine**) unose vendori tako što unose ponudenu cenu stavke, količinu, jedinicu mere itd. Svaka ponuda vezana je za jednu stavku, odnosno jedan vendor unosi najviše jednu ponudu po stavci. RFQVendor klasa predstavlja vendore koji imaju prava da vrše ponude.



Slika 7. Dijagram klasa RFQ-a

4. PRIKAZ IMPLEMENTIRANOG REŠENJA

Izborom kroz glavni menu **Management->Vendor Management** otvoriće se sledeća forma (Slika 8) sa listom kompanija koje je kupac uneo kao svoje interne poslovne partnere.

The screenshot shows the **VENDOR MANAGEMENT** form. It has a search bar with fields for `Vendor ID`, `Vendor name`, `Active?`, and `Is cross-reference?`. Below the search bar is a table with columns: `Vendor ID`, `Vendor name`, `Postal Site`, `City`, `State`, `ZIP/Postal Code`, `Is cross-reference?`, `Last Invite Date`, `Last Invite Status`, `Active?`, and `Actions`. The table shows one vendor: **DELL** with `DELL Shop` as the postal site, `New Sat Shop` as the city, `No` as the state, `23 10 2016` as the last invite date, `SENT` as the last invite status, `Yes` as active, and `DEL` as actions.

Slika 8. Lista internih kompanija kupca

Kroz ovaj modul korisniku se omogućava da unosi, modifikuje, šalje pozivnice ili briše stare.

Pošto nalog za kupovinu zahteva veliki broj podataka i komplikovane procese, bilo je potrebno napraviti što jednostavniju formu kako bi se krajnjim korisnicima olakšao rad. Vodeći se tom logikom potrebno je projektovati što jednostavniji korisnički interfejs koji bi zahtevao minimalan broj koraka koje korisnik mora da napravi.

Radi smanjivanje broja koraka, neka polja će biti automatski popunjena u zavisnosti od konteksta, odnosno u zavisnosti od izbora nekih drugih podataka koji sa sobom povlače informacije koje zahteva narudžba. Takođe, bilo je potrebno razdvojiti poglede na narudžbe, pa tako imamo `buyer view` i `seller view`.

Ovi pogledi se prikazuju u zavisnosti od ulogovane organizacije. Svaki pogled sadrži drugačije, prilagođene informacije.

Slika 9. Forma za unos narudžbe, buyer view

Zahtevi za ponudu su nešto složenije strukture od narudžbi. Na zahtev za ponudu možemo gledati kao predkorak za pravljenje naloga za kupovinu. Vendori će licitirati svoje ponude dok će kupac odabrati najpovoljnije.

Kada se ta licitacija završi, logično je da se vendorima koji su pobedili na licitaciji pravi nalog za kupovinu sa stavkama licitiranih ponuda.

Slika 10. prikazuje formu za unos tendera. Na dijagramu klasa je već pomenuto da se RFQ sastoji od stavki koje unosi kupac i ponuda koje unose vendor.

Slika 10. Forma za unos zahteva za ponudu

Kao PO i RFQ prolazi kroz određena stanja koja se mogu menjati na panelu sa mogućim opcijama. Na slici 10 možemo primetiti i “SEND TO VENDORS” i “PUBLISH”.

Ukoliko je kupac odabrao “PUBLISH” sve njegove interne kompanije imaju pravo da apliciraju na ovaj tender.

Oni će dobiti istu formu ali sa opcijom “Apply For Bid”. Od tog trenutka oni postaju deo zahteva za ponudu. Sledeći korak jeste da kada se završi proces apliciranja, kupac pošalje RFQ natrag vendorima kako bi započeli proces unošenja ponuda.

Kada se tender završi, kupac bira najpovoljnije ponude označavanjem svake ponaosob i menja RFQ u stanje AWARDED.

Tom aktivnošću vendori će biti obavešteni putem email adrese da su njihove ponude prihvaćene.

5. ZAKLJUČAK

Ovaj rad opisuje implementaciju web baziranog rešenja zasnovanog na B2B modelu elektronskog poslovanja za potrebe nabavki proizvoda i usluga. U radu su detaljno specificirani objekti koji se koriste za potrebe aplikacije. Opisani su elementi dijagrami klasa koji predstavljaju objekte poslovanja, njihove međusobne veze kao i njihova komunikacija.

6. LITERATURA

- [1] Material Design Lite
<https://getmdl.io/components/index.html>
- [2] Enciklopedija otvorenog sadržaja <https://sh.wikipedia.org/>
- [3] Spring Framework MVC
<http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>
- [4] Java Web Services
<https://docs.oracle.com/javaee/6/tutorial/doc/bnayn.html#bnayp>
- [5] RequireJS - <http://requirejs.org/docs/api.html>

Kratka biografija

Darko Vasić je rođen u Ljuboviji, 1990. god. Fakultet tehničkih nauka upisao je 2009. godine, odsek Računarstvo i Automatika. Osnovne akademske studije završio je 2013.

ANALIZA TROFAZNOG DIREKTNOG MATRIČNOG PRETVARAČA

ANALYSIS OF THREE-PHASE DIRECT MATRIX CONVERTER

Nenad Milenković, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad opisuje model matričnog pretvarača, direktnog AC/AC pretvarača učestalosti. Glavne teme koje se obrađuju u radu su karakteristike, topologija, princip rada, prednosti, nedostaci, metode upravljanja i modulacije, kao i mogućnosti primene ovog tipa pretvarača u industrijskim aplikacijama.

Abstract – This paper describes the matrix converters as direct AC/AC frequency converters. The main subjects are characteristics, topological structure, working principle, advantages and disadvantages, as well as potential for implementation of this type of converter in industrial applications.

Ključne reči: Matrični pretvarači, AC/AC pretvarači, energetska elektronika

1. UVOD

U moderno doba evidentna je stalno rastuća potreba za povećanom kontrolom i efikasnošću u svim oblastima energetike, pa tako i u oblasti električnih pogona. U prošlosti, kontrolisani električni pogoni su u najvećem broju bili izvođeni pomoću mehaničkih rotirajućih konvertora koji predstavljaju sklop od nekoliko spregnutih električnih mašina, kao što je primer dobro poznate Ward-Leonardova grupa. Danas se isključivo koriste statički konvertori energetske elektronike čija se implementacija zasniva na poluprovodničkoj prekidačima.

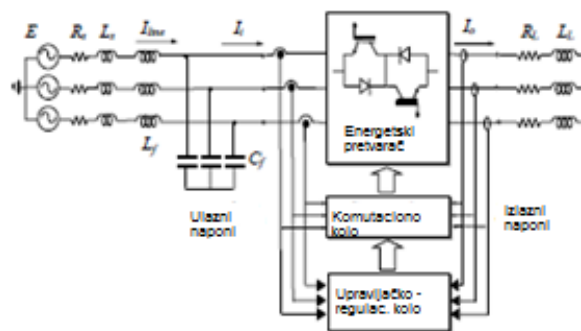
Matrični pretvarači ili konvertori predstavljaju direktne AC/AC pretvarače učestalosti čiju glavnu komponentu predstavlja skup od $m \times n$ bidirekcionalnih (bilateralnih) prekidača (m – broj ulaznih faza, n – broj izlaznih faza) koji imaju sposobnosti da provode i blokiraju struje i napona oba polariteta, zavisno od kontrolnih signala koje dobijaju. Zbog svoje direktne prirode, odnosno nedostatka DC kola sa skladištenjem energije, ovi pretvarači su pogodni za sisteme gde postojanje velikih pasivnih komponenti predstavlja problem, odnosno u slučajevima gde je jednostavno ekonomski isplativije imati kompletno poluprovodničko rešenje u poređenju sa konvencionalnim pretvaračima učestalosti. Nepostojanje potrebe za elektrolitskim kondenzatorima omogućava i duži radni vek, ali može predstavljati i ograničenje za pojedine mogućnosti pretvarača, kao i poteškoće pri radu u neidealnim uslovima u sistemu. Ipak, dizajn matričnih pretvarača je kompaktniji i mogu raditi sa boljom efikasnošću.

NAPOMENA:

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

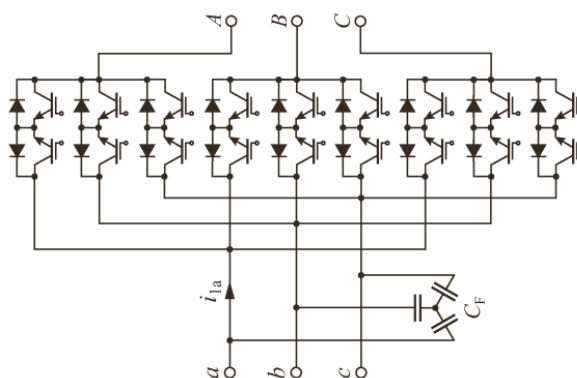
2. TOPOLOGIJA MATRIČNIH PRETVARAČA

Postoji nekoliko različitih topologija koje koriste bidirekcionalne prekidače i nemaju elemenata za skladištenje energije, a spadaju u matrične pretvarače u širem smislu. U ovom radu razmatra se trofazna varijanta matričnog pretvarača, odnosno trofazni direktni matrični konvertor, koji ima trofazni ulaz i izlaz, a sastoji se od 9 bidirekcionalnih prekidača, koji mogu svaku fazu ulaza povezati sa svakom fazom na izlazu. Ulazni terminali konvertora su priključeni na trofazni izvor, najčešće na mrežu, dok su izlazni terminali priključeni na strujno napajan potrošač (asinhroni motor) [1,2]. Na slici 1 prikazana je opšta šema jednog ovakvog pretvarača. Kapacitivni filter i induktivni filter predstavljeni na slici su suštinski neophodni delovi i njihova veličina je obrnuto srazmerna prekidačkoj frekvenciji konvertora.

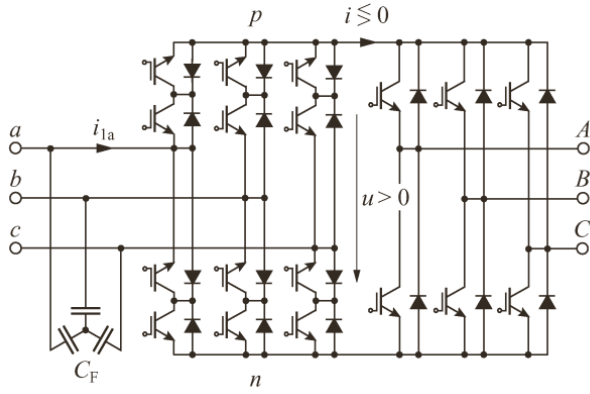


Slika 1. Opšta šema matričnog pretvarača.

Trofazni matrični pretvarači mogu biti u direktnoj i indirektnoj topologiji. Na slici 2 prikazana je direktna topologija, a na slici 3 primer indirektna. Konfiguracija indirektnog matričnog pretvarača predstavlja evoluciju topologije „back-to-back“ pretvarača, gde je ispravljač izveden sa bidirekcionalnim prekidačima, ali nema elementa za skladištenje energije u DC međukolu.



Slika 2. Topologija direktnog matričnog pretvarača.



Slika 3. Topologija indirektnog matričnog pretvarača.

Topologije direktnog i indirektnog matričnog pretvarača su ekvivalentne u smislu osnovnih funkcionalnosti. Njihova različitost u fizičkoj implementaciji odnosi se gotovo isključivo na izbor metode upravljanja prekidačima i drugačiju šemu komutacije [3], zbog čega se model indirektnog matričnog pretvarača neće posebno obrađivati u ovom radu.

3. PRINCIP RADA

Kako kod matričnog pretvarača ne postoje komponente za skladištenje energije, izlazni naponi moraju direktno biti generisani od ulaznih napona. Svaki od faznih izlaznih napona dobija se sekvencijalnim odabirom delova naponskog signala na ulazu. Učestalost odabiranja mora biti znatno veća, barem za red veličine, od frekvencije ulaznih i izlaznih signala, a dužina svakog odabira mora biti kontrolisana na način da srednja vrednost izlaznog naponskog signala prati željeni talasni oblik, kako bi generisani signali imali što manju distorziju. Posledica direktne prirode konvertora uslovljava da u svakom trenutku izlazni naponi treba da se uklape u talasne oblike napona na ulazu. Zbog ovog ograničenja maksimalna vrednost izlaznog napona koju matrični pretvarač može da generiše bez ulaska u oblast premodulacije iznosi $\sqrt{3}/2 \approx 0.87$ maksimalne vrednosti ulaznog napona [1]. Na sličan način su ulazne struje diktirane strujama opterećenja, odnosno sekvencijalnim odabirom delova strujnih signala na izlazu. Ukoliko se prekidačka učestanost pretvarača podesi na vrednost mnogo veću od frekvencije signala na ulazu i izlazu, dobija se da su signali struja u pretvaraču praktično idealnog sinusnog oblika (spektar se sastoji od osnovnog harmonika i harmonika bliskih prekidačkoj frekvenciji) [3]. Na slici 4 predstavljeni su karakteristični talasni oblici napona i struja na ulazu (slika 4.a) i izlazu (slika 4.b).

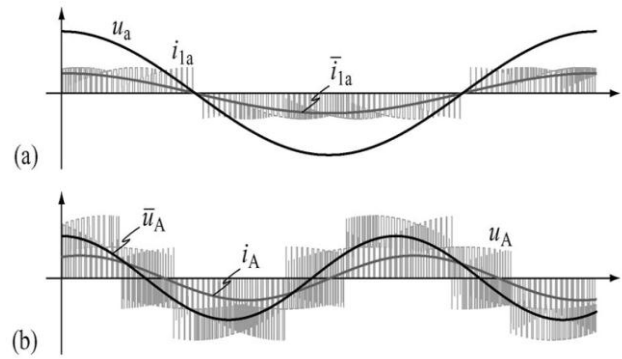
Na slici 5 dat je šematski prikaz matričnog pretvarača sa 9 bidirekcionalnih prekidača. Uzimajući u obzir da svaki od njih može biti u uključenom ili isključenom stanju, matrični pretvarač može teoretski biti u 2^9 odnosno 512 različitih stanja. Kako se uzima da se matrični pretvarač napaja sa naponskog izvora i opterećen je induktivnim potrošačem, nezavisno od izabrane metode upravljanja, dva osnovna pravila važe. Prvo, da se moraju izuzeti sva stanja pretvarača u kojima su faze izvora kratko spojene. Drugo, s obzirom na induktivnu prirodu potrošača, izlazne struje treba da budu kontinualne, tj. nikada ne bi smele biti prekinute, da ne bi došlo do indukovanja

velikog napona koji bi uništio prekidačku komponentu [1]. Uz primenu ovih pravila, ukupni broj mogućih stanja se svodi na 27 dozvoljenih stanja. Kao rezultat prekidači za svaku izlaznu fazu moraju biti kontrolisani prema izrazu:

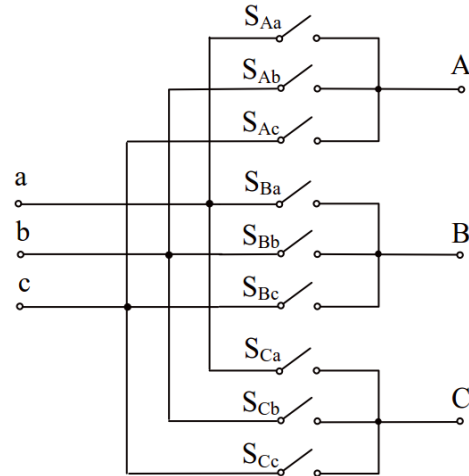
$$S_{ja} + S_{jb} + S_{jc} = 1, \quad j \in \{A, B, C\}, \quad (1)$$

gde je S_{jk} prekidačka funkcija bidirekcionalnog prekidača definisana kao:

$$S_{jk} = \begin{cases} 1 & S_{jk} \text{ zatvoren} \\ 0 & S_{jk} \text{ otvoren} \end{cases} \quad j \in \{A, B, C\}, k \in \{a, b, c\}. \quad (2)$$



Slika 4. Karakteristični talasni oblici veličina od interesa matričnog pretvarača (a) ulazni napon u_a , ulazna struja i_{1a} i njena srednja vrednost $\bar{i}_{1a}$ u toku perioda odabiranja T_p . (b) Izlazni napon u_A i njegova srednja vrednost $\bar{u}_A$ u toku perioda odabiranja T_p , izlazna struja i_A [3]



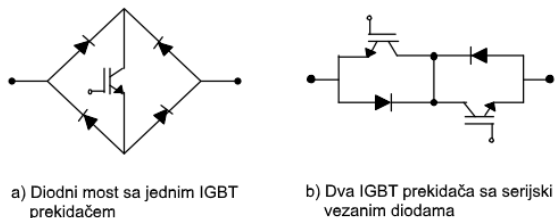
Slika 5. Šematski prikaz trofaznog matričnog pretvarača.

4. IMPLEMENTACIJA

Prvi ključni problem u implementaciji matričnog pretvarača predstavlja realizacija bidirekcionalnih prekidača. Trenutno, bidirekcionalni prekidač kao jedinstvena komponenta ne postoji na tržištu već se on mora realizovati kao kombinacija konvencionalnih, unidirekcionalnih prekidača. Neke od najčešće korišćenih konfiguracija su prikazane na slici 6. Drugi problem je

problem komutacije bidirekcionalnih prekidača. Kod komutacije sa jedne ulazne faze na drugu, dva pravila moraju biti ispoštovana kako prekidači ne bi bili ugroženi prekomernim intenzitetima struja ili napona [2]:

- 1) komutacija ne sme da izazove kratak spoj između dve ulazne faze,
- 2) komutacija ne sme da izazove prekid induktivnih izlaznih struja.

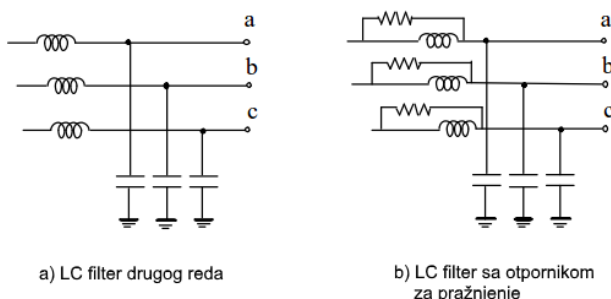


Slika 6. Neke od mogućih implementacija bidirekcionalnog prekidača [1]

Da bi bilo moguće ispuniti ova pravila, moraju se poznavati uslovi u kolu, a napon između prekidača u komutaciji ili izlaznu struju neophodno je meriti. Na osnovu merenja moguće je odrediti odgovarajuću sekvencu kombinacija prekidačkih stanja koja neće dovesti do kratkog spoja ili prekida izlaznih struja.

Iako se matrični pretvarači, zbog nedostatka elektrolitskih kondenzatora za skladištenje energije često pominju kao potpuno poluprovodničko rešenje, oni ipak zahtevaju određeni broj reaktivnih komponenti koje imaju ulogu filtera. Osnovni zadatak ovih filtera jeste da spreče značajnije oscilacije ulaznog napona konvertora u toku svakog PWM ciklusa, i takođe da spreče injektovanje neželjenih harmonika struje u izvor. Uzimaju se LC filteri drugog reda (slika 7), gde kondenzatori obezbeđuju stabilnost napona i (delimično) kompenzaciju, a prigušnice smanjuju nivo harmonika. Konfiguracija ovih filtera može biti različita i može zavisiti od različitih kriterijuma, a prvenstveno od prekidačke učestalosti i odabira strategije upravljanja [2]. Ulazni filteri statičkih pretvarača, nezavisno od njihove konfiguracije, moraju da ispunjavaju navedene zahteve:

- 1) eliminacija harmonika oko prekidačke učestalosti;
- 2) fazni pomeraj koji unosi filter mora biti minimalan;
- 3) stabilnost celokupnog sistema ne sme biti ugrožena.

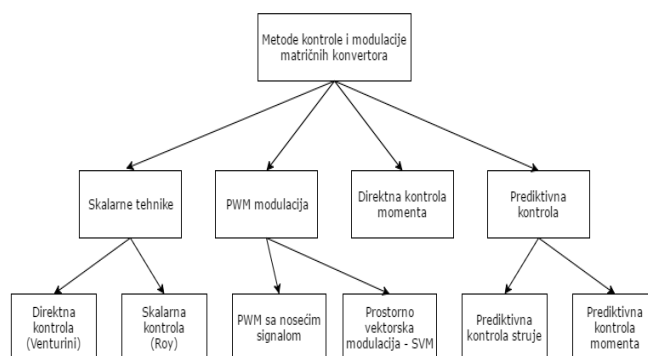


Slika 7. Vrste ulaznih filtera matričnog pretvarača [2]

5. METODE UPRAVLJANJA I MODULACIJE

Pregled i sistematizacija relevantnih metoda kontrole i modulacije razvijenih do sada za matrične pretvarače data

je na slici 8. Prva metoda sa slike (donji red) predstavlja poznatu Venturini metodu, odnosno metodu funkcije direktnog prenosa. U ovoj metodi se izlazni napon dobija kao proizvod ulaznog napona i prenosne matrice koja predstavlja konvertor. Druga metoda jeste skalarna kontrola koju je razvio Roy, koja koristi trenutni odnos napona ulaznih faza kako bi odabrala aktivna i nulta stanja prekidača konvertora. Treća metoda, odnosno PWM modulacija je tehnika prethodno razvijena i korišćena kod naponskih invertora. Najjednostavniji oblik ove metode je sa PWM sa nosećim signalom, dok malo elegantnije i pouzdanije rešenje predstavlja primena prostorno vektorske modulacije SVM. Trenutno najkorišćeniju metodu upravljanja i modulacije predstavlja SVM metoda zbog efikasnosti i mogućnosti koje se mogu ostvariti primenom ove metode. U novije vreme predstavljeno je i par modernih tehnika, kao što su prediktivna kontrola struje i momenta.



Slika 8. Pregled metoda kontrole i modulacije matričnih konvertora.

6. REZULTATI SIMULACIJE

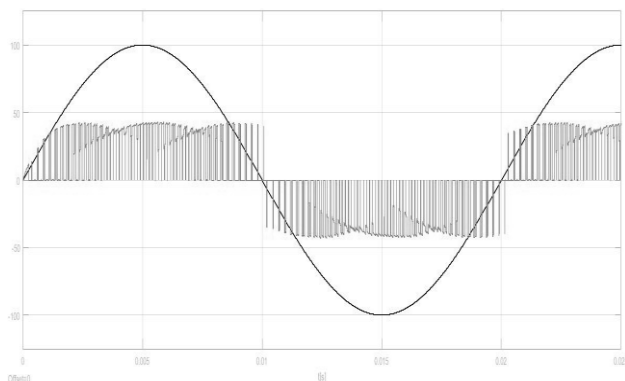
U ovom poglavlju dati su rezultati simulacije matričnog pretvarača, koji napaja induktivno opterećenje sa idealnim faktorom snage na ulazu. Matrični pretvarač modelovan je sa 9 bidirekcionalnih prekidača, realizovanih pomoću IGBT prekidača u konfiguraciji sa slike 2. Sistem se napaja sa idealnog trofaznog izvora frekvencije 50Hz. Upravljanje prekidačima zasniva se na prostorno vektorskoj modulaciji koja dozvoljava direktnu kontrolu izlaznih napona i ulaznih struja, čime je omogućena i kontrola ulaznog faktora snage. Model ne uključuje ulazne filtere zbog isticanja prirode signala generisanih samim konvertorom. Parametri induktivnog potrošača iznose 1Ω i 20mH , amplituda izlaznog napona iznosi 100V , dok je prekidačka frekvencija 4kHz . Na slici 9. i 10. prikazani su signali napona i struje po fazi na ulazu i izlazu matričnog pretvarača, a referentni signali su podešeni tako im je učestalost ista kao na ulazu, 50Hz . Sa slike 9 se može primetiti diskontinualna priroda signala struje na ulazu, što je posledica nedostatka ulaznog filtera, kao i idealni faktor snage.

7. PRIMENA MATRIČNIH PRETVARAČA

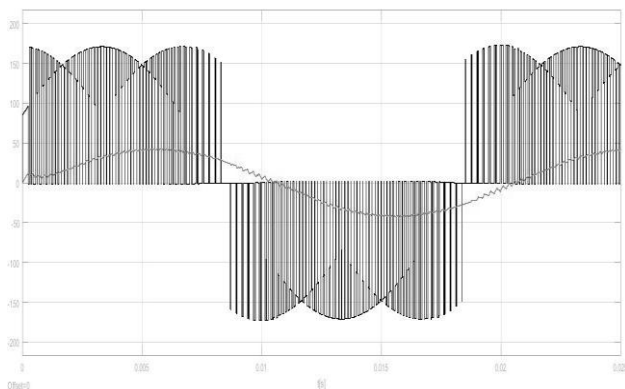
Matrični pretvarač predstavlja koncept koji teži minimalnoj internoj potrebi za skladištenjem energije i ova njegova osobina predstavlja ključni kriterijum primene matričnog pretvarača u električnim pogonima. Za AC/AC aplikacije koje zahtevaju određeni vid

skladištenja energije zbog intenzivne dinamike opterećenja, matrični pretvarač ne predstavlja optimalno rešenje. Idealni sistemi za ovaj konvertor predstavljaju manje dinamične sisteme koji većinu vremena rade sa 30% do 100% nominalnog opterećenja. S obzirom na bolje performanse matričnog pretvarača sa povećanjem prekidačke učestalosti u odnosu na „back-to-back“ konvertore, idealni sistemi u kojima se primenjuju matrični pretvarači bili bi kompresori, ventilatori, mikseri, pumpe i sistemi dizalica i liftova [3].

Poslednjih tridesetak godina matrični pretvarači se smatraju uređajima za koje se očekuje da nađu široku primenu u raznim industrijskim aplikacijama. Međutim, i pored intenzivnog rada na njihovom istraživanju i do sada postignutih rezultata, matrični pretvarači nisu uspeali značajnije da se probiju na tržištu. Trenutno jedini proizvođač komercijanih modela matričnih pretvarača je japanska kompanija „Yaskawa“, koja u ponudi ima nekoliko modela za različite raspone snaga. Pojavljuju se i drugi, manji proizvođači, koji koristeći nove vrste snažnih poluprovodničkih prekidača zasnovanih na SiC (Silicon Carbid) tehnologiji, traže uspešna tržišna rešenja. Oblasti primene matričnih pretvarača su i dalje tema istraživanja, a s obzirom na napredak tehnologije poslednjih godina raste i mogućnost primene matričnih konvertora u različitim aplikacijama, u koje spadaju i primene u posebno teškim okruženjima, poput podmornica i aviona. Takođe, matrični pretvarači se pokazuju kao pouzdana rešenja u primeni u pogonima vetrogeneratora.



Slika 9. Fazni signali ulaznog napona i struje matričnog pretvarača.



Slika 10. Fazni signali izlaznog napona i struje matričnog pretvarača.

8. ZAKLJUČAK

Mogućnosti primene matričnih pretvarača tema su mnogih istraživanja u proteklih tridesetak godina. Topologija ovih pretvarača pruža nekoliko prednosti u odnosu na konvencionalne kao što su sinusoidalni signali na ulazu i izlazu sa minimalnim harmonijskim izobličenjima i inherentni bidirekcionalni tok energije. U radu su predstavljene tehnologije direktnog matričnog pretvarača sa osvrtom na osnovne karakteristike, kao i mogućnosti primene, a dati su i problemi koji postoje u konkretnoj implementaciji. Obzirom na intenzivne aktivnosti i rezultate istraživanja strategija komutacije i algoritama upravljanja, masovna serijska proizvodnja i primena matričnih pretvarača u industriji se ubrzano približava, jer se za praktično većinu tradicionalnih problema ovih pretvarača uspešno pronaći dovoljno pouzdano, efikasno i ekonomski isplativo rešenje.

Ipak, najčešće korišćeni bidirekcionalni AC/AC pretvarač u industrijskim aplikacijama niskog napona i dalje ostaje konfiguracija poznata kao „back-to-back“ pretvarač (BBC). Oni koji se zalažu za propagiranje matričnog pretvarača ističu da direktni AC/AC pretvarači snage bez posrednih elemenata za skladištenje energije, odnosno DC međukola, predstavljaju kompaktnije rešenje, koje takođe doprinosi i dugovečnosti sistema, jer nema brzo potrošnih elemenata, pre svega elektrolitskih kondenzatora. Nasuprot njima, kritičari tvrde da matrični pretvarači ne pružaju znatno poboljšanje u odnosu na konvencionalna rešenja pre svega zbog svojih ograničenja, kao što su lošiji prenosni odnos i kompleksnost upravljanja, ne opravdavajući zbog toga komplikovaniju implementaciju matričnih pretvarača u odnosu na „back-to-back“ pretvarače [3].

9. LITERATURA

- [1] M. Matteini, „Control Techniques for Matrix Converter Adjustable Speed Drives“, Ph.D. Thesis, University of Bologna, Bologna, 2001.
- [2] P. Wheeler, J. Rodriguez, J. Clare, L. Empringham, A. Weinstein, „Matrix Converters : A Technology Review“, IEEE Trans. on Industrial Electronics, Vol.49, No.2, Sep.2002, pp.276-288.
- [3] T. Freidli, J.W. Kolar, J.Rodriguez, P. Wheeler, „Comparative Evaluation of Three-Phase AC–AC Matrix Converter and Voltage DC-Link Back-to-Back Converter System“, IEEE Trans. on Industrial Electronics, Vol.59, No.12, Dec.2012, pp.4487-4510.

Kratka biografija:



Nenad Milenković rođen je u Sremskoj Mitrovici 1989. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2016.god.



Vladimir Katić rođen je u Novom Sadu 1954. god. Doktorirao je na Univerzitetu Beogradu 1991. god. Od 2002. god. je redovni profesor Univerziteta u Novom Sadu. Oblasti interesovanja su energetska elektronika, obnovljivi izvori električne energije, električna vozila i kvalitet električne energije.

VIZUELNA KATEGORIZACIJA UZ POMOĆ KLJUČNIH TAČAKA VISUAL CATEGORISATION WITH BAG OF KEYPOINTS

Lazar Tatomirov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu predstavljamo rešenje problema prepoznavanja "pešaka" u video zapisu ili uživo putem kamere kao ulaznog uređaja. Prepoznati pešaci treba da budu uokvireni pravougaonikom kao rezultatna izlazna informacija prema korisniku.

Grubi plan koji dovodi do rešenja jeste da napravimo detektor objekta u slici, koji ćemo kasnije koristiti u svakom frejmu ulaznog video zapisa i to ne nad čitavim frejmom, nego jednim njegovim "prozorom".

Abstract – In this paper, we present the solution to the problem of recognizing "pedestrians" in video or live via the camera as an input stream. Identified pedestrians should be framed with drawn rectangle as resultant output information to the user. A rough plan that leads to a solution is to make the detector of objects in the image, which we'll use later in each frame of the input video clip and do not use it over entire frame, but one of its "windows".

Ključne reči: Prepoznavanja pešaka, Detektor objekta u slici

1. UVOD

Obrada slike je oblast koja izučava različite matematičke operacije nad digitalnim slikama kako bi se dobila unapređena slika ili se izvukla neka korisna informacija iz nje. To je oblik obrade signala koji za ulaz uzima sliku ili video zapis. Izlaz ili output obrade slike može biti slika ili skup karakteristika ili parametara koji se odnose na sliku. Većina tehnika za obradu slike uključuju tretiranje slike kao dvodimenzionalni signal i primenjuju se standardne tehnike za obradu signala. Slike se takođe obrađuju kao trodimenzionalni signali gde je treće dimenzija vreme ili z-osa. Kompjuterska vizija je interdisciplinarno polje koje se bavi obradom digitalne slike pomoću računara. Iz perspektive inženjera traži da automatizuje zadatke koje ljudski vizuelni sistem može.

Ovaj rad će se baviti užim poljem kompjuterske vizije pod nazivom "prepoznavanje objekata". Prepoznavanje objekata je tehnologija za pronalaženje i identifikovanje objekata u slici ili video sekvenci. Ljudi prepoznaju mnoštvo objekata u slici sa malo truda, uprkos činjenici da slika sa objektima može donekle da varira sa različitim uglovima posmatranja, u mnoštvu veličina čak i rotacijama. Ovaj zadatak je još uvek izazov za sisteme kompjuterske vizije.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Trpovski, vanr.prof.

Različiti pristupi su primenjivani već nekoliko decenija u dve kategorije :

Metode Pojavljivanja - objekti izgledaju drugačije pod različitim uslovima kao što su promene u osvetljenju ili boji, promene u pravcu posmatranja, promene u veličini ili obliku.

Metode bazirane na karakteristikama se trude da pronađu preklapanja između karakteristika objekata i karakteristika slike. Primarno ograničenje je da jedan položaj objekta mora da objasni sva izvodljiva preklapanja. Ovo su metode koje izvoze karakteristike iz objekata koje treba da bude prepoznat i slike koja se pretražuje. Tu su bitni uglovi, linearne ivice, površni pečevi (zakrpe).

U ovom radu razrađena je metoda prepoznavanja objekata u slici pod nazivom "vizuelna kategorizacija pomoću mnoštva ključnih tačaka". Ova metoda je bazirana na vektorima kvantizacije koji opisuju ne-varirajuće deskriptore ključnih tačaka u slici. Sama motivacija dolazi iz činjenice da ogroman broj aplikacija danas koristi slike i video zapise kad deo svog sadržaja. Ovo je dovelo do pregomilavanja ovakvog sadržaja u bazama podataka bez neke jasne klasifikacije niti načina da se ona proveriti. Takođe sama ideja da je moguće prepoznati objekat unutar matrice nekih brojeva bila mi je zapanjujuća te sam morao da zaronim dublje u ovu temu i implementiram neki od modernih detektora kako bi u mogućnosti mogao da ga iznova koristim za različite ciljeve u svom poslu. Detektor o kojem će biti više reči u narednim poglavljima moguće je "istrenirati" da prepoznaje različite objekte uz samo malo truda, odnosno pronalaženje dobrog dataset-a slika koje sadrže željeni objekat.

Izazov da će prepoznavanje da se vrši u video zapisu doveo je dodatni problem, odnosno optimalnog korištenja procesorskog vremena jer prepoznavanje treba da se odvije 30 puta u jednoj sekundi koliko danas većina zapisa ima frejmova u sekundi. Stoga problem je morao da se svede na minimum i izbor pristupa i njene implementacije morao je da procesorski efikasan.

Najbolji način da se problem svede na minimum nije da željeni objekat ne prepoznajete u odnosu na sebe, ili više drugih objekata, nego da ovaj objekat prepoznajete u odnosu na sve ostalo. Drugim rečima napravićemo dve klase gde ćemo prepoznavanje vršiti tako što ćemo željenu sliku svrstati u jednu od ove dve klase i tako napraviti odluku.

Za ovaj posao najbolji kandidat je algoritam pod nazivom "Metoda potpornih vektora" (eng. Support Vector

Machines) ili skraćeno SVM. Što se tiče detekcije bazirane na karakteristikama dva su kandidata zanimljiva, a to su algoritmi SIFT i SURF. SURF je daleko brži ali ima slabiju preciznost pri različitim potrebama, tako da je on pobednik.

Takođe, bilo bi izuzetno procesorski skupo da se svi deskriptori dobijeni SURF-m klasifikuju pomoću SVM-a tako da smo koristili popularno rešenje sa vokabularom od reči dobijenim klasterovanjem ovih deskriptora o čemu će biti više reči u sledećim poglavljima.

2. SURF

U ovom radu mi predlažemo novu detektor-deskriptor šemu koju smo nazvali SURF (Speeded-Up Robust Features). Detektor je zasnovan na na Heseovoj matrici [1], ali uz veoma malu aproksimaciju, baš kao što je DoG [2] u veoma maloj meri zasnovan na Laplasovom detektoru.

On se oslanja na integralne slike kako bi umanjio vreme izračunavanja pa ga stoga i nazivamo 'Fast-Hessian detector'. Sa druge strane, deskriptor opisuje distribuciju Haar-wavelet odziva u susedstvu ključnih tačaka. Ovog puta takođe koristimo integralne slike radi povećanja brzine.

Pored toga, iako su se koristile samo 64 dimenzije, umanjeno je vreme izračunavanja i pronalaženja sličnosti, a u isto vreme je uvećana robustnost.

Mi takođe predstavljamo novi korak- indeksovanje- zasnovan na Laplasovom detektoru, koji uvećava ne samo brzinu pronalaženja sličnosti, već i robustnost deskriptora.

3. K- MEANS

K-means je široko upotrebljavana tehnika klasterovanja koja teži da u što većoj meri umanjí kvadratnu razdaljinu između tačaka u jednom istom klasteru. Iako ne obezbeđuje potpunu tačnost, njena jednostavnost i brzina veoma su pogodne za praktičnu primenu.

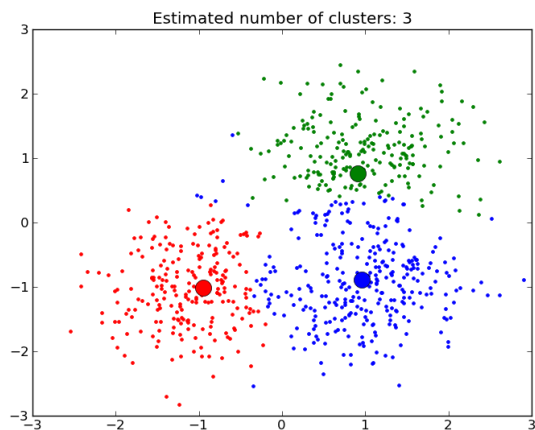
Proširujući k-means veoma jednostavnom tehnikom slučajnog izbora dobijamo algoritam koji je $\Theta(\log k)$ konkurentan sa optimalnim klasterovanjem. Početni eksperimenti pokazuju da naše proširenje poboljšava oboje, i brzinu i tačnost algoritma k-means, i to često veoma značajno.

Za k-means problem data nam je celobrojna vrednost k i skup tačaka podataka $X \subset \mathbb{R}^d$. Težimo da izaberemo k centre C na taj način da u što većoj meri umanjimo potencijal funkcije.

$$\varphi = \sum_{x \in X} \min_{c \in C} \|x - c\|^2 \quad (1)$$

Odabir ovih centara implicitno definiše klasterovanje - za svaki centar, određujemo jedan klaster koji će predstavljati skup tačaka podataka koji su bliži tom centru nego bilo kom drugom centru.

Primer ovakvog rezultata prikazan je na slici 1. Kao što je iznad naznačeno, pronalaženje tačnog rešenja za k-means problem spada u NP-teške probleme.



Slika 1. Primer klasterovanja sa 3 klastera nad podacima

4. Histogram gradijenata

Kada se radi sa velikim dimenzijama kao što je veoma čestu u aplikacijama kompjuterske vizije, potrebno je pronaći optimalan algoritam za pretraživanje suseda (eng nearest-neighbor search algorithm). U našem slučaju posle zadatog vokabulara uz pomoć k-means algoritma, svaki "patch" izvađen pomoću SURF algoritma koji ima 64 dimenzije, potrebno je uporediti sa vokabularom od 1000 centralnih patch-eva kako bi se dobio jedan histogram.

Ovakvo pretraživanje je procesorski izuzetno skupo, posebno u aplikaciji koja treba obradi 30 slika u jednoj sekundi. U ovom radu korišten je FLANN-BASED algoritam za pretragu (eng. Fast Library for Approximate Nearest Neighbors) [3] i deo je standardne interfejsa u biblioteci OpenCV za obradu slike. Ovaj algoritam je izuzetno brz što će zadovoljiti potrebe naše aplikacije iako kao rezultat ne garantuje uvek tačno pronađeni najbliži centar u vokabularu.

Kod aplikacija koja je naša gde je pronalaženje najbližeg suseda samo jedan deo unutar većeg niza algoritama koji rade sa verovatnoćama i ne uvek apsolutno tačnim rezultatima, ovaj algoritam koji daje 95% tačnosti savršen je izbor kao algoritam pretrage najbližeg centra. Svaki pronađeni najbliži centar povećava vrednost našeg histograma na poziciji tog centra u vokabularu za 1. Ovakav histogram ide kao ulazni parametar za razdvajanje klasa objekata u algoritam SVM o kojem će biti više reči u sledećem poglavlju.

5. SVM

Naš zadatak je da predvidimo da li test uzorak pripada ili ne jednoj od dve klase. Trening primere dobijamo u sledećoj formi: $\{X_i, Y_i\}, i = 1, \dots, n$ and $X_i \in \mathbb{R}^d, Y_i \in \{-1, +1\}$. Kovarijable (kovarijate) ili ulazni vektori nazivaju se $\{X_i\}$ varijablom a rezultirajuće (zavisne) varijable obeležavaju se sa $\{Y_i\}$.

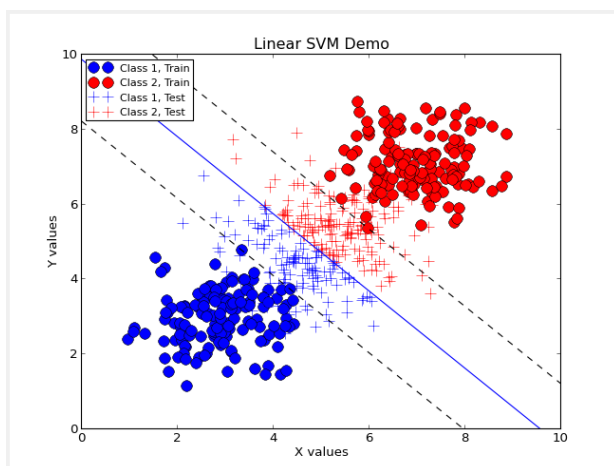
Smatramo veoma jednostavnim primerom onaj gde su podaci linearno separabilni. Takav je sledeći primer: Mogu da nacrtam pravu liniju $f(x) = W^T X - b$ tako da se svi slučajevi $Y_i = -1$ odvoje na jednu stranu i imaju $f(x) < 0$ a klase sa $Y_i = +1$ budu odvojene na

drugoj strani i imaju $f(x) > 0$. Pod uslovom da je ovo tačno, postizemo to da i nove test slučajeve možemo da klasifikujemo sledeći pravilo $Y_{test} = \text{sign}(X_{test})$.

Ipak, obično postoji neograničeno mnogo takvih hiperravni dobijenih malim izmenama datog rešenja. Kako vršimo odabir među svim ovim hiperravnima koje rešavaju problem razdvajanja naših trening podataka, ali mogu da imaju različite performanse kada su u pitanju novi test slučajevi.

Na primer, mogli bismo da izaberemo da povučemo liniju veoma blizu članova jedne specifične klase $Y = -1$. Intuitivno, kada stignu novi test slučajevi za klasifikovanje nećemo praviti mnogo grešaka što se tiče slučajeva koji bi trebalo da budu klasifikovani sa $Y = +1$, ali ćemo lako grešiti u onim slučajevima gde bi oni trebalo da budu klasifikovani sa $Y = -1$ (na primer, zamislmo da stiže nova grupa test slučajeva što izaziva male perturbacije podataka iz trening skupa).

Deluje logično da bi trebalo izabrati liniju razdvajanja koja bi bila najviše moguće udaljena od obe klase: $Y = -1$ i $Y = +1$, što znači- tačno na sredini, slika 2.



Slika 2. Linearni kernel razdvojen „najširoom ulicom“

6. KLIZEĆI PROZOR

Ovo je vrlo jednostavna metoda za lociranje detektovanih objekata u slici koja ima mnogo različitih optimizacija. Ona uzima jedan prozor predodređene veličine na startnoj poziciji u slici, prosledjuje ovaj prozor kao zasebnu sliku detektoru na detekciju, a zatim pomera prozor nekoliko piksela u stranu, i ponavlja postupak dok ne prodje kroz čitavu sliku kako je prikazano na slici 3.

Postavljaju se dva problema kod ovog pristupa. Prvi je preklapanje prozora kod detektovanja istog objekta, a drugi je detekcija različitih veličina objekata.

Prvi problem se rešava otklanjanjem prozora, postoje mnoge optimizacija na ovu temu međutima u ovom radju korišten je najjednostavniji, odnosno usrenjđavanjem suseda po lokaciji.

Problem kada je objekat veći na slici nego odabrani klizeci prozor se rešava puštanjem istog klizećeg prozora kroz više gausovih piramida originalne slike.



Slika 3. Primer kako prozor 3x3 treba da se pomera kroz sliku

7. ZAKLJUČAK

Za izradu ove aplikacije korištena je OpenCV biblioteka za C++ koja poseduje sve gore napomenute algoritme. Za trening fazu korišteni su dataset-ovi pešaka INRIA i PETA.

Aplikacija je trenirana na ovim uzorcima data setova jer našem kontekstu želimo da prepoznamo pešake u različitim okruženjima na putevima snimajući iz kola. Tako da ista ova aplikacija bi imala bolje rezultate sa nekim drugim trening datasetovima na primer u kontekstu nadzornih kamera koje imaju za cilj da pronađu pešake na uvek istom okruženju.

Za izradu vokabulara korišteno je 100 slika pešaka u različitim okruženjima i pod različitim osvetljenjem iz INRIA dataset-a.

Za izdvojene pešake „pozitive“ korišteno je 2032 slike iz kombinacije INRIA i PETA veličine 64x64 piksela, kao i „negativi“, odnosno slike svega drugog što nije pešak. Najveći izazov ove aplikacije je definitivno pronaći mesto gde imamo dovoljno tačno prepoznavanje pešaka u dovoljno brzom obradi. Kako god bi povećali tačnost prepoznavanja gubili bi na brzini, a kod nas 30 slika u sekundi treba biti obrađeno i to kada uvrstimo u to da jedna od ovih slika da bi bila obrađena zapravo je pretvorena u niz malih klizećim prozorom, pa još u više dimenzija sve ovo postaje priličan izazov.

Za brzinu najviše je pomogao brzi linearni SVM kernel kao i FLANN-BASED pretraživač suseda. A tačnost smo dobili pomoću „Cross-validation“, ponekad zvane „Rotation estimation“, ili unakrsna provera. Evo kako zapravo ovo funkcioniše.

Nad jednim vokabularom trenirali bi linearni SVM kernel sa različitim parametrima, kao i različitim parametrima za SURF. Nakon istreniranog uzorka aplikacije, započeo bi test nad klasifikovanom setu pozitivna i negativna. „false-positive“ je netačno pronađen pešak kao i netačno pronađen ne-pešak.

Nakon redom menjanja ovih SURF I SVM parametara uzimamo one koje daju najbolji procenat tačnih pozitivna i negativna. ovo se zove unakrsna provera.

4. LITERATURA

- [1] 1. Lindeberg, T., Feature detection with automatic scale selection.
- [2] Lowe, D., Distinctive image features from scale-invariant keypoints, cascade filtering approach.
- [3] Marius Muja, David G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, 2009.

Kratka biografija:



Lazar Tatomrov rođen je u Brčkom 1987. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i Telekomunikacija – Obrada signala odbranio je 2016. god.

**ZAKONSKI ASPEKT PLANIRANJA I IZGRADNJE OPTIČKIH KABLOVSKIH MREŽA
U REPUBLICI SRBIJI****LEGAL ASPECT OF PLANNING AND IMPLEMENTATION OF OPTICAL NETWORKS
IN REPUBLIC OF SERBIA**

Petar Stefanović, Željen Trpovski, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu dat je prikaz bitnih odlika „Pravilnika o tehničkim i drugim zahtevima prilikom izgradnje prateće infrastrukture potrebne za postavljanje elektronskih komunikacionih mreža, pripadajućih sredstava i elektronske komunikacione opreme pri izgradnji poslovnih i stambenih objekata“ i „Uputstva o realizaciji tehničkih i drugih zahteva pri izgradnji elektronske komunikacione mreže i pripadajućih sredstava u stambenim i poslovnim objektima“ (Pravilnik i Uputstvo), kao i zakonodavno-regulatornog okruženja u Republici Srbiji u kome su doneti.

Abstract – This paper provides main properties of Technical demands for building the optical infrastructure for the communication network in business and residential environment, as well as the corresponding laws in the Republic of Serbia.

Ključne reči: Optičke mreže, propisi.

1. UVOD

Infrastrukturni resurs koji obezbeđuje funkcionisanje elektronskih komunikacija i implementaciju informaciono-komunikacijskih tehnologija su elektronske komunikacione mreže (EKM). Zahvaljujući rapidnim tehnološkim promenama i razvoju čitavog niza novih usluga koje se većinom zasnivaju na korišćenju Interneta, uloga EKM se proširila i na sve segmente društva i dobila tako još veći značaj.

Povećanjem protoka informacija, uz njegovu simetriju koje sve češće postaje obavezujući uslov, proširen je skup sve zahtevnijih usluga za potrebe poslovnih i rezidencijalnih korisnika, čime je dodatno porasla uloga EKM kao komunikacione infrastrukture.

Odgovor operatora na sve veći broj zahtevnih usluga i rastuće zahteve i očekivanja korisnika je bio u transformaciji telekomunikacionih mreža zasnovanih na komutaciji kola (kanala) u mreže zasnovane na komutaciji paketa sa Internet protokolom (IP), koje se označavaju kao mreže sledeće generacije. Ova tranzicija je započela i velikim delom se i završila u okosnicama transportnih mreža.

U delu mreža za pristup još uvek postoji značajan udeo infrastrukture izgrađen kablovima sa metalnim provodnicima na delu između korisnika i komutacionog čvora koji je u vlasništvu dominantnog operatera.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željen Trpovski, vanr. prof.

Zahvaljujući konvergenciji medija i naglom porastu IP saobraćaja pretplatnička petlja izgrađena simetričnim paricama je dostigla svoj fizički limit i postala usko grlo, pa je izgradnja mreža za pristup uvođenjem kablova sa optičkim vlaknima postala imperativ.

Sa stanovišta regulative, u pogledu složenosti i potrebe da bude efikasna i delotvorna, najosetljiviji deo telekomunikacione infrastrukture predstavlja mreža za pristup. U njoj se često sukobljavaju interesi različitih operatora. Budući da je oblast optičkih komunikacija veoma široka u ovom radu akcenat je dat isključivo na zakonski aspekt planiranja optičkih mreža, analizi postojećih propisa u toj oblasti, kako kod nas tako i u svetu i sagledavanju neophodnih koraka da bi se ta oblast unapredila kod nas.

2. ISTORIJAT OPTIČKIH TELEKOMUNIKACIJA

Optičko vlakno je tanka staklena nit sačinjena od silicijuma ili od plastike. Staklo koje se koristi ima izuzetnu čistoću. Ne može se ni uporediti sa staklom na koje smo navikli u svakodnevnom životu. Staklo debljine nekoliko kilometara ima providnost običnog prozorskog stakla debljine 3-4 milimetra i time omogućava prostiranje svetlosti na velike daljine uz mala slabljenja signala.

Zahvaljujući otkriću Čarlsa Kaoa i Džordža Hokmana 1970. godine počeo je vrlo intenzivan razvoj optičkih komunikacija kada je tim stručnjaka iz kompanije "Corning Glass" proizveo optičko vlakno dužine stotinu metara. Široka oblast savremenih optičkih telekomunikacija intenzivno je počela da se razvija posle 1980. godine. Primenu optike u današnjem smislu pokrenuo je izum lasera 1958. godine, nakon toga poluprovodničkog lasera 1963. godine i nekoliko godina kasnije prvih optičkih vlakana. Dalji razvoj optičkih kablova doveo je do toga da 1988. godine bude postavljen prvi transatlantski optički kabl, sa regeneratorima na udaljenostima od 64 kilometra. Tokom osamdesetih godina uloženi su ogromni naponi da se otklone problemi vezani za popravku prekinutih optičkih kablova i da se poboljša tehnika njihovog postavljanja kako u zemlji tako i na dno mora.

Kao rezultat toga imamo 1996. godine postavljanje optičkih kablova preko Tihog okeana. Kod nas je postavljanje optičkih sistema počelo 1984 u Beogradu, a 1991. postavljena je optička veza Novi Sad - Sremski Karlovci.

Razvojem optičkog prenosa i uvođenjem SDH tehnike napravljen je veliki napredak u optičkim telekomunikacionim mrežama. Taj napredak se ogleda u mogućnosti da

se optičkim mrežama prenese mnogo veća količina informacija znatno većim brzinama. Sa napredkom digitalne tehnike prenosa razvijaju se i standardi kao što su SONET a ubrzo nakon toga i SDH, koje su dodatno uticale na dalji razvoj optičkih komunikacija. Danas se korisnicima pružaju nove razne vrste servisa što ne bi bilo moguće bez povećanja protoka i propusnog opsega, kakvi se nude u savremenim optičkim mrežama. Upravo je to ono što ističe optičke mreže kao kandidata za glavni prenosni medijum u bučnosti koja je već tu. Danas se optički kablovi koriste pri prenosu govornih signala, TV signala u kablovskim distributivnim sistemima, računarskih podataka, radarskih podataka i različitih mernih signala [1].

3. PLANIRANJE I REALIZACIJA OPTIČKIH KABLOVSKIH MREŽA

Kako bi se napravio dobar plan projekta, svaka odluka mora biti doneta na čvrstim i tačnim podacima. Zbog toga je od ključnog značaja imati tačne, pre svega geografske informacije, o regiji za koju se izrađuje plan. Softverski alati, nakon toga mogu koristiti te podatke kako bi napravili više različitih topologija mreže, pod različitim pretpostavkama, kako bi uporedili moguća scenarija i izabrali najrealniji. Najvažniji tipovi podataka potrebnih za pravilno planiranje mreže su: geografski podaci, specifikacija opreme i troškovi.

Geografske karakteristike regije, nad kojom se vrši planiranje optičke mreže moraju biti uzete u obzir tokom svih faza planiranja. Tipično se podaci topologije ulica prikupljaju pomoću različitih baza podataka geografsko informacionih sistema (GIS), mada neretko se koriste veb sajtovi za mapiranje i planiranje ruta kao što su maps.google.com i www.openstreetmap.org. Informacije o zgradama i stanovnicima tih zgrada je teže prikupiti. Ponekad se ove informacije dobijaju iz podataka dobijenih na višem nivou planiranja, npr. od lokalne elektrodistribucione kompanije. Ukoliko ne postoji validan izvor informacija, jedini način da se prikupe ove informacije jeste neposredno obilaženje lokacije i sagledavanje stanja na terenu i na osnovu toga realizovanje daljeg planiranja mreže.

Oprema za izgradnju FTTH mreže obuhvata aktivne komponente (multipleksere, pojačavače, optičke svičeve, media konvertere) i pasivne komponente (optičke distributivne ormane, spojnice, splitere, kablove, završne optičke kutije). Kod specifikacije opreme, potrebno je voditi računa o brojnim elementima kao na primer:

- kablovima koji se mogu instalirati u transportnoj, distributivnoj i pristupnoj mreži;
- dijametru kablovskih cevovoda kroz koje će biti uduvavani kablovi;
- opremi koja se ugrađuje u distributivnim tačkama, zgradama i kod krajnjeg korisnika;
- merenjima koje treba izvršiti pre puštanja mreže u rad.

Jedan od glavnih ciljeva pri planiranju optičke mreže jeste kontrola troškova, unutar datog seta ograničenja i zahteva koji se projektom moraju ispuniti. Kako bi se ovi troškovi što preciznije predvideli, moraju se sagledati svi aspekti izgradnje i održavanja optičke mreže. Projektant ima obavezu da odluči koji će korisnik biti konektovan na

koju optičko distributivnu tačku, kao i kakvo rešenje za menadžment optičkim kablovima treba koristiti u distributivnim tačkama. Mreža mora biti detaljno isplanirana sa optimalnim brojem optičko distributivnih tačaka. Takođe treba odrediti koji će se kablovi koristiti i kakva kablovska kanalizacija. Tehnički opis optičkih kablova treba da sadrži sažet opis primenjenih podzemnih i samonosivih optičkih kablova u mreži primarne raspodele. U tehničkom opisu treba dati pregledan crtež konstrukcije kabla na kojem su prikazani raspored cevčica sa optičkim vlaknima, rasteretni elementi, slojevi i materijali, kao i omotač, kojima se ostvaruju zahtevane tehničke karakteristike određene konstrukcije kabla. Tehnički opis treba da sadrži i tehničke karakteristike kabla sa relevantnim skicama, opisima i detaljima koji su bitni za ocenu karakteristika kabla. Za parametre koji su bitni za ocenu karakteristika kabla, u opisu se daje grafička i/ili tabelarna zavisnost njegove vrednosti u funkciji temperature, kako bi se u tehničkom rešenju mogla odrediti vrednost relevantnog parametra na potrebnoj temperaturi. U tehničkom opisu treba takođe navesti i očekivani radni vek kabla, pod pretpostavkom njegovog polaganja, eksploatacije i održavanja na propisani način, u kome se garantuje da će isti zadržati sve karakteristike i u propisanim ekstremnim nepovoljnim uslovima okruženja. Zajedno sa hardverskim zahtevima (u koje se ubrajaju i spojnice, spliteri, aktivni svičevi...), ova informacija se može iskoristiti za generisanje grubog predmera opreme i materijala koji može koristiti dobavljačima kao indikator količina koje bi trebalo isporučiti. Preporučljivo je koristiti automatske softverske alate, obzirom da oni mogu da obrade sve relevantne podatke, kao i njihove međuzavisnosti, u jednom koraku. U ovakvom okruženju, projektant određuje koji su parametri relevantni i koja ograničenja treba uvesti u alat za planiranje i optimizaciju mreže.

Sastavni deo planiranja i realizacije optičkih kablovskih mreža jeste i dokumentacija. Pored Idejnog projekta kablovskog distribucionog sistema od dokumentacije neophodan nam je Projekat izvedenog stanja, dozvola za polaganje optičkih kablova i rezultati merenja izvedene mreže. Obično se upravljanje projektom izvršava na sledeći način: Softver za detaljno planiranje mreže komunicira sa veb-baziranim sistemom na terenu, tako što mu šalje delove tog mrežnog dijagrama, pokazujući mu u svakom trenutku koji je sledeći korak. Ovakav vid komunikacije je moguć ukoliko na terenu imamo bežični sistem komunikacije, pomoću kojeg se tehničarima i inženjerima na terenu šalju zadaci, a oni odgovaraju izveštajima sa podacima o stanju na terenu i izvršenim zadacima.

4. ANALIZA PRAVNIH PROPISA IZ OBLASTI OPTIČKIH KOMUNIKACIJA U REPUBLICI SRBIJI

Brz razvoj elektronskih komunikacija je veoma značajan za Republiku Srbiju, pa je definisanje ciljeva razvoja elektronskih komunikacija od posebnog državnog interesa, pre svega zbog ekonomskog oporavka i rasta. Za ulaganja kompanija u telekomunikacionu infrastrukturu neophodno je da ta oblast bude zakonski uređena i da postoje državne strategije koje će davati smernice za dalji razvoj i ulaganja. Radi stvaranja pravnog okvira za

liberalizaciju tržišta i razvoja elektronskih komunikacija, postojeći propisi će se uskladiti sa međunarodnim i evropskim standardima, a cilju stvaranja uslova za zadovoljenje potreba krajnjih korisnika i operatora. Pravni okvir relevantan za razvoj elektronskih komunikacija u Republici Srbiji jesu sledeći zakoni:

- Zakon o elektronskim komunikacijama,
- Zakon o telekomunikacijama,
- Zakon o radiodifuziji i
- Zakon o planiranju i izgradnji.

Zakonom o elektronskim komunikacijama ("Službeni glasnik RS", broj 44/10) se uređuju uslovi i način za obavljanje delatnosti u oblasti elektronskih komunikacija [2]. Ciljevi i načela regulisanja odnosa u oblasti elektronskih komunikacija zasnivaju se na:

- obezbeđivanju uslova za ravnomeran razvoj elektronskih komunikacija na celoj teritoriji Republike Srbije,
- obezbeđivanju predvidivosti poslovanja i ravnopravnih uslova za poslovanje operatora,
- usklađivanju obavljanja delatnosti u oblasti elektronskih komunikacija sa domaćim i međunarodnim standardima,
- obezbeđivanju dostupnosti usluga univerzalnog servisa svim građanima u Republici Srbiji, uz zadovoljenje potreba specifičnih društvenih grupa, uključujući osobe sa invaliditetom, starije i socijalno ugrožene korisnike,
- obezbeđivanju međupovezivanja elektronskih komunikacionih mreža i usluga, odnosno operatora, pod ravnopravnim i uzajamno prihvatljivim uslovima,
- podsticanju konkurencije, ekonomičnosti i delotvornosti u obavljanju delatnosti elektronskih komunikacija,
- podsticanju racionalnog i ekonomičnog korišćenja numeracije i radiofrekvencijskog spektra,
- obezbeđivanju maksimalne koristi za korisnike elektronskih komunikacija, uključujući osobe sa invaliditetom, starije i socijalno ugrožene korisnike, naročito u smislu izbora, cene i kvaliteta,
- obezbeđivanju visokog nivoa zaštite interesa potrošača u odnosu sa operatorima, naročito obezbeđivanjem dostupnosti jasnih i potpunih informacija o cenama, uslovima pristupa i korišćenja (uključujući ograničenja) i kvalitetu javnih komunikacionih mreža i usluga, kao i efikasnim postupanjem po pritužbama na rad operatora,
- obezbeđivanju stalnog unapređenja kvaliteta usluga elektronskih komunikacija,
- obezbeđivanju mogućnosti krajnjih korisnika da, prilikom korišćenja javnih komunikacionih mreža i usluga, slobodno pristupaju i distribuiraju informacije, kao i da koriste aplikacije i usluge po njihovom izboru,
- obezbeđivanju visokog nivoa zaštite podataka o ličnosti i privatnosti korisnika,
- osiguravanju bezbednosti i integriteta javnih komunikacionih mreža i usluga.

Zakon o telekomunikacijama („Službeni glasnik RS”, br. 44/03, 36/06) je u skladu sa međunarodnim pravnim standardima. Njime se uređuju uslovi i način obavljanja delatnosti u oblasti telekomunikacija. Ovim zakonom je osnovana Republička agencija za telekomunikacije (RATEL) i definisana su njena prava i obaveze [3]. Ovim zakonom se još uređuju i pitanja koja se odnose na sprečavanje monopola, dodela dozvola za obavljanje delatnosti, regulacija tarifa telekomunikacionih usluga, interkonekcija mreža i operatora, zakup linija, međunarodne telekomunikacije i druga pitanja od značaja za funkcionisanje i razvoj telekomunikacija u Republici Srbiji.

Zakon o radiodifuziji ("Službeni glasnik RS", br. 42/02, 97/04, 76/05, 79/05, 62/06, 85/06, 86/06 i 41/09) uređuje uslove i način obavljanja radiodifuzne delatnosti, u skladu sa međunarodnim konvencijama i standardima, osnivanje Republičke radiodifuzne agencije, kao i ustanove javnog radiodifuznog servisa, utvrđivanje uslova i postupak za izdavanje dozvola za emitovanje radio i televizijskog programa, uređivanje i drugih pitanja od značaja za oblast radiodifuzije [4].

Zakon o planiranju i izgradnji ("Službeni glasnik RS", br. 72/09 i 81/09) uređuje uslove i način uređenja prostora, uređivanje i korišćenje građevinskog zemljišta i izgradnju objekata, uključujući i telekomunikacione objekte; vršenje nadzora nad primenom odredaba ovog zakona i inspeksijski nadzor, druga pitanja od značaja za uređenje prostora, uređivanje i korišćenje građevinskog zemljišta i za izgradnju [5]. Najvažniji podzakonski akt u ovoj oblasti je „Pravilnik o uslovima i normativima za projektovanje stambenih zgrada i stanova“ (objavljen u „Službenom glasniku RS“, broj 58/2012 od 13. 6. 2012. godine.), koga je donelo Ministarstvo životne sredine, rudarstva i prostornog planiranja. U članu 46, stav 2. pomenutog pravilnika se navodi da: „Sve stambene zgrade i stanovi moraju biti opremljeni elektronskom komunikacionom mrežom koja se planira, instalira, koristi i održava u skladu sa važećim propisima koji regulišu oblast elektronskih komunikacija“. Na osnovu ovog člana bliže uređivanje opremanja stambenih zgrada elektronskom komunikacionom mrežom prepušta se važećim propisima koji regulišu oblast elektronskih komunikacija (očigledno budućim, jer su postojeći u vreme usvajanja pomenutog pravilnika bili zastareli i prevaziđeni). Iz prethodne analize se može izvesti zaključak da su jedino u ZEK ugrađene odredbe u kojima je formulisana obaveza investitora da izgrade prateću infrastrukturu potrebnu za postavljanje elektronskih komunikacionih mreža, pripadajućih sredstava i elektronske komunikacione opreme do prostorija korisnika.

Strateški dokumenti relevantni za razvoj elektronskih komunikacija u Republici Srbiji su:

- Strategija razvoja elektronskih komunikacija 2010-2020 obuhvata pravne, institucionalne, ekonomske i tehničke aspekte razvoja u oblasti elektronskih komunikacija u Republici Srbiji,
- Strategija razvoja širokopojasnih mreža i servisa u Republici Srbiji do 2016. godine kao strateški cilj ističe razvoj širokopojasnog pristupa, razvoj alternativnih telekomunikacionih mreža,

liberalizaciju tržišta telekomunikacija i uz pad cena pristupa internetu,

- Strategija razvoja informacionog društva u Republici Srbiji ("Službeni glasnik RS", broj 87/06) obuhvata pravne, institucionalne, ekonomske i tehničke aspekte razvoja u oblasti IKTa, kao i bitan strateški cilj ističe razvoj širokopojsnog pristupa, osiguravanja bezbednosti mreža i razvijanje programa e-uprava, e-uključivanje, e-učenje, e-zdravlje i e-poslovanje, u skladu sa akcionim planom e-Evropa,
- Strategija razvoja elektronske uprave u Republici Srbiji za period od 2009. do 2013. godine ("Službeni glasnik RS", broj 83/09) obuhvata pravne, institucionalne, ekonomske i tehničke aspekte primene IKTa kojim se postiže efikasniji i efektivniji rad organa uprave i imalaca javnih ovlašćenja u funkciji vršenja vlasti, ekonomskog rasta i smanjenja tereta administracije.

5. PREGLED MEĐUNARODNIH STANDARDA I ZAKONSKE REGULATIVE U SVETU

U svetu najpoznatije regulatorno telo koje se bavi standardizacijom u telekomunikacijama je svakako Međunarodna telekomunikaciona unija (International Telecommunication Union - ITU). ITU je nizom preporuka definisala i uvela standarde u proizvodnju i rad sa optičkim kablovima. Na slici 1 mogu se videti preporuke koje se odnose na sisteme prenosa i digitalne sisteme i mreže.

ITU-T G-SERIES RECOMMENDATIONS TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS	
INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100-G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200-G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300-G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400-G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450-G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600-G.699
General	G.600-G.609
Symmetric cable pairs	G.610-G.619
Land coaxial cable pairs	G.620-G.629
Submarine cables	G.630-G.639
Free space optical systems	G.640-G.649
Optical fibre cables	G.650-G.659
Characteristics of optical components and subsystems	G.660-G.679
Characteristics of optical systems	G.680-G.699
DIGITAL TERMINAL EQUIPMENTS	G.700-G.799
DIGITAL NETWORKS	G.800-G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900-G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE - GENERIC AND USER-RELATED ASPECTS	G.1000-G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000-G.6999
DATA OVER TRANSPORT - GENERIC ASPECTS	G.7000-G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000-G.8999
ACCESS NETWORKS	G.9000-G.9999

Slika 1. ITU-T preporuke koje se odnose na sisteme prenosila i digitalne sisteme i mreže

U tim preporukama se nalazi i serija preporuka ITU-T G.650.1, ITU-T G.650.2, ITU-T G.650.3, ITU-T G.651.1, ITU-T G.652, ITU-T G.653, ITU-T G.654, ITU-T G.655, ITU-T G.656 i ITU-T G.657 koje se odnose na optičke kablove [6]. Sve kompanije koje se bave proizvodnjom optičkih kablova i opreme za rad sa optičkim kablovima se pridržavaju tih preporuka. Takođe i zakoni koji uređuju

ovu oblast u Republici Srbiji kao i drugim državama su usklađeni sa međunarodnim standardima.

6. ZAKLJUČAK

U Republici Srbiji transportne mreže su izgrađene od monomodnih optičkih kablova koji omogućavaju velike brzine prenosa podataka na velikim udaljenostima. Problem i dalje predstavljaju pristupne mreže jer se i dalje mahom koriste bakarni kablovi.

Za pružanje sve aktuelnijih širokopojsnih IP servisa, HDTV-a i video telefonije neophodno je da se krajnjim korisnicima omogući pristup sa protokom i nekoliko stotina Mbit/s. Iz tog razloga postojeće pristupne mreže moraju da se unaprede. Pouzdan način za to jeste masovnija upotreba FTTH (fiber to the home) mreža za pristup. U našoj zemlji većina operatora i dalje pristupne mreže bazira na tehnologiji bakarnih parica što onemogućava dalji razvoj širokopojsnog pristupa. Tehnologija optičkih mreža je u razvoju i ima sve veću upotrebu u celom svetu. Osobine optičkih mreža su bolje u mnogim segmentima od postojećih bakarnih mreža, ali su veliki troškovi implementacije razlog za njihovo sporo uvođenje u masovnu upotrebu. Sasvim je očigledno da je u rešavanju tog problema neophodna pomoć države i da je potrebno da se i u zakonskim aktima potencira na masovnoj upotrebi optičkih kablova u pristupnim mrežama (na primer da se prilikom izgradnje stambenih objekata ugrađuju optički kablovi do svakog stana). Republika Srbija bi trebala u svojim strategijama za razvoj telekomunikacija više pažnje da posveti upotrebi optičkih mreža jer zamenu pristupnih mreža treba posmatrati kao investiciju koja bi kasnije omogućila brži razvoj privrede i mogla da poveća BDP i do 2% na godišnjem nivou, jer bez razvijenih telekomunikacija nema ni razvoja privrede.

7. LITERATURA

- [1] http://sr.wikipedia.org/wiki/Fiber_optički_kabl, Septembar 2016
- [2] Službeni glasnik RS, Zakon o elektronskim komunikacijama
- [3] Službeni glasnik RS, Zakon o telekomunikacijama
- [4] Službeni glasnik RS, Zakon o radiodifuziji
- [5] Službeni glasnik RS, Zakon o planiranju i izgradnji
- [6] <https://www.itu.int/rec/T-REC-G/en>, Septembar 2016. godine

Kratka biografija:



Petar Stefanović rođen je u Novom Sadu 1988. god. Master rad odbranio je 2016. godine na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Optičke telekomunikacije.

**DINAMIČKO OBJAVLJIVANJE DOKUMENATA POMOĆU DOSTUPNIH
SOFTVERSKIH ALATA****THE DYNAMIC DOCUMENT PUBLISHING USING AVAILABLE SOFTWARE TOOLS**

Mirjana Macura, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Cilj ovog rada jeste da uporedi tradicionalno korišćen način na koji se upravlja sadržajem tehničkih dokumenata u softverskoj industriji, desktop publishing, sa topic-oriented načinom rada sa dokumentima. Kako bi se postigla veća brzina, fleksibilnost i efikasnost prilikom rada, neophodno je napustiti konvencionalne word procesore i preći na kreiranje dokumenata putem XML notacije, što se vrši specijalnim softverskim alatima. U ovom radu izvršeno je testiranje i poređenje odabranih softverskih alata komercijalnog tipa za kreiranje tehničkih dokumenata, koji podržavaju DITA standard: Adobe FrameMaker, MadCap Flare i XMetal.

Abstract – The aim of this paper is to compare desktop publishing as the commonly used method of managing technical documentation in the software industry, with the topic-oriented method of working with documentation. In order to increase the speed, flexibility and efficiency during work, it is necessary to leave behind the conventional word processors and move towards creating documents by XML notation, which can be achieved using special software tools. This paper presents the results of the testing and comparison of selected commercial software tools available for technical document creation, which provide support to the DITA standard: Adobe FrameMaker, MadCap Flare and XMetal.

Ključne reči: Desktop publishing, Topic, DITA, Dynamic publishing

1. UVOD

Izrada dokumentacije bitan je deo procesa proizvodnje i isporuke nekog proizvoda u softverskoj industriji, budući da postojanje odgovarajuće dokumentacije obezbeđuje dokumentovanje urađenog posla i omogućava lakšu razmenu znanja između stvaraoca i korisnika proizvoda.

Mana jednog od čestih oblika objavljivanja dokumenata, desktop publishing-a, osim nedostatka validacije ulaznih podataka, je što nije namenjen brzom objavljivanju dokumenata obimnog sadržaja, kao i nemogućnost automatizovanog dobijanja različitih dokumenata od istog sadržaja. Međutim, objavljivanje dokumenata topic-oriented metodom podrazumeva raspodelu sadržaja dokumenata na topic-e, tj. samostalne smislene celine kojima se može lako manipulirati.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Erdeljan, vanr.prof.

Sadržaj dokumenta se obično čuva u XML formatu jer se tako postiže lakše upravljanje sadržajem dokumenata, kao i njegova ponovna upotreba.

Dinamičko objavljivanje dokumenata zasniva se na topic-oriented metodi i može se definisati kao kombinacija softvera i procesa kojima se obezbeđuje kreiranje, obrada, objavljivanje i isporuka sadržaja (pod kojim se podrazumevaju tekst, slike, video, audio, kao i podaci koji potiču od drugih medija, koji se generišu u realnom vremenu) putem različitih kanala.

U praktičnoj primeni, autor dokumentacije treba da ima pristup repozitorijumu gde se čuva celokupna dokumentacija, što podrazumeva brzu reakciju u slučaju izmena, kao i brzu distribuciju izmenjenog sadržaja. Podelom sadržaja dokumenta na celine, odnosno topic-e, i njihovim povezivanjem u mape, omogućava se automatizovano kreiranje različitih dokumenata od zajedničkog sadržaja.

Dinamičko objavljivanje dokumenata se najčešće povezuje sa kreiranjem baza podataka u kojima se skladište topic-i kreirani u nekom od markup jezika, najčešće u XML jeziku.

2. ŽIVOTNI CIKLUS DOKUMENTA

Tehnička dokumentacija predstavlja vid dokumentacije kojom se opisuje način upotrebe, funkcionalnost i arhitektura nekog tehničkog proizvoda. U inženjeringu sistema tehnička dokumentacija može se grupisati u sledeće kategorije: specifikacije, tehnička uputstva, dokumenti gde je opisana arhitektura sistema.

U softverskoj industriji je često, osim za osnovni proizvod, neophodno kreirati dokumente i za projekte, kojima su obuhvaćene njihove specijalne verzije.



Slika 1. Tipičan životni ciklus dokumenta

Tipičan životni ciklus dokumenta, prikazan na slici 1, sastoji se iz 6 faza: kreiranja novog dokumenta (*create*), pripreme *draft* dokumenta, *review* faze (pregledanja dokumenta), objavljivanja dokumenta (*publish*), *update* faze (dorađivanja dokumenta - u tom slučaju se kreira novi *draft* dokument), i arhiviranja dokumenta (*archive*). Poštujući sve faze životnog ciklusa dokumenata, primenom dinamičkog objavljivanja moguće je poboljšati način na koji se oni kreiraju.

3. Topic-i, XML I DITA

Prema [1], *topic* je jedinica informacije koja ima naslov i sadržaj, dovoljno kratka tako da bude vezana za jednu temu, ili da odgovara na jedno pitanje, ali i dovoljno dugačka da ima smisla, odnosno da može da predstavlja samostalnu celinu. Dokumentacija zasnovana na *topic*-ima oslanja se na minimalistički pristup dizajniranju informacija, gde se za uspešno obavljanje nekog zadatka obezbeđuje minimalni broj instrukcija.

U okviru *topic*-oriented načina generisanja dokumenata moguće je kreirati različite vrste sadržaja, od kojih svaka obezbeđuje određen tip informacije za korisnike. Postoje 3 vrste *topic*-a: task, koncept i referenca. Prilikom primene u softverskoj industriji, koncept se koristi za opis osobina softvera, task prikazuje izvođenje procesa, dok referenca podrazumeva detalje o samom softveru, i često uključuje tabelarni prikaz.

Podela dokumenta na *topic*-e dovodi do sledećih prednosti [2]:

- Svaki od *topic*-a sadrži samo informacije potrebne da bi se razumeo jedan koncept, da bi se izvela jedna procedura, ili da bi se imao uvid u jednu referencu;
- Zbog njihove samostalnosti, pojedinačni *topic*-i mogu biti poslani na reviziju odmah nakon što nastanu;
- Kombinovano korišćenje *topic*-a u različitim dokumentima može biti automatizovano, što smanjuje vreme i troškove produkcije dokumenta;
- Male izmene proizvoda koji se opisuje u dokumentu utiču samo na pojedine *topic*-e, koji se mogu pojedinačno menjati;
- Zbog samostalnosti *topic*-a znatno je olakšana navigacija kroz dokument.

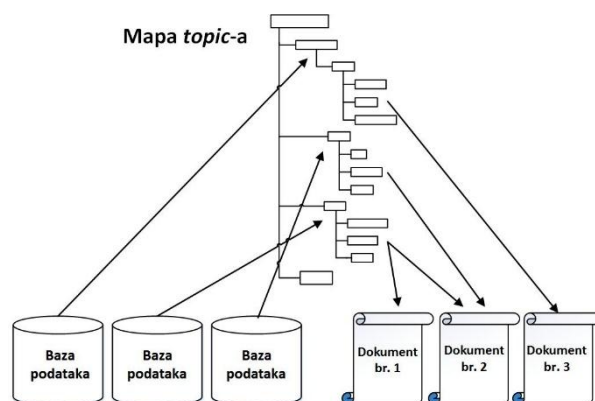
Pošto je strukturirani sadržaj lako podeliti na *topic*-e, za rad sa strukturiranim dokumentima često se koristi XML (Extensible Markup Language) *markup* jezik, koji predstavlja mehanizam za identifikovanje strukture u dokumentu [5]. On ne služi za prikazivanje podataka, već za manipulisanje njima. XML šeme koriste se za definisanje načina na koji se elementi i atributi zajedno uklapaju. Postoje dva glavna tipa ovih shema: DTD (Document Type Definition) i XML sheme.

DITA (Darwin Information Typing Architecture) predstavlja XML shemu koja se koristi za strukturirano pisanje [2]. Koristi se princip podele sadržaja dokumenta na samostalne celine, *topic*-e, koji se organizuju i povezuju u mape. Mapiranje informacija, kao vrsta

strukturiranog pisanja, predstavlja metodu za analizu, organizovanje i prikazivanje informacija.

DITA uključuje mehanizam koji omogućava kreiranje veza između *topic*-a bez stvaranja zavisnosti – takozvane tabele veza (eng. *relationship table*), čime se izbegava direktno povezivanje pojedinačnih *topic*-a. Ove tabele ulaze u sastav DITA mape.

Mapama je moguće kreirati indeks informacija, koji se nalazi izvan lokacija na kojima se te informacije nalaze. Mapa *topic*-a sadrži ključne koncepte opisane u bazama podataka ili dokumentima, i povezuje ih. Slika 2 prikazuje mapu koja se sastoji iz više *topic*-a (nalaze se u okviru stabla), koji su skladišteni u različitim bazama podataka, i služe za dobijanje više različitih dokumenata.



Slika 2. Mapa *topic*-a [2]

DITA standard je razvijen od strane OASIS DITA Tehničkog Komiteta. Zahvaljujući tome što je DITA otvoreni standard, nastali su brojni softverski alati koji ga podržavaju. Oni omogućavaju da se sadržaj dokumenta može jednostavno menjati, kao i da se taj sadržaj koristi za dobijanje dokumenata raznih formata.

Prema [3], grane industrije u kojima je DITA najviše zastupljena su: proizvodnja softvera, informacione tehnologije i servisi, telekomunikacije, industrija koja se bavi proizvodnjom poluprovodnika, i proizvodnja medicinskih uređaja.

4. ANALIZA KREIRANJA DOKUMENATA DESKTOP PUBLISHING NAČINOM RADA

Desktop publishing način rada sa dokumentima podrazumeva kreiranje i obradu dokumenata putem neke desktop aplikacije, najčešće *word procesora*. Microsoft Word je jedan od najpoznatijih *word procesora*, koji se često koristi i za kreiranje tehničke dokumentacije.

Ako se koristi kao *desktop publishing* aplikacija, Microsoft Word može postati nepodoban za izradu tehničke dokumentacije. Neki od razloga su:

- Mogućnost nestabilnog rada kod dokumenata sa velikim brojem stranica;
- Sadržaj u vidu teksta, slika, stilova, i formatiranja – sve se nalazi u jednom dokumentu, što otežava manipulaciju delovima dokumenta;
- Različito prikazivanje dokumenata u različitim verzijama softvera;

- Nema mogućnosti ponovne upotrebe sadržaja (osim kopiranjem).

Office Open XML, poznatiji pod imenom OOXML, je format Microsoft Word-a koji se bazira na XML-u, i služi za skladištenje dokumenata različitih vrsta (tekstualnih, tabelarnih, prezentacija), kao i grafikona, dijagrama, i drugih grafičkih materijala [4].

Kako Microsoft Word može da generiše XML, odnosno svaki word dokument je moguće sačuvati u XML formatu, trebalo bi ga smatrati za robustan alat za generisanje XML koda. Sadržaj ovakvog XML dokumenta na koji nije primenjen DITA standard je, međutim, težak za čitanje i nije namenjen za dalje izmene, te se za obradu dokumenata u ovom formatu koriste XML editori ili druge aplikacije namenjene radu sa XML kodom.

5. ANALIZA KREIRANJA DOKUMENATA TOPIC-ORIENTED NAČINOM RADA

Aplikacija za izradu tehničke dokumentacije koja je zasnovana na *topic-oriented* načinu rada treba da kao primarnu aktivnost ima kreiranje sadržaja dokumenta, a ne pisanje XML koda, mada u osnovi sadrži XML notaciju. Osim toga, ona bi trebalo da podržava specijalne elemente kao što su uslovi, koji služe za označavanje pojedinačnih *topic*-a ili njihovih delova, i na osnovu kojih se vrši filtriranje sadržaja u cilju dobijanja odgovarajućeg izlaznog dokumenta. Kreiranje izlaznih dokumenata različitih formata trebalo bi da bude jednostavno, a da pri tome oni budu bogatog sadržaja.

Kako bi se utvrdilo na koji način je moguće primeniti *topic-oriented* način rada sa dokumentima prilikom izrade tehničke dokumentacije u softverskoj industriji, izvršeno je testiranje i analizirani su alati za izradu tehničke dokumentacije koji podržavaju DITA standard: Adobe FrameMaker, MadCap Flare, i XMetal.

Za vreme testiranja utvrđeno je da se kod FrameMaker-a dokument ne deli na *topic*-e, već se samo prikazuje u strukturiranom obliku, dok se kod Flare-a i XMetal-a *topic-oriented* metoda sprovodi u potpunosti.

Prilikom testiranja softvera importovani su prethodno kreirani word dokumenti, kako bi se stekao uvid u proces konverzije dokumenata. Kod FrameMaker-a i Flare-a uspešno su konvertovani formati ulaznih dokumenata, dok je kod XMetal-a bila neophodna dvostruka konverzija, budući da može da otvori samo dokumente u DITA i XML formatu.

Mogućnost dobijanja većeg broja izlaznih dokumenata takođe je proverena prilikom testiranja. Na ovaj način je moguće od istog skupa *topic*-a dobiti dokument poput korisničkog priručnika, ali i *help* fajl za softver koji se opisuje. Najveći nedostatak uočen je kod FrameMaker-a i XMetal-a, koji nemaju mogućnost objavljivanja dokumenata u docx formatu.

Osim mogućnosti rada sa različitim ulaznim i izlaznim dokumentima, kod ove vrste softvera bitno je povezivanje sa različitim sistemima za upravljanje sadržajem, odnosno sa odgovarajućim repozitorijumom. Ovde se kao najbolji pokazao Flare, kod kojeg postoji mogućnost integracije sa

najvećim brojem različitih sistema za upravljanje životnim ciklusom softvera, poput Git-a, Microsoft TFS-a i Perforce-a.

XMetal se pokazao kao nepodoban za povezivanje jer je kod njega omogućena integracija samo sa nekim sistemima za upravljanje dokumentima koji se povezuju na određeni način.

5.1 Automatizacija

Kako se prilikom proizvodnje nekog softvera često smenjuju nove verzije proizvoda, potrebno je u skladu sa tim menjati i dokumentaciju. Čak i ako izmene nisu obimne, one utiču na veliki broj dokumenata, kako na one koji se tiču proizvoda, tako i na one projektnog karaktera. Poželjno je postići da se promene u sadržaju dokumenata postižu automatizovano, unošenjem izmene koja se odnosi na više dokumenata samo na jednom mestu.

Prilikom procesa automatizacije generisanja varijanti dokumenata od gotovih *topic*-a, kao polazni materijal korišćen je skup *topic*-a koji su povezani zajedničkom temom, odnosno koji su obuhvaćeni jednom mapom. Primenjenim uslovima određena su 4 dokumenta: jedan je predstavljao dokument vezan za sam proizvod, dok su ostali predstavljali projektne dokumente, što je uobičajeno prilikom kreiranja dokumentacije za neki softverski proizvod.

Dokument koji opisuje proizvod obuhvatio je najveći broj *topic*-a koji su se nalazili u okviru mape, dok su ostali dokumenti imali izmene u nekim delovima sadržaja.

Kod FrameMaker-a, strukturiranjem sadržaja dobijena je odgovarajuća hijerarhija između delova dokumenta, ali ne i željena samostalnost pojedinačnih celina. Pošto nije postojala podela na *topic*-e, određivanje uslova za stvaranje svakog od pojedinačnih dokumenata zahtevalo je pregledanje čitavog dokumenta i postavljanje potrebnih uslova na delove dokumenta koji treba ili ne treba da budu uključeni u izlazne dokumente. Dobijeni su željeni izlazni dokumenti, ali sama automatizacija nije bila uspešna.

Automatizacija u Flare-u je uspešno sprovedena, dobijeni izlazni dokumenti su se razlikovali shodno primenjenim uslovima. Svaki od dokumenata sadržao je veliki broj zajedničkih *topic*-a, dok su projektni imali i manji broj *topic*-a karakterističnih samo za specifični projekat. Tabele sadržaja su se automatski izmenile prilikom generisanja, u skladu sa uključenim *topic*-ima. Rezultatima testiranja vezanim za automatizaciju potvrđeno je da Flare u potpunosti podržava *topic-oriented* način rada sa dokumentima.

Kao posledica konverzije word dokumenata u oblik u kojem ga je moguće importovati (ili otvoriti) u XMetal-u, tj. u XML format, dobijen je dokument napisan u XML-u kojem nedostaje DTD (Document type Definition), tj. shema za definisanje strukture XML dokumenta. Posledica dvostruke konverzije formata je da tako dobijen izlazni dokument nije imao željene osobine i izgled. Automatizacija u XMetal-u nije bila uspešna.

6. KOMPARATIVNA ANALIZA ODABRANIH SOFTVERSKIH REŠENJA

Nakon testiranja odabranih softvera, izvršena je komparativna analiza. Kriterijumi koji su korišćeni prilikom ocenjivanja su izabrani da odgovaraju potrebama kompanije koja se bavi proizvodnjom softvera, te isporučuje svojim klijentima dokumentaciju u različitim oblicima i formatima.

Osim ocenjivanja opštih karakteristika softverskih alata, i utvrđivanja koji su zajednički elementi koje izabrani softverski alati imaju sa *word procesorima* (tj. sa Microsoft Word-om), za opisivanje karakteristika alata korišćeni su različiti formati.

U tabelama 1 i 2 prikazani su formati koji se često koriste kod tehničke dokumentacije u softverskoj industriji.

Tabela 3 sadrži neke od kriterijuma koji se tiču *topic-oriented* načina rada sa dokumentima, i odgovara

Tabela 3. Kriterijumi za *topic-oriented* način rada.

Kriterijumi	FrameMaker	Flare	XMetal
Stabilnost prilikom rada sa velikim dokumentima	+	+	-
Automatska podela na <i>topic</i> -e nakon importovanja	-	+	+
Automatsko generisanje tabele sadržaja	-	+	+
Automatsko osvežavanje referenci	+	+	+
Analiza mape <i>topic</i> -a pre objavljivanja dokumenta	-	+	-
Praćenje promena u sadržaju dokumenta	+	+	+
Poređenje dokumenata	+	-	-
Dodavanje komentara u <i>topic</i> -e	-	-	+

Kako tehnički dokumenti mogu imati veliki broj stranica, od alata se očekuje stabilnost prilikom rada. Pored toga, alat bi trebalo da prilikom importovanja podeli dokument na *topic*-e (na osnovu postojećih celina), kao i da za novonastalu mapu *topic*-a automatski generiše tabelu sadržaja. Analiza mape *topic*-a pre objavljivanja dokumenta obezbeđuje informacije o tome da li se svi *topic*-i nalaze u okviru indeksa mape i tabele sadržaja. Takođe, analizom mape proverava se povezanost linkova i referenci sa sadržajem dokumenta. Reference bi trebalo da se osvežavaju automatski, nakon svake izmene *topic*-a iz mape. Praćenje promena olakšava uvođenje verzija dokumenata, a poređenjem dokumenata postiže se skraćeno vreme neophodnog za izmene zajedničkog dela sadržaja. Dodavanje komentara u *topic*-e omogućava lakše sprovođenje njihove revizije.

Sabiranjem bodova nastalih prebrojavanjem pozitivnih ocena (+) po kolonama svih tabela sa kriterijumima korišćenih prilikom sprovođenja komparativne analize, utvrđeno je da je najbolji rezultat ostvario MadCap Flare, a veoma dobro se pokazao i Adobe FrameMaker. Analiza sprovedena pomoću kriterijuma pokazala je da XMetal nije pogodan za migraciju na *topic-oriented* način rada sa dokumentima, budući da nije zadovoljio veliki broj postavljenih kriterijuma.

7. ZAKLJUČAK

U ovom radu je prikazana analiza uobičajenog načina objavljivanja dokumenata, koja je upoređena sa modernijom, *topic-oriented* metodom, koja polako preuzima primat u oblasti tehničke dokumentacije. Analizom novog načina rada sa dokumentima, data je

potrebama migracije sadržaja dokumenata u kompaniji koja se bavi proizvodnjom softvera.

Tabela 1. Često korišćeni ulazni formati.

Format dokumenta	FrameMaker	Flare	XMetal
CHM	-	+	-
DITA	-	+	+
Word (docx)	+	+	-
PDF	+	+	-

Tabela 2. Često korišćeni izlazni formati.

Format dokumenta	FrameMaker	Flare	XMetal
CHM	+	+	+
DITA	-	+	+
Word (docx)	-	+	-
PDF	+	+	+

osnova za odabir pogodnog *topic-oriented* alata, koji je u skladu sa potrebama kompanije koja se bavi izradom softvera. Prilikom istraživanja testirane su aktuelne verzije za sledeće softverske proizvode: Adobe FrameMaker, MadCap Flare, i XMetal. Na osnovu predloženih kriterijuma koje alat treba da ispuni, najbolje rezultate postigao je MadCap Flare.

Istraživanje koje je sprovedeno u ovom radu ostavilo je mesta za dalja proširenja. Osim pitanja da li alat ima mogućnost integracije sa sistemom za upravljanje dokumentima, mogla bi se razmotriti mogućnost neke druge integracije koja bi omogućila svrsishodnije sprovođenje procesa automatizacije.

8. LITERATURA

- [1] OASIS Darwin Information Typing Architecture (DITA) Architectural Specification v1.0, OASIS Open, 2005.
- [2] <http://dita.xml.org/>, pristupljeno 20.6.2016.
- [3] Marion L., Catherine S., Jean Michel V.D.B., Laureen Z.: *DITA: A universal solution*, Paris Diderot University, 2015.
- [4] <http://officeopenxml.com/>, pristupljeno 20.6.2016.
- [5] <http://www.xml.com/>, pristupljeno 20.6.2016.

Kratka biografija:



Mirjana Macura rođena je u Novom Sadu 1990. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstvo – Energetska elektronika i električne mašine odbranila je 2016. godine.

**PROŠIRENJE DJANGO WEB APLIKACIJE U OKVIRU KROKI ALATA
EXTENDING DJANGO WEB APPLICATION WITHIN KROKI TOOL***Arpad Farkaš, Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je opisana implementacija parent-child panela baziranih na skicama poslovnih aplikacija, kreiranih pomoću generatora koda i dalje formiranih u razvojnom okviru Django. U skladu sa savremenim tehnologijama promenjen je i grafički korisnički interfejs veb aplikacije. Rešenje je integrisano u aplikaciju za skiciranje poslovnih aplikacija Kroki.

Abstract – This paper presents the implementation of parent-child panels with a code generator for Django framework based on sketches for business applications. The graphical user interface has been also updated in accordance with the contemporary technologies. The solution is integrated to an application for making sketches for business applications – Kroki.

Ključne reči: Kroki, Django framework, Python, generator, Model driven engineering, EUIS DSL

1. UVOD

U današnjem svetu elektronsko poslovanje sve više i više preuzima prostor i zbog toga se očekuje sve kraći rok razvoja softvera. Razvoj softvera vođen modelima, ukoliko se ispravno sprovede, omogućuje isporuku proizvoda u očekivanom roku. Implementacijom generatora koda za različite platforme, model može biti iskorišćen proizvoljan broj puta bez ikakvih izmena ili sa malim izmenama potrebnim da bi se specifikirale neke platformske specifičnosti.

Zadatak ovog rada je generisanje poslovnih aplikacija na bazi UML modela i skica korisničkog interfejsa razvijenih u okviru alata za vizuelno modelovanje poslovnih aplikacija Kroki [1]. Izabrana platforma je Django [2] u okviru Python [3] programskog jezika.

U sledećem poglavlju su prikazana slična rešenja iz ove oblasti. Treće poglavlje daje opis korišćenog standarda korisničkog interfejsa i dodatne informacije o tome kako je izvršeno mapiranje modelovanog entiteta na programski kod. Četvrto poglavlje opisuje značajnije korišćene tehnologije i alate koji se pojavljuju u Kroki sistemu. Peto poglavlje objašnjava bitne komponente aplikacije koje su razvijene pomoću Django razvojnog okvira. Šesto poglavlje sadrži opis rešenja zadatka, dok je u sedmom detaljno pokazana i objašnjena implementacija. Poglavlje osam sadrži primer aplikacije, generisan Kroki alatom. Deveto poglavlje sadrži zaključak rada zajedno sa smernicama daljeg razvoja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Gordana Milosavljević.

2. OPIS REŠENJA

U nastavku ovog poglavlja opisana je struktura i funkcionisanje KroGen podsistema. Prvi modul sistema je XML parser. Ovaj deo je zadužen za parsiranje XML datoteka koje kreira Kroki i za čuvanje svih potrebnih informacija u takvom obliku koji je pogodan za korišćenje u sledećim koracima. Posle parsiranja XML datoteka moraju biti dostupni svi potrebni podaci tako da generisanje bude moguće od strane različitih generatora koda. Ideja modularizacije je da promene budu omogućene u narednim modulima (adapter i generator), čak i ako su oni implementirani u drugim jezicima i na drugim platformama.

Drugi modul je Django adapter. Zadatak adaptera je da izvrši sve potrebne transformacije na podacima iz parsirane datoteke sa ciljem da olakša rad sa generatoru koda.

Treći modul procesa je generator koda. Pomoću njega se generiše datoteka sa izvornim kodom i dodatne datoteke sa konfiguracijom i sadržajem koje određuju vizuelni izgled veb stranica.

2.1 Parsiranje sadržaja xml datoteke

U toku parsiranja generišu se XML datoteke koje se zasnovane na EUIS DSL [4] jeziku. Ove datoteke sadrže sve potrebne podatke za generisanje aplikacija, kao što su podaci o menijima, forme i veze među formama kao i elemente i operacije koji se pojavljuju na formama. Nakon parsiranja, datoteke sadrže i hijerarhijsku organizaciju izvorne datoteke. Razlog za to je priprema za potencijalne promene u sledećim verzijama Django razvojnog okvira ili za modifikacije u Kroki alatu. Redosled parsiranja je sledeći: enumeracije, podaci o menijima, podaci o entiteima i podaci o panelima.

2.2 Pripremanje podataka za django generator koda

Ponekad se dešava da podaci koji su parsirani iz inicijalnog XML fajla nisu u potpunosti u odgovarajućem formatu da bi generator radio sa njima. Zbog ovih slučajeva je uveden adapter deo sistema. Podaci se prosleđuju generatoru samo ako su prošli preko adaptera.

2.3 Generisanje koda

Prvi korak u toku generisanja koda je brisanje postojećih datoteka koje su ostale od prethodnog generisanja. Datoteke se brišu zbog toga jer je generisanje uz čuvanje ručnih modifikacija izuzetno teško za implementaciju, pri čemu ručne izmene često ipak ne ostanu sačuvane. Umesto ručnih izmena generisanog koda, bolji pristup je razvoj strategije za integraciju ručno pisanog i generisanog koda, koji se nalaze u odvojenim datotekama.

Nakon brisanja, sledi generisanje strukture direktorijuma u projektu. Svi folderi se nalaze unutar foldera `generated/[imeProjekta]`, gde će „ime Projekta“ biti zamenjeno imenom generisanog projekta.

Posle toga, dolaze pogledi, modeli, forme i url adrese. Nakon toga se generišu Django HTML [5] šabloni i na kraju šabloni sa statičkim podacima. U standardnim panelima generišu se tri stranice za svaki entitet sa Freemarker šablonima za: pregled, unos novih podataka i za izmenu postojećih podataka.

Navedene funkcije je moguće redefinisati ili ih onemogućiti u toku modelovanja. Podaci o tome se takođe čuvaju u XML [6] datotekama. Za razliku od standardnih panela, u parent-child panelima generiše se jedna stranica koja sadrži prikaz child entita sa dodatnom funkcijom filtriranja.

2.4 Parent-child paneli

Za razliku od standardnog panela, parent-child paneli imaju samo jednu stranicu. Oni reprezentuju vezu između već postojećih entiteta i zbog toga nema potrebe za implementacijom dodavanja i brisanja parent-child entiteta. Međutim, neke operacije se mogu izvršiti i nad entitetima koji su povezani parent-child vezom.

Pošto ovakvi paneli reprezentuju veze, moguće je iskoristiti mnogo od ranije generisanih elemenata iz sistema, kao što su model baze i specifični delovi prikaza standardnih panela. Delovi koje je potrebno posebno generisati su pogledi (*view*) koji implementiraju logiku filtriranja i linkove u *urls.py* datoteci koji pokazuju na odgovarajuće poglede.

Hijerarhijska organizacija u okviru parent-child panela je dizajnirana po principu da iz bilo kog od dva ili više entita koji su povezani vezom „jedan prema više“ može da se odradi koji je parent (na strani veze gde je kardinalitet 1) a koji je child (na strani veze gde je kardinalitet „više“).

2.5 Ručno dodavanje koda

U poslovnim aplikacijama često se zahtevaju specifične funkcije čija implementacija se vrši ručno. Najčešće te funkcije nije moguće deklarativno napisati nijednim jezikom ili bi to zahtevalo toliko mnogo vremena da ne vredi uložiti vreme i trud u modelovanje. Pored toga, i generisanje bi takođe bilo previše sporo i glomazno.

Jedna od mogućnosti za dodavanje ručno implementiranog koda je da se u generisanom kodu odrede takozvane „zaštićene zone“ koje su osigurane od promene od strane generatora. Implementacija ovakvih generatora je komplikovana a čuvanje zaštićenih zona nije garantovano u svim situacijama (npr. ako neko zaboravi da označi početak ili kraj zaštićenih zona).

Ako je kod ručno implementiranih funkcija odvojen od generisanog koda, moguće je koristiti jedan o navedenih načina: nasleđivanje, proširivanje (delegacija) ili parcijalne klase i slično. Koristeći mogućnost Python jezika i strukturu Django razvojnog okruženja, ručno implementirani kod u takvim projektima se nalazi u posebnim datotekama pa je moguće koristiti navedene načine bez izmena generisanog koda.

2.6 Pokretanje aplikacije

U odnosu na prethodnu verziju Kroki aplikacije, napravljen je i taster za inicijalizaciju procesa generisanja koda i pokretanje generisane Django aplikacije na toolbaru. Klikom na taster, koraci se izvršavaju jedan po jedan i neposredno pre pokretanja Django servera izvršava se i migracija za kreiranje i modifikaciju baze podataka. Pokretanje Django aplikacije zahteva Python 3.4 i Django razvojni okvir verzije 1.8.5.

3. IMPLEMENTACIJA

U ovom poglavlju je detaljno opisana implementacija modula koji su navedeni u prethodnim poglavljima. Detaljno je opisano parsiranje i generisanje parent-child panela i gradivnih elemenata grafičkog korisničkog interfejsa koji koristi Bootstrap 3 biblioteku.

3.1 Parsiranje XML datoteka

Za parsiranje Kroki XML datoteka se koristi Java DOM parser biblioteka. Parsiranje podataka se vrši sekvencijalno, jer redosled parsiranja može uticati na rezultat. U prvom koraku se parsiraju enumeracije, pa meniji, zatim entiteti i standardni paneli i na kraju parent-child paneli. Prvi korak u procesu parsiranja je učitavanje podataka, pa DOM parser biblioteka transformiše tekst u DOM stablo i za svaki element stabla se generiše jedna enumeracija. Parsiranje parent-child panela se vrši na sličan način, samo sa tom razlikom da sada postoji više ugnježđenih XML elemenata. Ako u toku parsiranja parser naiđe na entite koji sadrže strani ključ drugog entiteta, proces se nastavlja sa drugom entitetom i nakon toga se parser vraća inicijalnom entitetu. Posle uspešnog parsiranja svi podaci će biti sačuvani u *DataModel* kontejner klasi.

3.2 Generator koda

Generator koda je realizovan preko *strategy* dizajn šablona. Pojedinačni delovi aplikacije su kreirani u odgovarajućim strategijama koje se pojedinačno dele na jednu generičku i apstraktnu strategiju. Prvo se inicijalizuje generator koda bez posebnih parametara, pa se nakon toga izvršava generisanje svake strategije, jedne za drugom. Poziva se *generate* metoda sa odgovarajućim anonimnim objektom za sve strategije pojedinačno. Glavni metod za generisanju prikazan i tačan redosled kreiranja se vidi takođe na Listing 1.

```
private void generateDjangoProject() throws
Exception {
    DjangoGenerator generator = new
    DjangoGenerator();
    generator.generate(new
    BasicFolderStructurePart());
    generator.generate(new
    ProjectSettingsPart());
    generator.generate(new ModelsPyPart());
    generator.generate(new FormsPyPart());
    generator.generate(new ViewPyPart());
    generator.generate(new URIsPyPart());
    generator.generate(new StaticFilesPart());
    generator.generate(new HomePyPart());
    generator.generate(new TemplatesPart());
    generator.generate(new CustomCodePart());
    generator.generate(new StaticTemplatePart());
}
```

Listing 1. Kod glavnog modula za generisanje

3.3 Generisanje views.py datoteka

Poslovna logika Django aplikacije je uvek implementirana u pogled (*view*) fajlovima, pa tako i u slučaju parent-child pogleda. Kao što je već spomenuto u prethodnim poglavljima, zadatak parent-child panela je da se prikažu entiteti koji učestvuju u vezi „jedan prema više“. Child paneli filtriraju svoj sadržaj na osnovu selektovanog reda u okviru parent panela. Pored toga, mogu se podešavati i prava pristupa pogleda tako da se zahteva logovanje korisnika, ako to još nije izvršeno.

U ovom fajlu su određeni i šabloni koji se moraju pozvati posle izvršavanja odgovarajućih metoda sa potrebnim parametrima za renderovanje sadržaja stranice.

3.4 Generisanje urls.py datoteka

Zadatak *urls.py* datoteke je da poveže url šablone sa odgovarajućim Python funkcijama iz fajla *views.py*. Pored toga, mogu da se podese razni linkovi koji nisu generisani ali su neophodni za rad aplikacije kao što su login stranica ili link za ručno pisani kod. Po definiciji, u Django sistemu svi linkovi moraju imati jedinstveno ime omogućavajući korišćenje ime linka u html šablonima u procesu razvoja, ali u toku renderovanja ime će biti zamenjeno odgovarajućom adresom. Url-ovi parent-child panela su generisani na isti način kao što su i za standardne panele.

3.5 Generisanje HTML Stranica sa parent-child panelom

Generisanje HTML stranica se vrši u dva koraka. U prvoj fazi se vrši definisanje Freemarker[7] šablona. U toku generisanja Django aplikacije na osnovu Freemarker šablona izvršava se kreiranje Django šablona koji se koristi za generisanje krajnje HTML stranice kad zahtevi stižu od pretraživača.

Oba obrađivača šablona podržavaju korišćenje hijerarhijske strukture u okviru šablona. Zahvaljujući ovom osobinom base.html služi kao roditeljski šablon za svaku stranicu. To je Django šablon, što znači da se u toku generisanja ne menja, samo se prekopira na odgovarajuće mesto u projektu.

Navigacioni modul koji je korišćen na svakoj stranici je implemetiran u navbar.html datoteci. Generisanje modula navigacije je bazirano na modelu i zbog toga sadržaj fajla zavisi od konkretnog entiteta u Kroki modelu. Modul za navigaciju je uključen u roditeljsku stranicu pomoću {% include "navbar.html" %} naredbe. Navbar.html je uključen u base.html a sve ostale stranice nasleđuju roditeljski base.html šablon i zahvaljujući ovome navigacija će biti prikazana na svakoj stranici.

Za generisanje parent-child stranice se koristi parentChild.ftl Freemarker šablon. Na osnovu ovog šablona u procesu generisanja kreiraju se toliko parent-child Django šablona koliko je bilo kreirano parent-child panela u Kroki modelu. Kad se pokrene veb aplikacija, nakon dobijenog zahteva od klijentskog pretraživača, traženi parent-child Django šablon će sakupiti potrebne podatke za one entitete koji moraju biti prikazani nakon renderovanja.

Šablon počinje sa nasleđivanjem base.html šablona a u narednom delu definiše se zaglavlje parent-child panela. Najveći deo sadržaja opisuje roditeljske i dečje panele a na kraju su zadata uključenja javascript biblioteke.

Deo koji se odnosi na panele obuhvata roditeljski panel sa svim entitetima koji su sačuvani u bazi. Ovaj deo šablona je naznačen komentarom <!--parent panel--> i generiše tabelu sa redovima koji odgovaraju pojedinačnim slogovima u bazi. Sa klikom na bilo koji red u tabeli svi dečji paneli će biti filtrirani na osnovu identifikatora reda. Filtriranje se vrši pomoću javascript funkcije getUrlParameter iz datoteke parentChild.js.

Za brisanje filtera koristi se taster Show all. Taster je prikazan u zaglavlju svim dečjim panelima i skida filtriranje sa svake tabele.

Deo šablona koji se bavi sa generisanjem dečjeg panele nalazi se između komentarima <!--child panel-->. Biće generisano toliko tabela koliko ima dečjeg panela povezanih sa roditeljskim panelom. Sve tabele sadrže slogove koje su usklađene u bazi a sadržaj tabela se menja ako je aktivirano filtriranje sa klikom u roditeljskom panelu.

Uključeni Javascript elementi na kraju šablona su odgovorni za funkcionisanje sajta na ispravan način. JQuery i Bootstrap[8] datoteke pomažu u prikazivanju novog izgleda sajta i obezbeđuju responzivne osobine. Dok manuelno napravljeni skriptovi omogućuju filtriranje i ispravno funkcionisanje menija.

3.6 Generisanje koda odgovornog za izgled stranice

Kod dizajniranja novog grafičkog interfejsa ciljevi su bili da bude savremeni, tako što prati sadašnje principe planiranja veb stranica, i da stranica bude responzivna. Responzivna osobina omogućuje da sa promenom veličine prozora pretraživač stranica dinamički menja sadržaj tako da skida, prikaže i promenjuje redosled elemenata. Takođe omogućuje ispravan prikaz i na mobilnim uređajima.

Iskoristeći nasleđivanje u korišćenim obrađivačima šablona sve je to napravljeno bez redundancije koda. Sve stranice su izgrađene tako da se navigacioni deo nalazi na gornjem delu, meniji se nalaze sa leve strane a preostali deo stranice služi za prikazivanje sadržaja.

4. PRIMER TEST APLIKACIJE

Ovo poglavlje sadrži primer kreiranja aplikacije na bazi modela koji je kreiran na master studijama. Služi kao dobar primer koji pokazuje rad alata na jednom realnom modelu. Dijagram je prikazan na Slika 1 i sadrži Player, Localization, Base i Match entitete kao i PlayerStateEnum i LocationStateEnum enumeracije.

Nakon skiciranja istog modela Kroki alatom, napravljen je nov dijagram proširen jednim parent-child panelom koji povezuje Player, Base i Match entitete. Ovaj diagram je prikazan na Slika 2

POREĐENJE PERFORMANSI NAPONSKI KONTROLISANIH RING OSCILATORA SA JEDNIM IZLAZOM I SA DIFERENCIJALNIM IZLAZIMA U CMOS TEHNOLOGIJI**PERFORMANCE COMPARATION OF SINGLE-ENDED AND DIFFERENTIAL VOLTAGE CONTROLLED RING OSCILLATORS IN CMOS TECHNOLOGY**

Jasmina Ćurčin, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu su predstavljene i poredene dve konfiguracije naponski kontrolisanog ring oscilatora sa jednim izlazom i naponski kontrolisanog ring oscilatora sa diferencijalnim izlazom, koji sadrže isti broj stepena u lancu, realizovanih u 0.25μm CMOS tehnologiji pomoću programskog paketa Cadence Virtuoso. Detaljno su opisane osobine naponski kontrolisanih oscilatora, a zatim je prikazano projektovanje dve vrste naponski kontrolisanih ring oscilatora. Prikazani su rezultati simulacija kao i poredenje performansi odabranih konfiguracija na osnovu dobijenih rezultata. Opisan je postupak postavke odabranih simulacija, kao i izrade i verifikacije leajuta. Predstavljena je alternativna konfiguracija koja sadrži neke poboljšane karakteristike u odnosu na prethodno odabrane konfiguracije.

Abstract – Design and performance comparison of single-ended and differential voltage controlled ring oscillator designed in 0.25μm CMOS technology using Cadence Virtuoso is presented in this paper. Theoretical background describes characteristics of chosen oscillators. Two configurations are designed and simulated. Results are compared and discussed and an alternative configuration is presented.

Ključne reči: ring oscilator, naponski kontrolisani ring oscilator, mikroelektronika

1. UVOD

Moderni komunikacioni sistemi zahtevaju generisanje stabilnog, periodičnog signala – takt signala, koji bi obezbedio vremensku bazu za operacije kao što su obrada signala, sinhronizacija, frekvenzijska sinteza, i tako dalje. Takt signal može biti ekstrahovan iz ulaznog signala, ali je u najvećem broju slučajeva generisan korišćenjem oscilatora, bilo postavljenog van čipa, bilo integrisanog na čipu. Oscilatori predstavljaju suštinski deo u mnogim elektronskim kolima, bilo da su korišćeni kao generatori takt signala, bilo da su korišćeni npr. za operacije u mobilnim uređajima, kao što je frekvenzijski prenos. Različita primena oscilatora zahteva i različite osobine oscilatora, a samim tim i različite načine projektovanja. Zbog velikih brzina, najpoželjnija konstrukcija oscilatora je ona koja može da se realizuje na samom čipu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Mirjana Damjanović.

Takođe, savremeni komunikacioni sistemi su zasnovani na operacijama koje se obavljaju korišćenjem različitih strukturnih blokova kod kojih je, zbog velikih brzina i performansi, veoma važno da su realizovani kao monolitna integrisana kola [1-2].

U ovom radu predstavljene su neke od mogućih konstrukcija naponski kontrolisanog oscilatora u konfiguraciji ring oscilatora. Objasnen je princip rada integrisanih oscilatora kao i potencijalni načini izrade oscilatora. Opisano je projektovanje dve vrste naponski kontrolisanih ring oscilatora korišćenjem programskog paketa Cadence. Prikazani su izgled šematika i leajuta za obe odabrane konfiguracije. Izvršene su simulacije i objašnjeni su koraci koji su upotrebljeni prilikom podešavanja alata *Analog Design Environment* programskog paketa Cadence i korišćenje simulatora *Spectre* za simulaciju kola radi obavljanja željenih analiza. Alternativna konfiguracija sa nekim poboljšanjima je predstavljena i diskutovani su dobijeni rezultati svih realizovanih konfiguracija.

2. PROJEKTOVANJE NAPONSKI KONTROLISANIH RING OSCILATORA**2.1. Jednostavni ring oscilator**

Ring oscilator predstavlja kombinaciju kaskadno povezanih stepena, koji se sastoje od kola za kašnjenje, gde je izlaz poslednjeg stepena putem povratne sprege povezan na ulaz prvog stepena[3].

Svaki stepen proizvodi kašnjenje, koje se označava sa t_p . Kada signal prođe kroz svaki od N stepena, on je fazno pomeren u vremenu za $N \cdot t_p$. Zatim, signal mora proći još jednom kroz svaki stepen da bi dostigao ostatak faznog pomeraja, rezultujući tako totalnom periodom od $2 \cdot N \cdot t_p$. Iz toga proizilazi da je perioda oscilovanja ovog kola definisana kao:

$$T = 2 \cdot N \cdot t_p \quad (1)$$

Gde N , dakle, predstavlja broj invertora, a t_p predstavlja kašnjenje kroz inverter.

Frekvencija oscilovanja je tada data kao:

$$f = \frac{1}{T} = \frac{1}{(2 \cdot N \cdot t_p)} \quad (2)$$

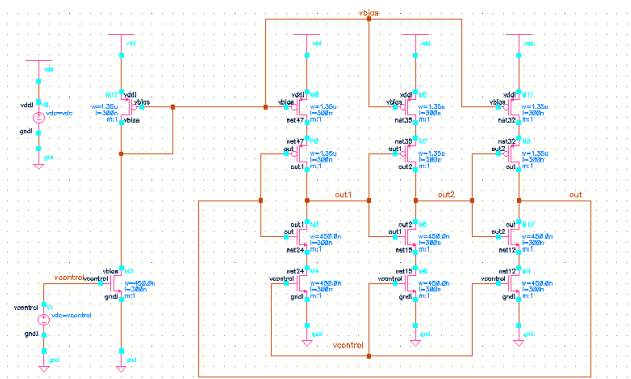
Pošto je u praktičnoj primeni oscilatora neophodno kontrolisati periodu oscilovanja, a samim tim i frekvenciju oscilovanja, pribegava se rešenju gde se frekvencija oscilovanja menja promenom kontrolnog

napona. Time se dolazi do rešenja da se frekvencija oscilovanja u ring oscilatoru kontroliše naponom V_{cntrl} .

2.2. Naponski kontrolisani ring oscilator sa jednim i sa diferencijalnim izlazom

Svaki stepen može biti ili sa jednim izlazom, ili sa dva komplementarna izlaza, tzv. diferencijalni [4].

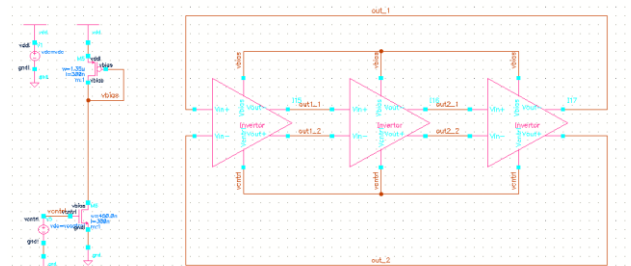
Na slici 1 je prikazano projektovano kolo, odnosno test benč naponski kontrolisanog ring oscilatora sa jednim izlazom. Da bi se obezbedilo da ovo kolo proosciluje, broj stepena za kašnjenje mora biti neparan.



Slika 1. Kolo naponski kontrolisanog ring oscilatora sa jednim izlazom

PMOS i NMOS tranzistori pomoću kontrolnih napona V_p i V_n , kontrolišu količinu struje koja će da protekne kroz inverter, odnosno kontrolišu struju pražnjenja kapacitivnosti invertora. Niža vrednost struje koja protiče kroz inverter rezultira dužom tranzicijom sa jednog nivoa signala na drugi, pa se time menja i frekvencija oscilovanja.

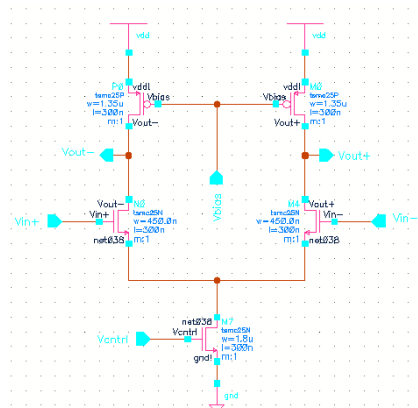
Na slici 2 je prikazano projektovano kolo, odnosno test benč, naponski kontrolisanog ring oscilatora sa diferencijalnim izlazom.



Slika 2. Kolo naponski kontrolisanog ring oscilatora sa diferencijalnim izlazom

Ovakva konfiguracija sadrži diferencijalne stepene za kašnjenje koji izgledaju kao što je prikazano na slici 3.

Diferencijalni ring oscilator može da poseduje i paran broj stepena, ukoliko kolo za kašnjenje ima istovremeno invertujući i neinvertujući izlazni signal. Povratna sprega u diferencijalnom VCO-u omogućava fazni pomeraj od 180° kroz oba kola za kašnjenje, jedan neinvertujući i drugi invertujući, čime dolazi do formiranja oscilacija.



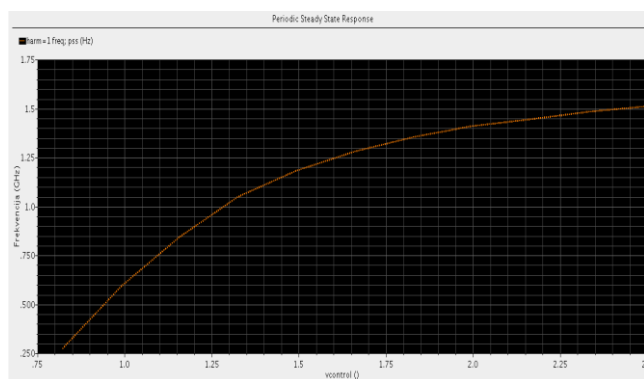
Slika 3. Diferencijalni stepen za kašnjenje

3. REZULTATI SIMULACIJA

Opseg frekvencija naponski kontrolisanog oscilatora se podešava promenom kontrolnog napona, $v_{control}$. Puštanjem tranzijene simulacije pri minimalnom i maksimalnom kontrolnom naponu može se odrediti opseg frekvencija oscilovanja projektovanih oscilatora.

Za naponski kontrolisani ring oscilator sa jednim izlazom, za $v_{control}$ u rangu od 0.75V do 2.5V dobija se opseg frekvencija od 202MHz do 1,5GHz. Ovo znači da je opseg podešavanja ovog naponski kontrolisanog ring oscilatora približno jednak 1,301GHz, a centralna frekvencija je jednaka 1,26GHz.

Na slici 4 prikazan je grafik zavisnosti frekvencije od promene napona $v_{control}$, korišćenjem PSS analize.

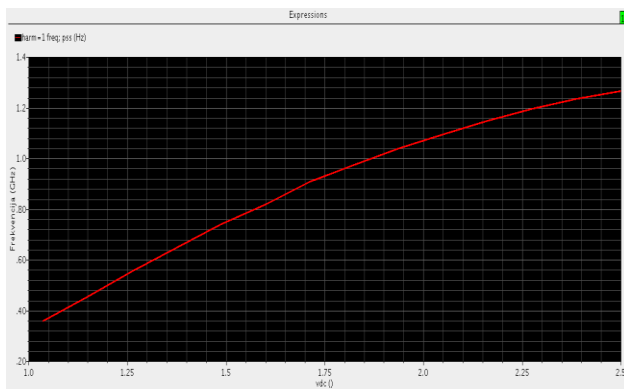


Slika 4. Grafik zavisnosti frekvencije od promene napona $v_{control}$.

Opseg u kojem je promena frekvencije linearna pri promeni kontrolnog napona je od 271MHz do 1GHz, pa je osetljivost za taj deo opsega, $KVCO(V_{cntrl})$, jednaka 1,62GHz/V.

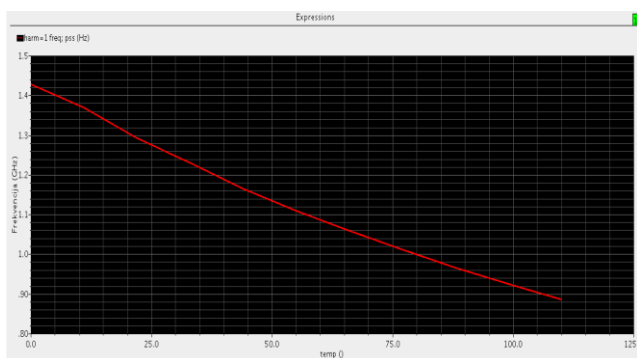
Uticaj promene napona napajanja na variranje centralne frekvencije je prikazano na slici 5.

Sa grafika se može uočiti da promenom napona napajanja centralna frekvencija varira od 357,9MHz do 1,268GHz, što znači da je osetljivost ovog oscilatora jednaka približno 0,62GHz/V.



Slika 5. Grafik promene frekvencije promenom napona napajanja.

Uticaj temperature na frekvenciju je prikazano na slici 6.



Slika 6. Grafik promene frekvencije promenom temperature.

Sa grafika se može primetiti kako promenom temperature od 0°C do 110°C frekvencija se smanjuje u opsegu od 1,428GHz do 0,886GHz, skoro potpuno linearno, što znači da je osetljivost u ovom slučaju oko 5MHz/°C.

Pomoću PSS simulacije se može predstaviti i šum faze, koji za ovu konfiguraciju iznosi od -92,2dBc/Hz do -96,83dBc/Hz pri ofsetu od 1MHz od centralne frekvencije.

Srednja vrednost potrošnje snage, za ceo opseg podešavanja, pri maksimalnoj vrednosti napona napajanja od 2,5V, kreće od 105,2μW do 858,2μW.

Sličnim postupcima dobijamo da je za naponski kontrolisani ring oscilator sa diferencijalnim izlazom frekvencija u opsegu od 554,7MHz do 2,44GHz., što znači da je opseg podešavanja ovog naponski kontrolisanog ring oscilatora približno jednak 1,301GHz, a centralna frekvencija je jednaka 1,56GHz.

Za razliku od prethodne konfiguracije, ova promena je linearna u celom opsegu, pa prosečna osetljivost iznosi 2,55GHz/V.

Podešavanjem vcontrol na njegovu srednju vrednost, gde je za nominalni napon od VDC=2,5V dobijena centralna frekvencija od 1,568GHz, dobijamo da frekvencija varira za oko 4,5% od svoje centralne frekvencije.

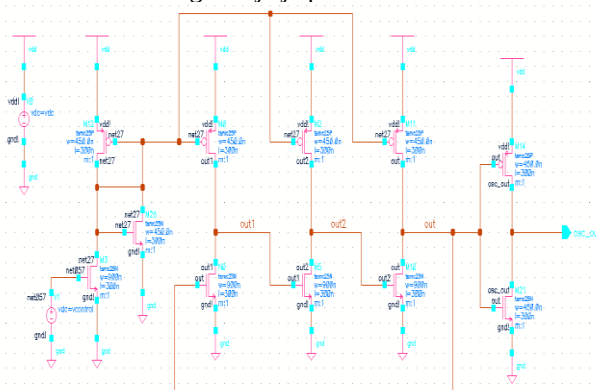
Promenom temperature od 0°C do 110°C frekvencija se smanjuje u opsegu od 1,812GHz do 1,16GHz, skoro linearno u celom opsegu podešavanja, što znači da je osetljivost ove konfiguracije jednaka oko 5,92MHz/°C

Pri minimalnom i maksimalnom kontrolnom naponu, šum faze se kreće od -88,48dBc/Hz do -89,56dBc/Hz na ofsetu od 1MHz od centralne frekvencije, a srednja vrednost snage potrošnje tokom celog opsega podešavanja frekvencije, pri maksimalnoj vrednosti napona napajanja od 2,5V, jednaka od 255,5μW do 1,63mW.

Kako su pokazali rezultati simulacija, diferencijalni ring VCO ima veću frekvenciju oscilovanja jer izlaz jednog stepena pobuđuje samo jedan gejst sledećeg stepena, dok izlaz single-ended stepena pobuđuje dva gejta. To znači da izlaz diferencijalnog ring VCO ima manju vrednost kapacitivnosti koju čine tranzistori svakog stepena, pa time postiže i veću frekvenciju oscilovanja.

S druge strane, ovim kompromisom je dobijena veća disipacija snage i veći fazni šum.

Alternativna konfiguracija je prikazana na slici 7.



Slika 7. Poboljšani dizajn naponski kontrolisanog ring oscilatora.

Pri vcontrol=0.75V, frekvencija iznosi 1,55GHz, a pri naponu vcontrol=2.5V iznosi 2,61GHz, što znači da je opseg podešavanja ove konfiguracije približno jednak 1,06GHz, a centralna frekvencija ovog oscilatora je jednaka 2,52GHz. KVCO(Vcntrl), za deo opsega podešavanja koji je linearan, iznosi približno 0,275GHz/V.

Promenom napona napajanja, frekvencija varira oko 78% od svoje centralne frekvencije, a vrednost osetljivosti je jednaka 1,35GHz/V, a za promenu temperature u rangu od 0°C do 110°C frekvencija se smanjuje u opsegu od 2,828GHz do 1,867GHz, što znači da je osetljivost ove konfiguracije na promenu temperature jednaka 8,73MHz/°C.

Vrednost šuma faze za ceo opseg podešavanja se kreće od -90,37dBc/Hz do -90,56dBc/Hz na ofsetu od 1MHz od centralne frekvencije, a prosečna potrošnja snage iznosi od 298μW do 1,605μW.

4. DISKUSIJA REZULTATA

Dobijeni rezultati pokazuju da ako se uporede vrednosti opsega podešavanja frekvencije, sve konfiguracije imaju veoma širok opseg podešavanja, s tim što predložena konfiguracija diferencijalnog ring naponski kontrolisanog oscilatora ima linearnu prenosnu karakteristiku u celom opsegu podešavanja.

Diferencijalni naponski kontrolisani ring oscilator, kao i alternativna konfiguracija mogu da dostignu više frekvencije oscilovanja, i to za skoro 1GHz, dok konfiguracija sa jednim izlazom može biti upotrebljena u aplikacijama koje zahtevaju nešto niže vrednosti frekvencije, kao što su PLL aplikacije korišćene u ultrazvučnim transiverima.

Predložena konfiguracija sa jednim izlazom i alternativna konfiguracija pokazuju dobre karakteristike što se tiče vrednosti izlazne amplitude, koja je skoro maksimalna, sa neprimetnim DC ofsetom u slučaju promene kontrolnog napona, dok je kod predložene konfiguracije za diferencijalnim izlazima, izlazna amplituda nešto manja, a takođe postoji i izvesni DC ofset pri promeni kontrolnog napona na ulazu.

Što se tiče osetljivosti na napon napajanja, kao što se može videti iz tabele, prednost ima diferencijalna konfiguracija u odnosu na druge dve, jer njena osetljivost ima mnogo manju vrednost, a zatim odmah iza nje, i alternativna konfiguracija, koja ima skoro duplo manju vrednost, nego konfiguracija sa jednim izlazom.

Ako se posmatra osetljivost na promenu temperature, prednost ima predložena konfiguracija sa jednim izlazom, jer ukoliko dođe do varijacije temperature za 1°C, i frekvencija će se promeniti za oko 5MHz, dok kod diferencijalne i alternativne za približno 6MHz i 8,7MHz, respektivno.

Upoređivanjem vrednosti šuma faze ovih konfiguracija, prednost svakako imaju ring oscilatori sa jednim izlazom, a posle njih odmah i data alternativna konfiguracija.

Prednost diferencijalnih ring oscilatora leži u tome što se šumovi, koji se unose putem napajanja, ili supstrata, pojavljuju kao zajednički na oba izlaza, pa su zatim poništeni već u sledećem stepenu. S tim u vezi, ova konfiguracija može biti primarna u integrisanim kolima sa velikim brojem digitalnih kola, iz razloga slabije osetljivosti na substratni šum i šum napona napajanja, kao i niži uneseni šum iz drugih kola na istom čipu [5].

Konfiguracije sa jednim izlazom se koriste u manje šumovitim okruženjima, pri potrebi za nižim frekvencijama, radi postizanja što boljih rezultata što se tiče fazno-šumnih karakteristika.

Alternativna konfiguracija može biti iskorišćena ukoliko je zahtevana jednostavnost, jer sadrži najmanje uređaja od druge dve konfiguracije, a može da dostigne više frekvencije od konfiguracije sa jednim izlazom, sa manjom potrošnjom snage od diferencijalne konfiguracije.

5. LITERATURA

[1] Paul R.Gray, Paul J. Hurst, Stephen H. Lewis, Roert G. Meyer : „Analysis and design of analog integrated circuits, 5th edition“, JohnWiley & Sons, Inc. United States of America, 2009.

[2] Behzad Razavi, : „Design of analog CMOS integrated circuits“, McGraw-Hill Book Co Singapore, 2001.

[3] A. Sharma, M. S. Manohar, G. Mishra, D. S. Rathore: „Compact Design of Ring and Voltage Controlled Oscillators for Wireless Devices“, *Electronics and Communication Engineering The LNM Institute of Information Technology, Jaipur*, International Journal of Engineering Research & Technology (IJERT), Vol. 2, No. 12, December 2013.

[4] Sanjay Agarwal, Ashish Kumar: „Phase-locked loops in an IC-based clockdistribution system“, Cypress Semiconductor Bangalore, 2013.

[5] Ali Hajimiri, Sotirios Limotyrakis and Thomas H. Lee: „Phase Noise in Multi-Gigahertz CMOS Ring Oscillators " , Center for Integrated Systems, Stanford, CA.

Kratka biografija:



Jasmina Ćurčin rođena je u Zrenjaninu 1986. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Mikroelektronika odbranila je 2016.god.

RAZVOJ INFRASTRUKTURE U MICROSOFT AZURE MICROSOFT AZURE INFRASTRUCTURE DEVELOPMENT

Milan Milosavljević, Željko Trpovski, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su opisani osnovni principi Cloud Computing-a, kao i neki od najvažnijih servisa koje nudi Microsoft Azure Cloud.

Abstract – This document describes the base principles of Cloud Computing and the most common services offered by Microsoft Azure.

Ključne reči: Azure, Cloud, Infrastruktura, Virtualizacija, CDN, Mreža

1. UVOD

Virtualizacija je današnji trend u IT industriji i imajući tu činjenicu u vidu, u radu se upoznajmo sa jednom od više platformi koje nude različite organizacije za pružanje usluga i Cloud servisa, a to je Microsoft Azure. Govorimo o korenima virtualizacije i šta je to zapravo Cloud. Upoznajemo se sa njegovim osnovnim prednostima i vrstama servisa koje Cloud nudi.

Upoznajemo se sa Windows Azure portalom, koje vrste korisnika postoje i koja je uloga svakog od njih. Govorimo o osnovnim funkcijama koje Azure portal nudi, o načinu funkcionisanja Azure-a, o svemu onome što korisnik ne vidi i o mestima gde se sve to fizički nalazi (data-centri), kao i šta je bitno da korisnik zna u vezi sa odabirom pravog Azure data-centra.

Pominjemo i neke od servisa koji znatno mogu da poboljšaju kvalitet i performanse kao i sigurnost sistema dizajniranog u Azure infrastrukturi.

2. VIRTUALIZACIJA I CLOUD

Većina velikih kompanija koristi razne vrste servera za potrebe poslovanja (Web serveri, FTP serveri, Domen kontroleri...). Ti serveri su umreženi međusobno preko rutera i svičeva.

Problem sa ovim pristupom gde imamo fizičke servere su određena ograničenja:

- Potrebno je imati stabilno napajanje svih komponenti (dodavanje rezervnih sistema za napajanje u slučaju da dođe do nestanka struje je dodatni trošak, naročito ako treba obezbediti veći broj servera),
- Potreban je stabilan sistem za hlađenje,
- Potrebna je stabilna internet konekcija,
- Sigurnost - ograničen fizički pristup prostoru gde se nalaze serveri.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Trpovski, vanr. prof.

Kako kompanija ne bi morala da brine o svim tim ograničenjima, ona obično angažuje provajdere koji će održavati njihovu infrastrukturu u svojim data-centrima (stabilni sistemi za napajanje, optimalno hlađenje i ventilacija, visoke brzine internet protoka, zaštita od prašine i visoka fizička sigurnost).

Sve se to prevazilazi virtualizacijom gde se na jednom 'velikom' serveru sa jakim CPU-om i više RAM memorije, pokreću virtuelne mašine manjih kapaciteta. Tako se može lakše odgovoriti na zahteve, a ovim pristupom se štedi i fizički prostor, jer je više fizičkih servera virtualizirano i spakovano u jedan.

To je osnovni princip kojim se vode moderni data-centri. Dakle, više 'velikih' fizičkih servera na kojima se nalaze po više 'manjih' virtuelnih servera.

Još jedna ideja jeste i ponuditi korisniku da sam koristi i kreira virtuelne mašine (servere) preko Web interfejsa, bez kontaktiranja provajdera. Ovim se zapravo postavljaju osnovni principi Cloud-a.

Cloud je zapravo evolucija već postojećih tehnologija i cela ideja Cloud Computing-a nije samo virtualizacija servera, već mnogo više.

Cloud pruža više vrsta servisa:

- SaaS (Software as a Service) - korišćenje softvera preko interneta. Primer su Office 365 i Gmail. To je prva vrsta servisa koju je Cloud ponudio.
- PaaS (Platform as a Service) – korišćenje softverskih alata bez instalacije na računaru. Jedan od primera je i sam Windows Azure, gde se mogu pokretati aplikacije preko portala (Visual Studio, SQL baze podataka).
- IaaS (Infrastructure as a Service) - najčešći oblik servisa, kada je reč o Cloud-u, gde provajderi pružaju mogućnost korisniku da sam kreira Virtuelne Masine (servere), pa čak i kompletan sloj infrastrukture.

Postoje Javni, Privatni i Hibridni Cloud. Servisi koje pruža Javni Cloud su, kao što i samo ime kaže, javno dostupni za korišćenje preko interneta. Privatni Cloud je ograničen na neku organizaciju (kompaniju), gde se čuvaju podaci i biznis aplikacije koje ne trebaju biti dostupne javnosti već samo unutar organizacije. Privatnim Cloud-om može upravljati sama organizacija ili neka iznajmljena, eksterna, kompanija.

Hibridni Cloud je kombinacija prethodna dva. Osetljivi podaci se čuvaju unutar firme, dok neke manje osetljive stvari mogu biti javne.

Cloud Computing–om se može smatrati i pristup i čuvanje podataka na internetu umesto na lokalnom računaru (hard disku). Na taj način se uvek može pristupiti istim podacima sa bilo kog računara (dovoljna je internet konekcija). To znači da se podaci čuvaju na nekom ‘udaljenom’ mestu, a sinhronizuju se sa povezanim uređajima.

Postoji više razloga zašto je Cloud postao trend u današnje vreme, a dinamička alokacija i dislokacija resursa je jedan od glavnih razloga.

Korisnik može vrlo jednostavno povećavati, odnosno, smanjivati performanse svojih virtuelnih mašina (scale up, scale down) pa čak i varirati njihov broj (scale-out, scale-in) u zavisnosti od trenutnih potreba. Time se lakše može odgovoriti na trenutne pikove i potrebe kompanije, a pri čemu one uštede dosta novca u odnosu na situaciju kada bi poslovali sa fizičkim serverima.

Takođe značajna prednost jeste i pristup resursima sa bilo kog mesta, bez obzira na lokaciju hardvera. Ovim je jasnije da je i samo održavanje i konfigurisanje mrežne infrastrukture lakše, a pri čemu se ne mora brinuti o licencama za pojedinačne servere.

Čuvanje podataka u Cloud-u je sigurnije. Teže je nepovratno izgubiti podatke, a o čuvanju rezervnih kopija korisnik ne mora da brine, već će to provajder obaviti za njega u skladu sa zahtevima.

Postoji više organizacija koje nude Cloud servise, a Windows Azure je samo jedna od njih.

Windows Azure, koji je Marta 2014-te preimenovan u Microsoft Azure, jeste Cloud Computing platforma koju je razvio Microsoft i koja služi za razvoj i održavanje aplikacija i servisa preko globalne mreže Microsoft-ovih data-centara.

Korisnik može brzo da razvije i održava svoju aplikaciju i to koristeći više različitih programskih jezika, alata i framework-a, kako onih koje je razvio sam Microsoft, tako i drugih, eksternih tehnologija.

Servisi koje pruža Microsoft Azure su PaaS i IaaS.

3. AZURE PORTAL

Azure Management Portal je korisnički interfejs preko koga korisnici mogu da kreiraju i upravljaju svojim servisima i aplikacijama.

Postoji više tipova korisnika u Azure portalu, a to su:

- Account Administrator
- Service Administrator
- Co-Administrator

Account Administrator kreira subskripciju (pretplatu) i organizuje način plaćanja.

Kada Account Administrator kreira subskripciju, on ujedno postaje i Service Administrator nad tom subskripcijom, ali može tu rolu dodeliti nekom drugom korisniku.

Service Administrator upravlja subskripcijom i resursima u njoj. Sve subskripcije moraju imati tačno jednog Service Administrator-a. On može odrediti još nekog korisnika da bude Co-Administrator u subskripciji.

Co-Administrator ima sva prava kao i Service Administrator, sem što ne može promeniti, niti izbrisati Service Administrator-a.

Subskripcija može imati više Co-Administrator-a.

U Azure portalu je moguće kreirati web aplikacije, virtuelne mašine (servere), mobilne servise, cloud servise, zatim virtuelne mreže u kojoj se mogu izolovati virtuelne mašine i time imati veća kontrola kad je u pitanju dozvoljena komunikacija između servera unutar mreže i spoljnog saobraćaja.

Takođe korisna komponenta je i ‘Azure Traffic Manager’ koji služi da rutira saobraćaj ka najbližem data-centru, a ukoliko najbliži data-centar nije dostupan iz bilo kog razloga, saobraćaj se dalje usmerava ka drugom data-centru. Upravljanje saobraćajem i pravilima rutiranja vrši korisnik.

Azure nudi više vrsta aplikacionih servisa koji služe da poboljšaju ili prošire opus aplikacija. Nama najinteresantniji modul iz ove sekcije je CDN (Content Delivery Network) koji služi za keširanje podataka kojima se najčešće pristupa i čime se povećavaju performanse.

4. KAKO FUNKCIONISE AZURE

Kada pričamo o komponentama ili servisima koje nudi Azure, pitamo se kako to sve funkcioniše i šta se tačno dešava u data-centrima.

Unutar data-centra nalazi se mnogo grupa mašina ili servera spojenih preko switch-a. Ta grupa se naziva klaster (**cluster**). Svim klasterima upravlja Fabric kontroler.

Možemo reći da je Fabric kontroler mozak Azure servisa koji analizira procese i donosi odluke. On upravlja svime što je u vezi sa virtuelnim mašinama (Load balancer, switch...). Svaka mašina ima instaliranog Fabric agenta koji trči u njoj i preko koga Fabric kontroler sakuplja podatke o virtuelnoj mašini.

Korisnik može kreirati više vrsta virtuelnih mašina, sa različitim operativnim sistemima i performansama (CPU, RAM...).

Kada korisnik kreira virtuelnu mašinu, on bira operativni sistem, a svim daljim update-ima i patch-evima na operativnom sistemu upravlja Fabric kontroler. On odlučuje gde će nova aplikacija trčati kao i na koji fizički server treba locirati novo-kreiranu virtuelnu mašinu, kako bi se optimizovalo korišćenje hardvera.

Pored upravljanja resursima, Fabric kontroler takođe prati ispravnost hardvera i vrši preraspodelu u slučaju pada sistema (kvara nekog od hardverskih komponenti).

Jasno je da je Fabric kontroler vrlo značajan deo Azure arhitekture.

5. AZURE DATA CENTRI

Kada pomislimo na Azure data-centar, zamišljamo velike zgrade sa velikim brojem servera. Upravo to i jesu Azure data-centri. Jednostavno, mora postojati prostor gde se čuvaju svi ti podaci. Microsoft ima data-centre raspoređene svuda po svetu odakle se upravlja svim Azure servisima.

Data-centri su podeljeni po regionima, a tačna lokacija se ne otkriva iz očigledno bezbednosnih razloga.

Kako izabrati pravi data-centar?

Biranje data-centra je deo strategije i treba obratiti posebnu pažnju na to. Prilikom svakog kreiranja Azure servisa, bilo da kreiramo Web Aplikaciju, Virtuelnu mašinu ili nešto drugo, potrebno je izabrati region.

Performanse

Treba izabrati data-centar koji će biti najbliži korisnicima aplikacije, jer se smanjuje brzina pristupa, a samim tim se povećavaju performanse. Dakle, treba analizirati tržište i postaviti aplikaciju tamo gde se očekuje najviše korisnika.

Cena

Cena usluga može da varira između data-centara. To su male varijacije, ali se na jednoj stvari može uštedeti: treba koristiti jedan region za sve servise koji međusobno komuniciraju, čime se izbegava naplata saobraćaja između data-centara.

Legalnost

Zakon o slobodi deljenja različitih informacija varira od države do države, tako da se i na to mora obratiti posebna pažnja.

Dostupni servisi

Nisu u svim regionima dostupni svi Azure servisi, uglavnom zbog hardverskih ograničenja. Lista dostupnih servisa po regionima je prikazana na Azure-ovoj Web stranici

<https://azure.microsoft.com/en-us/regions/services/>.

6. VIRTUELNE MREŽE (VIRTUAL NETWORK-VNET)

Isto kao i u slučaju kada imamo fizičke servere povezane u lokalnoj mreži, tako se i virtuelne mašine povezuju u VNET-u.

Dakle, servisi i virtuelne mašine koje pripadaju istoj virtuelnoj mreži mogu da komuniciraju međusobno preko privatnih IP adresa. U svakoj kreiranoj virtuelnoj mreži se može definisati više pod-mreža (subnet), onoliko koliko dozvoljava opseg IP adresa virtuelne mreže.

Kada kreiramo VNET u Azure-u, mi definišemo region kome ona pripada, i sve VM koje se kreiraju u toj mreži, takođe pripadaju tom regionu (nasleđuju). Zato je bitno unapred definisati strategiju.

Kada se svi serveri nalaze u virtuelnoj mreži kreiranoj u Azure, onda je to 'čista' Cloud virtuelna mreža (Cloud-only). Međutim, moguće je povezati VNET sa nekom lokalnom mrežom računara preko site-to-site tunela, i tada se radi o hibridnoj (Cross-Premises) mreži. Ograničenje je da se pri tom ne mogu preklapati adresni opsezi, a i ova vrsta mreže zahteva i VPN uređaj na strani lokalne mreže.

Adresni opsezi VNET-a se dodeljuju prema CIDR (Classless Inter-Domain Routing) notaciji. Dozvoljeni opsezi privatnih IP adresa su:

- 10.0.0.0/8: 10.0.0.0 – 10.255.255.255
- 172.16.0.0/12: 172.16.0.0 – 172.31.255.255
- 192.168.0.0/16: 192.168.0.0 – 192.168.255.255

Kreiranjem pod-mreža delimo mrežni adresni opseg na više pod-opsega. Na primer, mrežu sa adresnim opsegom 192.168.0.0/16 možemo podeliti na sledeće pod-opsege (pod-mreže):

- 192.168.1.0/24 gde se mogu adresirati Web serveri
- 192.168.2.0/24 gde se mogu adresirati FTP serveri
- ...

VNET kreiramo u Azure portalu tako što unapred znamo strategiju. Potrebno je neke stavi unapred definisati. Jednom kad se dodele resursi mreži, teško je ili možda i nemoguće izvršiti izmene.

Velike kompanije, naročito one koje posluju i pre same pojave ideje o virtualizaciji, uglavnom imaju već izgrađenu infrastrukturu i lokalnu mrežu fizičkih ili virtualiziranih servera u svojim poslovnim zgradama. Oni u najčešćem slučaju žele postepeno da migriraju svoje servere u Cloud i, naravno, žele da im svi serveri i dalje budu povezani međusobno. To je moguće (Hibridne mreže) i jedini uslov je postojanje Gateway-a na obe strane, a samim tim i VPN uređaja na strani lokalne mreže.

Takođe, mogu se povezati i dve virtuelne mreže, koje mogu pripadati kako različitim regionima, tako čak i različitim subskripcijama.

Postoje sledeći načini povezivanja:

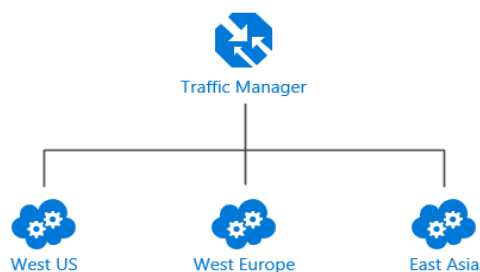
- **Point-to-Site** (Povezivanje jedne lokalne mašine, fizičke ili virtualne, sa VNET-om preko SSTP (Secure Socket Tunnel Protocol) protokola. Pristup virtuelnoj mreži se odobrava kroz sertifikat koji mora biti instaliran na svakoj lokalnoj mašini koja će biti konektovana na VNET.)
- **Site-to-Site** (Povezivanje dve mreže. Moguće je povezati VNET sa nekom lokalnom mrežom računara ili sa nekom drugom virtuelnom mrežom (VNET-to-VNET).

Site-to-site konekcija je brža od point-to-site konekcije i lakše se prenose podaci. Potreban je samo 'shared key' za pristup mreži i za razliku od point-to-site konekcije, nije potrebno instalirati sertifikate na svakoj lokalnoj mašini.

7. TRAFFIC MANAGER

Azure Traffic Manager služi da se distribuira saobraćaj između različitih Azure data-centra (više istih servera u različitim data-centrima). On koristi DNS (Domain Name System) da usmeri klijenta na najprikladniji server po metodi najbržeg traffic-routing-a (Slika 1).

Recimo da imamo jedan Web server lociran u nekom od evropskih Azure data-centar-a, a drugi Web server lociran u nekom od severno-američkih Azure data-centara. Preko Traffic Manager-a se klijenti iz, recimo Srbije, usmeravaju na prvi, a klijenti iz, recimo Kanade, usmeravaju na drugu Web server. Ovde je ipak bitno razumeti, da se odluka na koji server treba usmeriti klijenta, ne donosi na osnovu geografski najbližeg servera, već na osnovu najbržeg traffic-routing-a (sve zavisi od kvaliteta mreže).

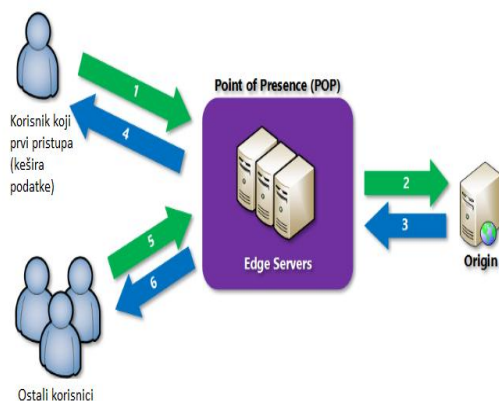


Slika 1. *Traffic Manager*

8. CDN

Jedan od načina da se poboljšaju performanse aplikacije ili website-a jeste keširanje određenih podataka. Azure koristi keširanje kako bi povećao brzinu Cloud servisa. CDN (Content Delivery Management) postavlja blob i druge statičke podatke u keš. Sam proces podrazumeva postavljanje podataka na strateški izabrano mesto i keširanje istog. Kao rezultat, korisnici dobijaju na raspolaganje najveći mogući bandwidth.

Kako CDN zapravo radi u praksi (Slika 2)?



Slika 2. *Opis rada CDN-a*

1. Kada korisnik zatraži neki fajl koristeći URL sa posebnim imenom domena (DNS), kao na primer, **www.ftn.com/neka-slika.jpg**, DNS ga usmerava na najbolju/najbližu POP (Point-of-Presence) tačku. POP sadrži servere gde se keširaju podaci (Edge serveri). Ovo je uglavnom POP koji je geografski najbliži korisniku.
2. Ako u POP serverima nema zahtevanog fajla u kešu, oni dalje kreiraju zahtev ka Origin-u (originalno mesto gde se file nalazi). Origin može biti Azure Web aplikacija, Azure Cloud servis ili bilo koji javno dostupan Web server. U našem primeru, to je **www.ftn.com** Web server.
3. Origin vraća zahtevani fajl (neka-slika.jpg) ka Edge serveru sa kog je primio zahtev. Origin sa fajlom, Edge serveru šalje i informaciju o tome koliko dugo je potrebno čuvati taj fajl u kešu, TTL (Time-to-Live).

4. Edge server šalje zahtevani fajl korisniku i kešira fajl sve dok ne istekne TTL. Ako se drugačije ne specificira, podrazumevani TTL je 7 dana.
5. Ostali korisnici mogu zahtevati taj isti fajl (URL) i mogu biti poslani na isti POP.
6. Ukoliko TTL nije istekao, Edge server vraća zahtevani fajl korisniku iz svog keša, bez daljeg zahteva ka Origin-u. Ovo je znatno brže i doprinosi ukupnom utisku korisnika.

Azure CDN servise nudi u saradnji sa partnerima kao što su 'Verzion' i 'Akamai'. Oni imaju POP lokacije geografski raspoređene svuda po svetu.

Jasno je da su prednosti korišćenja CDN-a povećana brzina (performanse), a činjenicom da se podaci keširaju na Edge serverima, smanjuje se broj zahteva ka Origin serveru, čime se znatno smanjuje i opterećenje samih Origin servera.

9. ZAKLJUČAK

U radu smo se upoznali nekim osnovnim servisima koje Azure nudi kada je u pitanju sloj infrastrukture. Objasnili smo kako se određeni servisi funkcionišu i to posebno svaki od njih. Jasno je, međutim, da je potrebna analiza osnovnih zahteva pre samog početka kreiranja infrastrukture, kako bi se sve upakovalo u jedno dobro i stabilno okruženje.

Naravno, ovo je samo deo onoga što Azure nudi. Postoji još mnogo Azure servisa vezanih za različite IT oblasti, a koji nisu usko vezani za infrastrukturu.

10. LITERATURA

[1] <https://docs.microsoft.com/en-us/azure/> , posećeno oktobra 2016. godine

[2] <https://www.tutorialspoint.com>, posećeno oktobra 2016. godine

Kratka biografija:



Milan Milosavljević rođen je u Raški 1987. god. Diplomski-master rad odbranio je 2016. godine na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacije



Željko Trpovski rođen je u Rijeci 1957. godine. Doktorirao je na Fakultetu tehničkih nauka 1998. god. Oblast interesovanja su telekomunikacije i obrada signala.

LOGIČKA KRIPTOANALIZA HEŠ FUNKCIJA

LOGICAL CRYPTOANALYSIS OF HASH FUNCTIONS

Vuk Šavija, Željens Trpovski, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Tema ovog rada jeste logička analiza kriptografskih heš funkcija. Analiza je rađena pomoću alata CryptLogVer za generisanje CNF (conjunctive normal form) koji su razvili P. Morawiecki i M. Srebrny [1]. CNF se potom proslijeđuje SAT rješavaču [2] koji zatim pokušava da riješi problem inverzije. Ovom metodom se pokazalo da je Kečakova heš funkcija sigurna na napad prikazan u radu.

Abstract – The subject of this paper is logical crypto-analysis of hash functions. The analysis has been done using the CryptLogVer tool, design to generate CNF (Conjunctive Normal Form) which was introduced and developed by P. Morawiecki and M. Srebrny [1]. CNF is then passed to the SAT solver [2] whose purpose is to solve the inversion problem. Using this method, Keccak hash function is proved to be safe for the presented attack.

Ključne reči: heš, analiza, SAT, CNF, keccak, inverzija

1. UVOD

Posljednjih godina heš funkcije postaju neizostavan faktor u kriptografskim sistemima. Široko su rasprostranjene i koriste se kod zaštite integriteta poruke, digitalnog potpisa...

Tema ovog rada je logička kriptanaliza heš funkcija. Pošto se danas u SHA3 protokolu koristi Kečakova heš funkcija, u tekstu koji slijedi biće opisan njen dizajn, i jedna metoda kriptanalize. Na kraju će biti predstavljeni rezultati ove metode koja koristi HDL kod za generisanje CNF formule i metode koja koristi programski jezik c++. Na samom kraju je zaključak i budući koraci u razvoju ove metode.

2. KEČAK HEŠ FUNKCIJA

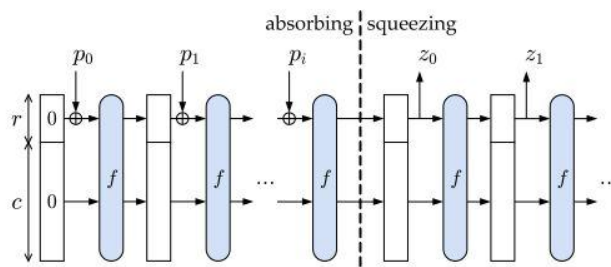
Kečakova heš funkcija je izabrana za SHA3, jer se pokazala kao najotpornija na moguće napade (inverzija, slaba otpornost kolizije i jaka otpornost kolizije).

2.1. Princip dizajniranja

Kečak (Keccak)[3] je familija heš funkcija koja je dizajnirana na principu sunderastih konstrukcija slika 1.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željens Trpovski, vanr. prof.



Slika 1 Sunderasta konstrukcija

Za razliku od mnogih heš funkcija koje koriste jaku funkciju kompresije, Kečak koristi duže stanje, a sigurnost se postiže korakom (iteracijom). Dva glavna parametra su r brzina heš funkcije (bitrate) i c kapacitivnost, čija suma definiše stanje nad kojim radi Kečak. Za SHA3 protokol dužina stanja je b je 1600 bita, dok se dužina heša označava sa n . Relacije između ovih parametara su: $r = b - c$, gdje je $c = 2n$. Vrijednost kapacitivnosti c mora da zadovolji sigurnosne zahtjeve, stoga ne može da se bira proizvoljno. Vrijednost kapacitivnosti je validna ako je kompleksnost napada na inverziju i slabu otpornost kolizije 2^n , a na jaku otpornost kolizije $2^{n/2}$, (pogledati tabelu 1). Različite vrijednosti za b i c daju kompromis između brzine i sigurnosti. Drugim riječima što je brža heš funkcija to je njena sigurnost manja. Takođe, važno je napomenuti da se parametar b bira tako da uvijek važi da je $r > n$, gdje je n maksimalno 512 bita.

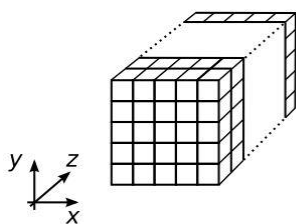
Tabela 1. Kompleksnost napada na sunder-funkcije. c - dužina "kapaciteta", n - dužina heša, L - dužina poruke čiji kolizioni par se traži.

	P-sunder	T-sunder
otpornost na inverzije	$\min(2^{c/2}, 2^n)$	$\min(2^c, 2^n)$
slaba otpornost na kolizije	$\min(2^{c/2}, 2^n)$	$\min(2^c/L, 2^n)$
jaka otpornost na kolizije	$\min(2^{c/2}, 2^{n/2})$	$\min(2^{c/2}, 2^{n/2})$

Kečak se odvija u dvije faze, apsorpcije (absorbing) i faze računanja heša (squeezing). U prvoj fazi nad ulaznim blokovima poruke dužine r se vrši operacija ekskluzivno ili sa prvih r bita stanja. Ova operacija se miješa sa ostalim operacijama Kečakove funkcije kompresije (u nastavku samo f). Druga faza počinje tek kada se svi blokovi ulazne poruke obrade, i u ovoj fazi se izbacuje prvih r bita kao heš. Faza se završava kada se dobije zadana dužina heša.

Podrazumjevanje vrijednosti parametara za Kečak su $r = 1024$ i $c = 576$, iz čega slijedi da dužina stanja funkcije kompresije $b = 1600$, gdje je ovaj niz predstavljen kao

trodimenzionalni $s[x][y][z]$ dimenzija $5 \times 5 \times 64$, a funkcija kompresije se sastoji od 24 runde, slika 2.



Slika 2 Ilustracija stanja funkcije Kečak

U slučaju da dužina ulazne poruke nije cjelobrojni umnožak dužine bloka, vrši se dodavanje nula (*padding*). Ono se razlikuje od ostalih heš funkcija zbog toga što kod sunderaste konstrukcije nije dozvoljeno, iz sigurnosnih razloga, da poslednji blok poruke bude ispunjen nulama, pa se stoga osim na kraj ulazne poruke, bit 1 dodaje i na kraj dopunjene poruke.

2.2. Logička analiza

Logička kriptanaliza heš funkcije svodi se na SAT-problem (*SAT - satisfiability*), odnosno na Bulov problem zadovoljivosti. Naime, ovaj problem se svodi na to da se za zadanu Bulovu formulu pronađe niz ulaznih promjenljivih koji daju na izlazu kao rezultat Bulove formule logičku 1. SAT je jedan od prvih problema koji je dokazan da pripada klasi *NP-kompletnih* problema [3]. Algoritmi, odnosno programi koji rješavaju ove probleme nazivaju se SAT-rješavači.

Jedan od najvažnijih koraka u napadu na kriptografske primitive pomoću SAT rješavača jeste generisanje CNF formule. Takva formula u potpunosti opisuje kriptografsku primitivu ili njen segment koja je meta napada. U logičkoj kriptanalizi generisanje CNF-a predstavlja najkompleksniji izazov. Postoji mnogo načina da se dođe do konačne CNF formule, što dovodi to toga da rezultati variraju (broj klauzula, prosječna dužina kaluzule i broj literala).

Ovaj problem se može riješiti korišćenjem programskog jezika C++, kao što su predstavili Jovanović i Jančić u [4]. Međutim, da bi se na ovaj način došlo do rješenja, potrebno je mnogo znanja iz oblasti programiranja. Prvo je potrebno da se napiše program koji u potpunosti opisuje kriptografsku primitivu. Zbog bolje efikasnosti poželjno je koristiti algoritme minimizacije kao što su Karnoove mape (*Karnaugh maps*), Kvin-Meklaski (*Quine-McCluskey*) ili espresso (*Espresso*) algoritam.

Kao rezultat dobija se CNF formula koja se dalje prosleđuje SAT rješavaču. Praksa je pokazala da je ovaj proces jako iscrpljujuć i skon greškama. Na sreću [1] su razvili alat koji u mnogome olakšava posao generisanja CNF formule.

Njihova metoda koristi dvije odvojene aplikacije. Prva je Quartus 2 – softver razvijen od strane Altera koji služi za analizu i sintezu HDL (*Hardware Description Language*) dizajna. Za potrebe generisanja CNF-a dovoljana je i besplatna verzija *Quartus II Web Edition*. Ovaj softver

ima ugrađen alat pomoću kog se generišu Bulove jednačine od prethodno opisane kriptografske primitive u programskom jeziku *system verilog*. Druga aplikacija neophodna za dobijanje CNF-a je *CryptLogVer* [5] koja vrši konverziju dobijenih Bulovih jednačina u CNF. Prije pozivanja SAT-rješavača, potrebno je još samo da se kod dobijenog CNF-a podesi određeni ulaz.

U ovom radu obrađen je napad na inverziju (*preimage attack*), što znači da je za zadanu vrijednost heša h potrebno pronaći poruku m takvu da važi: $f(m)=h..$

Napad se odvija u tri koraka:

1. Konverzija CNF-a pomoću *CryptLogVer*
2. Namještanje izlaznih bita (heša), kao i dijela ulaznih bita (*padding*)
3. Pokrenuti SAT-rješavač za zadani CNF

Sa ovim se napad završava. SAT-rješavač će pokušati da pronađe poruku m čija vrijednost heša odgovara zadanim bitima (drugi korak napada).

3. REZULTATI

Da bi se ispravno konvertovale Bulove jednačine dobijene pomoću Quartus 2 Web Edition u CNF neophodno je uraditi sledeće:

1. Na računaru pronaći direktorijum gdje je Quartus 2 instaliran i tamo pronaći folder quartus
2. Napraviti poddirektorijum *Projects* i unutar njega napraviti folder za željeni projekat (npr SHA1)
3. Unutar foldera napravljenog pod 2 kopirati kod napisan u *szstem verilog-u* kao i konverter *HDL2CNF.jar* (*CryptLogVer*) [6]
4. Iz komandne linije pozvati:

```
java -jar HDL2CNF.jar project_name
number_of_inputs number_of_outputs
```

Dobijena CNF datoteka je u DIMACS formatu koji je široko prihvaćen kao standardan format za SAT-rješavače. Ulazna datoteka može da sadrži komentare (linija počinje sa slovom c). Najvažniji dio ulazne datoteke predstavlja linija u kojoj je definisan broj promjenljivih i klauzula (*p cnf variables clauses*). Ostale linije određuju klauzulu: pozitivan literal je označen odgovarajućim pozitivnim brojem, dok je negativni literal označen odgovarajućim negativnim brojem. Poslednji broj u liniji bi trebao da bude nula.

Primjer:

c primjer *.cnf datoteke.

```
p cnf 4 3
1 -3 0
2 3 -1 0
3 -1 2 4 0
```

Od interesa su promjenljive koje predstavljaju ulazne, odnosno izazne bite. Na primjer, ako je generisan CNF za SHA1 (*sha1.cnf*) koji ima 512 bita na ulazu i 160 bita na izlazu (dužina heša), onda promjenljive od 1 do 512 predstavljaju ulazne bite, dok promjenljive od 513 do 672 su izlazni biti. Da bi se izveo napad inverzije treba da se

dodijele vrijednosti promjenljivih heša, i to pozitivan broj označava logičku jedinicu, dok negativan broj označava logičku nulu:

```
-513 0
-514 0
515 0
516 0
-517 0
...
672 0
```

Na kraju treba da se pokrene SAT-rješavač za ovako izmjenjenu *.cnf datoteku. Primjer poziva npr *PrecoSAT*:

```
./precosat -v sha1.cnf > results.txt
```

Pokazalo se da napad inverzije za kompletne *SHA1* i *SHA3* nije moguć, tj. dobijeni broj varijabli i klauzula je prevelik da bi postojeći SAT-rješavači mogli u realnom vremenu da pronađu rješenje.

U tabeli 2 su prikazani rezultati dobijeni ovom metodom za heš funkcije *SHA1* gdje je dužina ulaza 512 bita, a dužina heša 160 i *SHA3* 256/256, dok su u tabeli 3 prikazani rezultati dobijeni metodom [7].

Tabela 2 Rezultati HDL metode

	Broj klauzula	Broj varijabli
SHA 1	31431	186342
SHA 3	165339	772816

Tabela 3 Rezultati C++ metode [7]

	Broj klauzula	Broj varijabli
SHA1	63545	212878
SHA3	184341	625642

Pregledom rezultata može da se zaključi da obe metode daju veoma slične rezultate. Može se vidjeti da za *SHA1* metoda opisana u ovom radu daje bolje rezultate, tj. manji broj varijabli i klauzula koje su potrebne SAT-rješavaču da bi pronašao inverziju.

Sa druge strane za *SHA3* algoritam rezultati su pomješani. Dok se metodom opisanom u ovom radu dobija manji broj klauzula, metodom [7] se dobija manji broj varijabli.

4. ZAKLJUČAK

U ovom radu primjenjen je napad inverzije pomoću SAT-rješavača na *SH1* i *SHA3* algoritme kriptografskih heš funkcija. Pokazalo se da su ove funkcije i dalje dosta jake za ovaj tip analize. Iz rezultata se može zaključiti da ni ova metoda, a ni metoda [7] ne daju obećavajuće rezultate.

Logička kriptanaliza daje dobre rezultate, samo na redukovane heš funkcije.

Dalji pravac razvoja ove metode mogao bi da donese bolje rezultate, tako što bi se npr. konstruisali SAT-rješavači samo za ovakve tipove problema, da ne budu kao što je sada slučaj, opštenamjenski.

Drugi pravac logičke kriptanalize bi mogao da bude pravljenje novih algoritama koji u velikoj mjeri minimizuju broj potrebnih promjenljivih i klauzula da bi se neka kriptografska primitiva opisala i samim tim bi se pomoglo trenutnim SAT-rješavačima da brže pronađu rješenje.

5. LITERATURA

- [1] A SAT-based preimage analysis of reduced
- [2] Biere, A. 2009 P{re,i}coSAT@SC09. SAT 2009 competitive events booklet
- [3] https://en.wikipedia.org/wiki/Boolean_satisfiability_problem
- [4] D. Jovanović and P. Janičić, "Logical Analysis of Hash Functions," in FroCoS, ser. Lecture Notes in Computer Science, B. Gramlich, Ed., vol. 3717. Springer, 2005, pp. 200–215.
- [5] Morawiecki, P., Srebrny, M. and Srebrny, M. 2010. Towards CryptLogVer toolkit for logical cryptanalysis and verification. to appear.
- [6] <https://news.jchck.net/article/d694e6b3699959d4e074db6e82faf38a0ce17092>
- [7] M. Milić and V. Šenk, "Uniform Logical Cryptanalysis of CubeHash Function," Facta Universitatis Series : Electronics and Energetics, 2010. [Online]. Available: <http://factae.elfak.ni.ac.rs/fu2k103/9cubehash.pdf>

Kratka biografija:

Vuk Šavija rođen je u Kozarskoj Dubici 1987. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacije – obrada signala odbranio je 2016.god.

ISPITIVANJE POSTROJENJA ELEKTROLUČNE PEĆI „R-S SILICON“ TESTING OF SUBMERGED-ARC FURNACE FACILITY “R-S SILICON”

Dušan Galić, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Opisano je postrojenje elektrolučne peći za proizvodnju silicijum metala u Mrkonjić Gradu, Republika Srpska. Dat je prikaz tehnološkog procesa, glavni dijelovi postrojenja, kontrola i monitoring rada peći. Prikazani su rezultati mjerenja parametara kvaliteta električne energije, kao analiza dobijenih rezultata, te njihovo poređenje sa odgovarajućim standardima.

Abstract – In the thesis the 36 MVA submerged-arc furnace for production of silicon metal is described. An overview of technological process, main part of the facility, control and monitoring of working process is presented. Results of power quality parameters measurements are given, together with analysis of obtained results and then their comparison with standards

Ključne reči: Elektrolučna peć, kvalitet električne energije, viši harmonici

1. UVOD

Kvalitet električne energije obuhvata razne tehničke i ekonomske aspekte korišćenja električne energije i danas je jedan od najvažnijih parametara uspešnog i pouzdanog rada kako električnih uređaja (potrošača), tako i prenosnih i distributivnih električnih mreža. Jedan od najvažnijih parametara kvaliteta isporučenog napona jesu viši harmonici, koji su danas široko rašireni u elektroenergetskom sistemu [1].

Poznati su brojni negativni efekti prisustva viših harmonika u mreži, od kojih su najpoznatiji pojava rezonancije, oštećenja na kondenzatorskim baterijama, dodatni gubici u električnim mašinama, ometanje MTK signala i dr.

Pored energetske elektronske pretvarača, značajan izvor viših harmonika predstavljaju i elektrolučne peći. S obzirom da se radi o postrojenjima velikih snaga (nekoliko desetina MW), potrebno je posebnu pažnju posvetiti merama za eliminisanje ili umanjenje viših harmonika.

U radu će se razmatrati konkretno postrojenje elektrolučne peći za dobijanje silicijum metala firme „R-S Silicon“ u Mrkonjić Gradu (BIH). Cilj je da se izvrše merenja harmonika napona i struje i da se poređenjem sa postojećim evropskim standardima utvrdi da li ova peć predstavlja generator viših harmonika za električnu mrežu JP „Elektroprivreda“ Republike Srpske.

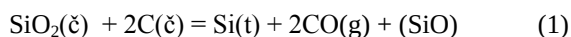
2. O POSTROJENJU „R-S SILICON“

Firma „R-S Silicon“ je osnovana 2012. godine u Mrkonjić Gradu od strane italijanske firme Metalleghe Spa sa sjedištem u Fleru, Breša. Nju karakterišu savremena tehnologija u procesu proizvodnje, visok nivo zaštite životne sredine i interna laboratorija, koja vrši striktnu kontrolu osnovnih sirovina i finalnog proizvoda. Gradnja fabrike silicijum metala počela je u maju 2014 godine i za samo 18 meseci uspešno završena investicija. Fabrika je projektovana i izvedena da zadovoljava najviše standarde zaštite životne sredine, primenjena je najmodernija i tehnološki najsavremenija oprema koja omogućava kvalitetno i sigurno obavljanje poslova.

Godišnja proizvodnja ove fabrike treba da dostigne 16.000 t silicijum metala visoke čistoće, koja se dobija iz potopljene elektrolučne peći snage 22 MW. Kao proizvod procesa dobija se tečni silicijum metal visoke čistoće, koji se prvo hladi, a zatim drobi na veličinu tražene granulacije ili u obliku kondenzovane silicijumske prašine.

3. OPIS TEHNOLOŠKOG PROCESA

Proizvodnja silicijum metala se može tretirati kao primarni kontinualni metalurški proces, gdje se silicijum metal proizvodi karbotermičkom redukcijom oksidne rude (kvarca) i gdje dolazi do reakcije metal-oksida sa ugljenikom iz koje se dobija metal i uljen-monoksid unutar otvorene rotirajuće elektrolučne peći [2]. Sirovine koje se koriste u proizvodnji se prvo mješaju, a zatim kontinualno dodaju u elektrolučnu peć odozgo, preko cilindričnih razgranato raspoređenih faza, gdje se one zagrijavaju i spuštaju prema vrućoj unutrašnjoj zoni peći. Ukupna hemijska reakcija unutar peći može da se napiše na sledeći način:



Silicijum-oksidi (SiO) je međuproizvod redukcije kvarca i ugljenika, pretvarajući se tokom procesa najvećim dijelom u silicijum. Tokom procesa peć rotira tako da približno napravi puni krug za 6-7 dana jer na taj način obuhvati rekativno područje oko sve tri elektrodne faze.

3.1 Sirovine

Sirovine korištene u procesu proizvodnje su: kvarc, drvenu ugalj, kameni ugalj i drvena sječka. Dobar kvarc, podoban sa proizvodnju mora da zadovoljava različite hemijske osobine koje se odnose na količinu silicijum dioksida te na količinu primjesa, takođe i fizičke osobine koje se odnose na granulaciju, parčanje, nečistoće. Udio SiO₂ mora da bude iznad 99,0%. Uporedo sa izborom najčistijeg kvarca proizvodnju treba osigurati po

NAPOMENA:

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

mogućnosti sa najčistijim ugljeničnim reducentima, koji imaju mali udio pepela. Kao reducenti najčešće se koriste drveni ugalj, petrol koks (kameni ugalj sa niskim udjelom pepela), drvena sječka kao dobar stabilizator rada peći. Primjenom usitnjenog drveta u industrijskoj proizvodnji silicijuma pronađen je izvrstan stabilizator za rad peći. Iako ima slabiju poroznost i reakcijsku sposobnost da bolje provodi električnu struju i lakše se grafitizira, on ima preimućstvo zbog malog sadržaja pepela, što ima veliki značaj za dobijanje silicijuma. Zbog svoje visoke provodljivosti, pogodno je da se miješa sa elektrootpornijim reducentima. Karbonske elektrode predstavljaju prepečene elektrode od amorfnog uglja i služe ne samo kao provodnici struje unutar peći nego i kao pravo redukciono sredstvo.

3.2 Priprema smjese i fizičko hemijski uslovi procesa

Ispravno podešenom smjesom sa fizičkog i hemijskog gledišta, kao i strogom kontrolom pogona mogu se postići veća iskorištenja kvarca i redukcionih sredstava kao i smanjenje specifične potrošnje električne energije (MWh po toni tečnog silicijum metala).

U kadi peći se pod uticajem električne energije odigrava oko desetak različitih procesa manje ili više povezanih. Kada se smjesa sa sirovinama ubaci u elektropeč, polako propada na dole pri čemu raste temperatura svih sastojaka smjese. Izvjesna količina reducenata se pali već na samoj površini i sagorjeva. Neposredno ispod površine dio silicijum-oksida se kondenzuje prema sledećoj reakciji



Prvi ozbiljniji procesi redukcije dešavaju se na temperaturama iznad 1.500 °C, dok pravo bitno pomjeranje ravnoteže u stranu reduciranja može se očekivati na temperaturi od 1.800 °C i pritisku od jednog bara. U početku se obrazuje nešto silicijum-oksida da zadovolji reakciju



Na nešto nižoj temperaturi od 1.600 °C silicijum karbid, kao nus-proizvod, biće obrazovan prema relaciji



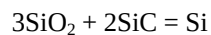
Kako je parcijalni pritisak SiO nizak, ukupna reakcija u ravnotežnoj tački, je vrlo bliska ovoj:



Obrazovanje karbida kao međuproizvoda u proizvodnji Si zavisi od hemijske aktivnosti ugljeničnog materijala i količine sastava materijala, koja sadrže Si. Što je veća hemijska aktivnost ugljeničnog materijala to je defektnija struktura SiC, što uzrokuje i višu hemijsku aktivnost u odnosu na okside Si. Isto se može konstatovati za termičku stabilnost karbida. Što je veća reakciona sposobnost ugljeničnog materijala upotrebljenog za sintezu silicijum karbida to on bolje sublimira na nižim temperaturama. Sa porastom temperature sadržaj SiO u gasu raste prema reakciji:



do temperature 1.800 °C kada silicijum postaje stabilan. U zoni iznad 1.800 °C mehanizmi reakcije nisu u potpunosti razjašnjeni, ali najvjerovatnije slijede reakcije:



4. INSTALACIJA I DIJELOVI POSTROJENJA ELEKTROLUČNE PEĆI

Glavni dijelovi postrojenja predstavljaju: transformatorska stanica 110/20 kV i sam proizvodni pogon, kojeg čine nekoliko cijelina: postrojenje rukovanja sirovinama, postrojenje elektrolučne peći, drobilično i filtersko postrojenje.

4.1 Transformatorska stanica 110/20 kV

Osnovni dijelovi trafostanice su: postrojenje 110 kV, transformacija 110/20 kV/kV, kompenzacija reaktivne energije, relejna kućica, komandno-pogonska zgrada u kojoj se nalazi 20 kV rasklopno postrojenje i ostali prateći objekti (jama za ulje sa kanalizacijom i dr.). Razodna postrojenja 110 kV izvedena su na otvorenom prostoru sa aparatim i sigurnosnim razmacima za spoljnu montažu i locirana tako da omogućava najpovoljnije uvođenje 110 kV dalekovoda. 110 kV postrojenje izgrađeno je sa jednim sistemom cjevnih sabirnica i četiri polja:

- Polje E01 – mjerno polje
- Polje E02 – DV polje 110 kV Mrkonjić Grad
- Polje E03 – transformatorsko polje 110/20 kV
- Polje E04 – DV polje 110 kV Jajce 1

Energetski transformator je prenosnog odnosa 110±10x1,5%/20 kV, snage 46 MVA, sprege YNd11 i zaštićen je sa obe strane odvodnicima prenapona. Postrojenje 20 kV smešteno je u posebnu prostoriju u komandno-pogonskoj zgradi energetskog objekta. Čelije su u „metal enclose“ tehnologiji, izvlačive, atestirane, slobodnostojeće, kompletno opremljene i sa ormarićem u vrhu za smeštaj sekundarne opreme. To su čelije tipa Siemens NXAIR-M sa jednim sistemom sabirnica naznačenog napona 24 kV i ima ih šest. Komenzacija reaktivne energije je mješovita, a postoji kompenzacija na srednjem naponu gdje su 4 kompenzatorske čelije od po 5,25 MVar. Kompenzacijsko postrojenje nalazi se u sklopu trafostanice i priključeno je na vodno polje kompenzacije 20 kV postrojenja.

4.2 Elektrolučna peć

Glavni dio fabrike "R-S Silicon" elektrolučna peć, koja se sastoji iz nekoliko funkcionalnih dijelova: kada peći zajedno sa rotacionim stolom, dimne kape, elektrodnih stubova koji drže i kontrolišu karbonske elektrode, sistema cijevnih, vodom hlađenih provodnika i tri monofazna transformatora sa preklopkom zvijezda/trougao. Napajanje peći električnom strujom kreće od SN čelije, odakle 20 kV kablovima dolazi do preklopke zvijezda/trougao, a odatle do tri transformatora snage 12 MVA. Ovi transformatori na sekundaru transformišu napon u rasponu od 120-290 V, a na sekundaru se nalazi i sistem vodom hlađenih provodnika. Monofazni transformatori su snage 12 MVA, odakle se može izvući zaključak da je nazivna snaga ove potopljene elektrolučne peći 36 MVA.

Najzanimljiviji dio ove peći se, nalazi se na niskonaponskoj strani, gdje se cijevni provodnici povezuju pomoću fleksibilnih bakarnih provodnika sa elektrodnim stubovima. Sekundarna strana pećnih transformatora je izvedena tako da postoji 16 izlaza od kojih osam odlazi ne jednu, a 8 na drugu elektrodu.

Povezujući sekundare sva tri transformatora na taj način dobija se sprega trougao na sekundarnoj strani. Kroz cijelu dužinu elektrodnih stubova postoje takođe cijevni provodnici, vodom hlađeni, koji se pružaju skroz do kontaktnih ploča u dnu elektrodnih stubova. Kontaktne ploče, zajedno sa pritisnim prstenovima, nalaze se u kontaktu sa elektrodom i vrše određeni pritisak dovoljan da se ne izgubi kontakt, a ni prevelik da se bi užarila sama elektroda. Na kontaktnim pločama se električna struja predaje okolini (elektrodi i smjesi unutar kade).

4.3 Postrojenje rukovanja sirovinama

Sirovine iz skladišnih prostora do trenutka ulaska u peć prolaze kroz postrojenje, koje čine dva sistema silosa zajedno sa trakastim transporterima. Mogu se razlikovati nekoliko funkcionalnih cijelina i to:

- Punjenje silosa sirovinama
- Doziranje pojedinačnih sirovina i njihovo miješanje
- Transport smjese u dnevne silose
- Doziranje materijala u peć

4.4 Drobilično postrojenje

Proizvedeni silicijum metal nakon hlađenja, ulazi u drobilicu, koja pomoću otvaranja i zatvaranja čeljusti ga drobi. Tako isitnjen proizvod odlazi na sistem trakastih transportera, koji ga dalje transportuje na dva vibraciona sita u zavisnosti od tražene granulacije, tj. na vibraciono sito VS-01 za granulaciju od 0-120, odnosno na VS-04 za granulaciju 0-50.

4.5 Filtersko postrojenje

Sistem za otprašivanje se sastoji od izmjenjivača toplote, dva ciklonska sistema, vrećastog filtera te ispušnog ventilatora i ima zadatak da smanji emisije prašine kao posljedicu rada elektrolučne peći pri proizvodnji Si-metala. Vrećasti filter ima središnji uvodni/ispušni distribucijski kanal da se gasovi dovedu na 12 komora. Najvećeg značaj za tretman izduvnih gasova, gasova iz peći te tretman prašine ima veliki ventilator, koji je smješten na prizemnom nivou i stvara podpritisak potreban za usis i prolaz vazduha kroz filter. Opremljen je senzorima temperature i vibracije, zvučnom izolacijom, kao i spojevima za uvod i ispuh. Iz 12 komora za filtraciju i prikupljanje prašine, ona prolazi kroz transportni sistem. Ono što povezuje filter sa transportnim sistemom su 12 rotacionih dozatora na koji se nastavljaju trake za odvod prašine do dva silosa u kojima se odvija proces densifikacije. Proces densifikacije se odvija na taj način da postoje za svaki od silosa otpuhivač, koji uduvava vazduh u donji dio silosa i tako miješa silikatnu prašinu u njemu.

5. MONITORING RADA SISTEMA

Kontrola rada peći se odvija u kontrolnoj prostoriji na prvom spratu zgrade, poznatom kao operatorski sprat. Ekipe koje borave i vrše kontrolu rada su radnici koji rade na punjenju i raspoređivanju materijala na površini peći, te vrše pregled parametara pomoću kojih se vrši kontrola optimalnog rada peći. Takođe, iz ove prostorije se vrši kontrola filterskog postrojenja i postrojenja rukovanja sirovinama.

Kontrola 36 MVA potopne elektrolučne peći izvedena je pomoću Siemens PLC-a i SCADA sistema. SCADA sistem se sastoji od PLC-a, dve „human-machine interface“ operatorske stanice i računara sa „Autofurn“ programom. PLC je Siemens serije 300 sa ulazno-izlaznim jedinicama u pogonu (*remote I/O boards*). Interfejs je razvijen pomoću WinCC 7.2 SCADA sistema. Centralna procesorska jedinica je proizvođača Siemens tipa CP319 2 DP/PN. Mreža se dalje grana na nekoliko mrežnih prstenova koji su povezani pomoću optičkih link modula (*optical link module OLM*). Može se primjetiti četiri različite pod-mreže, prva povezuje ulazno izlazni ormar polja u trafostanici (*remote I/O board*) sa oznakom RT-03, druga ormar koji napaja svo pomoćno napajanje peći (ventilatore, hidraulične pumpe itd), treća prikuplja signale sa pećnih transformatora, sistema elektroda, sistema vodom hlađenih cijevi, iz postrojenja za tretman vode i postrojenja za rukovanje sirovinama, a četvrta signale sa drobiličnog i filterskog postrojenja. Sve ove mreže su povezane putem optičkih modula sa glavnim PLC-om.

6. MJERENJE VIŠIH HARMONIKA

Mjerenja kvaliteta električne energije, prvenstveno viših harmonika, vršeno je pomoću analizatora Fluke 435 (*Power quality analyzer, Fluke 435*) u periodu od 29.08.2016 od 17:00h sve do 05.09.2016 do 17:00h. Mjerenje je ukupno trajalo 169 sati. Instrument je bio priključen u sredjenaponskoj ćeliji J02, koja predstavlja dolazno napajanje, na sekundarnoj strani mjernih strujnih i naponskih transformatora (slika 1).

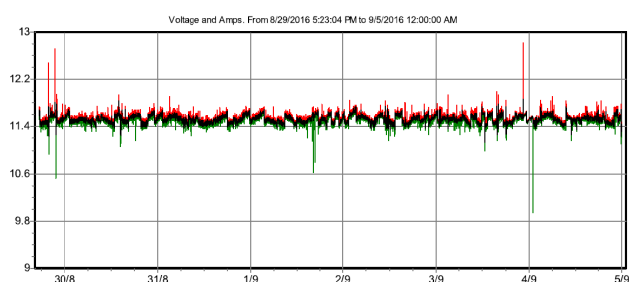


Slika 1. Način priključenja mernog uređaja.

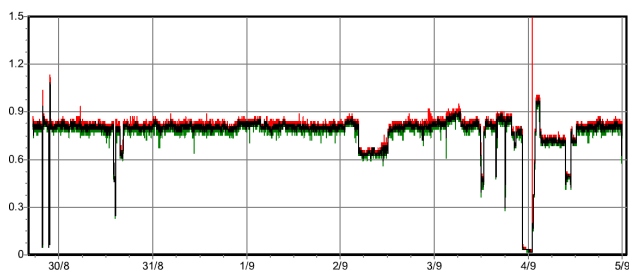
Od velikog broja zabeleženih parametara, kao prikaz rezultata mjerenja biće pokazane vrijednosti napona i struje jedne faze, te njihovo harmonijsko izobličenje.

7. REZULTATI MJERENJA I POREĐENJE SA STANDARDIMA

Napon je meren na 20 kV sabirnici, gdje fazni napon iznosi 11,5 kV. Rezultati su prikazani na slici 2. U ovome slučaju dominantna je jedna faza, jer su ostale imale slične vrednosti. Na slici 3 prikazani su rezultati mjerenja struje elektrolučne peći, koja je uglavnom bila nominalne vrednosti, osim u nekoliko slučajeva, od kojih je specifičan 4. 9., kada je pogon bio skoro u praznom hodu. Pri normalnim pogonskim uslovima, bez uzimanja u obzir prekida napajanja, 95% 10-min. srednjih efektivnih vrednosti napona nedelnog intervala merenja mora biti u opsegu $U_n \pm 10\%$, odnosno u opsegu 10,38 kV - 12,69 kV, dok preostalih 5% može biti u rasponu $U_n \pm 15\%$, odnosno u opsegu 9,81 kV - 13,27 kV [3]. Posmatrajući grafik sa slike 2 može da se zaključi da je napon u zadatim granicama.



Slika 2. Napon na 20 kV sabirnici (7 dana mjerenja).



Slika 3. Struja elektrolučne peći (7 dana mjerenja).

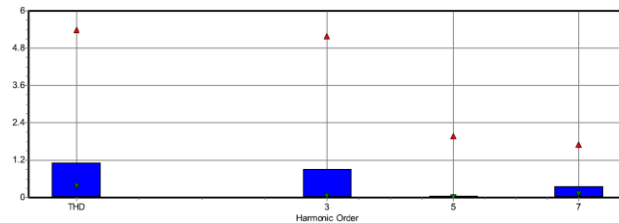
Harmonijskom obradom signala napona i struje dobijene su vrednosti viših harmonika prikazane na slikama 4 i 5. Na slici 4 prikazani su treći, peti i sedmi harmonik napona, zajedno sa ukupnim harmonijskim izobličenjem (engl. *Total Harmonic Distortion, THD*). Posmatrajući vrednosti THD po fazama zaključuje se da one iznosi oko 1 % i da je najveći uticaj 3-ćeg harmonika napona. Vrijednost ovog harmonika je nešto malo manja od 1 %, dok je 5-tog i 7-mog oko 0,5%.

Ako se posmatra grafik harmonika struje izraženih u procentima (slika 5), vidi se da su izraženi treći, peti i sedmi harmonik struje, a dato je THD struje. Posmatrajući THD po fazama zaključuje se da ono iznosi oko 4,5% i da je izraženiji treći harmonik struje, dok peti i sedmi harmonik struje imaju slične vrijednosti. Vrijednost trećeg harmonika je nešto manja od četiri procenta negdje oko 3,5 %, a peti i sedmi harmoni iznose 0,5 %.

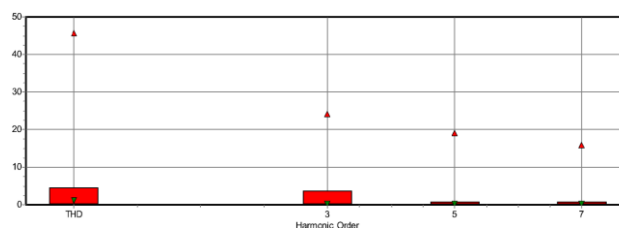
Na području Evropske unije se kvalitet napona osigurava u skladu sa standardom EN 50160, koji je usvojen i u Srbiji pod oznakom SRP EN 50160 [3]. Osim napona, za kvalitet električne energije je bitan i karakter struje opterećenja, ali on nije definisan sa EN 50160. Preporuke koje posmatraju kvalitet struje u distributivnim mrežama su one sa područja elektromagnetne kompatibilnosti

(EMC), ali su to IEC standardi (engl. *International Electrotechnical Commission, IEC*) IEC 61000-3-2, IEC 61000-3-4 i IEC 61000-3-6.

Sve izmerene vrednosti harmonika napona su znatno ispod maksimalno dozvoljenih standardom EN 50160 [3], dok su dobijene vrednosti harmonika struja takođe ispod odgovarajućih limita iz IEC standarda [4].



Slika 4. Viši harmonici napona jedne faze



Slika 5. Viši harmonici struje jedne faze.

8. ZAKLJUČAK

Mjerenje kvaliteta energije u postrojenju elektrolučne peći „R-S Silicon“ pokazuje da su vrijednosti efektivnih vrednosti napona i viših harmonika napona i struje nalaze unutar propisanih graničnih vrijednosti definisanih preporukama EN50160.

9. LITERATURA

- [1] V. Katić, „Kvalitet električne energije – viši harmonici“, Edicija tehničke nauke – Monografije, Broj 6, Fakultet tehničkih nauka, Novi Sad, 2002.
- [2] I.J. Barker, A.B. Stewart, „Inductive reactance and the operation of large submerged-arc furnaces“, *Journal of The South African Institute of Mining And Metallurgy*, March 1980, pp.123-128.
- [3] EN 50160:1999, "Voltage characteristics of electricity supplied by public distribution systems", CENELEC, 1999.
- [4] IEC 61000 Family of standards, „Electromagnetic Compatibility“, Geneva, 1995.

Kratka biografija:



Dušan Galić rođen je u Banja Luci 1990. god. Master studije upisuje 2013 god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2016.god.



Vladimir Katić je doktorsku disertaciju odbranio na Univerzitetu u Beogradu 1991. god. Redovan profesor Univerziteta u Novom Sadu je od 2002. god. za oblast Energetska elektronika, mašine i pogoni i OIE. Prodekan je FTN-a i šef Katedre za energetska elektroniku i pretvarače. Oblasti interesovanja su energetska elektronika, kvalitet električne energije i obnovljivi izvori električne energije.

OSOBINE SAVREMENIH SISTEMA ZA VIDEO NADZOR

MODERN SURVEILLANCE SYSTEMS' PROPERTIES

Nenad Protić, Željko Trpovski, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su iznesene razlike između sistema video nadzora sa analognim i sa digitalnim prenosom signala. Opisane su karakteristike i prednosti IP sistema za video nadzor, kao i kamere i oprema koja se u njima koristi. Predstavljene su osobine video servera i softvera za upravljanje videom, mrežne tehnologije u sistemima video nadzora, kao i CCTV sistem.

Abstract – This paper surveys the differences between surveillance systems with analog signal transmission and surveillance systems with digital signal transmission. This paper describes performances and advantages of IP surveillance systems as well as cameras and other used equipment. Video server performances, network technology in surveillance systems and CCTV systems are presented as well.

Ključne reči: Video nadzor, kamera, server, IP, CCTV.

1. UVOD

Savremeno doba praćeno je sve bržim razvojem tehnologije, interneta i računarskih mreža, te smo svedoci stalnog prelaska sa analognih na digitalne sisteme. U skladu sa tim, razvijaju se i elektronske komponente, u okviru kojih je jako značajno i korisno pomenuti razvoj IP kamera. IP kamere mogu se primenjivati u različitim granama društva, a u ovom radu akcenat je stavljen na njihovu primenu u oblasti sistema za video nadzor. Ovakav tip kamera nastao je pomenutim prelaskom na digitalne sisteme i potiskivanjem zastarelih analognih sistema video nadzora.

Sam termin IP (Internet protokol) kamera podrazumeva tip digitalne video kamere koja omogućava primanje i slanje podataka preko računarske mreže i interneta. Video nadzor u čiju svrhu se koriste ovakve kamere pruža mogućnosti kao i klasični analogni sistemi video nadzora, uz razliku da postoje nove funkcije i opcije, nedostupne u analognim sistemima.

Danas je pojam video nadzora, normalna pojava na svakom uglu. Video nadzor predstavlja način na koji se može nadgledati određeni prostor, bilo da se radi o zaštiti od krađe, kontroli zaposlenih, bezbedonosnih razloga ili o nečemu drugom.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Trpovski, vanr. prof.

Kroz ovaj rad dat je osvrt na savremene sisteme za video nadzor, CCTV sisteme i njihovu uporednu analizu, navedene su vrste sistema za video nadzor i njihove osnovne karakteristike i govori se konkretno o IP sistemu video nadzora, koji je najviše istaknut u okviru celokupnog rada, kao najpopularniji sistem nadzora koji je u upotrebi u različitim oblastima. Prikazane su osnovne mrežne tehnologije koje se mogu koristiti u jednom sistemu, obuhvatajući LAN, internet i bežične mreže, i dat je kraći prikaz CCTV sistema za video nadzor i njihovih osnovnih karakteristika.

2. SISTEMI ZA VIDEO NADZOR

Na osnovu tehnologije prenosa signala sistemi video nadzora mogu se podeliti na sisteme sa analognim prenosom signala, sisteme sa digitalnim prenosom signala u visokoj rezoluciji i IP sisteme video nadzora.

Video nadzor sa analognim prenosom signala podrazumeva da se obrada slike vrši sa optičkog čipa kamere i taj čip se šalje u konvencionalnoj tehnologiji kompozitnog video signala. Kod analognog sistema prisutna su ograničenja koja se odnose na rezoluciju i kvalitet slike, jer se u signalu javljaju smetnje, kao posledica prirode kompozitnog signala. Video signal se šalje putem koaksijalnog kabla određene dužine. Ukoliko se ta dužina povećava, dolazi do gubitka kvaliteta signala. Rezolucije koje se mogu postići ovakvim sistemom video nadzora dostižu do 960 H (horizontalnih piksela), a dalje se prelazi u HD sistem.

HD sistem video nadzora je sistem u kome se vrši digitalni prenos slike kroz koaksijalni kabl u visokoj rezoluciji. Njegove glavne prednosti ogledaju se u mnogo većoj rezoluciji i kvalitetu slike bez gubitaka. Ono što čini dobru stranu ovakvog sistema jeste prenos digitalnog sistema kroz standardni koaksijalni kabl koji se koristi u analognom sistemu video nadzora, pa se može izvršiti zamena celog ili jednog dela sistema analognih kamera, a da se pri tome ne mora menjati postojeća kablovska instalacija. Sa druge strane, loša strana jeste ograničenost dužine kabla, jer se može preneti signal dužine do 50 metara.

IP sistem video nadzora podrazumeva da se slika obrađuje i vrši njen prenos putem TCP/IP protokola, tj. putem računarske mreže. Ovaj sistem omogućava da se vrši prenos bez gubitaka i da se vrše prenosi jako visokih rezolucija, dozvoljava da se snimanje slike izvodi na većem broju servera, te integraciju sa drugim sistemima (kontrola pristupa, alarmni sistem...). Rezolucije koje se u ovim sistemima mogu ostvariti su veoma visoke, idu do nekoliko megapiksela.

3. IP SISTEM VIDEO NADZORA

IP video nadzor je sigurnosni sistem koji omogućava pregledanje snimaka ili njihovo gledanje uživo preko mreže (LAN ili WAN). Najznačajniji deo celokupnog sistema na kome se sve zasniva jeste računarska mreža, preko koje se vrši slanje i prijem informacija. Preko žične ili bežične mreže digitalizovani video signal može da se pošalje na bilo koju lokaciju u svetu. Uz upotrebu odgovarajućeg softvera i uz pristupne šifre sistem IP video nadzora može se kontrolisati i pregledati prethodno snimljen materijal sa bilo kog mesta koje ima pristup računarskoj mreži.

Kod ovog sistema nadzora komunikacija je dvosmerna - IP kamera može da šalje video, audio podatke ili druge vrste podataka, ali pored toga, na osnovu primljenih instrukcija može npr. da otvori vrata ili da uključi alarm. Sistemu u isto vreme može pristupiti veći broj korisnika i tako pregledati snimljeni materijal [2].

Ono što ovaj sistem čini rasprostranjenim danas jeste niz rednosti koje poseduje u odnosu na analogne sisteme:

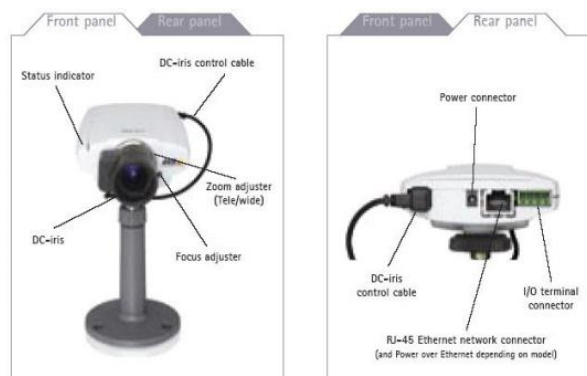
- Pristup sa udaljenih lokacija
- Manja cena celokupnog sistema uz povećanu funkcionalnost
- Jednostavnost i ušteda upotrebom PoE (*Power Over Ethernet*) napajanja, uz mogućnost da se kamere napajaju preko istog kabla koji se koristi za prenos podataka.
- Visok kvalitet slike
- Mogućnost upozorenja putem elektronske pošte
- Podrška za inteligentne detekcije pokreta, prepoznavanje oblika, brojanje objekata, ljudi itd.
- Modularnost, skalabilnost i fleksibilnost - mogućnost integracije IP video nadzora sa drugim sistemima poput kontrole pristupa, alarmnih sistema, upravljanja građevinama, upravljanja saobraćajem.

Osnovne komponente IP sistema za video nadzor čine mrežna (IP) kamera, video server, PC i odgovarajući softver za kontrolu sistema [5].

4. IP KAMERE

IP kamere ili mrežne kamere istovremeno predstavljaju i kameru i kompjuter u istom kućištu. To je najnapredniji vid kamera. One koriste postojeće računarske mreže i samim tim olakšava se projektovanje instalacija bilo koje dužine. Njihovo funkcionisanje može biti potpuno samostalno (bez snimača) i ne postoji ograničenje kada se radi o maksimalnoj rezoluciji. Glavni delovi IP kamera su: objektiv, senzor slike, jedan ili više procesora i memorija.

Procesori se koriste da bi se izvršilo procesiranje slike, komprimovanje slike, njena analiza i da bi se omogućio rad na računarskoj mreži. Senzor slike je u stvari fotosenzor na osnovu koga se optička slika pretvara u elektronski oblik signala. Memorija kamere služi za čuvanje operativnog sistema, a u lokalnu memoriju skladište se snimljene video sekvence. Na slici 1. dat je primer jedne IP kamere.



Slika 1. Primer IP kamere

IP kamere se mogu podeliti na osnovu više karakteristika. Prema obliku kućišta i pokretljivosti kamera razlikuju se:

Fiksne kamere - predstavljaju tradicionalni dizajn kamere. Njihovo vidno polje (*Field of View*) određuje se u trenutku kada se kamera montira, a objektiv može biti fiksni i varifokalni.

Fiksne kupolne ili fiksne DOME IP kamere - njihova je prednost što kamera može biti usmerena u proizvoljnom pravcu, a kućište pruža i efikasnu zaštitu od pomeranja, pokrivanja, promene pravca ili defokusiranja kamere.

PTZ IP kamere (*Pan Tilt Zoom*) - nude funkcionalnost mrežnog videa uz mogućnosti promene pravca snimanja objekta ili prostora po horizontali (*Pan*) i vertikalni (*Tilt*) i promene zuma (*Zoom*). Pokretanje ovih kamera jednostavno se vrši putem računara povezanog na računarsku mrežu, a sve komande šalju se preko istih mrežnih kablova kojima se vrši slanje videa.

PTZ kupolne (*DOME*) IP kamere - odlikuju se punom fleksibilnošću uz 360° neprekidnu rotaciju i napredan mehanički dizajn kojim se omogućava da se pomeranje kamere izvodi kontinualno. Osim toga, kod ovih kamera prisutna je i mogućnost optičkog zuma do 35 puta, te veoma brze promene pravca snimanja objekta ili prostora, te promene zuma (*pan/tilt/zoom*.)

Savremeni sistemi nadzora razvijaju se ugradnjom megapikslnih senzora u kamere. Ova tehnologija u mrenim kamerama pruža veću rezoluciju video snimaka u odnosu na CCTV - omogućava vidljivost detalja i identifikovanje ljudi i objekata. Prilikom kombinovanja megapikslnih sa standardnim mrežnim kamerama, čija se funkcionalnost ističe na polju drugih osobina (veliki optički zum, velika osetljivost na svetlost, niska cena) ostvaruju se uslovi za efikasne, pouzdane i povoljne instalacije video nadzora.

5. VIDEO SERVER I SOFTVER ZA UPRAVLJANJE VIDEOM

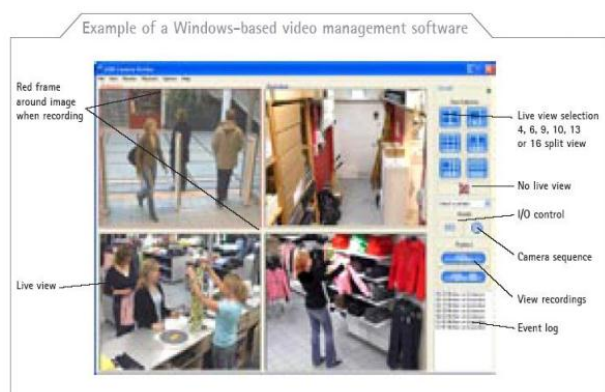
Video server naziva se još i video enkoder. Uz pomoć njega vrši se kretanje sistemom bez odbacivanja postojeće analogne kamere, odnosno, digitalizuje video signal iz analogne kamere i šalje ga u sistem putem TCP/IP protokola. Ostvaruju se nove funkcije analogne kamere, tako da više ne postoji potreba za koaksijalnim kablovima, monitorima i DVR - ima.

Video server najčešće sadrži od jednog do četiri analogna porta za konekciju sa analognim kamerama, te *Ethernet*

port za povezivanje na mrežu. Poput IP kamera sadrži i kompresioni čip i operativni sistem, pa se podaci iz analognog oblika konvertuju u digitalni video, a zatim se taj video šalje i snima preko računarske mreže.

Softverom za upravljanjem videom obezbeđuju se osnovne funkcije praćenja, analize, snimanja i upravljanja videom. Odvija se na *Windows* ili *Unix/Linux* serveru, a da bi se gledale aplikacije mrežnog videa potreban je standardni *web browser*, koji to omogućava preko mrežnog interfejsa ugrađenog u mrežnu kameru ili video server.

Postoji i mogućnost posmatranja jedne ili većeg broja kamera istovremeno. Na slici 2. prikazan je izgled prozora softvera koji se koristi za upravljanje videom.

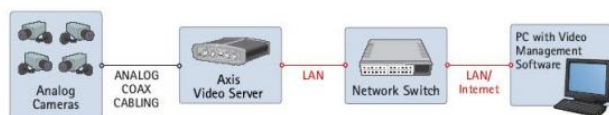


Slika 2. Izgled softverskog prozora za upravljanje videom [2]

6. FUNKCIONISANJE SISTEMA VIDEO NADZORA

Sistemi video nadzora koji koriste video servere predstavljaju prave sisteme mrežnog videa. U jednom ovakvom sistemu video nadzora koriste se video server, mrežni *switch* (uređaj koji upravlja protokom podataka u delovima lokalne mreže) i PC, koji poseduje softver za upravljanje videom.

Blok šema funkcionisanja ovakvog sistema mrežnog videa predstavljena je na slici 3.

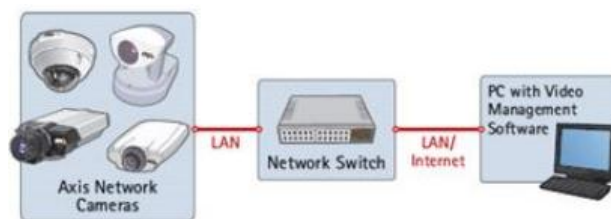


Slika 3. Blok šema sistema mrežnog videa koji koristi video server [2]

Sistem video nadzora koji koristi IP kamere (slika 4.) funkcioniše tako što IP kamera, koja je sama kombinacija kamere i računara, vrši digitalizaciju, kompresiju slike i uspostavljanje konekcije na mrežu.

Koriste se mrežni *switch* - evi da bi se video preneo na IP baziranu mrežu i onda se snima na standardnom PC računaru koji ima softver za upravljanje videom.

Prednosti ovog sistema su upotreba *Ethernet* - a i *wireless* - a, kamere jako visokih rezolucija, *Pan/tilt/zoom*, audio, digitalni ulazi i izlazi preko Internet protokola zajedno sa videom, i fleksibilnost i skalabilnost sistema.



Slika 4. Blok šema sistema mrežnog videa koji koristi IP kamere [2]

Prva faza u funkcionisanju sistema video nadzora jeste **formiranje slike**. Na kvalitet slike utiče nekoliko različitih faktora, kao što su odabir optike ili senzora slike, moć obrade i uspešnost algoritama prilikom postupka obrade [1]. Značaj senzora slike kamere jeste u tome što se na osnovu njega vrši pretvaranje svetlosti u oblik električnog signala. Razlikuju se dve tehnologije za senzor slike prilikom gradnje kamere [3]:

- **CCD** (Charged Coupled Device), koji su formirani upotrebom specijalne tehnologije za proizvodnju kamera i
- **CMOS** (Complementary Metal Oxide Semiconductor), zasnovani na standardnoj tehnologiji koja je već prisutna u memorijskim čipovima, kao npr. kod PC računara.

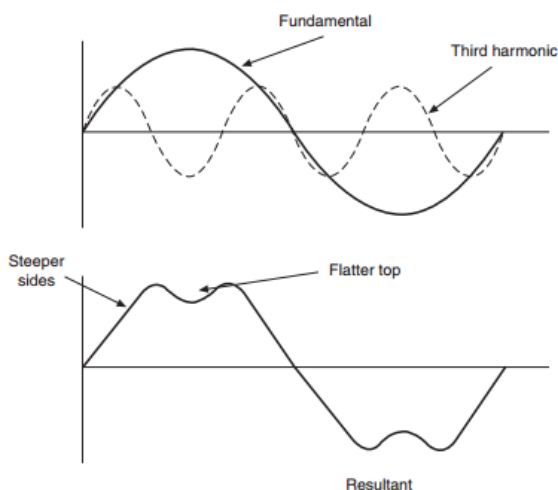
Nakon formiranja sledi **skeniranje slike**. Postoje dve tehnike koje se mogu izdvojiti kada se govori o kreiranju videa, a to su skeniranje poluslike, tj. skeniranje preplitanjem i progresivno skeniranje [2]. Skeniranje poluslike predstavlja tehniku slaganja linija, u okviru koje senzor slike proizvodi sliku sačinjenu od dva para linija - parnih i neparnih. Kod progresivnog skeniranja cela slika se skenira na principu linija po linija na svakih 25 – 30 sekundi, pa snimljene slike nisu podeljene na posebna polja. Da bi se primenila ova tehnika skeniranja potrebno je da su na raspolaganju monitori najvećeg kvaliteta.

Kompresija je potrebna da bi sistem video nadzora upravljao veličinom fajlova za skladištenje i gledanje koji se prenose mrežom. Kompresija redukuje količinu podataka koja se šalje kroz mrežu, jer se na osnovu zahtevane širine protoka i skladištenja podataka može zaključiti da nekomprimovani video nije praktičan i da je skup. Kompresiju slike moguće je obaviti sa i bez gubitaka. Razlikuju se dva pristupa metoda kompresije, a to su kompresija mirne slike (JPEG, JPEG 2000) i video kompresija (M – JPEG, MPEG i H.264 ili MPEG - 4) [2].

7 CCTV

CCTV (Closed Circuit Television) predstavlja sisteme za video nadzor izgrađene da daju informacije o jednoj lokaciji u vidu video zapisa i da pritom budu dostupni na drugom mestu. Video signal u CCTV sistemu sastoji se iz više komponenti kako naizmenične tako i jednosmerne struje i frekvencije je u opsegu od 0-5.5 MHz. Ovaj signal se ponaša slično kao radio signal.

Modifikacija signala (Slika 5.) vrši se zbog toga što je potrebno dobiti dobru sliku visoke rezolucije, što se postiže upravo zaoštavanjem ivica signala i visokim odzivnim vremenom samog signala.



Slika 5. Modifikacija analognog signala [4]

Kamere u CCTV sistemu zahtevaju konvertor koji transformiše upadnu svetlost i električne signale. Kod savremenih digitalnih uređaja ovu ulogu vrše CCD kola (eng. Charge Coupled Device – elementi sa spregnutim naelektrisanjem). CCD je silikonski uređaj koji može da skladišti električni napon. Čip koji sadrži više ovih uređaja poredanih u tačno definisanom poretku može se koristiti da skladišti analogni video ili audio signal i da njima manipuliše. Sabiranje naelektrisanja ne može da se vrši u području bliskom fotodiodi, zbog potrebe da se naelektrisanja prikupi dok se prethodno akumulirano otpušta za potrebe emisije trenutnog frejma. Zato naelektrisanja treba prebaciti u neko trenutno skladišno mesto [4].

U industriji video nadzora i dalje najzastupljeniji monitori jesu CRT (cathode ray tube) koji se koriste još od 1940-ih. Od tada, katodna cev je pretrpela mnoge modifikacije kako ali je princip rada i dalje isti. Površina ekrana je pokrivena osnovnim elementima, tzv. fosfornim tačkama ili trakama. Na kraju katodne cevi nalazi se elektronski top, odnosno tri topa (po jedan za svaku boju - crvenu, plavu i zelenu), koji šalju elektronski snop u pravcu svake pojedinačne tačke i u zavisnosti od intenziteta zraka sa svakog od tri topa, dobija se željena boja u željenoj nijansi. Snop se usmerava elektromagnetnim poljem promenljive jačine koji su postavljeni sa strane katodne cevi.

Kako bi video nadzor bio efikasniji potreba se javila za snimanjem i arhiviranjem snimaka kao dokaznog materijala o zabeleženim aktivnostima. Najčešće korišćeni načini snimanja video materijala u CCTV i IP jesu DVR (Digital Video Recorder) i NVR (Network Video Recorder). Dok DVR vrši digitalnu kompresiju analognog signala, a snimke smešta na hard disk, NVR arhivira digitalne slike direktno sa IP mreže.

6. ZAKLJUČAK

Napretkom tehnologije sve je izvesnije da će IP video nadzor potpuno zameniti analogne sisteme, iako se analogni sistemi i dalje u velikoj meri koriste prvenstveno zbog niske cene. IP sistemi su skuplji jer su delovi izgrađeni od kvalitetnijih materijala kako bi pružio bolje performanse, i duži životni vek celog sistema. Uprkos

tome što su za male poslodavce analogni sistemi i dalje prihvatljivija opcija, poslednjih godina zabeležen je veliki broj onih koji su sa analognog sistema prešli na IP i razlozi za to su višestruki.

Prvi razlog je svakako bolji kvalitet slike, odnosno veća rezolucija. Analogni sistemi se i dalje proizvode po starom standardu, dok IP kamere iz dana u dan napreduju. IP kamera koja je najnižeg kvaliteta i dalje daje sliku jednaku najboljoj analognoj kameri, ako ne i bolju.

Sledeća prednost IP sistema je u tome što je eliminisan rizik prekida konekcije usled kvarova ili oštećenja kabla. Analogni sistemi i dalje moraju da se povežu koaksijalnim kablom, od svake kamere do video rekordera koji zatim beleži sliku na DVR ili na VHS kasetu (koje su i dalje u upotrebi u malim sistemima zbog niske cene). IP kamere nisu ograničene ovim povezivanjem, one samo moraju biti povezane na internet mrežu. IP sistem nije ograničen brojem kamera koje se mogu priključiti (za razliku od analognog koji ima ograničen broj portova za kamere). Kako se signal prenosi putem mreže, materijal je dostupan istovremeno na više lokacija, i šalje se brže. Takođe, signal koji se šalje može se na više načina zaštititi, bilo enkripcijom ili korišćenjem zaštitnih programa poput firewall-a.

Iako je sistem napredniji, IP sistem je i dalje lakši za instalaciju. Proizvođači se trude da omoguće što je lakše postavljanje sistema, te su skoro na nivou jednostavnosti sa web kamerama za kućne računare. Za razliku od analogne kamere, IP kamera zahtev samo jedan kabl koji će biti zadužen za transmisiju signala i za napajanje, kao i za dodatne kontrole (zumiranje, rotiranje, fokusiranje...). Kako tehnologija napreduje, primećujemo i pad u ceni IP sistema, jer je oblast koja se razvija i konkurencija je sve veća na tržištu, te tako postaju sve dostupniji.

7. LITERATURA

- [1] Anthony C. Caputo, *Digital Video Surveillance and Security*, Published by Elsevier Inc, Burlington 2010.
- [2] Shu Tao, Apostotopoulos J., Guerin R. IBM T.J. Watson Res.Center, Hawthorne, NY, *Real-Time Monitoring of Video Quality in IP Networks*, IEEE 2009
- [3] Luo Hui, Sch. of Inf. Eng., East China Jiaotong Uni., Nanchang, China. *Study on Real-Time Video Monitoring System Based on BSP - 16*, IEEE June 2010
- [4] Joe Cieszynski, IEng MIEE (elec) Cert. Ed. CGI, *Closed Circuit Television Second edition*, Oxford, 2004
- [5] www.viser.edu.rs

Kratka biografija:



Nenad Protić rođen je u Zrenjaninu 1986. god. Master rad odbranio je 2016. godine na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacije

PRIMENA OPTIČKIH KOMUNIKACIJA U INDUSTRIJSKIM SISTEMIMA
OPTICAL COMMUNICATIONS APPLIED IN INDUSTRIAL SYSTEMSMiljan Rogan, Željko Trpovski, *Fakultet tehničkih nauka, Novi Sad***Oblast - ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je dat pregled mnogih aspekata modernih optičkih sistema prenosa koji koriste SDH tehnologiju za multipleksiranje i prenos digitalnih signala. Iznesene su razlike u prenosu podataka optičkim kablom u odnosu na bakarnu paricu kao i prednosti prenosa signala optičkim vlaknom u odnosu na bakarnu paricu. U radu su prikazane osnovne karakteristike SDH tehnologije, i govori se o perspektivi SDH tehnike, mogućoj integraciji sa DWDM tehnikom multipleksiranja, ATM i IP tehnikama prenosa.

Abstract – This paper surveys the many aspects of modern optical transmission systems based on SDH technology. Presented the differences in data transfer optic cable as compared to copper cables as well as the advantages of fiber optic signal transmission compared to copper pair. This paper surveys basic characteristics of SDH technology and a possible evolution of SDH technology toward integration with DWDM, ATM and IP technologies is discussed.

Ključne reči: Optika, optičko vlakno, PDH i SDH mreža, IP, ATM, Ethernet.

1. UVOD

Ubrzani razvoj telekomunikacija i uopšte komunikacija pratio je potrebu za prenosom sve veće količine informacija na velike razdaljine. Tehnologije kao što su koaksijalni kabl, prenos podataka putem satelita, a koje su korišćene do unazad 20 godina, zbog ograničenja u kapacitetu morale su biti unaprijeđene. Jednostavno, potreba za količinom prenesenih informacija ubrzo je prevazišla mogućnosti pomenutih tehnologija. Rješenje je pronađeno u upotrebi optičkih vlakana u prenosu informacija.

Iako su vlakna instalisana prije dvadeset i više godina bila namijenjena relativno skromnom kapacitetu prenosa, na osnovu novih saznanja o optičkom prenosu, ista vlakna se bez značajnijih fizičkih zahvata mogu koristiti za prenos ekstremno velikih kapaciteta.

Prvobitno su se u optičkim sistemima, čiji je planirani kapacitet bio skroman u odnosu na današnje zahtjeve, postavljena jednomodna vlakna sa skokovitim indeksom prelamanja. Planirani kapacitet je bio nekoliko stotina Mbit/s, međutim, danas se ista vlakna koriste za prenos nekoliko stotina Gbit/s.

Oblast optičkih komunikacija je veoma široka, tako da je akcenat u ovom radu stavljen isključivo na sinhronu digitalnu hijerarhiju (SDH).

U posljednjih deset godina SDH predstavlja dominantnu tehnologiju za multipleksiranje i prenos digitalnih signala u transportnoj mreži, od govora i podataka do video signala. Nastala iz potrebe da se prethodna PDH (Plesiohrona digitalna hijerarhija) tehnologija zameni jeftinijim i fleksibilnijim rješenjima kada su u pitanju multipleksiranje /demultipleksiranje osnovnih digitalnih signala (pritoka), centralizovani nadzor i upravljanje, itd.

2. PREDNOSTI OPTIČKIH VLAKANA U ODNOSU NA BAKARNU PARICU

Optičko vlakno posjeduje veoma veliki prenosni kapacitet, ali isto tako i veoma mali stepen prigušenja signala što sve rezultuje relativno visokom $B \cdot L$ proizvodu, odnosno proizvodu prenosnog kapaciteta i maksimalne udaljenosti na koju se taj kapacitet može prenijeti. Na Fiber optički kablovi imaju veliki propusni opseg sa brzinom protoka od 40 Gbps dok se u bliskoj budućnosti očekuje da ta cifra naraste i preko 100 Gbps.

Fiber optički kablovi su u potpunosti imuni na elektromagnetsku interferenciju (EMI), radio frekvencijsku interferenciju, atmosferska pražnjenja. Ako su ispravno dizajnirana, fiber optička vlakna su imuna na efekte radijacije. Optički kablovi obezbjeđuju potpunu galvansku izolaciju između oba kraja vlakna, odnosno galvansko razdvajanje instalacija.

Dok signal putuje duž optičkog vlakna on gubi snagu. Za povećanje dometa na krajnjim tačkama vlakna primjenjuju se pojačavači i ripiteri. Za jednostavne fiber optičke sisteme moguća je razdaljina od 5 km bez pojačavanja signala dok za visoko kvalitetne sisteme ta razdaljina je veća. U slučaju kabla sa upređenim paricama moguće je prostiranje signala do 2.5 km između ripitera brzinom od 4 Mbps.

Korišćenjem koaksijalnog kabla moguće je prostiranje signala do 25 km između ripitera brzinom manjom od 1 Mbps. Upoređujući ih sa ostalim vrstama kablova, fiber optički kablovi imaju znatno manju težinu i veoma mali prečnik. Skoro je nemoguće 'zakačiti' se na fiber optički kabl i neopaženo pristupiti informacijama koje se prenose.

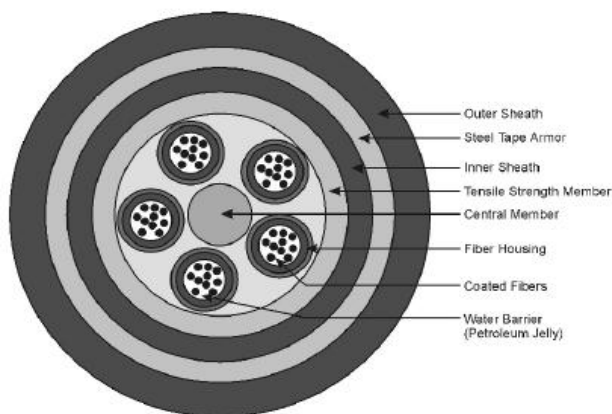
Potrebno je fizički pristupiti vlaknu da bi se izvukle informacije, a to će dovesti do smanjenja nivoa signala i povećanja nivoa greške što se lako detektuje. Dakle, sa trenutno dostupnom tehnologijom, fiber optički sistemi su sistemi sa visokim stepenom sigurnosti [1].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Trpovski, vanr. prof.

3. KONSTRUKCIJA OPTIČKIH KABLOVA

Osnovni strukturni elementi koji se koriste u kablovima su: centralni član, snažni član, kućište vlakna, vodena barijera kao i omotač kabla. Neki kablovi zahtijevaju i dodatni zaštitni oklop. Struktura optičkog kabla je prikazana na slici 1.



Slika 1. Struktura optičkog kabla [1]

Kao što i sam naziv kaže, centralni član je struktura u središnjem dijelu kabla. Obično se pravi od fiberglasa, moguće je i da sadrži čelik. Optički kabl treba da sadrži relativno neelastičan snažni član, postavljen čitavom dužinom kabla, da bi izdržao opterećenje i prekomjerno rastezanje vlakna.

Za vrijeme proizvodnje vlakna dobijaju primarni sloj zaštite koji štiti površinu vlakna od mehaničkih i hemijskih uticaja. Vlakna zatim dobijaju sekundarnu ili bafer zaštitu. To može biti ili labava (slobodna) bafer zaštita, obično za spoljašnju upotrebu ili čvrsti bafer za unutrašnju upotrebu. Slobodno baferovanje, koristeći labave (slobodne) cijevi ili konstrukciju sa prorezanim jezgrom, koristi se za većinu spoljašnjih aplikacija gdje je vjerovatno da će vlakna biti izložena silama naprezanja ili temperaturnim promjenama.

Konstrukcija sa slobodnom cijevi koristi tvrdi, glatku fleksibilnu cijev čije su unutrašnje dimenzije mnogo veće od prečnika vlakna. Jedno ili više vlakana može da se instalira unutar cijevi tako da može slobodno da se kreće unutar nje. Cijev izoluje vlakna od bilo kakvih spoljašnjih uticaja. Konstrukcija sa slobodnom cijevi može da sadrži do 12 vlakana po cijevi, koristeći obojene cijevi i vlakna radi lakše identifikacije.

Tip konstrukcije sa prorezanim jezgrom koristi prorezani plastični strukturni član koji okružuje centralni snažni član. U prereze u jezgri se smještaju vlakna što im omogućuje radijalno pomjeranje i izolaciju od spoljašnjih sila. Ovo minimizuje naprezanja vlakana i samim tim i njihove gubitke uslijed mikrosavijanja.

4. SDH OPTIČKI SISTEMI PRENOSA

SDH (*Synchronous Digital Hierarchy*) mreže predstavljaju transportne mreže namjenjene za transport raznovrsnih signala poput IP saobraćaja, ATM ćelija, PDH (*Plesiochronous Digital Hierarchy*) struktura, itd. Za SDH mreže se može reći da predstavljaju transportne mreže druge generacije, razvijene da obezbijede veće protoke i fleksibilnije upravljanje i nadgledanje mrežom u

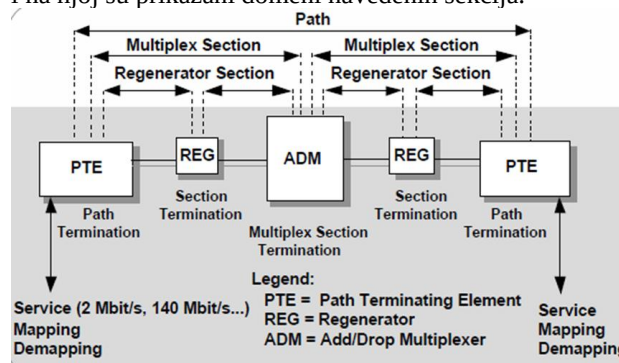
odnosu na transportne mreže prve generacije, odnosno PDH mreže [2]. Osnovni problem PDH mreža bila je njihova nefleksibilnost i kao posledica neekonomičnosti pri izradi mrežnih PDH čvorova zbog potrebe za kompletnim multipleksiranjem i demultipleksiranjem. Pri tome, PDH mreže nisu davale mogućnost kvalitetnog upravljanja i nadgledanja mreže. Takođe, težilo se smanjenju razlika u standardima između evropske i američke strane i kao rezultat nastale su standardizovane SDH i SONET tehnologije koje su veoma slične.

Slika 2. prikazuje definiciju PDH i SDH nivoa hijerarhije. SDH koristi isti princip hijerarhija kao i PDH tehnologija. STM (*Synchronous Transport Module*) je skraćenica za sinhroni transportni modul. Za razliku od PDH mreža, SDH je sinhronizovana mreža u kojoj su svi čvorovi sinhronizovani na isti referentni takt, čime se značajno smanjuje učestanost preskoka.

Nivo Hijerarhije	Ukupan protok [Mb/s]	Nivo Hijerarhije	Ukupan protok [Mb/s]
E1	2.048	STM-1	155.52
E2	8.448	STM-4	622.08
E3	34.368	STM-16	2488.32
E4	139.264	STM-64	9953.28
E5	564.992	STM-256	39813.12

Slika 2. PDH i SDH hijerarhija [3]

Kao i većina drugih telekomunikacionih mreža, i SDH je bazirana na slojevitoj strukturi pri čemu se razlikuju regeneratorska sekcija, multipleksna sekcija i sekcija puta (navođenje je dato od najnižeg sloja ka najvišem sloju). Na slici 3. prikazana je jedna tipična dionica u SDH mreži i na njoj su prikazani domeni navedenih sekcija.

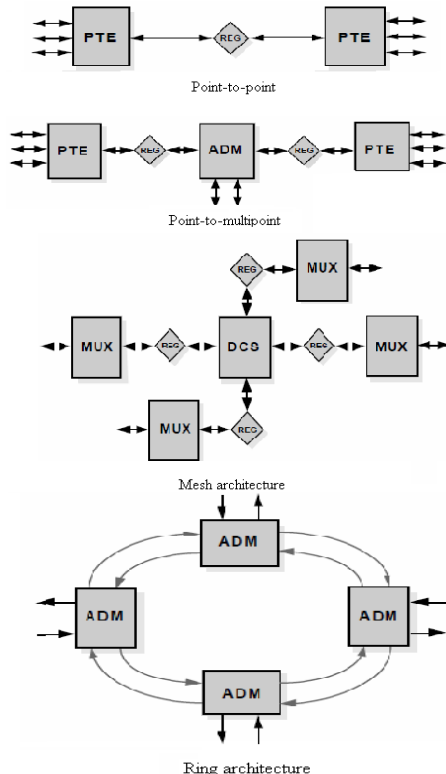


Slika 3. Definicije sekcija u SDH mreži [3]

PTE (*Path Termination*) predstavlja uređaj koji omogućava priključenje korisnika tj. njihovih tokova (na primer, ATM tokovi, IP saobraćaj, PDH strukture) i multipleksiranje tih tokova u SDH strukturu koja se prenosi preko SDH mreže (pod SDH strukturom podrazumjevamo STM-i okvir, pri čemu se ti okviri šalju kontinualno jedan za drugim).

REG predstavlja regeneratore koji vrše osvežavanje signala pod čime se podrazumjeva rekonstrukcija vrijednosti bita i takta sa kojim ti rekonstruisani biti treba da se šalju na izlazu regeneratora i oni su potrebni na dugačkim dionicama. ADM (*Add/Drop Multiplexer*) predstavlja tzv. 'add/drop' multipleksere kojima se omogućava izvlačenje pojedinih korisničkih tokova iz SDH strukture [2].

Kod SDH mreža postoji više topologija. Na primer, tačka-tačka topologija gde su dva PTE elementa direktno povezana ili preko jednog ili više regeneratora. Isto tako može se kreirati i tačka-više tačaka topologija gde se dodatno između PTE elemenata ubacuje i ADM, slika 4. Takođe, moguće je kreirati i meš topologiju. Može se kreirati i zvezda topologija ako se koriste DXC uređaji. Međutim, najpopularnija topologija koja se dominantno koristi u praksi jeste prsten topologija zbog omogućavanja implementacija efikasnih mehanizama zaštite od kvarova u mreži.



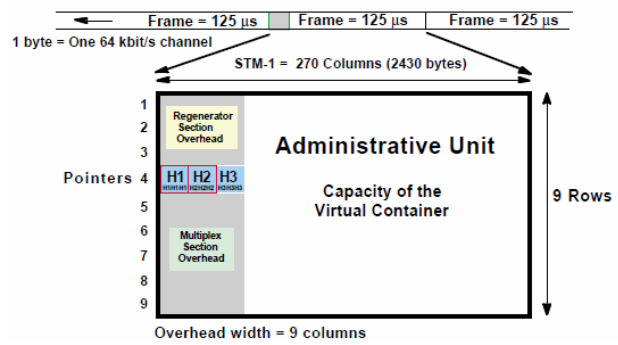
Slika 4. Topologija SDH mreže [3]

Struktura STM-1 okvira je prikazana na slici 5. STM-1 okvir predstavlja najniži nivo hijerarhije. Sastoji se iz 2430 bajtova ($9 \times 270B = 2430B$), pri čemu jedan STM-1 okvir traje $125\mu s$.

Okvir se tipično prikazuje kao pravougaonik od devet redova, gde svaki red ima 270 bajtova. Kao što se na slici vidi okvir sadrži zaglavlje koje se sastoji od tri dijela,

- RSOH (Regenerator Section Overhead) predstavlja zaglavlje koje je relevantno na nivou regeneratorske sekcije,
- MSOH (Multiplex Section Overhead) predstavlja zaglavlje koje je relevantno na nivou multipleksne sekcije
- pokazivač POH (Path Overhead) dio je relevantan na nivou sekcije puta.

Pri tome, u POH djelu se može nalaziti jedan ili više pokazivača u zavisnosti od strukture korisnog dijela. Pokazivač pokazuje na početak korisničke strukture koja je spakovana u korisnom dijelu i neophodan je jer korisnička struktura može da pluta u korisnom dijelu tj. njen početak ne mora da se uvijek nalazi na istoj poziciji. Korisni deo sadrži korisničke strukture, tj. korisničke tokove koji se prenose kroz SDH mrežu unutar STM-1 okvira.

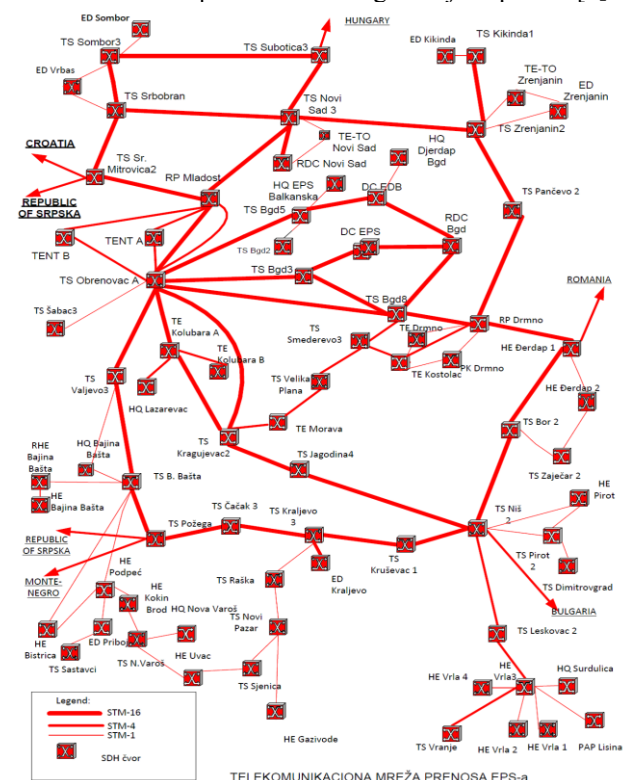


Slika 5. STM-1 okvir [3]

STM okvir nivoa N se dobija multipleksiranjem STM okvira nižeg nivoa N/4. Na primer, STM-4 okvir se dobija multipleksiranjem četiri STM-1 okvira, a STM-16 multipleksiranjem četiri STM-4 okvira. Pri tome, multipleksiranje se vrši na nivou bajtova.

5. PRIMENE SDH U KOMUNIKACIONOJ MREŽI

Mreža prenosa zasnovana na SDH tehnologiji izgrađena je na telekomunikacionoj optičkoj mreži u čijim je krajnjim tačkama instalirana SDH terminalna i multipleksna oprema. Mreža na magistralnom nivou sadrži 80 tačaka. Na slici 6. dat je prikaz svih tačaka-čvorova sa instaliranom terminalnom opremom i telekomunikacionim kapacitetima za odgovarajuće pravce [4].



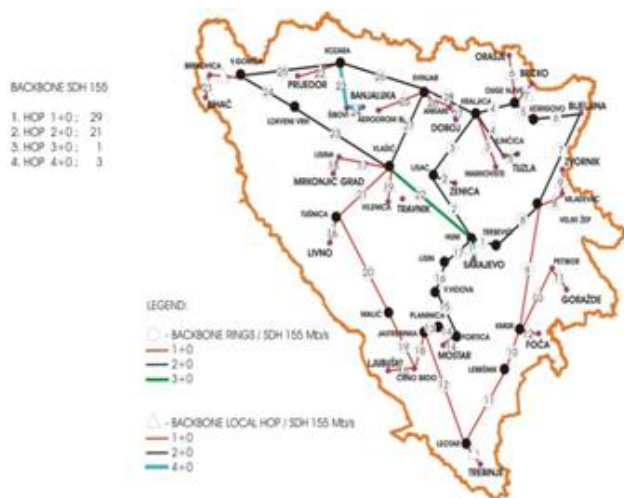
Slika 6. Mreža prenosa EPS-a sa SDH čvorovima i kapacitetima linkova [4]

Za svrhu prenosa poslovnih, tehničkih i govornih podataka izabrana je SDH tehnologija (Synchronous Digital Hierarchy) koja je u vrejme iniciranja projekta i instaliranja opreme bila dominantna u svetu za ovakve primjene. Kapaciteti na glavnim pravcima su nivoa STM-16 (2.488 Gb/s), na manje važnim STM-4 (622 Mb/s), a

na nekim perifernim i antenskim nivoa STM-1 (155 Mb/s).

Okosnicu paketske mreže elektroprivrede Srbije čine pet savremenijih paketskih uređaja (core rutera), smeštenih na pet lokacija u elektroprivrednim objektima, povezanih u full-mesh strukturi preko STM-4 interfejsa na SDH uređajima. U pristupnom delu mreže povezano je 30 lokacija pri čemu se na 24 lokacije zbog zahtevane velike raspoloživosti nalaze dva nezavisna rutera povezana na različite rutere u okosnici paketske mreže, a na 6 lokacija po jedan ruter. Od navedenih 30 lokacija na 19 lokacija su pristupni ruteri postali voicegateway-i, odnosno, IP telefonske centrale, ugrađivanjem odgovarajućih kartica za povezivanje sa postojećim TDM centralama i javnom mrežom.

Na slici 7. prikazana je topologija mreže za potrebe povezivanja institucija Bosne i Hercegovine na području cijele Bosne i Hercegovine. Mreža se sastoji od pet prstenova, koristi se frekvencijski diverzitet, a prstenovi su međusobno povezani u glavni prsten, tako da raspoloživost mreže iznosi 99,99 %. Kapacitet mreže na glavnim linkovima iznosi 622 Mbps, zavisno od broja kanala koji se koriste ($n \times 155$ Mbps).



Slika 7. SDH mreža javne uprave BiH [4]

Osnovne karakteristike uređaja koji je korišten za realizaciju SDH su:

- frekvencijski opseg 7 GHz i 11 GHz
- point-to-point sistem,
- velike udaljenosti, do 90 km,
- veliki kapacitet, STM1-155 Mb/s,
- sistemi 1+0, 2+0, 2x2+0 i
- ethernet interface (only).

6. ZAKLJUČAK

U ovom radu dat je pregled osnovnih karakteristika fiber optičkih vlakana. Detaljno je opisana konstrukcija optičkih vlakana kao i mehanizmi povezivanja vlakana i problemi koji se mogu pri tome javiti. Takođe, u radu je dat pregled mnogih aspekata modernih optičkih sistema prenosa koji koriste SDH tehnologiju za multipleksiranje i prenos digitalnih signala. Prikazane su osnovne

karakteristike SDH tehnologije: razvoj standarda, hijerarhijska podela na protoke, struktura rama, postupak multipleksiranja. Prikazani aspekti SDH optičkih sistema pokazuju sve prednosti njihovog korišćenja u modernim transportnim mrežama: visok nivo standardizacije, centralizovani nadzor i upravljanje, "samoizlečivost", mogućnost preusmeravanja saobraćaja, fleksibilnost u odnosu na signale (servise) koje može da prihvati.

Moguća je evolucija transportne mreže ka optičkoj transportnoj mreži (OTN) u kojoj se pored DWDM tehnike koriste optički prekidači (OS), optički add-drop multiplekseri (OADM) i cross-connect na nivou talasnih dužina (WXC). Pristup ovoj mreži može da bude bez korišćenja SDH uređaja tj. na definisanom optičkom interfejsu ali i uz njihovu upotrebu.

Posebno je perspektivna varijanta IP sa MPLS (*Multi-protocol Label Switching*) protokolom koji se tek razvija i koji treba da obezbedi stalni kvalitet za sve usluge uvođenjem mogućnosti kontrole i preusmeravanja saobraćaja.

To znači da se usluge mogu rangirati po zahtevanom kvalitetu prenosa, tako da usluge koje ne trpe veća kašnjenja (govor, video) dobiju prioritet u rutiranju ili dodeli propusnog opsega.

Sve više se koristi i direktan pristup na nivou Ethernet 1 Gbit/s i 10 Gbit/s interfejsa, bez korišćenja IP protokola, zbog privlačne cene.

7. LITERATURA

- [1] D. Bailey, E. Wright, "Practical Fiber Optics". 2003, Oxford and J. Zubia and J. Arrue, "Plastic Optical Fibers: An Introduction to Their Technological Processes and Applications". 2006, Bilbao
- [2] A. V. Palle, "SDH (Synchronous Digital Hierarchy) & Its Architecture," Int.J. Sci. Res. Dev., vol. 1, no. 7, pp. 1–5, 2013
- [3] <http://www.cisco.com/c/en/us/support/docs/optical/synchronous-digital-hierarchy-sdh/28327-sdh-28327.html>, <http://www.tek.com/document/fact-sheet/sdh-telecommunications-standard> i Vladimir Čorak, *Uvod u SDH mreže*, VF-TEL, 1995.
- [4] <http://www.eps.rs> i infoteh.etf.unssa.rs.ba/zbornik/2008/radovi/B-I/B-I-18.doc

Kratka biografija:



Miljan Rogan rođen je u Stocu 1986. god. Diplomski-master rad odbranio je 2016. godine na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacije



Željen Trpovski rođen je u Rijeci 1957. godine. Doktorirao je na Fakultetu tehničkih nauka 1998. god. Oblast interesovanja su telekomunikacije i obrada signala.

SOFTVERSKO REŠENJE ZA UPRAVLJANJE PODACIMA O DISTRIBUTIVNOJ MREŽI SOFTWARE SOLUTION FOR MANAGING DATA IN DISTRIBUTION NETWORK

Mladen Gavrić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U okviru ovoga rada opisana je aplikacija koja omogućava vođenje evidencije o postojećim elementima u mreži, kao i vođenje evidencije o novim elementima koji se dodaju kako sama mreža raste. Aplikacija predstavlja pregled inventara, lokaciju pojedinačnih elemenata kao i detalje o svakom elementu. Korišćenjem ove aplikacije, lako se može doći do informacija o tipu svakog elementa i njegovim tehničkim karakteristikama.

Abstract – The presented work describes application that allows keeping records of existing elements in the distribution network, as well as keeping records of new elements that are added to the network which itself grows. The application provides an overview of the inventory, the location of individual elements as well as details about each element of distribution network. By using this application, we can easily get information about the type of each element and its technical characteristics.

Ključne reči: Microsoft .NET, MS SQL Server, Elektrodistributivna mreža.

1. UVOD

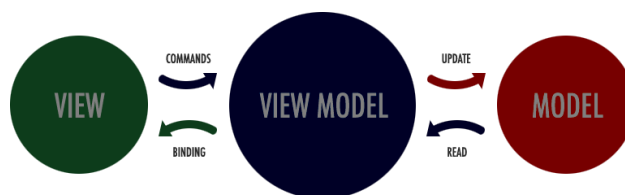
Elektrodistributivna mreža se može sastojati od nekoliko miliona elemenata. Neophodno je čuvanje podataka o svim ugrađenim elementima. Usled stalnog porasta zahtjeva za električnom energijom i za kvalitetnijim snabdjevanjem, u elektrodistributivnu mrežu se stalno ugrađuju novi elementi. Kvarovi koji se neminovno dešavaju u elektroenergetskom sistemu oštećuju opremu koja se s vremena na vrijeme mora mijenjati. Zbog svega navedenog, neophodno je kreirati aplikaciju koja omogućava čuvanje i upravljanje podacima o elektrodistributivnoj mreži.

2. OPIS TEHNOLOGIJA

Za realizaciju ove aplikacije korišćen je MVVM patern. Beneficije koje nudi ovaj patern su ogromne koje objedinjuje oba WPF i Silverlight development. Prednosti korišćenja ovoga paternu ogledaju se u sledećim segmentima, fleksibilan kod promjene korisničkog interfejsa bez refaktorisiranja logike u bazi, čvrsto razdvajanje između aplikacione logike i UI čini aplikaciju lakšu za testiranje, održavanje i razvijanje, tokom razvoja aplikacije, programeri i dizajneri mogu da

rade nezavisno i istovremeno na svim komponentama i druge..

Model-View-ViewModel patern može se koristiti na svim XAML platforma. Njegova namjena je da obezbedi jasnu podelu problema između korisničkog interfejsa i njegove logike.



Slika 1 – Grafički prikaz MVVM patern-a

Tri osnovne komponente u MVVM obrazcu, View, Model and View-Model.. Svaka komponenta je odvojena jedna od druge i ima nezavisnu ulogu.

2.1 Microsoft SQL Server

Microsoft SQL server predstavlja program za skladištenje relacionih baza podataka. Osnovna svrha programa je skladištenje i manipulisanje nad podacima sadržanim u bazama. Za razvoj ovog sistema i skladištenje i čuvanje podatak korišćen je Microsoft SQL server 2005 Express Edition.

2.2 Windows Communication Foundation (WCF)

Windows Communication Foundation je servisno orijentisani model razmjene poruka, koji omogućava programima da komuniciraju preko računarske mreže ili lokalno. Alat koji u sebi uključuje set biblioteka razvijenih za distribuirano programiranje. WCF je često korišćen samo za razvojno servisno orijentisane arhitekture. Klijenti mogu da koriste servise tako da jedan servis može upotrebljavati više klijenta kao deljeni servis. Karakteristika servisa je da je sastavljen od cjelina čije su komponente slabo povezane.

2.3 Windows Presentation Foundation (WPF)

Windows Presentation Foundation je grafički podsistem za rendovanje korisničkog interfejsa u aplikaciji zasnovanim na Windows-u. WPF, predhodno poznat kao Avalon, je inicijalno izdat kao deo .NET Framework 3.0. WPF koristi DirectX. WPF pokušava da obezbedi konzistentan model programiranja za izradu aplikacija i nudi jasnu razliku između korisničkog interfejsa i poslovne logike. Kao takav, on nosi blisku sličnost sa sličnim XML orijentisanim objektima modelima. WPF koristi XAML, izveden iz XML-a, kako bi definisao i povezao različite UI elemente. WPF aplikacije mogu biti

NAPOMENA:

Ovaj rad je proistekao iz master rada čiji mentor je bio Marin Gostimirović.

razvijene kao samostalni desktop programi, ili ugradjeni objekti u nekom website-u.

3.ELEMENTI DISTRIBUTIVNE MREŽE

Osnovni elementi elektrodistributivne mreže koji su obrađeni u ovome radu su:

- Čvorovi – Sabirnice
- Sekcije
- Transformatori
- Prekidačka oprema

Ova četiri elementa predstavljaju minimalan skup entiteta pomoću kojih se može opisati jedan elektroenergetski sistem. Svaki od navedenih elemenata ima svoje karakteristike, koje su generalizovane u kataloge radi lakše pretrage. Ta osobina se može iskoristiti za uštedu memorijskoga prostora prilikom čuvanja podataka o mreži: proizvodjački podaci su smješteni u kataloge koji se mogu iskoristiti za opisivanje velikog broja istih elemenata. Konkretnim elementima je pridružena oznaka (referenca) kataloga. Više elemenata ima referencu ka istom katalogu.

3.1 Opis kataloga

Katalozi predstavljaju skup informacija koje uglavnom dostavlja proizvođač opreme. Prema tome, nisu prikazani svi podaci iz proizvodjačke specifikacije, nego samo podskup podataka od interesa.

3.1.1 Katalog Sekcija

Sekcija – nadzemni vodovi su opisani podužnim parametrima:

- Resistansa (r)
- Reaktansa (x)
- Sesceptansa (b) i
- Kapacitivnost (g);

Kao i naponskim nivoom izolacije.

3.1.2 Katalog Transformatora

- Nominalni napon primarnog namotaja
- Nominalni napon sekundarnog namotaja
- Impedansa transformatora svedena na primar (R, X)
- Imdansa magnećenja (R0, X0)

3.1.3. Katalog rasklopne opreme

Sva prekidačka oprema modelovana je preko parametara:

- Nominalni napon
- Maksimalna struja koju prekidač može da provede.

3.2.Modeli konkretnih elemenata

Modeli konkretnih elemenata koji se nalaze u mreži sadrže:

- Jedinstvene identifikatore
- Reference ka drugim elementima
- Reference ka katalogu

Pored navedenih, mogu sadržati dodatne specifikacije informacija. U nastavku je dat pregled modela po tipu elementa:

3.2.1 Čvorovi

Čvorovi predstavljaju elemente elektroenergetski sistem u kojima se stiče više elemenata više elemenat. To mogu biti visokonaponski, srednjenaponski ili niskonaponske sabirnice.

- Jedinstveni identifikatori(ID)
- 2D geografske koordinate
- Oznake (referencu) naponskog nivoa
- Snagu potrošnje

3.2.2. Sekcija

Sekcije – nadzemni vodovi su elementi elektroenergetskog sistema koji služe za prijenos električne energije. Model sekcije u ovome radu:

- Jedinstveni identifikatori
- Reference ka katalogu
- Reference ka čvorovima između kojih je priključen
- Dužinu u metrima

3.2.3.Transformatori

Model transformatora sadrži sledeće podatke:

- jedinstveni identifikator
- referencu kataloga
- reference čvorova između kojih je priključen-

3.2.4. Prekidačka oprema

Model prekidačke opreme obuhvata podatke:

- jedinstveni identifikator
- referencu kataloga
- referencu čvorova između kojih je priključen
- status prekidačke (otvoren/zatvoren).

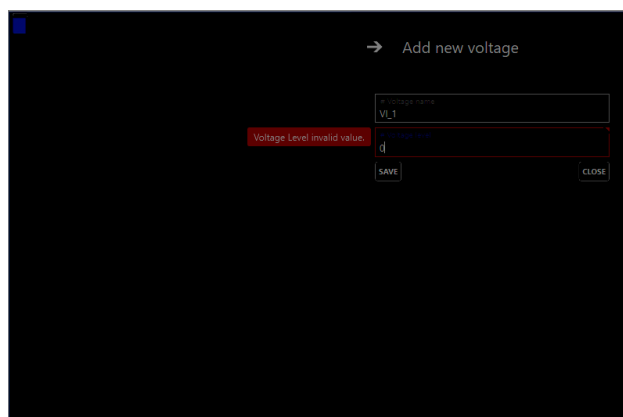
4. APLIKACIJA

Aplikacija je razvijena na Visual Studio-u, koji je službeni IDE za razvoj Windows aplikacija.

4.1 Opis aplikacije

Za svaku tabalu u bazi kreirana je po jedna stranica, koja je direktno sa povezana sa ViewModel.cs klasom u kojoj su sadržane sve metode. Metode koje su implementirane na svakoj stranici su metode koje se odnose na CRUD operacije, učitavanje stranice, verifikacija, pretragu i sortiranje elemenata na stranici.

Svaki element distributivni mreže referenciran je na naponski nivo. Nazivni naponi elektroenergetskih mreža(i oprema uopšte) su standardizovani, ali u različitim državama se ne koriste svi standardni naponski nivoi. U distributivnoj mreži Srbije se koristi naponi nazivnog napona 35, 20, 10 i 0,4kV ili ako su u pitanju industrijska postojanja za koje su standardni naponi 6, 3, 1kV, samim tim su i uvedena ograničenja za unos naponskog nivoa.



Slika 2-Dodavanje naponskog nivoa.

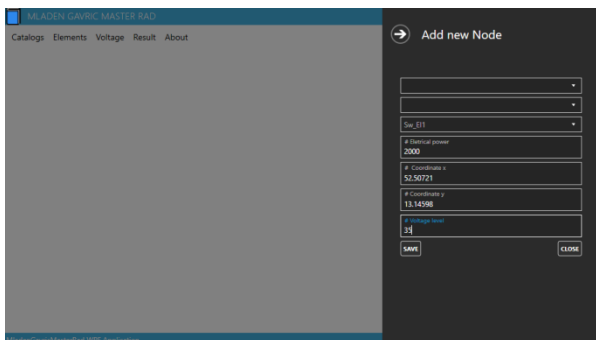
Korinik ima mogućnost dodavanja novih elemenata i mogućnost izmene podataka o postojećim elementima. Dodavanje svakog elementa se vrši preko posebne forme.

Da bi se izmene uspešno sačuvala, potrebno je uneti osnovne podatke o elementima. Da bi se sprečile greške, različite provjere (validacije) se izvode prilikom dodavanja elemenata. Na taj način se korisniku sugerise pravilan unos podataka ali se i sprečava loš unos koji bi mogao ugroziti rad aplikacije.

Prvo bi trebalo kreirati kataloge i elemente koji se pojavljuju kao reference u drugim elementima (npr. čvorovima). Očekuje se da korisnik rijetko dodaje nove kataloge, jer se obično ugrađuje ista oprema.

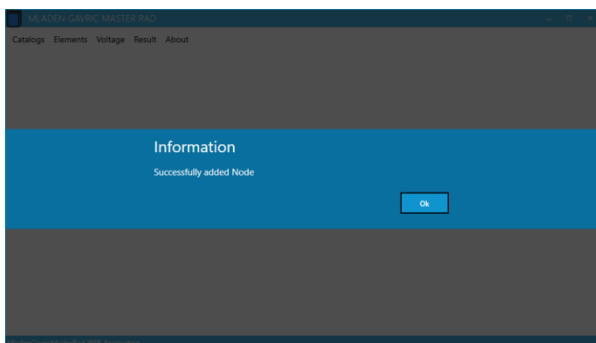
4.1.1 Validacija

Validacije zavise od tipa elemenata i dizajnirane su u skladu sa standardnim praksama i poslovanjem elektrodistributivnih preduzeća. Zajednička validacija za sve elemente jeste unos imena i naponskoga nivoa.



Slika 3-Dodavanja novog Node

Za dodavanje novih čvorova postoji validacija postojanja naponskoga nivoa i kordinata. Za slučaj dodavanja novih sekcija moraju biti ispunjeni sledeći uslovi, da sekcija mora biti na istom naponskome nivou. U suprotnom, imao bi se kratak spoj. Čvorovi moraju da budu naponskoga nivoa jednakoga ili nižeg od naponskoga nivoa iz kataloga sekcije.



Slika 4-Poruka uspešno dodanog čvora.

Validacija dodavanja transformata izvršena je prema sledećem kriterijumu, da odabir čvorova mora biti na odgovarajućim naponskim nivoima koji su navedeni u katalogu transformatora.

Validacija za dodavanje prekidača je sledeća da čvorovi između kojih se dodaje prekidač moraju biti na istom naponskome nivou, u suprotnome bi se imao kratak spoj. Aplikacija nudi mogućnost grafičkoga prikaza elemenata koji trenutno postoje u mreži. Odabirom opcije iz menija moguće je prikazati histograme koliko kojih elemenata postoji u mreži.

5. ZAKLJUČAK

Aplikacija je prilagodjena mogućnostima personalnih računara i savremenih operativnih sistema. Predstavlja samo jedan mali dio u primeni savremenih informacionih tehnologija u vođenju distributivnih preduzeća i predstavlja jedan mali dio osnove za dalji razvoj aplikacija koje omogućavaju proračune gubitaka energije, analizi topologije i drugih stanja elektrodistributivnih mreža čiji se proračun izvode iz kompleksih algoritama i matematičkih procedura.

6. LITERATURA

- [1] "Desktop-application", Internet: <http://www.pcmag.com/encyclopedia/term/41158/desktop-application>, poslednji pristup 23.10.2016
- [2] "Microsoft", Internet: <https://sh.wikipedia.org/wiki/Microsoft>, poslednji pristup 23.10.2016
- [3] "Microsoft Visual Studio", Internet: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio, poslednji pristup 23.10.2016
- [4] "Common Intermediate Language", Internet: https://en.wikipedia.org/wiki/Common_Intermediate_Language, poslednji pristup 23.10.2016
- [5] "Windows Communication Foundation", Internet: https://en.wikipedia.org/wiki/Windows_Presentation_Foundation, poslednji pristup 23.10.2016
- [6] "ModelViewViewModel", Internet: <https://msdn.microsoft.com/en-us/library/hh848246.aspx>, poslednji pristup 23.10.2016
- [7] Internet: <http://mahapps.com/>, poslednji pristup 23.10.2016
- [8] Internet: <https://code.msdn.microsoft.com/windowsapps/Chart-Control-in-WPF-c9727c28>, poslednji pristup 23.10.2016

Kratka biografija:



Mladen Gavrić rođen je 1987. godine u Tuzli. Diplomski-master rad na Fakultetu tehničkih nauka u Novom Sadu iz oblasti Elektrotehnike i računarstva odbranio je 2016. godine.

**UPOREDNA ANALIZA PRIMENE NEWTON/RAPHSON-ovog POSTUPKA I
KOMPENZACIONE METODE NA SUBPRENOSNIM MREŽAMA**

**COMPARATIVE ANALYSIS OF APPLICATION OF NEWTON/RAPHSON'S
PROCEDURE AND COMPENSATION BASED METHOD ON SUBTRANSMISSION
NETWORKS**

Marko Petković, Nikola Vojnović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu obrađen je Newton/Raphson-ov postupak proračuna tokova snaga kao i postupak sumiranja struja i korekcija napona (Kompenzacioni metod) za proračun simetričnih tokova snaga radijalnih mreža i mreža sa petljama. Model mreže je izveden primenom metode potencijala čvorova kao i direktnom primenom strujnog i naponskog Kirchhoff-ovog zakona. Na osnovu algoritama za proračun tokova snaga koji su izvedeni, napisan je računarski program za proračun tokova snaga mreža sa proizvoljnim brojem čvorova i petlji. Izvršena je uporedna analiza vremena izvršenja i tačnosti proračuna na različitim primerima.

Abstract – This paper deals with symmetrical power flow calculations of radial and meshed networks using Newton/Raphson's and current summation and voltage calculation method (Compensation based method). The network model is derived by method of nodes potential and direct application of Kirchhoff's current and voltage laws. The corresponding calculation algorithms for power flow calculation are derived. Based on those algorithms a computer program is also developed with an arbitrary number of nodes and loops. The comparative analysis of execution time and accuracy of calculations are checked on various examples.

Ključne reči: Subprenosna mreža, Distributivna mreža, Newton/Raphson-ov metod, Gauss/Seidell-ov metod, Kompenzaciona metoda, Kirchhoff-ovi zakoni, metoda potencijala čvorova, proračun tokova snaga.

1. UVOD

Proračuni stacionarnih tokova snaga najčešći su proračuni elektroenergetskih mreža. Sastoji se od sledeće četiri komponente: matematički model mreže, matematički metod za proračun modela mreže, utvrđivanje algoritma za proračun stanja mreže i računarski program napisan u skladu sa algoritmom.

Subprenosne mreže jesu trofazne mreže sa relativno velikim brojem petlji – “upetljane”. Matematički modeli tih mreža napisani su saglasno sa metodom nezavisnih napona. Ti postupci su orijentisani na čvorove. Njihova rešenja su zasnovana na klasičnom Newton/Raphson-ovom metodu za rešavanje sistema nelinearnih jednačina [1].

NAPOMENA:

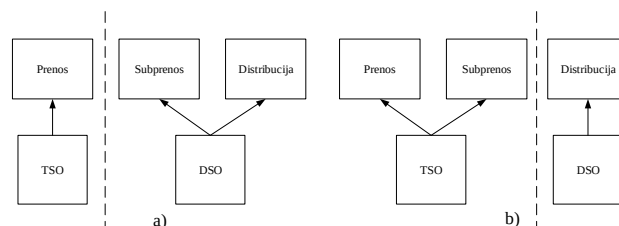
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Strezoski, red. prof.

Sa malim brojem izuzetaka, distributivne mreže jesu radijalne ili “mreže sa malim brojem petlji”. U osnovi većine postupaka proračuna tokova snaga ovih mreža leže nepoznate struje grana mreže. Zbog toga se kaže da su ti postupci orijentisani na grane. Njihova rešenja su zasnovana na metodi sumiranja struja kada su u pitanju radijalne mreže ili Kompenzacionoj metodi kada su u pitanju mreže sa petljama.

U delu 2 je opisana struktura subprenosnih mreža i primeri upravljanja ovim delom mreže. U delu 3 su opisani postupci proračuna tokova snaga primenjenih u radu. U delu 4 su dati primeri proračuna. Delovi 5 i 6 sadrže zaključke i korišćenu literaturu, respektivno.

2. SUBPRENOSNE MREŽE

Subprenosni sistemi predstavljaju deo električne mreže koji napajaju distributivni deo mreže. Težnja je da se subprenosna mreža u većini slučajeva pridruži preduzećima za distribuciju električne energije. Takav pristup upravljanja elektroenergetskih sistema otvara nova pitanja u oblasti softvera za upravljanje distributivnim delom mreže, koji bi se trebao istovremeno primeniti i na subprenosni deo mreže. Primeri pripadnosti subprenosnog dela distribuciji ili prenosu prikazani su na slici 1.



Slika 1 – Subprenosni deo mreže priključen a) distribuciji, b) prenosu

TSO (Transmission system operator) – predstavlja operatera koji je zadužen za prenos i upravljanje električnom energijom od proizvodnih postrojenja preko prenosne mreže do regionalnih ili lokalnih distribucija.

DSO (Distribution system operator) – predstavlja operatera koji je zadužen za distribuciju i upravljanje električnom energijom od završetka prenosne mreže do krajnjih potrošača.

3. PRORAČUN TOKOVA SNAGA

Cilj proračuna tokova snaga predstavlja proračun vektora stanja (fazora napona svih čvorova) mreže. Eksitaciju mreže, koja se unapred specificira, čine kompleksni napon korena i kompleksne snage potrošnje/proizvodnje potrošača

priključenih u preostalim čvorovima. Time je režim mreže određen, ali nije poznat. Distributivne mreže za razliku od prenosnih i subprenosnih, karakteriše radijalan pogon i relativno slaba upetljanost. Metode koje se koriste u distributivnim mrežama razlikuju se od onih koje se koriste u prenosnim i subprenosnim mrežama. Većina poznatih iterativnih postupaka koji se koriste u prenosnim i subprenosnim mrežama bazirani su na matričnom postupku, dok se u distributivnim mrežama koriste specijalizovani algoritmi koji su orijentisani ka granama [4].

3.1. Model i algoritam proračuna tokova snaga Newton/Raphson-ovog iterativnog postupka

Newton/Raphson-ov metod se sastoji od linearizovanja sistema nelinearnih jednačina koje se rešavaju oko utvrđene tekuće (h -te) aproksimacije rešenja. Tako dobijen sistem linearnih jednačina rešava se po korekcijama tekuće aproksimacije rešenja. Korigovana tekuća aproksimacija predstavlja novu – „bolju“ aproksimaciju rešenja sistema nelinearnih jednačina. Prva – početna aproksimacija rešenja ($h = 1$) bira se iskustveno (intuitivno), ili se generiše nekim jednostavnim postupkom.

Osnovni blok dijagram Newton/Raphson-ovog postupka prikazan je na slici 2. Dakle, nakon formiranja ulaza vrši se proračun debalansa aktivnih i reaktivnih snaga u čvorovima. Potom se ispituju uslovi konvergencije i u slučaju da su ispunjeni, proračun se završava. U suprotnom, prelazi se na sledeću iteraciju.

Celokupni režim elektroenergetskog sistema može da se rekonstruiše ako se poznaju modul i fazor napona svakog čvora. Te veličine predstavljaju stanje razmatranog sistema.

$$\mathbf{X}_s = [\theta_1, V_1, \theta_2, V_2, \dots, \theta_n, V_n]^T, \quad (1)$$

Vektor nepoznatih veličina vektora stanja glasi:

$$\mathbf{X} = [\theta_2, V_2, \dots, \theta_n, V_n]^T, \quad (2)$$

gde je sa n označen broj čvorova.

Problem tokova snaga je formulisan tako da je kompleksni napon balansnog čvora (njegov ugao θ_i^{spec} i modul U_i^{spec}) specificiran (zadat). Uvažavajući to, matematički model tokova snaga EES glasi:

Trivijalni deo

$$\theta_i = \theta_i^{spec}, \quad (3)$$

$$U_i = U_i^{spec}, \quad (4)$$

gde su:

U_i^{spec} – unapred zadat (specificiran) modul kompleksnog napona balansnog čvora,

θ_i^{spec} – unapred zadat (specificiran) ugao kompleksnog napona balansnog čvora.

Dominantni deo

$$P_k = g'_k(\mathbf{X}), \quad k=2, 3, \dots, n \quad (5)$$

$$Q_k = g''_k(\mathbf{X}), \quad k=2, 3, \dots, n \quad (6)$$

gde su:

P_k i Q_k – unapred zadate injektirane snage u čvorovima $k=2, 3, \dots, n$

$g'_k(\mathbf{X}), g''_k(\mathbf{X})$ – Predstavljaju sumu aktivnih (reaktivnih) snaga, respektivno, koje granama otiču iz čvora k .

Raspregnuti deo

$$P_I = g'_I(\mathbf{X}) \quad (7)$$

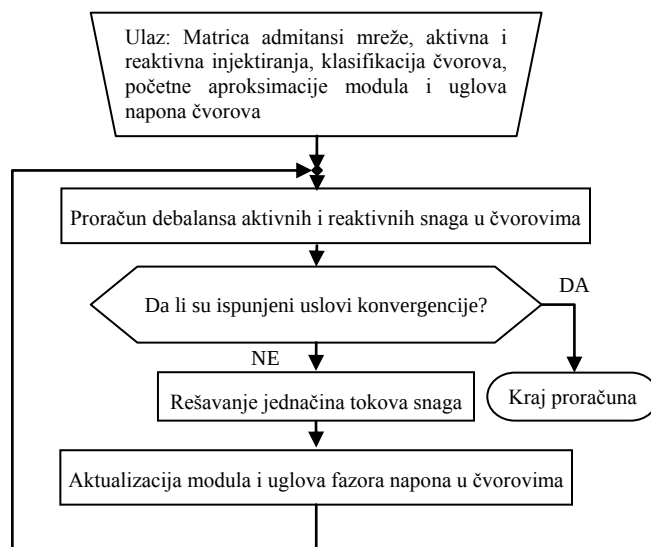
$$Q_I = g''_I(\mathbf{X}) \quad (8)$$

gde su:

P_I i Q_I – nepoznate injektirane aktivne i reaktivne snage balansnog čvora

$g'_I(\mathbf{X}), g''_I(\mathbf{X})$ – Predstavljaju sumu aktivnih (reaktivnih) snaga, respektivno, koje granama otiču iz balansnog čvora.

Dok su u relacijama trivijalnog dela eksplicitno iskazane dve komponente vektora stanja (ugao i modul napona prvog – balansnog čvora), dotle su preostale $(2n-2)$ komponente izražene implicitno, pomoću $(2n-2)$ relacije. Ako se ovih $(2n-2)$ relacija proglase sistemom jednačina, sa poznatim levim stranama ($P_k, Q_k, k=2, 3, \dots, n$) koje treba da se reše po već opisanom vektoru nepoznatih veličina $\mathbf{X}$, tada je skupom od $2n$ relacija opisan celokupni vektor stanja sa $2n$ komponenti. Relacije (7 i 8) mogu da se iskoriste za izračunavanje nepoznate injektirane aktivne i reaktivne snage balansnog čvora.



Slika 2 – Blok dijagram Newton/Raphson-ove metode [5]

3.2. Modeli algoritam proračuna tokova snaga Kompenzacione metode

Kao osnovni algoritam za proračun tokova snaga u radijalnim distributivnim mrežama (orijentisan ka granama) koristi se metoda sumiranja struja poznatija kao Shirmohammadi-ev algoritam. Algoritam počinje inicijalizacijom, nakon koje započinje iterativni postupak. U svakoj iteraciji postoje tri koraka: 1) proračun injektiranih struja za svaki čvor; 2)

proračun struja po granama (“zamena unazad”) i 3) proračun napona u čvorovima (“zamena unapred”). Nakon toga, vrši se proračun debalansa aktivnih i reaktivnih snaga i ispituju se uslovi konvergencije. Ukoliko su svi uslovi konvergencije ispunjeni, iterativni postupak se završava, u suprotnom prelazi se na sledeću iteraciju.

Proračun simetričnih tokova snaga trofaznih mreža sa petljama se sastoji od postupka sumiranja struja i korekcija napona, kao i od *kompensacije za petlje*.

Da bi mreža sa petljama mogla da se tretira isto kao radijalna mreža, potrebno je da se otvore sve njene petlje. Izbor mesta gde se otvaraju petlje slobodan je [2, 3].

Otvaranjem svih petlji, mreža se svodi na radijalnu mrežu – radijalizovana mreža. Režim u toj mreži se razlikuje od režima mreže sa petljama (originalne mreže). Da bi režim u radijalizovanoj mreži bio isti sa režimom originalne mreže, potrebno je da se izmene topološke strukture mreže, koje su se desile njenom radijalizacijom, kompenzuju odgovarajućim kompenzacionim strujama – supstituišu idealnim strujnim generatorima.

Kompensacione struje tih generatora se injektiraju u čvorove u kojima su petlje otvorene. Pošto su režimi isti, umesto modelovanja i rešavanja mreže sa petljama, može da se napiše i reši matematički model radijalizovane mreže sa insertovanim odgovarajućim idealnim strujnim generatorima u čvorovima otvaranja petlji.

Model takve mreže mora da se proširi sa relacijama koje opisuju jednakost napona parova čvorova nastalih otvaranjem petlji. Te relacije se zamenjuju odgovarajućim Thévenin-ovim relacijama.

Proračun mreže sa p petlji u $(h+1)$ iteraciji koristeći se Gauss/Seidel-ovom metodom može se predstaviti kao:

$$\hat{I}_k^{h+1} = \frac{\hat{S}_k}{\hat{U}_k^{h*}} + \hat{J}_{ck}^h + \hat{y}_{ok} \hat{U}_k^h + \sum_{j \in a_k} \hat{I}_j^h, \quad k = n+p, \dots, 3, 2 \quad (9)$$

$$\hat{U}_k^{h+1} = \hat{U}_K^{h+1} - \hat{z}_k \hat{I}_k^{h+1}, \quad k=2, 3, \dots, n+p \quad (10)$$

$$\Delta \hat{J}_c^h = \hat{Y}_N (\hat{U}_1^{h+1} - \hat{U}_2^{h+1}), \quad \left[\hat{Y}_N = (\hat{Z}_T)^{-1} \right] \quad (11)$$

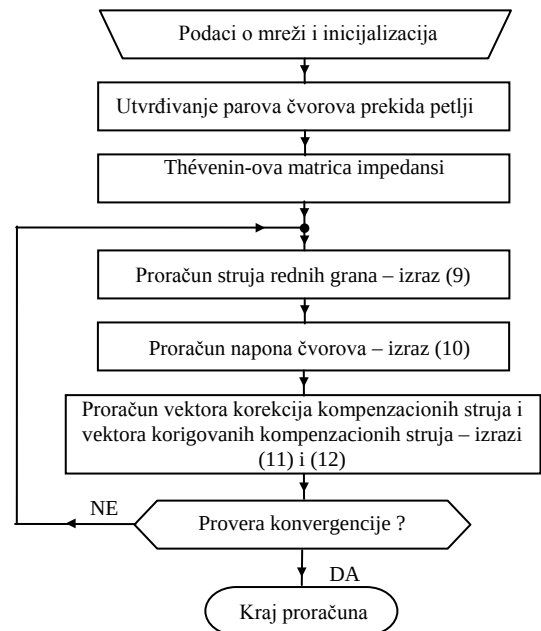
$$\hat{J}_c^{h+1} = \hat{J}_c^h + \Delta \hat{J}_c^h \quad (12)$$

Sa $\hat{U}_1^{h+1}$ i $\hat{U}_2^{h+1}$ označeni su vektori $(h+1)$ -vih aproksimacija napona originalnih i novogenerisanih čvorova u kojima se otvaraju petlje, respektivno, oba dimenzija $p \times 1$.

Elementi na istim pozicijama ta dva vektora odgovaraju paru čvorova otvaranja jedne petlje (originalni i novogenerisani).

Sa $\hat{Y}_N$ je označena Norton-ova matrica admitansi, odnosno inverzna Thévenin-ova matrica impedansi $\hat{Z}_T$ dimenzija $(p \times p)$ gde je p broj petlji. Sa $\Delta \hat{J}_c^h$ je označen vektor korekcija h -tih aproksimacija kompenzacionih struja $\hat{J}_c^h$, a sa $\hat{J}_c^{h+1}$ vektor korigovanih – $(h+1)$ -vih aproksimacija tih struja.

Njihove dimenzije su $p \times 1$. Proračun se završava u h -toj iteraciji, kada su zadovoljeni kriterijumi konvergencije. Blok dijagram algoritma prikazan je na slici 3.



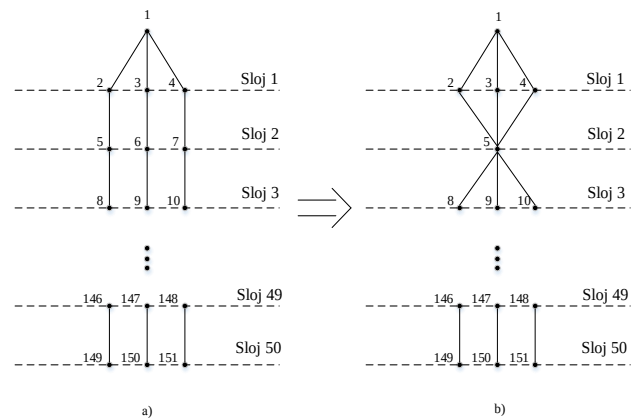
Slika 3–Blok dijagram Kompenzacione metode

4. UPOREDNA ANALIZA REZULTATA

U ovom delu data su dva primera proračuna.

4.1. Uticaj povećanja broja petlji na vreme izvršenja

U ovom primeru razmatra se 20kV mreža sa 3 fidera i 50 čvorova po fideru. Na ovom tipu mreže vrši se dodavanje petlji između svih čvorova koji pripadaju jednom sloju i to se ponavlja (počevši od drugog sloja) na svakom drugom sledećem sloju sve do poslednjeg. Primer kreiranja mreže prikazan je na slici 4.

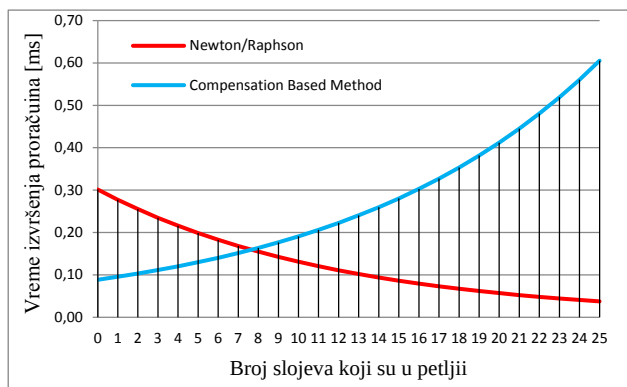


Slika 4 – a) Radijalna mreža sa 3 fidera i 50 čvorova po fideru, b) Primer upetljavanja jednog sloja

Parametri svih sekcija su isti. Ukupna potrošnja svih čvorova iznosi (6 MW, 3 MVar) i ravnomerno je raspoređena u svim čvorovima.

Na slici 5 prikazana je zavisnost vremena izvršenja proračuna od broja slojeva koji su upetljeni. Kriva koja opisuje vreme izvršenja Newton/Raphson-ovog proračuna prikazuje pad vremena sa porastom broja petlji, dok kriva koja opisuje Kompenzacionu metodu prikazuje porast vremena sa porastom petlji. Na ovom primeru pokazano je da Kompenzaciona metoda ubedljivo prednjači u odnosu na Newton/Raphson-a kada je u pitanju mali broj

petlji, to i predstavlja domen njene primene (distribucija sa malim brojem petlji), od trenutka kad je procenat upetljanosti veći od 16% (na prikazanoj mreži) Newton/Raphson preuzima primat kada je u pitanju vreme izvršenja.



Slika 5 – Zavisnost vremena izvršenja proračuna od broja slojeva koji su upetljani

U tabeli 1 dati su rezultati proračuna u kojima figurišu napon i njemu pripadajući fazni stav, za čvorove poslednjeg sloja date mreže. Isto tako prikazano je vreme izvršenja kao i broj iteracija.

Na osnovu rezultata prikazanih u tabeli 1, primetno je da se vrednost napona u slučaju postojanja petlji povećala u odnosu na vrednosti napona u slučaju radijalne strukture kada su u pitanju oba proračuna.

U_{last} [V], θ_{last} [°] – moduo napona čvorova u poslednjem sloju i pripadajući fazni stav.

Tabela 1 – Rezultati proračuna za mrežu 3*50 sa petljama

Mreža	Newton/Raphson			
	U_{last} [V]	θ_{last} [°]	Broj iteracija	Vreme [ms]
3*50, Radijalna	19909.735	-0.06795	2	0.3015
3*50, 8 slojeva u petlji	19913.134	-0.06539	2	0.1548
Mreža	Kompensaciona metoda			
	U_{last} [V]	θ_{last} [°]	Broj iteracija	Vreme [ms]
3*50, Radijalna	19909.389	-0.06828	2	0.0884
3*50, 8 slojeva u petlji	19912.803	-0.06572	4	0.1636

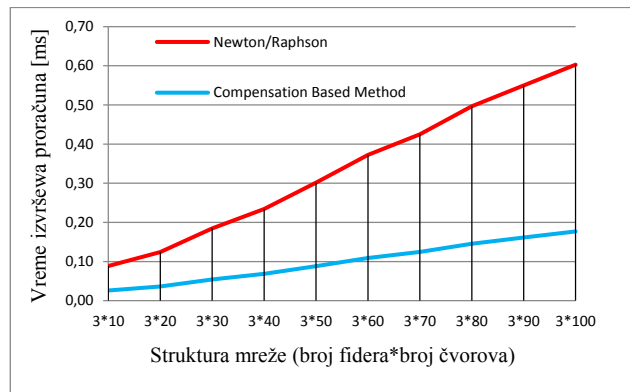
4.2. Uticaj povećanja dimenzija radijalne mreže na vreme izvršenja

U ovom primeru razmatra se 20kV mreža sa 3 fidera i 10 čvorova po fideru, radijalna. U svakom sledećem proračunu dodaje se po 10 čvorova po fideru sve dok se ne dođe do mreže sa 3 fidera i 100 čvorova po fideru. Cilj je da se vidi kako povećanje mreže utiče na dužinu trajanja proračuna kada je u pitanju radijalna mreža.

Parametri svih sekcija su isti. Ukupna potrošnja svih čvorova iznosi (6 MW, 3 MVar) i ravnomerno je raspoređena u svim čvorovima.

Na slici 6 prikazana je zavisnost vremena izvršenja proračuna od broja čvorova po fideru. Povećanjem broja

čvorova po fideru raste i vreme izvršenja proračuna. U datom primeru, rast broja čvorova po fideru (povećanje mreže) dovodi do linearnog rasta vremena izvršenja oba proračuna.



Slika 6 – Zavisnost vremena izvršenja proračuna od broja čvorova po fideru (radijalna mreža)

5. ZAKLJUČAK

U uporednoj analizi rezultata proračuna dobijenih primenom Kompenzacione metode i Newton/Raphson-ovog iterativnog metoda uočeno je minimalno odstupanje rezultata što potvrđuje tačnost oba proračuna. Kada je u pitanju vreme izvršenja proračuna na različitim primerima, dolazi se do zaključka da je ono u direktnoj vezi sa brojem petlji koje se javljaju u mreži. Kompenzaciona metoda ima prednost u brzini kada su u pitanju mreže sa malim brojem petlji (tipično za distribuciju), dok Newton/Raphson ima prednost u mrežama sa većim brojem petlji (tipično za subprenos i prenos).

6. LITERATURA

- [1] V. C. Strezoski: *Analiza elektroenergetskih sistema*, skripta, Fakultet tehničkih nauka Novi Sad, 2011.
- [2] D. Shirmohammadi, H. W. Hong, A. Semlyen, G. X. Luo: A Compensation-Based Power Flow Method for Weekly Meshed Distribution and Transmission Network; *IEEE Trans. on PS*, Vol.3, No.2, May 1988, pp. 753-762.
- [3] G. X. Luo, A. Semlyen: Efficient Load Flow for Large Weekly Meshed Distribution and Transmission Network; *IEEE Trans. on PS*, Vol.5, No.4, November 1990, pp. 1309-1316
- [4] V.C.Strezoski: *Osnovi Elektroenergetike*; Fakultet tehničkih nauka u Novom Sadu, Novi Sad, 2014.
- [5] M.S.Čalović, A.T.Sarić: *Osnovi analize elektroenergetskih mreža i sistema*; Akademski misao, Beograd, Tehnički fakultet, Čačak; 2004.

Kratka biografija:



Marko Petković rođen je u Kikindi 1990. godine. Živi u Novom Sadu. Diplomski rad je odbranio 2014. godine na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi.

ANALIZA DETEKTORA I DESKRIPTORA ZA KLASIFIKACIJU OBJEKATA U SLICI I PRIMENA U OKVIRU ANDROID PLATFORME**ANALYSIS OF DETECTORS AND DESCRIPTORS FOR CLASSIFICATION OF OBJECTS FROM IMAGES AND THEIR USE ON AN ANDROID PLATFORM**

Ivan Lazić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu dat je kratak prikaz poznatih metoda izdvajanja tačaka od interesa, kao i izdvajanja obeležja iz njih. Potom na osnovu ideja ASIFT algoritma uvođenja afine invarijantnosti, pokušani su isti principi i na klasama binarnih deskriptora sa dodatnim formiranjem obeležja putem modela vreće reči. Takođe je formirana baza slika odabranih zgrada Novog Sada nad kojom je izvršen krosvalidacioni postupak za dobijanje ocene ove metode u odnosu na originalnu. Na osnovu dobijenih rezultata realizovan je softver u vidu android aplikacije koja će koristiti najbolje pokazane metode direktno na uređaju, koristeći OpenCV biblioteke.

Abstract – This paper provides a brief overview of the more known methods of keypoint detection and its description. Then, based on the ASIFT algorithm, the same principles were tested on the binary family of descriptors, with additional construction of features using the bag of words model. Additionally, a dataset comprising of selected buildings in Novi Sad was formed, on which the crossvalidation process was run (in order to evaluate the new methods). Based on the results, a software was implemented in the form of an android application which will use the best resulting methods directly on the device, with the use of OpenCV libraries.

Ključne reči: *Detekcija tačaka od interesa, Izdvajanje obeležja, Model vreće reči, Klasifikacija, Android*

1. UVOD

Poslednjih godina, mobilni telefoni polako zamenjuju konvencionalne alate i uređaje time što imaju mogućnost da izvršavaju sve kompleksnije i raznovrsnije operacije. Jedna od novijih primena je u okviru tzv. augmentisane realnosti, motivisana činjenicom prenosive prirode mobilnog uređaja, kao i postojanjem ugrađene kamere.

Prve AR aplikacije u okviru rešavanja problema iz kompjuterske vizije bazirale su se na pružanju informacija o objektima putem analize i klasifikacije obeležja sa slike. Uprkos porastu mogućnosti mobilnih telefona, većina aplikacija ovog tipa se zasniva na korišćenju kamere uređaja za akviziciju podataka, koji se zatim putem bežične mreže šalju bazi na obradu, koja potom vraća rezultate.

Cilj ovog rada biće osvrst na neke od dosadašnjih metoda izdvajanja tačaka od interesa sa slika, kao i metoda njihovog opisivanja, sa fokusom na binarnim deskriptorima i njihovim mogućim proširenjima, zbog efikasnosti njihovog računanja i manje memorijske zavisnosti. Softverska implementacija rešenja će se sastojati iz dva dela, od kojih će se jedan baviti problemom treniranja na desktop računaru u offline režimu, a drugi online deo, implementiran na android platformi, koji će vršiti obradu podataka i klasifikaciju direktno na uređaju.

2. DESKRIPTORI SA REALNIM VREDNOSTIMA

U okviru ovih deskriptora spadaju popularni SIFT (Scale Invariant Feature Transform) [1] i SURF (Speeded-up Robust Features) [2] detektori i deskriptori, kao i unapređenje SIFT algoritma na afine promene ASIFT (Affine-SIFT) [3].

2.1. SIFT

Detekcija ovog algoritma se vrši na bazi laplasijana gausijana, s tim da SIFT dodatno uvodi aproksimaciju njega sa razlikom gausijana. Robustnost na promenu veličine ostvaruje time što kreira piramidalnu reprezentaciju po veličinama, nad kojom vrši detekciju, a robustnost na rotaciju ostvaruje tako što računa orijentaciju kumulativnog gradijenta za posmatrani region u okolini tačke od interesa.

Izdvajanje obeležja iz datih tačaka od interesa SIFT vrši tako što posmatra region veličine 16x16 piksela oko tačke, koji je potom izdvojen na 16 4x4 podregiona i računa gradijent i orijentaciju za svaki. Za svaki podregion potom pravi histogram od 8 podeoka, gde svaki podeok odgovara uglu od 45 stepeni. Konačni deskriptor predstavlja vektor sa svim histogramima i čini ga ukupno 128 realnih brojeva.

2.2. SURF

Slično SIFT algoritmu SURF vrši aproksimaciju laplasijana gausijana sa box filterima. Takođe koristi piramidalnu reprezentaciju veličina za uvođenje robustnosti na promenu veličine. Robustnost na rotaciju uvodi tako što posmatra odzive na Har vejevle u datom regionu tačke od interesa. Rezultate potom mapira u xy koordinatni sistem i pokretnim prozorom dimenzije kružnog isečka sa uglom od 60 stepeni računa kumulativnu orijentaciju odziva, gde će maksimalna dobijena vrednost služiti pri normalizaciji na orijentaciju.

Za izdvajanje obeležja posmatra se kvadratni region u okolini tačke od interesa koji je izdvojen na 16 jednakih delova. Po njima se potom računaju odzivi na Har vejevle gde se po regionu formira vektor od 4 elementa. Na

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dejan Vukobratović, vanr. prof.

kraju, konačni deskriptor je vektor od ukupno 64 realnih vrednosti.

2.3. ASIFT

Nastao kao modifikacija SIFT algoritma, u kome uvodi potpunu robustnost na affine promene u slikama. Koristeći SIFT algoritam koji normalizuje promene rotacije i simulira promenu veličine, ASIFT dodatno simulira promene orijentacije kamere, što po autorima algoritma dodatno povećava robustnost detekcije i na perspektivne transformacije. Suštinska ideja algoritma je da se uradi SIFT detekcija i deskripcija za svaku afino transformisanu sliku za određen broj parametara.

Rezultati ovog algoritma daju motivaciju i za njegovu primenu na drugim algoritmima, pod uslovom da imaju zadovoljenu normalizaciju nad rotacijom i simulaciju nad veličinom.

3. DESKRIPTORI SA BINARNIM VREDNOSTIMA

Uprkos tome što su SIFT i SURF davali odlične rezultate, imali su veliku kompleksnost pri računanju. U nastojanju da se smanje brzine računanja, razvijeni su deskriptori binarnog tipa, koji su brzi za računanje i imaju uporedive rezultate sa SIFT i SURF algoritmom.

3.1. ORB

ORB (Oriented FAST and rotated BRIEF [4]) je sastavljen od dva algoritma za detekciju i izdvajanje obeležja.

Detekcija se vrši putem FAST (Features from Accelerated Segment Test [5]) algoritma, poređenjem intenziteta posmatranog piksela sa pikselima raspoređenim po krugu oko njega, prečnika 3. Ukoliko se nađe 9 njih koji zadovoljavaju uslov da su svi veći (ili manji) od centralnog za prag T , posmatrani centralni piksel postaje tačka od interesa. Takođe, slično SIFT algoritmu, konstruiše se piramida veličina radi uvođenja robustnosti na promenu veličine, a orijentacija dobijene tačke od interesa se računa preko momenta regiona.

Izdvajanje obeležja se radi slično BRIEF (Binary Robust Independent Elementary Features [6]) algoritmu. Za region u okolini tačke od interesa radi se upoređivanje intenziteta nad parovima piksela određenim šablonom i kao rezultat jednog poređenja se dobija vrednost od jednog bita. U okviru ORB algoritma vrši se ukupno 256 ovakvih poređenja, što daje kao deskriptor vektor od 256 bita.

3.2. BRISK

BRISK (Binary Robust Invariant Scalable Keypoints [7]) radi na sličnom principu kao i ORB.

Detekcija se vrši putem modifikovanog AGAST (Adaptive and Generic corner detection based on the Accelerated Segment Test [8]) algoritma, koji je baziran na optimizovanom FAST algoritmu u vidu stabla odluke za bržu detekciju, sa dodatnim uvođenjem piramide veličina.

Za formiranje vektora obeležja, BRISK posmatra okolinu tačke od interesa pomoću predefinisano šablona. U okviru njega se formiraju parovi koji su na većim rastojanjima, koji služe za procenu orijentacije datog regiona; i parovi koji su na kraćim rastojanjima, čijim upoređivanjem dobijamo konačni deskriptor. Ukupno, formira se 512 parova na kraćim rastojanjima, što odgovara vektoru obeležja od 512 bita.

3.3. FREAK

FREAK (Fast Retina Keypoint [8]) predstavlja samo algoritam za formiranje vektora obeležja. Za detekciju tačka od interesa autori predlažu korišćenje detektora iz BRISK algoritma.

Vektor obeležja, slično BRISK algoritmu, formira se posmatranjem parova određenih predefinisanim šablonom. Specifično za ovaj algoritam, koncentracija tačaka šablona je veća u sredini i opada sa porastom udaljenosti.

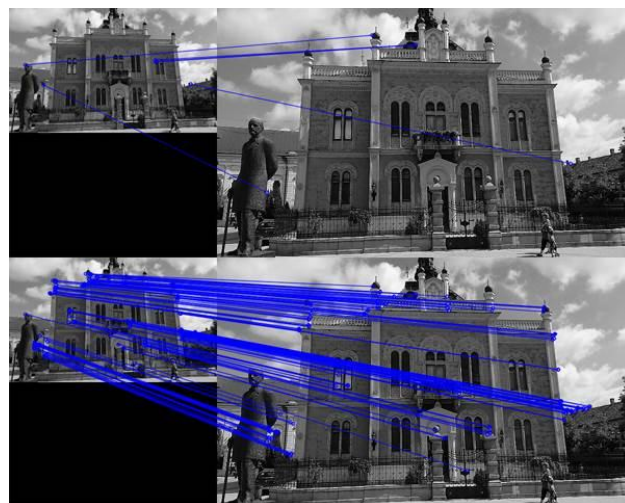
Za uvođenje robustnosti na rotaciju, FREAK posmatra skup od 45 parova tačaka koji su simetrično postavljeni u odnosu na centar i računanjem lokalnog gradijenta dobija meru orijentisanosti posmatranog regiona.

Kao konačni vektor obeležja posmatra se 512 posebno odabranih parova piksela koji daju deskriptor od 512 bita.

3.4. LATCH

LATCH (Learned Arrangements of Three Patch Codes [9]) je jedan od novijih algoritama za izdvajanje obeležja iz tačaka od interesa. Kao novinu, LATCH uvodi posmatranje trojki piksela (od kojih je jedna nazvana "sidro") i njihovih regiona veličine 7×7 piksela, gde se upoređuje Frobenijusevo rastojanje razlike regiona sidra i jednog piksela sa rastojanjem razlike regiona u okolini drugog piksela i sidra. Rezultat od 512 poređenja se stavlja u vektor koji formira konačni deskriptor. Dodatno, robustnost na rotaciju se uvodi u zavisnosti od orijentacije tačke od interesa, tako što se šablon remapira na datu orijentaciju.

Međutim, autori LATCH algoritma ne uzimaju u obzir promene u odnosu na veličinu, što značajno utiče na performanse deskriptora. Stoga je za ovaj algoritam, u okviru implementacije ovog rada, dodatno uvedena konstrukcija piramide veličina, po uzoru na ORB piramidu. Rezultat uparivanja tačaka od interesa pre i posle ove modifikacije je prikazan na slici 1.



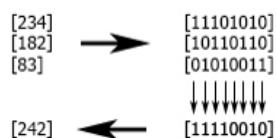
Slika 1. Uparivanje tačaka sa originalnim i modifikovanim LATCH algoritmom

4. MODEL VREĆE REČI

Model vreće reči [10] je prvobitno predložen kao način pojednostavljenja reprezentacije tekstualnih dokumenata za jednostavniju klasifikaciju. Ideja je da se na osnovu fiksnog rečnika mogu formirati obeležja koja će biti ravnopravna, nezavisno od veličine tekstova u formi histo-

gramske reprezentacije pojavljivanja reči iz rečnika. Stoga, možemo imati dva dokumenta različite dužine, ali sa sličnim i uporedivim obeležjima ukoliko imaju istu sadržinu.

Generalizacija ovog modela omogućuje i primenu u okviru kompjuterske vizije. U ovom slučaju "reči" su vektori obeležja dobijeni iz navedenih algoritama, a "rečnik" se formira klasterizacijom svih dobijenih deskriptora. U slučaju realnih deskriptora (SIFT i SURF), klasterizacija se vrši standardnim k-means algoritmom, dok u slučaju binarnih deskriptora, potrebno je uvesti modifikaciju. U tom slučaju, slično ideji autora iz [11], klasterizacija će biti u obliku modifikacije k-means algoritma, s tim da se odluka za pojedinačno obeležje centra klastera dobija po principu glasanja, prebrojavanjem svih obeležja koji spadaju pod njega, a za meru rastojanja se koristi Hamingovo rastojanje. Ilustracija ovakvog načina određivanja centra je prikazana na slici 2.



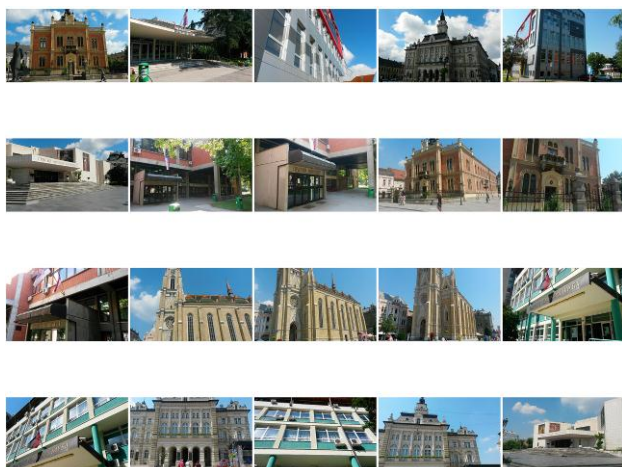
Slika 2. Ilustracija ideje glasanja

Za datih 8 obeležja iz 3 deskriptora (koji su grupisani u celobrojne vrednosti zbog memorijske efikasnosti) metodom glasanja odlučujemo vrednosti 8 obeležja centra, koji potom ponovo formiraju celobrojnu vrednost.

Međutim, autori iz [12] sugerišu da je dovoljno dobro koristiti i k-means algoritam za konstrukciju rečnika.

5. EKSPERIMENT I REZULTATI

Za potrebe testiranja formirana je baza slika od 9 klasa, gde svaka sadrži 50 slika pojedine zgrade veličine 800x480 piksela, koje su slikane iz relativne blizine sa varirajućim uglom posmatranja, gde su neke od njih prikazane na slici 3.

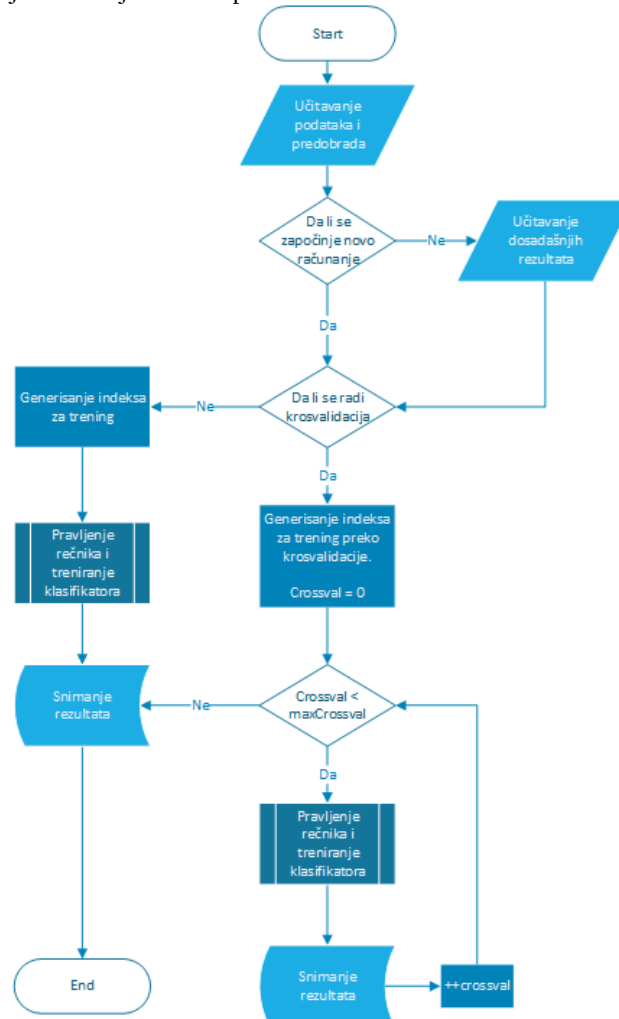


Slika 3. Primer nekih od prikupljenih slika

Slike su prikupljene pomoću android uređaja putem posebno razvijene android aplikacije koristeći OpenCV biblioteke. Programsko rešenje je podeljeno u dva dela: offline i online.

4.1. Offline

Offline deo se odvija na desktop računaru i za zadate parametre vrši testiranje algoritama putem metode krosvalidacije ili kreira potrebne datoteke za android aplikaciju. Programsko rešenje je ostvareno u okviru Microsoft Visual okruženja, koristeći OpenCV biblioteke, čija je pojednostavljena šema prikazana na slici 4.



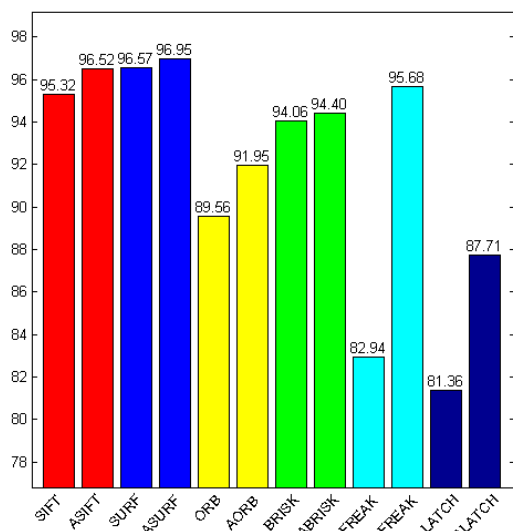
Slika 4. Pojednostavljena šema programskog rešenja offline dela

Pri svakom obliku detekcije, gledalo se da imaju približno sličan broj tačaka od interesa. Rezultati na slici 5 prikazuju da uvođenje funkcije za afinu invarijantnost poboljšava performanse ne samo SIFT već i drugih testiranih deskriptora, potvrđujući osnovnu ideju rada o korišćenju ove modifikacije za binarne deskriptore i to u formi klasifikacije putem modela vreće reči. U okviru ove baze slika FREAK je pokazao najbolje rezultate od svih ostalih binarnih deskriptora.

4.2. Online

Online deo se izvršava u okviru implementirane android aplikacije, što je ostvareno u okviru Eclipse okruženja sa dodatnim OpenCV bibliotekama.

Primer rezultata klasifikacije dat je na slici 6.



Slika 5. Grafik tačnosti klasifikacije sa SVM koristeći algoritme bez i sa dodatkom funkcije za afine transformacije za rečnik dimenzije 500



Slika 6. Primer rezultata klasifikacije na androidu

6. ZAKLJUČAK

U ovom radu opisano je rešenje problema klasifikacije objekata, uz prikaz jednostavnog sistema za treniranje i testiranje. Takođe su ispitane mogućnosti današnjih mobilnih telefona pri procesovanju ovih algoritama direktno na uređaju, čime se isključuje zavisnost od internet mreže. U praktičnoj implementaciji na android platformi potrebno je postići što veću tačnost, vodeći računa o memorijskoj zahtevnosti. Dodatno treba voditi računa i o vremenu potrebnom za izvršavanje, što je i bila glavna motivacija ispitivanja i korišćenja binarnih deskriptora. Bitan faktor je takođe i izbor deskriptora. Poboljšanja su uvek moguća u vidu dodatne predobrade, s tim da je ispitano samo mali skup parametara pri procesu generisanja obeležja i klasifikacije.

7. LITERATURA

- [1] Lowe, D. G. (1999), „Object recognition from local scale-invariant features“, Proceedings of the Seventh IEEE International Conference on Computer Vision
- [2] Bay, H., Tuytelaars, T., & Van Gool, L. (2006), „SURF: Speeded up robust features“
- [3] Morel, J.-M., & Yu, G. (2009), „ASIFT: A New Framework for Fully Affine Invariant Image Comparison“, SIAM Journal on Imaging Sciences
- [4] Rublee, E., & Bradski, G. (2011), „ORB - an efficient alternative to SIFT or SURF“
- [5] Rosten, E., Porter, R., & Drummond, T. (2010), „Faster and better: A machine learning approach to corner detection“, IEEE Transactions on Pattern Analysis and Machine Intelligence
- [6] Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010), „BRIEF: Binary robust independent elementary features“
- [7] Leutenegger, S., Chli, M., & Siegwart, R. (2011), „BRISK: Binary Robust Invariance Scalable Keypoints“, Proceedings of the International Conference on Computer Vision
- [8] Alahi, A., Ortiz, R., & Vandergheynst, P. (2012), „FREAK: Fast retina keypoint“, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition
- [9] Levi, G. (2015), „LATCH : Learned Arrangements of Three Patch Codes“
- [10] Tsai, C.-F. (2012), „Bag-of-Words Representation in Image Annotation: A Review“, ISRN Artificial Intelligence.
- [11] Grana, C., Borghesani, D., Manfredi, M., & Cucchiara, R. (2013), „A fast approach for integrating ORB descriptors in the bag of words model“, IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics
- [12] Markatopoulou, F., Pittaras, N., & Papadopoulou, O. (2015), „A Study on the Use of a Binary Local Descriptor and Color Extensions of Local Descriptors for Video Concept Detection“

Kratka biografija:



Ivan Lazić rođen je u Sremskoj Mitrovici 1989. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Obrada signala odbranio je 2016.god.

**AUTOGENERISANJE LISTE RIZIČNIH ISPADA ZA POTREBE ANALIZE
SIGURNOSTI ELEKTROENERGETSKOG SISTEMA****AUTOGENERATION OF RISK-BASED OUTAGE LIST FOR CONTINGENCY
ANALYSIS OF POWER SYSTEMS**

Branko Đukić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu prikazan je metod za autogenerisanje liste rizičnih ispada, koja bi se potom koristila kao ulaz za analizu sigurnosti, čime bi se ona ubrzala i usredsredila samo na potencijalno kritične ispadе. Predloženo je formiranje kriterijuma rizika na osnovu verovatnoće ispada i njegovih mogućih posledica po rad sistema. Algoritam je testiran na test prenosnoj mreži i rezultati su prikazani u vidu dobijene liste ispada.

Abstract – In this paper a method for autogeneration of risk-based outage list is presented. This list is later used as an input to contingency analysis, improving its computational speed and efficiency. Criterion for risk assessment based on probability of outages and their effect on system operation is proposed. Algorithm is tested on test transmission network and resulting outage list is provided.

Cljučne reči: Analiza sigurnosti, lista ispada, procena rizika.

1. UVOD

Problem složenosti i vremenske zahtevnosti analize sigurnosti elektroenergetskog sistema (EES) predmet je mnogih radova i razvijeni su različiti metodi za njegovo prevazilaženje. Pored metoda rangiranja i skeniranja liste ispada korišćenjem aproksimativnih proračuna, sve češće je i uvođenje probabilističkog pristupa, tj. procene verovatnoće ispada. U ovom radu je predstavljena ideja autogenerisanja liste ispada za potrebe analize sigurnosti na osnovu kriterijuma rizika. Pod rizikom ispada podrazumeva se verovatnoća da do njega dođe, kao i moguće posledice po rad sistema. Analiza sigurnosti bi se potom vršila samo nad tom listom, čime bi se izbeglo nepotrebno analiziranje mnogih nekritičnih ispada i vreme izvršavanja značajno skratilo.

2. ANALIZA SIGURNOSTI

Pod pojmom sigurnosti EES podrazumeva se sposobnost sistema da bez većih posledica podnese neke nepredviđene, ali verovatne poremećaje iz unapred formirane liste mogućih poremećaja [1]. Dakle, sistem je siguran ako pri bilo kom poremećaju sa specificirane liste, njegove komponente nisu opterećene preko svojih ograničenja, naponi u čvorovima i učestanost u sistemu ne odstupaju od specificiranih vrednosti preko granica tolerancije. Poznavanje efekata poremećaja je neophodno

radi određivanja preventivnih mera u cilju sprečavanja kaskadnih (lančanih) poremećaja. Ta lista "verovatnih" poremećaja koje treba analizirati se može sastavljati na osnovu kombinacije istorijskih podataka i inženjerskog iskustva mrežnih operatora. Međutim, za veće prenosne mreže ovaj zadatak postaje previše zahtevan, pa je neophodno razviti algoritam za autogenerisanje te liste. Ona ne sme biti previše dugačka, pošto su tehnike za proračun režima sistema nakon ispada bazirane na proračunu tokova snaga, pa kako bi se analiza vršila prihvatljivo brzo lista mora sadržati samo potencijalno kritične ispadе [1].

S obzirom da se od sistema očekuje da osim jednostrukih ispada "preživi" i neke od verovatnijih višestrukih, na listi je potrebno da se nađu i takvi ispadи. Međutim, broj kombinacija ukupnog broja elemenata u sistemu je $N!/(k!(N-k)!)$, pa se zahteva i isto toliko uzastopnih simulacija. Na primer, za mrežu od 3000 elemenata, broj mogućih dvostrukih ispada je skoro 4.5 miliona [1]. Proračun tokova snaga sa uvažavanjem svakog ispada pojedinačno bi bio spor i zahteva puno računarskih resursa, do te mere da je neizvodljiv već za malo veće sisteme [2]. Da bi se rešio problem ovako visoke složenosti, razvijene su razne aproksimativne metode, zasnovane na ideji da velika većina mogućih ispada ne izaziva veće posledice i nije ih potrebno detaljno analizirati. Prema tome, pre same analize sigurnosti, neophodno je obraditi ispadе nekim bržim metodom, kako bi se izdvojili samo oni koji su potencijalno kritični.

Za te potrebe, brojne su metode rangiranja, koje su karakteristične po tome što ne daju informaciju da li ispadи izazivaju narušavanje nekih ograničenja, već samo kvantifikuju kritičnost ispada relativno u odnosu na druge, performance index-ima. Potom, da bi se proverila kritičnost, potrebna je kompletna analiza proračunom tokova snaga. Problem kod ovih metoda je što nisu pouzdane i moguć je efekat maskiranja [3]. S druge strane, metode skeniranja predstavljaju složenije rešenje, ali obradom ispada aproksimativnim metodama proveravaju kritičnost i time se prevazilazi pomenuti problem maskiranja. Zbog svojih prednosti, ovim metodama je posvećeno više istraživanja i razvijeni su različiti efikasni algoritmi, kako za jednostruke, tako i za višestruke ispadе.

Da bi oni imali smisla, moraju biti mnogo brži od kompletne analize sa AC proračunom tokova snaga, pa se primenjuju razne aproksimacije i heurističke tehnike, koje bi trebale neznatno da degradiraju preciznost, ali i da donesu osetnu uštedu na vremenu i računarskim

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Andrija Sarić, red. prof.

resursima. U prvim radovima režimi sistema nakon ispada su dobijani pomoću proračuna DC tokova snaga [3], ili prve iteracije brzog dekoplovanog proračuna tokova snaga [4]. Proračun DC tokova snaga je neiterativan i mnogo brži postupak od AC proračuna, a rezultati su zadovoljavajuće preciznosti. Mana u odnosu na AC proračun je u nemogućnosti provere naponskih ograničenja. Još veća efikasnost se može postići proračunom *distributivnih faktora*. Ovi faktori predstavljaju osetljivost grana mreže na ispad drugih grana. Pošto zavise samo od topologije mreže, ne menjaju se sve dok je topologija ista, pa ako su već izračunati skeniranje upotrebom istih je još brže.

Razvijena su i mnoga proširenja ovih metoda na višestruke ispadе. Jedna od takvih ideja je smanjivanje skupa ograničenja koje je potrebno proveravati za ispadе. Predloženo je iterativno pomeranje skupa ograničenja, od najstrožijih ka najblažim, i time postepeno smanjivanje skupa ispada koje je potrebno analizirati [1]. Slična ideja podrazumeva iterativno sužavanje granica distributivnih faktori, od najšireg (od -1 do 1) ka užim, konzervativnijim intervalima. Utvrđeno je da se već u prvim iteracijama izbacili veliki broj nekritičnih ispada, kao i veliki broj ograničenja za koje nema potrebe proveravati ostale, potencijalno kritične ispadе [5]. Ovakvi zaključci dovode do ideje *pre-screeninga*, odnosno da je i za *screening* potrebno uraditi pripremu i smanjiti skup ispada koji treba da se skenira.

3. PROBABILISTIČKI PRISTUP

Tradicionalno, sigurnost se postiže konzervativnim pristupom, uz pomoć determinističkih tehnika, u kojima se ne razmatra kolika je verovatnoća da do nekog poremećaja dođe, već samo njegove moguće posledice. Da bi EES u svakom trenutku radio sa zadovoljenim kriterijumom sigurnosti, određuje se strog skup granica, kojim se značajno umanjuje fleksibilnost sistema, nedovoljavajući potpuno iskorišćenje kapaciteta elemenata. Pojava deregulisanog tržišta, uz konstantan rast potrošnje električne energije, dovodi do potrebe za prelaskom na fleksibilniju i konkurentniju eksploataciju sistema. Imajući ovo u vidu, neizbežno se nameće unapređenje postojećih metoda i zamena tog strogo determinističkog pristupa analizi sigurnosti probabilističkim, koji će uzimati u obzir i verovatnoće poremećaja, kako bi se izbeglo da neki poremećaji koji su malo verovatni dovedu do isuviše pesimističnog određivanja pogonskih ograničenja.

Iako je probabilistička priroda sigurnosti EES-a bila prepoznata i jasna od same pojave prvih modernih sistema, obično je bila zanemarivana. Čak i neki praktični pokušaji koji su davali pozitivne rezultate, nisu bili šire prepoznati, pa se i u današnjoj praksi ovom problemu i dalje prilazi uglavnom deterministički. Ipak, u novijim radovima dobijaju nezanemarljivu pažnju, ali zbog složenosti problema relativno sporo se prihvataju i implementiraju. U [6] rizik je definisan kao proizvod verovatnoće i efekta ispada na rad sistema. Dok se u determinističkom pristupu sigurnosne granice proračunavaju na osnovu najkritičnijih ispada, u ovom radu se uzima u obzir rizik od svih ispada sa specificirane liste. Time se dobijaju fleksibilnije granice i omogućava

se bolja eksploatacija sistema, što je veoma važno za konkurentnost na tržištu. Dakle, sam metod za proračun režima sistema nakon ispada se ne razlikuje, pa se prelazak na probabilistički pristup ne mora strogo smatrati zamenom starog, determinističkog, već radije kao ekstenzija istog. Ovakav pristup omogućava da se premosti razdvojenost sigurnosti i ekonomičnosti EES-a, predlažući načine za ekonomsko kvantifikovanje sigurnosti, kakvo je neophodno u uslovima deregulisanog tržišta [6].

4. FORMIRANJE KRITERIJUMA RIZIKA

Ideja ovog rada je da se pre detaljne analize ispada na brz način obrade svi ispadi i izdvoji skraćena lista, na osnovu jednog integralnog kriterijuma koji bi predstavljao meru rizika od pojave nekog poremećaja i njegovih posledica po sistem [7]. Da bi ta lista bila "realistična", moraju se obuhvatiti i verovatnoće ispada. Postoji više mogućnosti za uvažavanje verovatnoće ispada elemenata i koja će se primeniti uglavnom zavisi od podataka koji su na raspolaganju, kako istorijskih, tako i kataloških podataka o samim elementima [6]. Posebno ukoliko se obrađuju i višestruki ispadi, kojih je prevelik broj, važno je izdvojiti samo one koji zaista mogu da se dogode. Međutim, prostim množenjem verovatnoća ispada svih elemenata koji ga sačinjavaju ne uzima se u obzir njihova fizička ili električna udaljenost. Da bi se prevazišao ovaj problem, u [7] se predlaže uvođenje pojma *kredibiliteta* ispada, kojeg bi sačinjavala dva faktora: verovatnoća da se ispad dogodi (*likelihood*, *probability*) i verodostojnost, koliko je ispad realističan (*plausibility*). Sama lokalna priroda uzroka ispada govori u prilog tome da verodostojnost u mnogome zavisi od bliskosti elemenata u ispadu [7].

Pored kredibiliteta ispada, kriterijum rizika potrebno je da obuhvati i bitnost elementa. Kritičnost neće biti obuhvaćena, čime se ovaj rad razlikuje od drugih radova o proceni rizika. Ovde se rizik koristi za potrebe autogenerisanja liste, nad kojom će se tek kasnije vršiti provera kritičnosti, skeniranjem ili detaljnom analizom. Time se značajno ubrzava ceo proces, što je i glavni cilj.

U ovom radu se često naglašava reč "procena", s obzirom da se pojmovi poput "rizika", "kredibiliteta", "verodostojnosti", "bitnosti" i slični ne mogu smatrati egzaktnim [7]. Koliko god da se unapređuju primenjene tehnike za ovu svrhu, uvek će biti neegzaktni i podložni promenama i neizbežni su slučajevi gde je diskutabilno koji su prihvatljivi rezultati, da li je neki ispad rizičan ili ne i u kojoj meri. Zbog toga je ključna "procena" i ovakvi radovi služe pre kao otvaranje ili nastavak diskusije o problemu analize sigurnosti, nego kao njeno razrešenje ili zatvaranje.

Integriranje bitnosti u kriterijum rizika pre svega omogućava da se odbace elementi za koje se na osnovu inženjerskog iskustva može smatrati da njihov ispad ne bi nosio velike posledice. Tu spadaju slabo opterećeni elementi, elementi koji se nalaze u dobro upetljenim delovima, gde bi njihovo opterećenje lako mogli da preuzmu susedni elementi i sl. Takođe, ovim se operatoru skreće pažnja koji elementi bi eventualnim ispadom nosili teške posledice, što zbog velikog opterećenja istih, što zbog stvaranja novih ostrva u sistemu. Naročito su rizične situacije ako ispad dovodi do odvajanja generatora od

glavne mreže, čime se rizikuje njihov ispad iz sinhronizma i druge teže posledice. Za detekciju ostrva koristiće se laka i efikasna metoda analize grafa mreže. Broj generatora u novim ostrvima ne daje mnogo informacija, jer se ne može znati koliko su to veliki generatori i koliko su bili eksploatisani u trenutku odvajanja od glavne mreže. Prema tome, sumiraju se odate snage svih generatora koji se nađu u novim ostrvima. Kao i provera opterećenja, i provera pojave novih ostrva pri ispadu nije naročito vremenski zahtevna, a daje vrlo korisne informacije o bitnosti elementa za rad sistema.

Sada je konačno kompletiran kriterijum bitnosti, pa se može dati formula za rizik od ispada:

$$Risk = Probability \cdot Importance, \quad (1)$$

gde su:

Risk - rizik koji nosi ispad;

Probability - verovatnoća ispada elementa;

Importance - bitnost elementa.

Formula je dakle slična kao i u [6], s tim da ne obuhvata kritičnost, jer se pri autogenerisanju rizične liste ne proračunavaju režimi nakon ispada. Bitnost je formirana kao:

$$Importance = Loading \cdot Islanding, \quad (2)$$

gde su:

Loading - opterećenje elementa (u MW);

Islanding - pojava novih ostrva u sistemu, računa se kao:

$$Islanding = New_Islands \cdot \sum_i (P_{gen,i} + Q_{gen,i}), \quad (3)$$

gde su:

New_Islands - broj novih ostrva u sistemu nakon ispada;

P_{gen,i} - aktivna snaga koju *i*-ti generator odaje mreži;

Q_{gen,i} - reaktivna snaga koju *i*-ti generator odaje mreži.

Algoritam podrazumeva iteriranje po toj listi *N*-1 ispada, pri čemu se tekući element kombinuje sa elementima ispod sebe na listi – *i*-ti sa (*i*+1)-im, *i*-ti sa (*i*+2)-im itd. Za svaku kombinaciju proračunava se rizik simultanog ispada, na sličan način kao i za jednostruke ispade. S obzirom da već za sisteme srednje veličine broj mogućih dvostrukih ispada dostiže red veličine miliona, korisno je definisati minimalni rizik kao uslov za stavljanje *N*-2 ispada na listu. Time bi se izbeglo dobijanje liste prevelikih dimenzija, uštedelo na memorijskim resursima kao i vremenu samog autogenerisanja liste. Kada se naiđe na prvi par kome je rizik manji od te vrednosti, prelazi se na kombinacije sa sledećim elementom i tako dok se ne prođe kroz celu *N*-1 listu. Kada je iteriranje završeno, lista se sortira i u zavisnosti od dužine liste koju korisnik traži, prosleđuje se toliko najrizičnijih ispada. Ovim nisu proverene sve moguće kombinacije, ali se pokazuje da se time značajno štedi na vremenu. Jedino u slučaju da korisnik specificira dužinu veću nego što je dužina dobijene liste, nastavlja se dalje sa kombinovanjem elemenata i tada se obrađuju svi mogući *N*-2 ispadi.

Što se tiče procene rizika, vrši se slično kao za jednostruke ispade, s tim da se proverava i bliskost elemenata u ispadu. Bliskim će se smatrati: dve grane koje imaju bar jedan zajednički krajnji čvor, dva šanta priključena na isti čvor i grana i šant, ukoliko je šant priključen upravo na jedan od krajnjih čvorova grane. S obzirom da se koristi matematički

model sistema koji je u obliku grafa mreže, provera ovih uslova je trivijalna i ne usporava čitav proces autogenerisanja liste. S druge strane, ovim se ignorišu kombinacije bliskih elemenata na listi u odnosu na udaljene i tako operatoru daje informacija koji *N*-2 ispadi zaslužuju više pažnje i detaljniju analizu.

5. REZULTATI PRIMENE NA TEST SISTEMU

Predloženi algoritam za autogenerisanje rizične liste ispada primenjen je na prenosnom test-sistemu. Od elemenata koji su od interesa za analizu, ovaj sistem sadrži 93 sekcije (u tabelama SEC), 68 transformatora (TR), 89 interkonektivnih vodova (SL), 33 generatora (GEN), 6 kondenzatorskih baterija (CAP) i 2 statička VAR sistema (SVC). Dakle, postoji 391 različit *N*-1 ispad i 76245 *N*-2 ispada. Ovo je isuviše veliki broj za detaljnu obradu, a i operatorima je potrebna mnogo kraća lista, kako bi mogli lakše da je analiziraju. Prema tome, jasno je da je neophodno postojanje nekih od metoda pripreme za ovu analizu - metode skeniranja, autogenerisanje liste i sl. Dužina liste rizičnih elemenata je predviđena da bude opcija koju korisnik unosi, u zavisnosti od nivoa detaljnosti analize koju treba da vrši. Ovde će biti izdvojeno 25 najrizičnijih ispada:

Tabela 1: Autogenerisana lista 25 rizičnih ispada

Ispad	Vero- vatnoća	Pove- zanost	Optere- ćenje	Novih Ostrva	Snaga generatora u novim ostrvima (u MW)	Rizik
N-1_SEC_108	0.005		57.59	1	216.80	124.85
N-1_TR_250	0.001		72.85	1	277.24	18.308
N-2_SEC_192 & SEC_196	0.000025	1	139.62	1	216.80	15.135
N-2_SEC_108 & SEC_229	0.000025	1	68.25	2	216.80	11.097
N-2_SEC_108 & SEC_107	0.000025	1	66.54	2	216.80	10.820
N-2_SEC_108 & SEC_690	0.000025	1	65.98	2	216.80	10.728
N-2_SEC_108 & SEC_210	0.000025	1	62.43	2	216.80	10.151
N-2_SEC_108 & SEC_211	0.000025	1	61.90	2	216.80	10.065
N-2_SEC_108 & SEC_694	0.000025	1	91.09	1	216.80	9.874
N-2_SEC_108 & SEC_698	0.000025	1	91.03	1	216.80	9.867
N-2_SEC_108 & SEC_704	0.000025	1	59.86	2	216.80	9.733
N-1_SEC_129	0.005		13.20	1	72.85	9.614
N-2_SEC_193 & SEC_195	0.000025	1	130.57	1	72.85	4.756
N-2_SEC_108 & SL_478	0.000010	1	107.35	1	216.80	4.564
N-2_SEC_129 & SEC_139	0.000025	1	104.80	1	72.85	3.817
N-2_SEC_108 & TR_250	0.000005	1	90.60	2	277.24	3.786
N-2_SEC_129 & SEC_140	0.000025	1	102.45	1	72.85	3.732
N-2_SEC_108 & SL_477	0.000010	1	81.80	1	216.80	3.546
N-2_SEC_108 & SL_469	0.000010	1	78.50	1	216.80	3.403
N-2_SEC_108 & SL_470	0.000010	1	65.88	1	216.80	2.856
N-2_SEC_108 & SEC_194	0.000025		159.10	2	216.80	2.586
N-2_TR_250 & SEC_694	0.000005	1	66.52	1	277.24	1.844
N-2_TR_250 & SEC_698	0.000005	1	66.46	1	277.24	1.842
N-2_TR_250 & SEC_229	0.000005	1	43.68	2	277.24	1.816
N-2_SEC_108 & SEC_191	0.000025		165.09	1	216.80	1.789

Verovatnoće ispada povezanih elemenata su pri integrisanju u kriterijum rizika pomnožene sa 10. Ovo se vrši s ciljem da se simultanim ispadima takvih parova elemenata pridaje veći prioritet na listi i operateru skrene pažnja na njih, jer predstavljaju "kredibilnije" N-2 ispace od parova udaljenih elemenata. Prema rezultatima prikazanim u tabeli 1 može se zaključiti da ovo pojačavanje u dovoljnoj meri podiže ovakve ispace na listi (treća kolona).

S obzirom da je verovatnoća simultanih ispada udaljenih elemenata veoma mala, takvi parovi koji su se našli na ovoj listi su uglavnom sačinjeni od visoko opterećenih elemenata, ili elemenata čiji pojedinačni ispadi dovode do nastanka novih ostrva u sistemu. Stoga ovakvi ispadi na listi više ukazuju operatoru koji su to elementi pojedinačno rizični, nego što ukazuju na rizik od simultanog ispada.

Dakle, iako je N-1 ispada naizgled malo na ovoj listi, N-2 ispadi sadrže još mnogo takvih i praktično predstavljaju samo uređene parove rizičnih N-1 ispada. S druge strane, "pravi", kredibilni N-2 ispadi su posebno naglašeni da su to parovi povezanih elemenata i da ih je potrebno posmatrati kao celinu. Neretko bar jedan od elemenata koji ih sačinjava sam po sebi nije rizičan, a simultanim ispadom mogu imati negativne posledice po rad sistema.

Dalje, u kriterijum rizika uzeto je u obzir i opterećenje elementa, kao i eventualni nastanak novih ostrva u sistemu. Rezultati su pokazali da su uz pomoć ovog kriterijuma uspešno identifikovani rizični elementi u delovima mreže gde ona nije upetljena.

6. ZAKLJUČAK

Jedan od glavnih problema analize sigurnosti je složenost i sporo izvršavanje, naročito u slučajevima velikih mreža sa mnogo elemenata. Za prevazilaženje ovoga, razvijene su razne metode skeniranja, u cilju da se od liste svih elemenata sistema dobije mnogo kraća lista potencijalno kritičnih ispada. S obzirom da je osim jednostrukih potrebno obraditi i neke od verovatnijih višestrukih ispada, i sam proces skeniranja može biti spor, naročito ako su u pitanju velike mreže sa mnogo elemenata. Zbog toga je i za skeniranje potrebno razviti neki oblik pripreme, kako bi se smanjila ulazna lista.

U ovom radu je predstavljena ideja autogenerisanja takve liste, koja treba da sadrži jednostruke i dvostruke ispace koje je potrebno analizirati. Ono po čemu se ovaj metod razlikuje od uobičajenih metoda pripreme za analizu ispada u EES, predstavlja činjenica da se uopšte ne proverava kritičnost ispada, već se rangiraju na osnovu formiranog kriterijuma rizika. Ovo čini proces mnogo bržim i jednostavnijim, što su upravo karakteristike koje su najviše i poželjne. Potom se pri skeniranju i kompletnoj analizi proverava kritičnost, sa mnogo kraćom ulaznom listom i time mnogo brže.

Uključivanjem verovatnoće ispada, dat je prioritet elementima sistema sa višim prosečnim stopama ispada, pa se detaljna i vremenski iscrpna analiza vrši samo nad ispadima koji zaista mogu da se dese. Zbog lokalne prirode spoljašnjih uzroka ispada, uvažena je mnogo veća verodostojnost simultanih ispada bliskih elemenata u odnosu na parove udaljenih elemenata. Rezultujuća lista

ovim postaje mnogo realističnija i pri vrhu sadrži uglavnom takve "kredibilne" N-2 ispace.

Ovaj metod autogenerisanja liste je nova ideja, osmišljena kao sam početak celog procesa – da se za sve elemente prvo proceni samo rizik koji bi njihovi ispadi nosili, bez provere njihove kritičnosti, što ovaj metod čini mnogo bržim. Potom se kriterijum kritičnosti može integrisati u kriterijum rizika, što moderan, probabilistički pristup analizi sigurnosti i nalaže. Ovaj rad se po tome i razlikuje od drugih radova baziranih na kriterijumu rizika, što predlaže razdvajanje procene njegovih činilaca u različite metode, čime se lista ispada u više koraka skraćuje i celokupan proces ubrzava.

Prema tome, moguća poboljšanja procene rizika za potrebe autogenerisanja liste treba ograničiti samo na modelovanje verovatnoće i kriterijum bitnosti elemenata. Ova dva činioca nisu egzaktna i uvek postoji prostor za diskusiju o mogućnostima za njihovu kvalitetniju procenu. Stoga, ovaj rad, kao i mnoge druge radove iz ove oblasti zasnovane na probabilističkom pristupu, nikako ne treba posmatrati kao zatvaranje ove teme, već naprotiv, kao otvaranje ili nastavak diskusije o optimizaciji rešavanja problema analize sigurnosti EES-a.

7. LITERATURA

- [1] P.A. Kaplunovich: *Efficient algorithm for online N-2 power grid contingency selection*, Massachusetts Institute of Technology, February 2014.
- [2] C.M. Davis, T.J. Overbye: *Multiple element contingency screening*, *IEEE Trans. on Power Systems*, 3, (26), pp. 1294-1301, 2011.
- [3] F. Albuyeh, A. Bose, B. Heath: *Reactive Power Considerations in Automatic Contingency Selection*, *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-101, no. 1, pp 107-112, January 1962.
- [4] S. Vemuri, R.E. Usher: *On-line automatic contingency selection algorithms*, *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-102, no. 2, pp. 346-354, February 1983.
- [5] F. Galiana: *Bound estimates of the severity of line outages in power system contingency analysis and ranking*, *IEEE Trans. on Power Systems*, vol. PAS-103, no. 9, pp. 2612-2624, 1984.
- [6] H. Wan: *Risk-based security assessment for operating electric power systems*, Iowa State University, Retrospective Theses and Dissertations, Paper 12621, 1999.
- [7] http://www.nerc.com/files/concepts_v1.0.2.pdf, *Reliability Concepts: Credible Contingencies*, North American Electric Reliability Corporation, December 2007.

Kratka biografija:



Branko Đukić rođen je u Banja Luci 1992. godine. Fakultet tehničkih nauka u Novom Sadu upisao je školske 20011/2012. godine, smer energetika, elektronika i telekomunikacije. Na osnovnim studijama diplomirao je 2015. godine, smer elektroenergetski sistemi. Master rad iz oblasti Elektrotehnike i računarstva odbranio je 2016. godine.

REALIZACIJA PROGRAMSKE PODRŠKE ZA GENERISANJE PROJEKTNE STRUKTURE I PROGRAMSKOG KODA ZA MODUL ESP8266**REALIZATION OF SOFTWARE TOOL FOR GENERATING PROJECT STRUCTURE AND SOURCE CODE FOR MODULE ESP8266**

Vladimir Rončević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – u radu je opisana implementacija programske podrške za generisanje i razvoj projekata, konfiguracionih datoteka i programskog C koda za modul ESP8266 (proizvođač Espressif). Programska podrška je realizovana u programskom jeziku Perl u Linux radnom okruženju.

Abstract – This paper describes the implementation of a software tool for generating and development projects, configuration files and embedded C codes for ESP8266 module (manufacturer Espressif). A software tool was implemented with the programming language Perl in the Linux operating environment.

Ključne reči: Generisanje C koda, ESP8266 modul, bežična mreža.

1. UVOD

Proces koji treba da se realizuje pre početka razvoja projekta za ugradne sisteme (eng. Embedded Systems) je stvaranje i podešavanje razvojnog okruženja kao i razvoj programskog koda koji se koristi u fazi implementacije rešenja. Podešavanje razvojnog okruženja je težak i dug pripremni proces kada proizvođač hardverskog modula nema podršku u vidu integrisanog razvojnog okruženja koji objedinjuje sve u sklopu programske podrške (eng. Integrated Development Environment - IDE). U tim slučajevima potrebno je napraviti programsku podršku koja će automatizovati i ubrzati proces generisanja projektne strukture i programskog koda. Projektna struktura je sačinjena od direktorijuma, konfiguracionih datoteka, datoteka sa pravilima za prevođenje programskog koda u mašinske instrukcije (eng. Makefile) i datoteke sa programskim kodovima. Realizovana programska podrška iz ovog rada vrši generisanje projektne strukture i programskog koda striktno za modul sa oznakom ESP8266. Modul se u industriji koristi za bežično povezivanje (eng. Wireless connection - WiFi) više uređaja ili elektronskih sklopova u jednu lokalnu mrežu (eng. Local Area Network - LAN). Razvoj programskog koda za modul se generiše u datotekama koje su formatirane po pravila za C programski jezik, koji zahteva specijalno razvojno okruženje i skup programskih alata (eng. Xtensa Toolchain Set). Centralna procesorska jedinica modula se bazira na 32-bitnom procesoru sa redukovanim skupom mašinskih instrukcija (eng. Reduced Instruction Set Computing - RISC).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Ivan Mezei.

Oznaka procesora je Xtensa LX106 (proizvođač Tensilica, u sklopu kompanije Cadence Intenacionalne).

2. PRINCIP RADA I ANALIZA STRUKTURE PROGRAMSKE PODRŠKE

Osnovni skup programskih alata za Xtensa procesore se lako može instalirati pod Linux okruženjem, ali samo upravljanje skupom alata može biti kompleksno. Posle uspešne instalacije Xtensa skupa programskih alata, potrebno je dodati lokaciju instalacionog direktorijuma u sistemsku promenljivu pod imenom PATH za korisnika koji treba da koristi programsku podršku. Za korišćenje programske podrške u Linux radnom okruženju potrebno je da bude instaliran Perl interpreterski prevodilac sa standardnim paketima (eng. Perl Package Modules) kao i dodatnim paketima, programski paket za rad sa XML datotekama (XML::Simple) i programski paket za tabelarni prikaz podataka (Text::Table) [3].

Za pokretanje programske podrške potrebna je konzola sa komandnom linijom. Iz komandne linije pokreće se programska podrška sa dva argumenta koji predstavljaju ime projekta i putanja do direktorijuma u kome treba da se generiše projektna struktura. Programska podrška vrši obradu ulaznih podataka i nastavlja sa kolektovanjem podataka za konfiguraciju projekta. U narednim koracima korisnik unosi parametre za definisanje pravila, koja se primenjuju pri generisanju finalnog binarnog koda (eng. Binary Fromat, .BIN) koji se prebacuje u memoriju modula. Parametri modula koji se prikupljaju: boot_version (selekcija boot verzije modula), boot_mode (selekcija boot moda), spi_speed (selekcija brzine rada), spi_mode (selekcija modula) i spi_size_map (selekcija veličine memorije). Nakon podešavanja parametara generiše se projektna struktura sledećeg izgleda:

projekat/

- bin/
- include/
- ld/
- lib/
- Makefile
- src/
- build.sh
- build_env.sh
- Makefile
- tools/

Kolektovani parametri za razvoj modula se upisuju u datoteku ~/projekat/src/build_env.sh. Parametrizacija se izvršava sledećom komandom iz konzole:

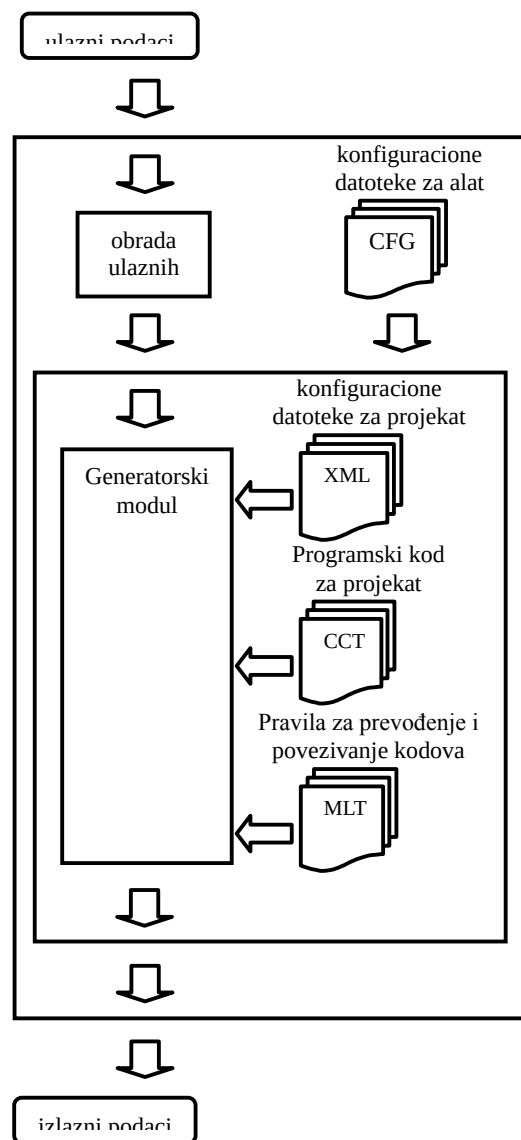
```
$ source build_env.sh
```

Direktorijum ~/projekat/bin/ je rezervisan za skladištenje binarnih datoteka koje će se prebaciti u memoriju modula. Direktorijum ~/projekat/include/ je rezervisan za datoteke sa zaglavlja (eng. Header Source files). Direktorijum ~/projekat/ld/ sadrži datoteke sa instrukcijama za program poveziivač (eng. Linker) koji je zadužen za povezivanje prevedenih programskih datoteka u jedinstvenu izvršnu binarnu datoteku [6]. Direktorijum ~/projekat/lib/ sadrži prevedene programske datoteke od strane proizvođača koje se koriste u fazi razvoja projekta. Datoteka ~/projekat/Makefile je datoteka sa instrukcijama za prevođenje programskog koda u finalni binarni kod. Direktorijum ~/projekat/src/ sadrži datoteke sa programskim kodom za bežični modul. Datoteka ~/projekat/build.sh je datoteka sa parametrima za pokretanje procesa prevođenja programskog koda u finalni binarni kod. Generisanje finalne binarne datoteke se izvršava sledećom komandom iz konzole:

```
$ ./build.sh
```

Datoteka ~/projekat/src/Makefile sadrži instrukcije za prevođenje programskog koda iz direktorijuma src/ u prevedenu statičnu arhivu datoteka (eng. Static Archive Library) [7]. Statična prevedena arhiva iz ~/projekat/src/ direktorijuma je samo jedan deo prevedenog koda koji treba da se poveže sa prevedenim kodom iz direktorijuma ~/projekat/lib/ od strane programa za povezivanje. Direktorijum tools/ sadrži dodatna uputstva za prevođenje i povezivanje programskog koda u finalnu binarnu reprezentaciju (pravila su napisana od strane proizvođača Espressif) [4].

Programska podrška je sačinjena od programskog koda koji je oformljen korišćenjem Perl programskog jezika, verzije 5.18.2 [9]. Programska struktura je pravljena modularno u proceduralnom obliku, zarad lakše, ubrzane buduće modifikacije i proširenja sa minimalnom izmenom koda. Struktura programske podrške (slika 1. *Struktura programske podrške*) se sastoji od modula za obradu ulaznih podataka i kolektovanih parametara, konfiguracionih datoteka za programsku podršku (CFG datoteke), generatorskog modula, konfiguracionih datoteka za projekat (XML datoteke), datoteka sa šemama za programski kod (CCT datoteke), datoteka sa šemama za prevođenje programskog koda u mašinske instrukcije (MLT datoteke) na osnovu kojih se formiraju izlazni podaci, odnosno datoteke sa programskim kodom i pravilima za prevođenje. Kolektovani parametri sa ulaza se selektivno odabiraju, svi parametri se čuvaju u konfiguracionim datotekama alata. Najodgovorniji modul je generatorski modul koji je zadužen za generisanje projektne strukture i svih pratećih datoteka. Generatorski modul je zadužen za formiranje finalne konstrukcije programskog koda. Programske strukture (segmenti programskih kodova) se formiraju čitanjem šema iz CCT datoteka (eng. C Code Template) i kombinacijom sa parametrima koji se čitaju iz XML datoteka [2]. Strukture za prevođenje (instrukcije za prevođenje) se formiraju na sličan način kao i programski kodovi, sa razlikama da se koriste druge šeme koje se čuvaju i čitaju iz MLT datoteka (eng. Makefile and Linker Scripts). Dodatni parametri se čuvaju u XML datotekama. Na sledećoj slici predstavljena je struktura programske podrške:



slika 1. *Struktura programske podrške*

3. GENERISANJE PROGRAMSKOG KODA I INSTRUKCIJA ZA PREVOĐENJE KODA

XML datoteke su formirane po W3C's XML 1.0 specifikaciji u UTF-8 formatu [8]. XML datoteke sadrže tipove i identifikatore promenljiva, imena i tipove funkcija, imena i parametre makro funkcija i niz drugih parametara koji se koriste za formiranje programskih struktura. CCT datoteke se baziraju na standardnoj proširivoj šemi u Perl programskom jeziku (eng. Perl Template Toolkit), koja je prilagođena za formiranje C programskog koda [1]. MLT datoteke se baziraju na istoj ideji, konstrukcije MLT datoteka su prilagođene za Makefile pravila.

Na sledećem primeru ilustrovan je princip generisanja jedne strukture koje se koristi u okviru projekta za bežičnu mrežu, strukturna promenljiva tipa sa imenom mesh_http_url_table. Struktura se sastoji od tri člana. Formiranje strukturne promenljive, šema za struktuiran tip podatka:


```

struct [% name %] {
    [% FOREACH structure_member = vars -%]
    [% structure_member.type %]
    [% structure_member.identifier %];
    [% END %]
};

```

Struktura ima sopstveni identifikator koji označava ime novog tipa podatka, u ovom slučaju `mesh_http_url_table`. Svaki član strukture je promenljiva koja ima tip i identifikator. Članovi strukture su organizovani kao niz podataka koji se čuvaju u XML datoteci za strukturu. Formiranje parametara za članove strukture, svaki član ima tip i identifikator koji se organizuju u sledećem obliku:

```

<?xml version="1.0" encoding="UTF-8" ?>
<genesp>
    <item
        id="Variable"
        type="Struct"
        name="mesh_http_url_table"
        init="none"
        file="include/mesh_http.h">
        <struct_member>
            <identifier>url</identifier>
            <type>const char *</type>
        </struct_member>
        <struct_member>
            <identifier>url_handler</identifier>
            <type>mesh_http_url_handler</type>
        </struct_member>
        <struct_member>
            <identifier>url_arg</identifier>
            <type>const void *</type>
        </struct_member>
    </item>
</genesp>

```

XML datoteka ima dodatne attribute koji dodatno opisuju strukturu za generisanje identifikacioni opis (eng. id), tip opisa (eng. type), ime opisa (eng. name), inicijalizacija opisa (eng. init) i ime datoteke u kojoj se opis formira (eng. file). Programska podrška posle učitavanja i obrađivanja konfiguracionih datoteka generiše strukturu sledećeg oblika:

```

struct mesh_http_url_table {
    const char *url;
    mesh_http_url_handler url_handler;
    const void *url_arg;
};

```

Korisničko lice za izmenu sadržaja `mesh_http_url_table` strukture treba da menja samo sadržaj XML datoteke, dodavanjem ili uklanjanjem parametara u XML datoteci. Izmena CCT šablona se sprovodi samo u slučaju potrebe dodavanja ključnih reči u definiciju strukture (kao što su `typedef`, `static` itd.).

Princip generisanja programskog koda funkcije se bazira na sličnom principu, konstruktna šema čuva se u CCT datoteci, parametri kao što su ime funkcije, tip funkcije i argumenti funkcije čuvaju se u XML datoteci, segment

koda koji funkcija treba da izvrši formira se od niza drugih parametara koji se čuvaju u drugim XML datotekama, koji se čitaju i konstruišu pre segmenta za funkciju. Generisanje makro definicija, makro funkcija, definicija novih tipova, enumeracija i svih ostalih programskih struktura realizuje se primenom istog principa sa CCT i XML datotekama.

Programski kod generisan pomoću programske podrške iz ovog rada je kodiran po UTF-8 formatu u latiničnoj formi. Pored programskog koda generišu se i prateći komentari vezani za određeni deo koda kao pojašnjenje funkcionalnosti. Komentari za programski kod se čuvaju u posebnim XML datotekama, odvojeno od programskog koda. Komentar određenog segmenta programskog koda se formatira u bloku za komentar programskog jezika C (eng. Block Comment) [5].

Programska podrška u okviru projektne strukture pored programskog koda generiše instrukcije i uputstva za prevođenje programskog koda u mašinske instrukcije. Makefile datoteke sa pravilima su zadužene za prevođenje svih datoteka sa programskim kodovima u nove datoteke sa segmentiranim mašinskim kodovima, odnosno objektna datoteke (eng. Object Files). Objektna datoteke se povezuju sa prevedenim programskim kodovima od strane proizvođača u narednim koracima pri formiranju finalne prezentacije izvršnog mašinskog koda. Pored pravila za prevođenje programskog koda u objektna datoteke, čuvaju se pravila za povezivanje objektnih datoteka. Makefile datoteke se čuvaju u UTF-8 formatu i latiničnoj formi.

Pravila, raspodela memorije, memorisjke adrese modula, startne tačke izvršavanja programskog koda (imena funkcija) za program povezač se čuvaju u specijalnim datotekama (LD datoteke, koje se nazivaju linker skripte). Proces formiranja LD datoteke se vrši pomoću parametara iz XML datoteke i konstruktna šeme koja se čuva u MLT datoteci (eng. Makefile, Linker Script Templates). Proces generisanja LD datoteka se sprovodi na isti način kao i proces generisanja programskog koda uz primenu drugačijih konstruktnih šema. MLT datoteka sadrži šemu za konstrukciju direktiva, uputstva za prevođenje i povezivanje objektnih datoteka. MLT datoteke se čuvaju u UTF-8 formatu i latiničnoj formi. Komentari za uputstva prevođenja su izostavljena.

Programska podrška tokom generisanja projektne strukture i programskog koda sa instrukcijama za prevođenje ispisuje tekstualne poruke o uspešno odrađenim fazama. Po završetku svih faza generisanja korisničko lice će biti upućeno o narednim koracima za razvojnu fazu projekta (kako se koristi, kako se pokreće proces prevođenja programskog koda u finalnu datoteku sa mašinskim kodom, kako se može kopirati u memoriju modula).

4. TESTIRANJE

Programska podrška je testirana u Linux okruženju openSUSE Leap 42.1 sa instaliranim sistemskim paketima, Perl interpreterskim prevodiocem, Perl standardnim programskim paketima, Perl programskim paketima za rad sa XML, CCT i MLT datotekama kao i osnovnim skupom alata za razvoj programskog C koda za Xtensa procesore.

Od strane programske podrške je generisana predviđena projektna struktura sa direktorijumima i datotekama. Generisana je struktura datoteka sa formatiranim programskim kodom po pravilima C programskog jezika. Generisana je struktura datoteka za prevođenje programskog koda u finalnu datoteku sa mašinskim instrukcijama. Programski kod generisan od strane programske podrške je implementacija za modul koji radi u bežičnoj mreži. Svaki modul periodično šalje paket informacija posle povezivanja u bežičnoj mreži.

Bežična mreža može biti sačinjena od više modula (eng. Mesh Wi-Fi Network). Svaki modul koji je u mreži sadrži listu adresa svih modula u mreži. Po prijemu paketa modul parsira informacije iz paketa i prosleđuje paket na destinaciju, odnosno adresu koja je upisana u paketu. Programski kod je organizovan od programskih datoteka modula (eng. C Source Code Files) i programskih datoteka zaglavlja (eng. Header Source Code Files).

Programski kod je preveden u finalnu datoteku, formatiranu u binarnom obliku za ESP8266 modul. Binarna datoteka je prebačena u memoriju više modula. Moduli su međusobno komunicirali, razmenjivali pakete.

5. ZAKLJUČAK

Programska podrška je tokom test faze uspešno generisala projektnu strukturu na osnovu dva ulazna parametra i selekcijom dodatnih parametara. Dodatni parametri za projekat se učitavaju i odabiraju selektivno na upit iz komandne linije u konzoli. Dodatni parametri se mogu menjati ili dodavati kroz konfiguracione XML datoteke. Kao modifikacije za neki budući rad može se razmišljati o drugačijem pristupu oko definisanja dodatnih parametara. Korisničko lice bi moglo kroz manji broj upitnih faza da selektuje skup dodatnih parametara, parametri bi mogli da se definišu kao klasa ili grupa parametara. Kao još jedno potencijalno rešenje koje bi moglo da se koristi, odnosno implementira u budućem radu je i selekcija između više verzija projekata sa gotovim strukturama programskog koda. Trenutno programska podrška ima samo jednu projektnu strukturu koja se generiše. Može se razmišljati o obezbeđivanju više implementacija za modul (modul koji može da funkcioniše kao jedinica koja komunicira samo sa određenim modulima u mreži ili modul koji će biti glavna jedinica, odnosno jedinica koja je zadužena za analizu i statistiku mreže sa koje se može vršiti nadgledanje celokupne mreže).

6. LITERATURA

- [1] Andy Wardley, “*Template-Front-end module to the Template-Toolkit*”, <http://search.cpan.org/~abw/Template-Toolkit-2.26/lib/Template.pm>, posećeno avgusta 2016. godine
- [2] Grant McLean, “*XML::Simple-An API for simple XML files*”, <http://search.cpan.org/~grantm/XML-Simple-2.22/lib/XML/Simple.pm>, posećeno avgusta 2016. godine
- [3] Anno Siegel, “*Text::Table-Organize Data in Tables*”, <http://search.cpan.org/~shlomif/Text-Table-1.131/lib/Text/Table.pm>, posećeno avgusta 2016. godine
- [4] Espressif Inc., “*Getting started guideline ESP8266*”, <https://espressif.com/en/support/explore/get-started/esp8266/getting-started-guide>, posećeno avgusta 2016. godine
- [5] GNU Project, “*The GNU C Reference Manual*”, <https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html>, posećeno avgusta 2016. godine
- [6] GNU Project, “*The GNU linker*”, ftp://ftp.gnu.org/old-gnu/Manuals/ld-2.9.1/html_mono/ld.html, posećeno avgusta 2016. godine
- [7] GNU Project, “*Makefile (make build automation)*”, <https://www.gnu.org/software/make/manual/make.html#toc-Overview-of-make>, posećeno avgusta 2016. godine
- [8] W3C, “*Extensible Markup Language (XML) 1.0 (Fifth Edition)*”, <https://www.w3.org/TR/REC-xml/>, posećeno avgusta 2016. godine
- [9] Perl 5 Porters, “*Perl 5 version 24.0 documentation*”, <http://perldoc.perl.org/>, posećeno avgusta 2016. godine

Kratka biografija:



Vladimir Rončević rođen je u Beogradu 1987. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Mikroprocesorski sistemi i algoritmi odbranio je 2016.god.

UPRAVLJANJE RAZMENAMA SNAGE U POVEZANIM ELEKTROENERGETSKIM SISTEMA SA UVAŽAVANJEM DISTRIBUIRANOG BALANSNOG ČVORA

AREA INTERCHANGE CONTROL IN A POWER FLOW PROBLEM WITH RESPECT TO DISTRIBUTED SLACK BUS

Vesna Čičković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad opisuje jednu metodologiju koja procenjuje upravljanje razmenama snage (RS) između regulacionih oblasti (RO) povezanih elektroenergetskih sistema (EES) u proračunu tokova snaga sa uvažavanjem distribuiranog balansnog čvora (DBČ), koristeći Newton-Raphson-ov metod. Pokazano je da se jednačine vezane za upravljanje RS mogu uključiti u sistem jednačina tokova snaga, imajući u vidu raspodelu debalansa aktivne snage u sistemu na više regulacionih generatora (RG) koji igraju ulogu DBČ-a. Predložena metoda je testirana na modifikovanoj IEEE test mreži od 118 čvorova.

Abstract – This paper describes one methodology to evaluate the area interchange control (AIC) in a power flow problem with respect to distributed slack bus (DSB) using the Newton-Raphson method. It is shown that the equations of the AIC can be incorporated into the system of equations of the power flow problem considering also a distribution of real power imbalance in the system to many regulating units which represents DSB. The proposed method has been tested using modified IEEE 118-bus system.

Cljučne reči: upravljanje razmenama snage, distribuirani balansni čvor, tokovi snaga

1. UVOD

Povezani EES su dosta pouzdaniji, ekonomičniji i fleksibilniji od izolovanih sistema. Pogodnosti rada sistema u interkonekciji obuhvataju uštede u proizvodnji električne energije, kao i ostvarivanje veće operativne rezerve u slučaju kritičnih ispada pojedinih agregata ili naglih povećanja potrošnje [1-4].

Darrieus-ov princip neintervencije [3,5] jedan je od osnovnih principa pogona interkonekcije, koji kaže da se bilans aktivnih snaga u svakom podsistemu interkonekcije održava prvenstveno sopstvenim resursima, poštujući pritom unapred dogovorenu (specificiranu) razmenu aktivne snage sa ostalim podsistemima interkonekcije.

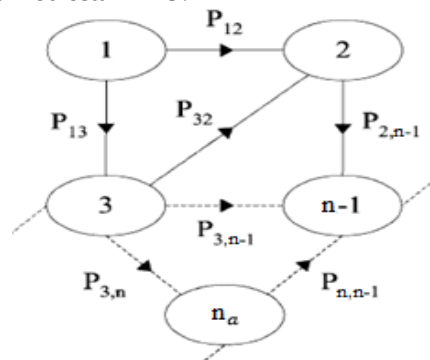
DBČ je pojam koji se vezuje za skup više regulacionih agregata sa mogućnošću promene proizvodnje aktivne snage u cilju pokrivanja debalansa iste. Motivi za uvođenje DBČ-a su ograničenost klasičnog balansnog čvora u pogledu proizvodnje reaktivne snage, zanema-

rivanje raspoložive rezerve aktivne snage ostalih agregata, kao i nerealna predstava EES-a nastala primenom standardnog balansnog čvora u proračunu tokova snaga.2.

2. TEORIJSKE OSNOVE

2.1. Inicijalna razmatranja o snagama razmena

Slika 1 ilustruje EES sa n oblasti međusobno povezanih različitim brojem interkonekcija (vodovi i/ili transformatori). Za svaku RO unapred je dogovorena RS sa svakom od ostalih RO.



Slika 1. Ilustracija snaga razmena između RO [5]

Time je određena i sumarna RS svakog podsistema n sa susedima: $PI_n, n = 1, \dots, n_a$. Snaga razmene je pozitivna ako RO odaje snagu susedima, a u suprotnom slučaju ona je negativna. Sumarna razmena n -te RO utvrđuje se kao algebarska suma njene razmene sa svakim drugim članom interkonekcije:

$$PI_n = \sum_{\substack{i \in \Omega_n \\ j \notin \Omega_n}} P_{ij}, \quad (1)$$

gde je Ω_n skup svih čvorova koji pripadaju n -toj RO, a P_{ij} tok aktivne snage u grani koja spaja čvorove i i j pri referentnom smeru od čvora i ka čvoru j . Za tako definisane sumarne snage očigledno važi relacija:

$$\sum_{n=1}^{n_a} PI_n = 0. \quad (2)$$

RS za sve oblasti osim n_a -te date su sledećim izrazima:

$$\begin{aligned} PI_1 &= P_{12} + P_{13} \\ PI_2 &= -P_{12} - P_{32} + \dots + P_{2,n-1} \\ PI_3 &= -P_{13} + P_{32} + \dots + P_{3,n-1} + P_{3,n} \\ &\vdots \\ PI_i &= \dots \\ &\vdots \\ PI_{n-1} &= -P_{2,n-1} - P_{3,n-1} - P_{n,n-1} + \dots \end{aligned} \quad (3)$$

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Andrija Sarić, red.prof.

Očigledno da je snaga n_a -te oblasti unapred određena izrazima (2) i (3). Ako su u pitanju dve RO, onda pozitivna RS jedne jednaka je negativnoj RS druge RO.

U dispečerskom centru svakog podsistema kontinualno se stiču informacije o sopstvenoj ukupnoj snazi razmene električne energije sa susedima. Saglasno sa principom neintervencije, te snage moraju da se održavaju sve dok se ponovnim dogovorom ne specificiraju druge [5].

2.2. Participacioni faktori i distribuirani balansni čvor

Klasični balansni čvor u proračunima tokova snaga u suštini obavlja dve funkcije: 1. predstavlja referentni čvor s obzirom na uglove fazora napona, 2. predstavlja u opštem slučaju beskonačni izvor ili ponor (re)aktivne snage i služi da pokrije ukupan debalans snaga u EES-u.

U onim sistemima u kojima je npr. potrebno zadržati podjednake inkrementalne troškove proizvodnje u nekim generatorskim čvorovima (uključujući tu i onaj u balansnom čvoru), ideja sa jedinstvenim balansnim čvorom očigledno ne ispunjava dati kriterijum. Drugim rečima, ukoliko generator priključen na balansni čvor jedini pokriva debalans snage, dobiće se neoptimalno stanje EES-a.

Umesto jedinstvenog generatora koji pokriva celokupan debalans snage, u realnim sistemima postoji određen broj generatorskih jedinica koje učestvuju u sekundarnoj regulaciji sa ciljem balansiranja proizvodnje sa gubicima, potrošnjom i u opštem slučaju sa uvozom/izvozom aktivne snage.

Krajnji cilj ovih agregata jeste da održe sistemsku učestanost na nominalnoj vrednosti (u Evropi 50 Hz) na račun promene odate aktivne snage. Količina te odate snage zavisi prvenstveno od veličine debalansa, odnosno od participacionih faktora k_i koji su pridruženi tim generatorima na osnovu kombinovanih ekonomskih kriterijuma i kriterijuma pouzdanosti.

U [6-9] participacioni faktori se uglavnom definišu na dva načina. Prvi je dat obrascem:

$$k_j = P_{gjM} / \sum_j P_{gjM} \quad (4)$$

gde je P_{gjM} maksimalna snaga j -tog agregata, $j \in \beta_s$, a β_s predstavlja skup svih čvorova na koje su priključeni RG. Pri tome važi relacija:

$$\sum_j k_j = 1. \quad (5)$$

Drugi način za definisanje participacionih faktora je dat relacijom:

$$k_j = \frac{P_{gj}^{loss}}{P_{loss}} \quad (6)$$

gde su P_{gj}^{loss} gubici „asocirani“ j -tom generatoru, a P_{loss} ukupni gubici snage u sistemu. Pod gubicima asociranim pojedinačnim generatorima se bukvalno misli na delove mreže i njihove gubitke za čije pokrivanje su zaduženi ti generatori.

Ovaj način određivanja participacionih faktora se više primenjuje i u distributivnim mrežama.

U ovom radu, u slučaju da neki od RG-a prilikom pokrivanja debalansa dostignu limit (gornji ili donji) proizvodnje aktivne snage definisan pogonskim ograničenjima, za preostale generatore se preračunavaju novi faktori učešća na osnovu njihove trenutne rezerve:

$$k_j = P_{gjrez} / \sum_j P_{gjrez} \quad (7)$$

gde je $P_{gjrez} = P_{gjmax} - P_{gj}$, odnosno $P_{gjrez} = P_{gj} - P_{gjmin}$. Odluka da li će se rezerva aktivne snage generatora P_{gjrez} posmatrati u odnosu na minimalnu ili maksimalnu vrednost proizvodnje generatora zavisi od znaka debalansa. Ukoliko je debalans veći od nule, jasno je da agregati treba da povećaju svoju baznu proizvodnju, pa se rezerva aktivne snage generatora računa kao $P_{gjrez} = P_{gjmax} - P_{gj}$. Slično je i sa negativnim debalansom. I u ovom slučaju neophodno je da važi relacija (5). Jasno je da sva prethodna razmatranja važe i za faktor učešća σ RG u lokalnim regulacionim čvorovima RO, koji određuje učešće tih RG u pokrivanju debalansa nastalog usled promene RS te RO.

Potrebno je istaći da se generatorima koji su dostigli limit dodeljuju participacioni faktori učešća jednaki nuli. Na taj način ovi generatori ne učestvuju više aktivno u ostatku proračuna (njihova proizvodnja je konstantna i jednaka maksimalnoj ili minimalnoj vrednosti).

3. FORMULACIJA PROBLEMA

Potrebno je u zadatoj mreži (sa poznatom topologijom, parametrima i potrošnjom) opteretiti izvore tako da budu zadovoljene dogovorene RS između RO od kojih se sastoji zadati interkonektivni EES. U ovom radu će se posmatrati opšti slučaj kada svaka RO sadrži proizvoljan broj lokalnih regulacionih čvorova (tj. RG) i svakoj od njih raspodeliti razliku između dogovorene i trenutne vrednosti RS na lokalne regulacione čvorove, prema unapred zadatim koeficijentima učešća σ .

Drugi deo problema vezan je za uključenje DBČ-a u proračun tokova snaga u prethodno zadatoj mreži. U ovom slučaju, zanemaruju se održavanje i kontrola RS između RO, tj. posmatra se sistem u celini. Potrebno je da se celokupni debalans aktivne snage (tj. gubici) podele na više generatora prema unapred zadatim koeficijentima učešća k_j .

Konačno, zahteva se kombinacija kontrole RS sa uvažavanjem DBČ-a. Pojedini generatori na ovaj način mogu imati i $\sigma > 0$ i $k_j > 0$.

4. REŠENJE PROBLEMA

4.1. Model tokova snaga sa upravljanjem RS-a

Kako bi se u jednoj RO mogla kontrolisati razmena sa ostatkom sistema, potrebno je u njoj definisati jedan ili više lokalnih regulacionih čvorova, tako da se promenama injektiranja aktivne snage u njima održava željena snaga razmene. U vezi sa tim, proizvodnja aktivne snage u takvom k -tom čvoru P_{kgen} proglašava se za nepoznatu veličinu, pa izraz za injektiranje aktivne (računate) snage u čvoru k glasi:

$$P_k^{cal} = P_k^{cal}(\theta, V) - \sigma_k^n \Delta PA_n, \quad (8)$$

pri čemu važi $P_{kgen} = P_{kgen0} + \sigma_k^n \Delta PA_n$. P_{kgen0} predstavlja konstantnu baznu (učestvuje pri pokrivanju potrošnje) snagu generatora u k -tom čvoru, σ_k^n označava faktor učešća k -tog generatora u pokrivanju debalansa n -te RO, ΔPA_n predstavlja ukupni priraštaj aktivne snage u svim lokalnim regulacionim čvorovima n -te RO, a $P_k^{cal}(\theta, V)$ standardni obrazac [9] za izračunavanje injektirane aktivne snage u čvoru k , koja je funkcija od modula (V) i uglova (θ) napona u čvoru k i u čvorovima incidentnim čvoru k . Deo snage injektiranja u čvoru k koji je nepromenljiv tokom proračuna je dat sa:

$$P_k^{spec} = P_{kgen0} - P_{kload}, \quad (9)$$

gde je P_{kload} eventualna potrošnja u čvoru k . Trenutni mismatch snage u iteraciji h dat je izrazom:

$$\Delta P_k^{(h)} = P_k^{spec(h)} - P_k^{cal(h)}. \quad (10)$$

Jednačina (10) posle linearizacije data je kao:

$$\Delta P_k = \sum_j A + \sum_i B + C, \quad (11)$$

gde je $A = \frac{\partial P_k^{cal}}{\partial \theta_j} \Delta \theta_j$, $B = \frac{\partial P_k^{cal}}{\partial V_i} \Delta V_i$, $C = \frac{\partial P_k^{cal}}{\partial \Delta PA_n} \Delta(\Delta PA_n)$, a j pripada skupu klasičnih čvorova PV, PQ i skupu lokalnih regulacionih čvorova, dok i pripada skupu PQ čvorova, dok je k -ti čvor deo n -te RO. $\Delta(\Delta PA_n)$ označava priraštaj (u tekućoj iteraciji) injektirane aktivne snage u svim lokalnim regulacionim čvorovima n -te RO.

Osetljivost oblika $\frac{\partial P_k^{cal}}{\partial \Delta PA_n}$ ima vrednost $-\sigma_k^n$, pa se zaključuje da što je veće σ_k^n , veći je i uticaj RG-a u pokrivanju debalansa nastalog usled promene snaga razmena njegove RO.

Da bi se kompletno uvažila kontrola RS, potrebno je uvažiti specificirane RS, tj. dodati nove jednačine u klasični model tokova snaga definisan u [9]. RS za n -tu RO je definisana kao suma tokova snaga po svim granama koje je povezuju sa ostatkom sistema [relacija (1)]. Trenutni mismatch RS u iteraciji h dat je izrazom:

$$\Delta PI_n^{(h)} = PI_n^{spec(h)} - PI_n^{(h)}. \quad (12)$$

Debalans za tu RO je posle linearizacije dat obrascem:

$$\Delta PI_n = \sum_j \frac{\partial PI_n}{\partial \theta_j} \Delta \theta_j + \sum_i \frac{\partial PI_n}{\partial V_i} \Delta V_i, \quad (13)$$

gde je $n = 1 \dots n_a - 1$, a numeracija je vezana za oznake sa Slike 1. Primetno je da je $n \neq n_a$, jer se specificirana RS ne zadaje za referentnu oblast n_a .

4.2 Model tokova snaga sa uvažavanjem DBČ-a

Slično kao u prethodnom paragrafu, da bi se pokrio debalans aktivne snage (sada jednak isključivo gubicima) neophodno je da postoje određeni RG sa faktorima učešća $k_j > 0$ koji ustvari čine DBČ. Izraz za aktivnu snagu RG priključenog u j -ti čvor koji je deo DBČ-a je:

$$P_{jgen} = P_{jgen0} + k_j \Delta G, \quad (14)$$

gde je ΔG regulaciona greška (debalans aktivne snage), a k_j faktor učešća j -tog RG u pokrivanju ΔG . Imaju se potpuno analogni izrazi sa obrascima (8)-(11), pri čemu je:

$$P_j^{cal} = P_j^{cal}(\theta, V) - k_j \Delta G, \quad (15)$$

a $C = \frac{\partial P_k^{cal}}{\partial \Delta G} \Delta(\Delta G)$. Umesto skupa lokalnih regulacionih čvorova sada se biraju RG na nivou celokupnog sistema. Referentni čvor u odnosu na uglove fazora napona može biti proizvoljni čvor iz sistema.

4.3 Model tokova snaga sa upravljanjem RS-a i uvažavanjem DBČ-a

Pokazuje se da je relativno jednostavno definisati model za proračun tokova snaga koji uključuje kombinaciju kontrole RS sa uvažavanjem DBČ, koristeći „spajanje“ jednačina iz prethodna dva paragrafa u jedan globalni model.

Promenljive stanja su sada i $\Delta(\Delta G)$ i niz od $n_a - 1$ vrednosti tipa $\Delta(\Delta PA_n)$. „Tačke spajanja“ dva modela predstavljaju RG sa $\sigma > 0$ i $k_j > 0$, pa je P_j^{cal} u takvim čvorovima dato sa:

$$P_j^{cal} = P_j^{cal}(\theta, V) - k_j \Delta G - \sigma_j^n \Delta PA_n. \quad (16)$$

Konačan matrični oblik kombinovanog modela za proračun tokova snaga označen je sa relacijom:

$$F(X^{(h-1)}) = J(X^{(h-1)}) \cdot \Delta X^{(h)}, \quad (17)$$

gde je:

$$J(X^{(h-1)}) = \begin{bmatrix} \frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial \Delta G} & \frac{\partial P}{\partial V} & \frac{\partial P}{\partial \Delta PA} \\ \frac{\partial Q}{\partial \theta} & 0 & \frac{\partial Q}{\partial V} & 0 \\ \frac{\partial PI}{\partial \theta} & 0 & \frac{\partial PI}{\partial V} & 0 \end{bmatrix}, \quad (18)$$

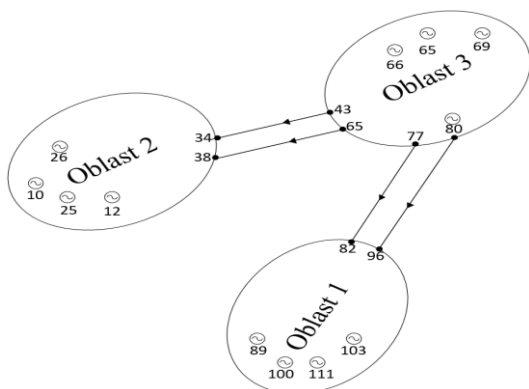
a $F(X^{(h-1)}) = [\Delta P \ \Delta Q \ \Delta PI]^T$ i $\Delta X^{(h)} = [\Delta \theta \ \Delta(\Delta G) \ \Delta V \ \Delta(\Delta PA)]^T$. $\Delta X^{(h)}$ označava vektor nepoznatih priraštaja promenljivih stanja u h -toj iteraciji. Treba primetiti da upravljanje RS-a, kao i uvažavanje DBČ-a nema uticaja na reaktivne snage u sistemu, pa su odgovarajuće osetljivosti u J matrici jednake nuli.

Takođe, PI predstavlja niz sačinjen od suma tokova snaga po granama izračunatih za pojedinačne RO, pa zavisi samo od θ i V , što je primetno iz 4. reda modifikovane Jakobijan matrice [relacija (18)].

5. PRIMENA

Proračuni tokova snaga sa kombinovanim modelom upravljanja RS-a i uvažavanjem DBČ-a ilustrovani su na modifikovanoj (sa samo 4 interkonekcije) IEEE test mreži od 118 čvorova. Uprošćen prikaz te mreže, gde kružići unutar oblasti predstavljaju RG dat je na Slici 2.

Potrebno je ostvariti uvoz aktivne snage iz Oblasti 3 u Oblast 1 od 150 MW, a u Oblast 2 od 250 MW. Referentni čvor s obzirom na uglove fazora napona je 69. Pritom, u Tabeli 1 su dati faktori učešća RG, σ i k , kao i bazne i maksimalne snage tih RG.



Slika 2. Prikaz modifikovane IEEE mreže od 118 čvorova sa indeksima RG-a i graničnih čvorova

Tabela 1. Ulazni podaci za RG

Čvor	σ	k	$P_{gbaz}[MW]$	$P_{gmax}[MW]$
89	0.4	0	607	650
100	0.2	0	252	300
103	0.2	0	40	60
111	0.2	0.2	36	80
10	0.15	0.2	450	650
12	0.3	0	85	150
25	0.15	0	220	310
26	0.4	0	314	400
65	0	0.1	391	450
66	0	0.1	392	450
69	0	0.2	381	550
80	0	0.2	471	600

Tabela 2. Rezultati proračuna za RG

Ind.čvora	$\sigma_{2.it}$	σ_{fin}	$P_{g2.it.}[MW]$	$P_{gfin.}[MW]$
89	0.4	0.4	540.8	577.3
100	0.2	0.2	218.9	237.1
103	0.2	0.2	6.89	25.1
111	0.2	0.2	14.01	35.3
10	0.689	0.689	436.29	470.6
12	0	0	150	150
25	0.311	0.311	259.9	222.9
26	0	0	400	400
65	0	0	396.56	398.1
66	0	0	397.56	399.1
69	0	0	392.12	395.2
80	0	0	488.12	491.2

Tabela 3. Tokovi snaga po interkonektivnim granama

Od čvora	Ka čvoru	$P_{grane}[MW]$	$P_{gub}[MW]$
Rockhill (34)	S.Kenton (43)	-19.46	0.23
EastLima (38)	Musknum (65)	-230.54	5.24
Logan (82)	Turner (77)	-72.23	1.61
Baileys (96)	CabinCrk (80)	-77.77	2.3

Proračun je konvergirao u 6 iteracija sa tačnošću od 10^{-6} . Rezultati su prikazani u Tabelama 2 i 3. RG iz RO 1 smanjili su svoju proizvodnju u odnosu na baznu vrednost, jer se povećao uvoz snage sa ~90 MW na 150 MW. Slično, RG iz RO 2 povećali su proizvodnju, jer se smanjio uvoz iz RO 3 sa početnih ~417 MW na 250 MW. Pritom, RG u čvorovima 12 i 26 su već u drugoj iteraciji dostigli maksimum snage proizvodnje, pa su njihovi faktori σ postali nula, a RG-ma u čvorovima 10 i 25 proračunati su novi σ koeficijenti naspram sopstvenih rezervi koji su održani do kraja proračuna (Tabela 2).

RG u balansnoj oblasti su pokrili debalans nastao usled promena RS i gubitaka aktivne snage u celokupnom sistemu. Njihovi limiti nisu dostignuti, pa nije bilo potrebe za preračunavanjem koeficijenta k .

6. ZAKLJUČAK

U ovom radu opisan je mogući način proračuna tokova snaga sa kontrolom RS i uvažavanjem DBČ-a. Klasični Newton-Raphson-ov postupak se modifikuje, kako bi se iterativno, na osnovu faktora učešća σ i k određivali doprinosi RG u pokrivanju debalansa aktivne snage, nastalog usled promene tokova snaga na interkonektivnim vodovima i nepoznatih gubitaka na nivou kompletnog EES-a.

Pokazano je da se pritom zadržava kvadratna konvergencija ka rešenju, što je velika prednost ovog postupka. U slučaju da pojedini generatori dostignu neko od pogonskih ograničenja, njihova proizvodnja se zadržava na zatečenoj maksimalnoj ili minimalnoj vrednosti, dok se automatski vrši preraspodela opterećenja na generatore sa dovoljnom količinom rezerve aktivne snage, prema proračunatim faktorima učešća σ odnosno k .

7. LITERATURA

- [1] www.entsoe.eu, 25th November, 2016.
- [2] www.ems.rs, 25th November, 2016.
- [3] M.S. Čalović, A.T. Sarić, P.Č. Stefanov: *Eksplotacija elektroenergetskih sistema u uslovima slobodnog tržišta*, Tehnički fakultet u Čačku, 2005.
- [4] D.P.Kothari, I.J.Nagrath: *Power System Engineering*, 2nd edition, McGraw-Hill, New Delhi, 2008.
- [5] V.Strežoski: *Analiza elektroenergetskih sistema II*, skripta, Fakultet Tehničkih Nauka, Novi Sad, 2010
- [6] M.J.D. Santos, J.L.R. Pereira, et al.: *A New Approach for Area Interchange Control Modeling*, IEEE Transactions on Power Systems, Vol. 19, No. 3, August 2004.
- [7] C. Keerati: Application of Distributed Slack Bus Power Flow to Competitive Environments, Power Engineering Conference, December 2007, pp 1-6.
- [8] M.Z. Kamh, R. Iravani: *A Sequence Frame-Based Distributed Slack Bus Model for Energy Managment of Active Distribution Networks*, IEEE Transactions on Smart Grid, Vol. 3, No. 2, June 2012
- [9] M.S. Čalović, A.T. Sarić: *Osnovi analize elektroenergetskih mreža i sistema*, Akademski misao i Tehnički fakultet u Čačku, Beograd, 2004.

Kratka biografija:



Vensa Čičković rođena je u Požarevcu 1992. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – smer Elektroenergetski sistemi odbranila je 2016. god.

SOFTVERSKO REŠENJE ZA POSLOVNO IZVEŠTAVANJE U APOTEKARSKOM SISTEMU**SOFTWARE SOLUTION FOR BUSINESS REPORTING IN THE PHARMACY SYSTEM**

Nikola Stamenković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu predstavljen je proces razvoja Data Warehouse sistema i naprednog sistema za poslovno izveštavanje, u okviru apotekarskog sistema. Opisan je način na kojim je isprojektovan i implementiran ETL proces prilikom izgradnje Data Warehouse baze, kao i na koji način je nad njom izmodelovan BI izveštajni sistem sa kojim će se izvršavati upiti.

Abstract – This paper introduces the process of developing a Data Warehouse system and advanced systems for business intelligence within pharmacy system. We describe the way in which ETL process is developed and implemented while developing a Data Warehouse, as well as the way of modeling BI repository on which queries will be executed.

Cljučne reči: Data Warehouse, ETL, Reporting

1. UVOD

Savremene kompanije koje nastoje da uspešno i tačno rešavaju poslovne zadatke koje im nameće tržište, moraju posedovati složene informacione sisteme, koji im umnogome pomažu u njihovom rešavanju. Transakcione baze podataka (*Online Transaction Processing*, OLTP) ovih sistema sadrže veliku količinu podataka, u kojima broj torki varira od par stotina hiljada, pa sve i do više desetina, pa i stotina miliona. Da bi se ovako obiman skup podataka mogao realno sagledati, i da bi se iz njega lakše izvlačile informacije, kompanije pribegavaju softverskim rešenjima koja spadaju u grupu alata poslovne inteligencije.

Termin poslovna inteligencija predstavlja mnogo širi pojam od samih alata, zato što obuhvata skup metodologija, procesa, arhitektura i tehnologija koje transformišu neobrađene podatke u razumljive i korisne informacije. Ovako dobijene informacije su ključni faktor u procesu donošenja poslovnih odluka, koje mogu biti krucijalne za uspešno poslovanje same kompanije.

Motivacija za nastanak ovog rada je nizak stepen upotrebljivosti i kvaliteta prikazanih informacija na izveštajima u okviru poslovnog izveštavanja u apotekarskom sistemu. S obzirom na to da je osnovna delatnost apotekarskog sistema nabavka i prodaja, potrebno je da poslovne odluke koje se donose u okviru ovog sektora budu donešene u što kraćem vremenskom periodu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji je mentor bio dr Ivan Luković, red. prof.

Kao rezultat projekta kreiranja naprednog izveštajnog sistema, omogućeno je krajnjim korisnicima napredno izveštavanje koje ima za cilj unapređivanje njihovog biznisa. Razvoj izveštajnog sistema obuhvata razvoj Data Warehouse baze podataka nad polaznom OLTP bazom kao i kreiranje BI izveštajnog sistema. Pored uvoda i zaključka, rad sadrži tri poglavlja. U drugom poglavlju, dat je prikaz izvorišne OLTP baze i odredišne Data Warehouse baze apotekarskog sistema, prikazane su njihove konceptualne šeme i opisan je način na koji je izmodelovan i implementiran proces izvlačenja, transformacije i punjenja podataka (*Extract, Transform, Load*, ETL). U trećem poglavlju opisan je razvoj BI izveštajnog sistema nad kreiranom Data Warehouse bazom podataka. U četvrtom poglavlju su opisane osnovne funkcionalnosti centralnih šifarnika koji imaju za cilj podršku celog BI apotekarskog sistema. Takođe, prezentovane su osnovne komponente njegove arhitekture.

2. RAZVOJ DATA WAREHOUSE BAZE PODATAKA ODABRANOG APOTEKARSKOG SISTEMA

U ovom poglavlju će biti prikazano rešenje isprojektovane i implementirane Data Warehouse baze podataka odabranog podsistema apotekarskog sistema u okviru jedne kompanije. Prelazak na Data Warehouse bazu podataka ima za cilj unapređenje performansi upita izveštajnih alata, koji će kasnije biti izvršavani nad ovom bazom. Ubrzanje upita bilo je postignuto zahvaljujući prelasku na model Data Warehouse baze podataka, što doprinosi bržem odzivu sistema na zahteve alata za upite. Još jedna od karakteristika ovakvog sistema su denormalizovane dimenzije, u okviru kojih se podaci svih nivoa hijerarhije nalaze unutar iste tabele, čime se izbegava spajanje tabela prilikom kreiranja upita. Upotreba posebnih vrsta indeksa karakterističnih za Data Warehouse sisteme, takođe znatno ubrzava pretragu podataka.

Za potrebe modelovanja šeme izvorišne OLTP baze i odredišne Data Warehouse baze upotrebljen je alat *Microsoft Management Studio 2012*. Za implementaciju celokupnog Data Warehouse sistema iskorišćen je *Microsoft*-ov sistem za upravljanje bazama podataka *SQL Server 2012*.

2.1. Specifikacija izvorne OLTP baze podataka

Kao izvor podataka, za potrebe razvoja Data Warehouse baze, iskorišćen je deo OLTP baze polaznog informacionog sistema. Podaci iz polazne baze podataka su preuzeti sa produkcionog servera kompanije u originalnom obliku, a kasnije su u okviru ETL procesa

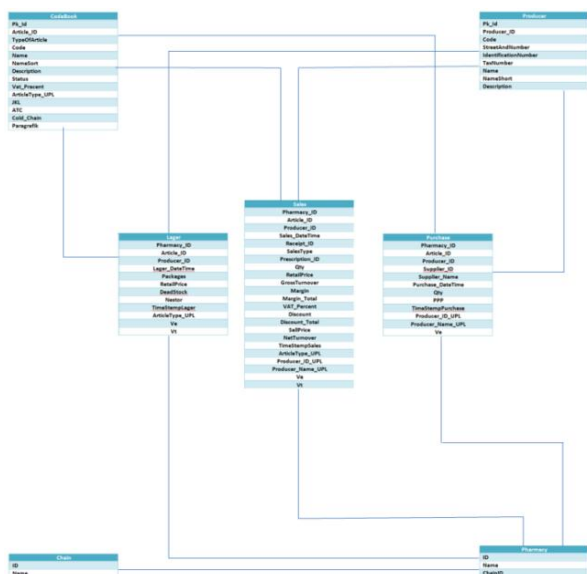
filtrirani i prilagođeni potrebama ciljnog *Data Warehouse* sistema.

2.2. Specifikacija ciljne *Data Warehouse* baze podataka

U ovom odeljku dat je detaljan opis *Data Warehouse* šeme baze podataka.

Prilikom projektovanja *Data Warehouse* sistema izmodelovane su tabele činjenica koje su denormalizovane i obezbeđuju čuvanje velikih količina podataka, a sve u cilju maksimalnog povećanja performansi sistema.

U nastavku će biti opisani elementi *Data Warehouse* šeme i njeni grafički prikaz (slika 1). Tabela dimenzija *CodeBook* sadrži podatke o šifarniku artikla i povezan je sa tabelom činjenica preko dimenzionog ključa *Article_Id*. Svi neophodni atributi šifarnika artikla su opisani u ovoj tabeli dimenzija. Tabela dimenzija *Producer* je dimenzija šifarnika proizvođača i povezan je sa tabelom činjenica preko dimenzionog ključa *Producer_ID*. Tabela dimenzija *Chain* je dimenzija šifarnika lanaca apoteke, ona predstavlja sve pojedinačne apoteke u sklopu lanca koje koriste dati sistem.



Slika 1. Grafički prikaz *Data Warehouse*

Tabela dimenzija *Pharmacy* je dimenzija šifarnika apoteka koja detaljno opisuje koje apoteke postoje u sistemu. Data tabela je povezana sa dimenzijom *Chain*. Tabela *Lager* je jedna od tri tabele činjenice *Data Warehouse* baze podataka. Ona sadrži podatke o stanju artikla u apoteci na nivou stavke. Data tabela Činjenica povezana je sa sledećim dimenzijama: *CodeBook*, *Producer*, *Pharmacy*. Tabela *Lager* sadrži podatke o šifri organizacione jedinice, šifri artikla, šifri proizvođača, datum lagera, količina, maloprodajna vrednos, mrtvi lager, nagomilani artikal, datum inserta podataka, artikal tip.

Tabela *Prodaja* je jedna od tri tabele činjenice *Data Warehouse* baze podataka. Ona sadrži podatke o prodaji artikla u apoteci na nivou stavke. Data tabela činjenica povezana je sa sledećim dimenzijama: *CodeBook*, *Producer*, *Pharmacy*. Tabela činjenica prodaje sadrži podatke o šifri organizacione jedinice, šifri artikla, šifri proizvođača, datumu i vremenu prodaje, broju računa,

tipu prodaje (R- recept, RV- vojni recept, C-slobodna prodaja), šifri recepta, količini artikla, maloprodajnoj ceni, marži, vrednosti PDV – a, popustu na kasi, ukupnom popustu, prodajnoj ceni, neto prometu, bruto prometu, ukupnoj marži, datumu ubacivanja podataka, tipu globalnog artikla, globalnoj šifri proizvođača, globalnom nazivu proizvođača.

Tabela *Nabavka* je jedna od tri tabele činjenica *Data Warehouse* baze podataka. Ona sadrži podatke o nabavci artikla u apoteci na nivou stavke. Data tabela činjenica povezana je sa sledećim dimenzijama: *CodeBook*, *Producer*, *Pharmacy*. Tabela *Nabavka* sadrži podatke o šifri organizacione jedinice, šifri artikla, šifri proizvođača, šifri dobavljača, nazivu dobavljača, datumu nabavke, količini, nabavnoj ceni, datumu i vremenu inserta podataka, šifra proizvođača globalno, nazivu proizvođača globalno.

2.3. Opis ETL procesa

ETL proces obuhvata učitavanje podataka iz izvorišne baze podataka, njihovu transformaciju i krajnje učitavanje u određenu *Data Warehouse* bazu podataka. Ovaj proces je realizovan uz pomoć *Microsoft Management Studio 2012* TSql programskog jezika. Komunikacija između *database* servera obezbeđena je pomoću *Linqed* servera. U nastavku će detaljnije biti opisano na koji način je izvršeno mapiranje za dimenzione i činjenične *Data Warehouse* tabele. Podaci o artiklima za dimenziju *CodeBook* dobijaju se iz tabele *m_Artikal* izvorne baze podataka. Da bi se dobila validna torka koju treba učitatiti u ovu dimenziju, neophodno je izvršiti operaciju spoja između tabele *m_Artikal* i *m_Artikal_i* nakon čega su dobijeni podaci spremni za kopiranje u ciljnu tabelu.

Za dimenziju *Producer* podaci se dobijaju iz tabele *m_Partner* izvorne baze podataka. Takođe, neophodno je izvršiti operaciju spoja između tabele *m_Partner* i *m_Partner_i* nakon čega su dobijeni podaci spremni za kopiranje u ciljnu tabelu.

Dimenzija *Pharmacy* popunjava se podacima iz izvornih tabele *m_OrgJedinica*, *m_OrgJedinica_u* i *m_OrgJedinicaPodatak*. Da bi se dobili podaci koje treba učitati neophodno je izvršiti operaciju spoja. Dimenzija *Chain* popunjava se unosom podataka kroz sql upit direktno nad bazom podataka.

Tabela činjenice *Prodaja* kao izvor podataka koristi tabele: *PosTr_RacunStavka*, *PosTr_RacunZaglavlje* i *posTr_RacunStavkaProizvodnje*, *m_OrgJedinicaPodatak* i *m_Artikal*. Pored podataka o prodaji koji se preuzimaju iz datih tabele urađena je i transformacija datih podataka. Maloprodajna cena se računa na taj način što se od prodajne cene oduzme iznos poreza. Ovakva preračunata vrednost smešta se u obeležje *RetailPrice*. Marža se računa tako što se od prodajne cene oduzme iznos poreza, nakon toga oduzme i nabavna cena. Vrednost marže smešta se u obeležje *Margine*. Vrednost popusta se računa tako što se od maloprodajne cene oduzme prodajna cena. Ovakvo preračunata vrednost smešta se u obeležje *Discount*. Bruto promet se računa tako što se količina množi sa maloprodajnom cenom. Vrednost bruto prometa smešta se u obeležje *GrossTurnover*. Ukupna marža se računa tako što se količina množi sa maržom. Vrednost ukupne marže smešta se u obeležje *Margin_Total*. Ukupan popust se računa tako što se količina množi sa

popustom. Ovako preračunata vrednost smešta se u obeležje Total Discount. Prodajna cena se dobija se kada se od maloprodajne cene oduzme popust. Ovako preračunata vrednost smešta se u obeležje SellPrice. Neto promet dobija se kada se količina množi sa prodajnom cenom. Vrednost neto prometa smešta se u obeležje NetTurnover. Podaci za datu dimenziju se prebacuju svakodnevno. Takođe u datoj dimenziji ukoliko je postojao nalog za ispravku za dati dan podaci se prenose sa negativnim predznakom za količinu pa samim tim i vrednosti su negativne prilikom transformacije.

Tabela činjenice Zalihe kao izvor podataka koristi tabele: rt_Dokument, rt_Dokument_2, rt_ZalihaStavkaDok, m_OrgJedinicaPodatak i m_Artikal, m_Partner_u. Pored podataka o stanju zaliha koji se preuzimaju iz datih tabela urađena je i transformacija datih podataka. U procesu transformacije računa se maloprodajna cena tako što se od prodajne cene oduzima iznos poreza. Polje mrtvi lager dobija se kada se primene uslovi da je prodaja u poslednjih šest meseci nula i da je trenutno stanje veće od nule gde se isključuju artikli koji su novi u poslednjih šest meseci. Sporohodni podatak se dobija primenom formule trenutni lager veći ili jednak prodaji artikla u poslednjih tri meseca i artikli koji ne pripadaju grupi mrtvog lagera. Dati podaci za datu dimenziju se prebacuju svakodnevno, posle zatvorenog dana u apoteci. Ukoliko je postojao u danu dokument 'povrat robe dobavljaču' podaci se prenose sa negativnim predznakom.

Tabela činjenice Nabavka kao izvor podataka koristi tabele: rt_Dokument, rt_Dokument_2, m_Artikal, rt_ZalihaStavkaDok, rt_StavkaDokKalkElementi, m_Partner_Naziv, m_Partner_u i m_OrgJedinicaPodatak. Dati podaci kao i kod dimenzija Zalihe i Prodaje se prebacuju svakodnevno, posle zatvorenog dana u apoteci.

3. IZVEŠTAVANJE IZ PODSISTEMA APOTEKARSKOG SISTEMA

Jedan od zadataka ovog rada je napredno izveštavanje u apotekarskom sistemu. Ovo poglavlje detaljno opisuje funkcije koje su implementirane i način upotrebe tih funkcija. U nastavku poglavlja se govori o korisnicima ovog rešenja, koje funkcije su im na raspolaganju i dat je izgled formi koje pomažu pri realizaciji pomenutih funkcija. Korisnici ovog sistema su vlasnici ili menadžeri apoteka koji ove funkcije koriste radi praćenja prodaje, zaliha, nabavke a u cilju poboljšanja poslovanja. Svim korisnicima dostupne su sve funkcije sistema. Napredni izveštaji su podeljeni u tri osnovne biznis komponente: Prodaja, Zalihe i Nabavka. O svakoj navedenoj komponenti korisnik može da prati odgovarajuće parametre. U svim delovima izveštajnog sistema korišćene su iste intuitivne ikonice koje korisnik može iskoristiti za jednostavnije praćenje podataka.

3.1. Forma izveštaja

Svi izveštaji na izveštajnom sistemu su prikazani u istoj formi.

Za izabrani parametar/parametre, apoteku/lanac, mesec vide se:

1. Mesečna vrednost
 1. za tekuću godinu,
 2. prethodnu godinu i

3. razlika mesečne vrednosti za tekuću i prethodnu godinu, izražena u procentima.

2. Kumulovana vrednost od početka godine

1. za tekuću godinu,
2. prethodnu godinu i
3. razlika kumulovane vrednosti za tekuću i prethodnu godinu, izražena u %.

Kod izveštaja koji se odnose na prodaju i zalihe vidljive su strelice koje svojim brojem i bojom ukazuju na to da li su vrednosti bolje ili lošije u odnosu na prethodnu godinu. Strelice crvene boje pokazuju pad, a zelene rast u odnosu na prethodnu godinu.

Broj strelica zavisi od razlike u vrednostima za tekuću i prethodnu godinu:

- rast/pad 1-5% → 1 strelica,
- rast/pad 5-10% → 2 strelice i
- rast/pad >10% → 3 strelice.

3.2. Prijava na sistem

Izveštajni sistem se nalazi na web stranici na kojoj korisnik unosi svoje korisničko ime i lozinku radi provere njegovog identiteta (Slika 5).



Slika 5. Prijava na system

Ukoliko je prijava uspešna, korisniku se prikazuje kontrolna tabla koja ima tri celine: Kontrolna tabla, Apoteka i Lanac.

Na kontrolnoj tabli (slika 6) vidi se pregledniji prikaz parametara prodaje, zaliha i nabavke.

Kontrolna tabla									
Podsumi									
Mesec									
Dec 2014									
Prodaja (P)									
Zalihe	22,389,028	5,007,080	-19,426	217,207,212	192,238,138	-4,978			
Nabavka	4,817,272	1,010,103	-19,426	42,207,492	38,407,211	-4,876			
Lanac	1,480,100	244,289	-42,094	11,162,519	11,140,512	-22,006			
Prodaja (Z)	3,605	707	-19,426	35,007	37,203	2,196			
Zalihe (Z)									
Prodaja (Z)	3,605	707	-19,426	35,007	37,203	2,196			
Nabavka (Z)	4,817,272	1,010,103	-19,426	42,207,492	38,407,211	-4,876			
Lanac (Z)	1,480,100	244,289	-42,094	11,162,519	11,140,512	-22,006			

Slika 6. Kontrolna table

Glavni meni ima četiri celine: Kontrolna tabla, Apoteka, Lanac i Plan.

Na nivou apoteke, kao i na nivou lanca vide se kao posebne celine: Prodaja, Zalihe i Nabavka, u okviru kojih korisnik može pratiti odgovarajuće parametre.

U delu Plan postoje dva pod nivoa:

- Apoteka i
- Lanac.

4. JEDINSTVENA LISTA ŠIFARNIKA (UPL)

U ovom poglavlju detaljno je opisan servis koji se koristi kao podrška pri održavanju šifarnika koji su potrebni da bi napredni izveštaji sa što boljom preciznošću pokazali validnije podatke. Takođe, u nastavku poglavlja se govori o korisnicima ovog rešenja tj. zaposlenima u organizaciji koji su odgovorni za uparivanje centralnog šifarnika i šifarnika apoteke, kao i o tome koje funkcije su im na raspolaganju. Dat je izgled formi koja pomaže pri realizaciji pomenutih funkciji.

4.1. Prijava rada

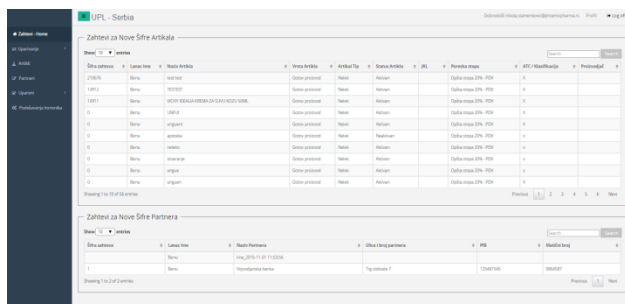
Centralni šifarnik se nalazi na web stranici gde korisnik unosi svoje korisničko ime i lozinku radi njegove provere identiteta.

Ukoliko je prijava uspešna, korisniku se prikazuje glavni prozor (slika 8) odakle može da pristupa funkcijama aplikacije.

Pri pokretanju u glavnom prozoru aplikacije korisnik može da bira između tri opcije: Zahtevi - Home, Uparivanje, Artikli, Partneri, Upareno i Podešavanje korisnika.

4.2. Zahtevi – Home

Nakon odabira opcije Zahtevi – Home, ili prilikom prvog inicijalnog ulaska u aplikaciju pojavljuju se dve tabele sa podacima. Jedna pod nazivom Zahtevi za nove šifre artikala, a druga pod nazivom Zahtevi za nove šifre partnera. Ukoliko zahtevi postoje potrebno je otići na opciju Uparivanje.



Slika 8. Zahtevi-home

Izborom opcije Uparivanje\Artikli otvaraju se dve tabele koje služe za mapiranje (uparivanje) artikla. Korisnik dati zahtev koji je poslat uparuje sa Centralnim šifarnikom artikla.

Izborom opcije Uparivanje\Partneri takođe se otvaraju dve tabele koje služe za mapiranje (uparivanje) partnera. Korisnik kao i kod zahteva Artikla uparuje na isti način i Partnera.

U odeljku Artikli se nalazi centralni šifarnik Artikla gde se korisnicima nude sledeće opcije za manipulaciju šifarnika : Obriši, Izmeni i Novi.

U odeljku Partneri se nalazi centralni šifarnik Partneri gde se korisnicima nude sledeće opcije za manipulaciju šifarnika : Obriši, Izmeni i Novi.

5. ZAKLJUČAK

U ovom radu predstavljen je proces razvoja *Data Warehouse* sistema i naprednog sistema za poslovno izveštavanje, u okviru apotekarskog sistema. Opisan je isprojektovan i implementiran ETL proces prilikom izgradnje *Data Warehouse* sistema. Nakon završene

implementacije izveštajnog sistema, kreirani su izveštaji koji demonstriraju način na koji on funkcioniše. Podsystem koji je iskorišćen kao izvor podataka izveštajnog sistema, je podsystem jedne apoteke.

Takođe opisani su, pored izveštajnog sistema, servisi koji se koriste kao podrška pri održavanju šifarnika u apoteci.

Prilikom kreiranja konceptualne i implementacione *Data Warehouse* šeme baze podataka iskorišćen je alat Microsoft SQL Server. Za kreiranje izveštaja upotrebljeni su programski jezik i *Bootstrapframework*. Takođe, ista tehnologija je korišćena prilikom projektovanja servisa koji se koristi kao podrška za održavanje šifarnika.

Neki od daljih pravaca razvoja izveštajnog sistema bi bili:

- Pored postojećih izveštaja za nabavku koji su opisani u radu, potrebno je napraviti novi set izveštaja za nabavku gde bi se video ostvareni rabat apoteke ka dobavljaču.
- Pravljenje forme za unos plana, na nivou lanca i apoteke, pomoću koga bi se pratilo njegovo ostvarenje.
- Kreiranje upozorenja i automatskog dostavljanja izveštaja upotrebom automatskih notifikacija na određene promene u sistemu. Time, korisnici ne bi bili u stalnoj obavezi da prate rezultate na izveštajima. Nakon određenog događaja, sistem bi obavestio korisnika putem *e-mail* poruke, telefona, ili upozorenja ispisanog na izveštaju, i tek nakon toga korisnik bi preuzeo određene korake za rešavanje prijavljenog problema.

Realizacijom naprednog izveštajnog sistema unapređen je proces donošenja poslovnih odluka prilikom analize izveštaja. Zahvaljujući interakciji sa izveštajem, korisnicima je omogućen prikaz podataka, čime je u velikoj meri skraćen put dolaska do prave informacije.

6. LITERATURA

- [1] I. Luković, Baze podataka, materijali sa predavanja, Univerzitet u Novim Sadu, Fakultet tehničkih nauka, 2012
- [2] Itzik Ben-Gan, Microsoft SQL Server 2012 T-SQL Fundamentals, 2012
- [3] Itzik Ben-Gan, Dejan Sarka, Adam Machanic, Kevin Farlee, T-SQL Querying, 2015
- [4] Microsoft Virtual Academy, dostupno na: <https://mva.microsoft.com/>, 01.01.2016
- [5] Microsoft MSDN library.

Kratka biografija:



Nikola Stamenković rođen je 27. aprila 1987 godine u Beogradu, Srbija. Završio je Fakultet tehničkih nauka, osek Računarstvo i automatika, usmerenje primenjene računarske nauke i informatika. Osnovne akademske studije završio je 2013 godine. Master studije je upisao iste godine, na Fakultetu tehničkih nauka u Novom Sadu, gde je položio sve ispite predviđene planom i programom.

**ДЕТЕКЦИЈА, АНАЛИЗА И КОНВЕРЗИЈА ОПИСА ДОГАЂАЈА У ПАМЕТНИМ
ЕЛЕКТРОЕНЕРГЕТСКИМ СИСТЕМИМА****DETECTION, ANALYSIS AND CONVERSION OF EVENT DESCRIPTIONS IN SMART
GRIDS**

Срђан Цвијановић, Факултет техничких наука, Нови Сад

Област - ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај - Рад приказује анализу водећих решења на пољу *Security Information and Event Management (SIEM)* алата. Рад такође описује имплементацију алгоритма за детекцију и пријаву значајних догађаја у софтверском решењу који се примењује у паметним електроенергетским системима.

Abstract – This paper presents an analysis of the leading solutions in the field of *Security Information and Event Management (SIEM)* tools. It also describes the algorithm for the detection and reporting of significant events in the software solution which is used in smart grids.

1 УВОД

Циљ овог рада је упоредна анализа *Security Information and Event Management (SIEM)* система, као и детекција, анализа и складиштење значајних догађаја у паметним електроенергетским системима. Заштита информација у електроенергетским системима представља веома тежак посао, нарочито ако су системи велики, са већим бројем рачунара, мрежне опреме и стручњака укључених у рад. У таквим системима, заштиту је нарочито потребно спроводити у анализи логова, односно дневника догађаја које стварају оперативни системи, апликације, различита мрежна опрема и разни други програми и уређаји. Ручна анализа дневника догађаја често може да буде мукотрпна, посебно код великих система где је потребно анализирати велику количину података. *Security Information and Event Management (SIEM)* системи за управљање сигурносним информацијама и догађајима баве се решавањем управо овог проблема. Анализом дневника догађаја прикупљених из разних извора које електроенергетским систем користи, *SIEM* може утицати на његову сигурност. Овом анализом може се доћи до сумњивих података, те по њима извршити прилагођавање алата за заштиту од потенцијалних напада. *SIEM* се може поделити на две компоненте. Прва је управљање информацијама (*Security Information Management, SIM*), а друга управљање догађајима (*Security Event Management, SEM*). У раду ће се најпре говорити о *SIEM*-у и водећим решењима у којима се он примењује. Структура рада конципирана је тако, да поред уводног дела у коме су изложене уводне напомене и кратак увод у тему, садржи још шест поглавља:

НАПОМЕНА:

Овај рад је проистекао из мастер рада чији ментор је био др Имре Лендак, доцент.

- У другом делу детаљно је описан систем за сигурносним информацијама и догађајима *Security Information and Event Management (SIEM)*. У овом делу говори се и о томе како сигурносни логови обезбеђују повећање сигурности система, управљању дневницима догађаја као и о њиховом складиштењу. Дат је и кратак увод о *Smart Grid* систему.
- Трећи део описује водећа *SIEM* решења.
- Четврто поглавље приказује коришћене алате и технологије.
- У петом делу описан је алгоритам за детекцију догађаја од интереса, односно механизам обраде догађаја.
- Шести део је закључак овог рада.

2 АНАЛИЗА ПРОБЛЕМА

Ради што бољег разумевања тематике овог рада, изложен је кратак опис *SIEM* система и њихових компоненти. Поред тога је дат кратак опис *Smart Grid* система и дневника догађаја и начини њиховог генерисања у оквирима истих.

2.1 Security Information and Event Management

SIEM представља систем за управљање сигурносним догађајима на темељу прикупљања, анализе и приказа информација присутних у дневницима догађаја. Прикупљањем догађаја из електроенергетског система и њиховом анализом може се побољшати сигурност самог система. То је управо оно чиме се *SIEM* бави. Као што је већ наведено у уводном делу рада, *SIEM* је могуће поделити у две категорије, управљање информацијама (*SIM*) и управљање догађајима (*SEM*).

2.1.1 Управљање информацијама (SIM)

SIM представља део *SIEM*-а који се бави прикупљањем информација из разних извора и њиховим складиштењем. Након складиштења, потребно је омогућити приступ информацијама ради њиховог даљег коришћења, односно њихове анализе и обраде. За то се користе алати за управљање логовима (енгл. *Log Management Tools*).

2.1.2 Анализа догађаја (SEM)

Главни задатак *SEM* је анализа дневника догађаја у реалном времену, ради правовременог откривања и спречавања потенцијалних претњи. *SEM* је део *SIEM*-а који се бави анализом претходно централизовано

ускладиштених догађаја. За овакву врсту анализе користе се алати који догађаје анализирају аутоматски. Електроенергетски системи су често велики и стога емитују огромну количину догађаја.

Њихово ручно анализирање би одузело јако пуно времена и било би финансијски неисплативо. Због тога је пожељно користити алате за аутоматски анализу.

2.2 Повећање сигурности анализом дневника догађаја

Дневници догађаја користе се за проналажење и отклањање грешака који се јављају у систему. Поред тога, могуће их је користити и са аспекта сигурности, рецимо у праћењу активности корисника система. Овакви дневници догађаја представљају сигурносне логове чијим коришћењем се сигурност система може побољшати у великој мери.

Сигурносни логови могу бити генерисани од стране различитих елемената у оквиру Smart Grid система, нпр. сигурносних програма, оперативних система или апликација.

3 ВОДЕЋА SIEM РЕШЕЊА

Ово поглавље описује водећа *SIEM* решења. Постоји велики број *SIEM* решења од којих су нека комерцијална, а нека бесплатно доступна.

Неки од најпознатијих комерцијалних решења су:

- *Symantecov Security Information Manager*,
- *RSA enVision*,
- *Q1 Labs Log Manager*,
- *HPE ArcSight*,
- *IBM QRadar*, односно
- *Splunk*.

Комерцијална решења се претежно користе у већим организацијама, док су бесплатне имплементације погодне за мање организације. Неки познатији примери бесплатних решења су:

- *Pivot Point Security OSCAR Free*
- *CorreLog Server SIEM*
- *NetWrix Event Log Manager Free*
- *AlienVault Open Source SIEM*, односно
- *Cyberoam iView*.

Детаљније ће бити описани: *AlienVault Open Source SIEM*, *Splunk Free*, *ArcSight*, *QRadar* и *CorreLog Server SIEM*. За анализу догађаја коришћени су алати „*CorreLog Server SIEM*“ и „*Splunk*“.

3.1 Splunk

Splunk је један од најпопуларнијих алата за управљање догађајима. Поред комерцијалне верзије *Enterprise*, постоји и бесплатна верзија *Free*. Као и већина бесплатних верзија овакве врсте софтвера, *Free* је значајно ослабљен у односу на комерцијалну верзију, али првих шездесет дана нуди потпуно једнаке функционалности. Ова верзија је ограничена на 500 MB дневно.

Верзија *Enterprise* може да обрађује и анализира и терабајте података те је веома погодна за велике организације. Подржава генерисање извештаја у више формата, нпр. текстуални, табеларни или графички облици. Садржи велики број апликација на које се може разложити. Могуће је инсталирати апликације по жељи, односно прилагодити их потребама. Неке од апликација су: „*Splunk for Cisco*“, „*Web Page Monitor*“, „*Splunk for Windows*“, „*Splunk for Unix and Linux*“, „*Splunk for FS*“, итд..

3.2 CorreLog Server SIEM

CorreLog Server SIEM је производ који садржи основне функционалности за спровођење пуне *SIEM* имплементације. Заснован је на стандардном методу прикупљања и анализе података. *CorreLog Server SIEM* представља флексибилно решење за јачање безбедности. Садржи пакет са алатима и функционалностима које омогућавају безбедно чување података. Подржава управљања логовима и њихову корелације у реалном времену. Укључује претрагу велике брзине и извештавање.

3.3 HP ArcSight

ArcSight спада у једно од најпознатијих комерцијалних решења *SIEM* система. Компонента *Logger* је алат који обједињује претраживање, извештавање, анализу и алармирање на основу логова. Компанија нуди различите алате:

- *Enterprise Security Manager (ESM)* – спроводи корелациону анализу догађаја у читавој организацији/систему.
- *Logger* – претраживање и прикупљање логова. Доступан је за бесплатно преузимање. Не имплементира *SIEM* него искључиво управљање логовима. Има подршку за *Common Event Format (CEF)* проширив, текстуално базиран формат.
- *Express* – управљање логовима и корелациону анализу.
- *IdentityView* – праћење активности корисника.
- *Connectors* – прикупљање података са различитих извора.

Главни проблем *SIEM* окружења је интеграција. Произвођачи различитих уређаја често имају различите формате дневника догађаја и њихова међусобна интеграција може да буде дуготрајан и тежак посао. *CEF* је стандардни формат који решава овај проблем интеграције.

4 АЛГОРИТАМ ЗА ДЕТЕКЦИЈУ ДОГАЂАЈА ОД ИНТЕРЕСА

У овом поглављу описан је ток података од њиховог прикупљања и обраде до анализе употребом изабраних решења за анализу догађаја. Решење је развијено употребом *Microsoft Visual Studio Express 2015* окружења. Коришћен је *SevenZipSharp.dll* фајл. Подаци од интереса налазе се у логовима које апликације генеришу. Како бисмо дошли до њих потребно је одабрати логове и из њих ишчитати податке.

Након одабира датотеке у којој се налазе догађаји од интереса, догађаји се филтрирају и прослеђују путем стандардног улаза *Python* скрипти написаној у *PyScripter for Python 2.7* окружењу. У скрипти су наведени адреса рачунара на који се логови прослеђују и порт на коме је сигурносни алат конфигурисан.

5 АНАЛИЗА ДОГАЂАЈА ПРИМЕНОМ ОДАБРАНИХ SIEM РЕШЕЊА

У овом одељку извршена је анализа догађаја применом одабраних *SIEM* решења. Решења различитих произвођача имају различите методе претраживања и анализирања догађаја.

Овај рад обухвата анализу догађаја применом *Splunk* и *CorreLog Server SIEM* алата.

5.1 Анализа *Splunk* решења

Након инсталације *Splunk* алата потребно је конфигурирати порт намењен за примање догађаја и одабрати протокол за размену порука.

За претраживање логова користи се велики број команди, од којих посебно издвајамо команду *stats*. Она ради на принципу *sql*-а и креира статистике над атрибутима догађаја. Уколико ову команду користимо без *by* клаузуле, резултат претраживања биће један ред. Комплетна синтакса команде *stats*:

```
stats      [partitions=<num>]      [allnum=<bool>]
[delim=<string>] ( <stats-agg-term>... | <sparkline-agg-term>... ) [<by-clause>]
```

Stats команда користи велики број функција за прорачуне над догађајима:

```
avg() | c() | count() | dc() | distinct_count() | earliest() |
estdc() | estdc_error() | exactperc<int>() | first() | last() |
latest() | list() | max() | median() | min() | mode() | p<in>() |
perc<int>() | range() | stdev() | stdevp() | sum() | sumsq() |
upperperc<int>() | values() | var() | varp().
```

Пре саме претраге, корисно је одабрати временски период у оквиру кога су догађаји које претражујемо настали. То се може урадити кликом на дугме *All time*. Резултате претраге могуће је сачувати у неколико облика: Report, Dashboard Panel, Alert, Event Type. То је могуће урадити избором опције "Save as".

5.2 Анализа *CorreLog Server SIEM* решења

Након инсталације „*CorreLog Server SIEM*“ алата отвара се његово окружење у Интернет претраживачу. Сервер је конфигуриран на порту који је унесен приликом инсталације.

Порт за комуникацију путем протокола аутоматски се подешава на вредност 514. Избором опције *Messages* из главног менија, приказују се сви догађаји ускладиштени у алату.

Главни мени садржи ставку *Dashboards* која приказује различите статистике у систему. *CorreLog Server SIEM* попут осталих *SIEM* алата садржи моћан подсистем за креирање извештаја.

Одабиром ставке главног менија *Reports* пружа се могућност креирања извештаја у више формата.

Ставка менија *Correlation* служи за креирање тригера и обавештења. Избором подставки *Actions* и *AddNew*, креира се нова акција.

6 УПОРЕДНА АНАЛИЗА

У овом поглављу рада ће бити приказане могућности двосмерне интеграције *CorreLog* и *Splunk* решења.

6.1 Слање *CorreLog* података у *Splunk*

Овај начин интеграције омогућава редуцију претходно ускладиштених података у *CorreLog*-у и њихово прослеђивање у *Splunk*. Овако можемо потенцијално смањити трошкове избацивањем података које није могуће једноставно одбацити, а немају високу важност. На пример, овакав начин интеграције може смањити обим *firewall* података.

6.2 Слање *Splunk* података у *CorreLog*

У овом контексту *CorreLog* решење представља надзор за *Splunk*. Прати се безбедност и статус *Splunk* система, а *CorreLog* обезбеђује слање извештаја, упозорења и обавештења крајњим корисницима.

7 ЗАКЉУЧАК

У овом раду извршена је анализа водећих решења на пољу *Security Information and Event Management (SIEM)* алата и представљен је опис алгорита за детекцију и пријаву значајних догађаја у софтверском решењу који се примењује у паметним електроенергетским системима.

SIEM системи омогућавају корисницима управљање дневницима догађаја, њихово прибављање, складиштење, претраживање као и обрађивање. У данашње време могуће је обрађивати терабајте података.

Ручна анализа података врши се на малој количини информација. Овим је анализа логова достигла значајан напредак.

Прво је извршена детекција догађаја од интереса у дневницима догађаја електроенергетског система. Затим су ти догађаји преведени у формате који одговарају већини *SIEM* решења. На крају, догађаји су послати путем *TCP* или *UDP* протокола на адресу и порт на коме је конфигурирано *SIEM* решење.

Ови системи теже даљем развоју у смислу повећања тачности при анализи информација, повећања доласка до корисних информација добијених из дневника догађаја као и поједностављењу коришћења. Многа *SIEM* решења имају високу цену, те се у последње време јављају бесплатни алати који омогућавају пружање услуга мањим системима.

Зато је алат *Splunk* један од најпопуларнијих алата ове врсте. Он у својој бесплатној верзији нуди алате прилагођене потребама широког дијапазону корисника.

8 РЕФЕРЕНЦЕ

- [1] Linda Briesemeister, Steven Cheung, Ulf Lindqvist, Alfonso Valdes SRI International, Menlo Park, CA, Detection, Correlation, and Visualization of Attacks Against Critical Infrastructure Systems, 2010 Eighth Annual International Conference on Privacy, Security and Trust.
- [2] Igor Anastasov, Danco Davcev, *SIEM* Implementation for Global and Distributed Environments, University "Ss. Cyril and Methodius", the Faculty of Computer Science and Engineering (FCSE) Skopje, Macedonia.
- [3] SIEM Overview, <http://securityinformationeventmanagement.com/siem.html>, август 2016.
- [4] Splunk Documentation, <http://docs.splunk.com/Documentation/Splunk>, август 2016.

Кратка биографија:



Срђан Цвијановић је рођен 23.09.1992. у Бањалуци. Године 2011. завршио је гимназију „Лаза Костић“ у Новом Саду. Исте године уписао се на „Факултет Техничких Наука“ у Новом Саду, смер „Рачунарство и аутоматика“. Основне академске студије завршио је 2015. године. Одмах након тога уписао је мастер академске студије на смеру „Примењено софтверско инжењерство“.

ANALIZA ANOMALIJA U RADU CROSS-PLATFORM SOFTVERSKIH REŠENJA U OKVIRU SMART GRID SISTEMA**ANOMALY ANALYSIS OF CROSS-PLATFORM SOFTWARE SOLUTIONS IN SMART GRID SYSTEMS**

Nikola Vidović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Cilj rada jeste da prikaže softversko rešenje za analizu anomalija i prikupljanje interesantnih događaja iz log fajlova, u kojima većina informacionih sistema beleži važne događaje. Pošto moderni informacioni sistemi kreiraju velike količine takvih podataka, njihova ručna obrada bi bila vremenski zahtevna i podložna ljudskim greškama. Zbog toga se javlja potreba za alatom koji vrši automatizovanu obradu. U radu će biti prikazano rešenje za analizu log fajlova primenljivo u cross-platform rešenjima razvijenih za Smart Grid sistem.

Abstract – The goal of this paper is to present a software solution for anomaly analysis and collecting interesting events from the log files, which are used by most information systems. As modern information systems produce large volumes of log data, their manual analysis would be time-consuming and error-prone. Therefore, it is necessary to develop solutions which automate log analysis. The paper presents a log analysis solution developed by the author for cross-platform, Smart Grid systems.

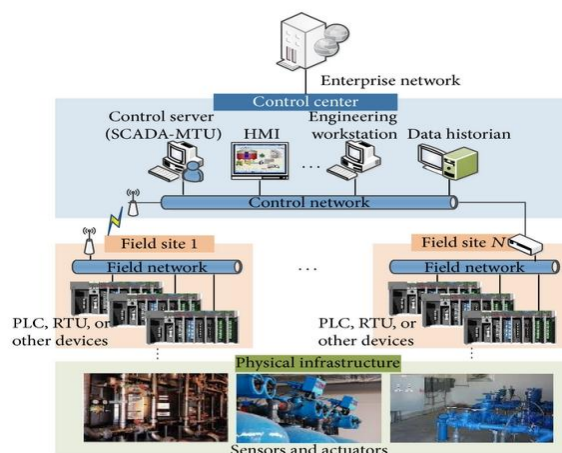
1. UVOD

Smart Grid je elektroenergetska mreža, koja koristi digitalne i ostale napredne tehnologije za nadgledanje i upravljanje proizvodnje, transporta i distribucije električne energije na putu od izvora do krajnjih potrošača. Koristeći informacione i komunikacione tehnologije, Smart Grid prikuplja podatke i automatski deluje na osnovu informacija o proizvodnji i potrošnji, u cilju unapređivanja efikasnosti, ekonomičnosti i održivosti proizvodnje i distribucije električne energije.

Za upravljanje Smart Grid sistemom se koriste akviziciono upravljački sistemi (eng. *Supervisory Control and Data Acquisition* – SCADA). Cilj akviziciono-upravljačkih sistema je obezbeđivanje efikasnog nadzora i upravljanja nad proizvoljnim fizičkim procesom, korišćenjem računara i programske podrške. Prikupljeni podaci se obično mogu videti na jednom ili više SCADA

host računara koji su locirani na centralnom ili master sajtu. Svaki proces u industriji u kome postoji potreba za povećanjem stepena automatizacije je odličan kandidat za primenu SCADA sistema i mreža. Osnovni zadatak akviziciono upravljačkog sistema smatra se ciklična akvizicija digitalnih odmeraka fizičkih veličina, koje

određuju stanje fizičkog procesa. SCADA sistemi podržavaju kako digitalne tako i analogne tipove signala. Uobičajeni analogni signali koje SCADA sistem nadgleda i kontroliše su napon, struja, snaga, nivoi, temperature, pritisci, protoci i brzina. Tipični digitalni signali koji se nadgledaju i kontrolišu su prekidači, zaštitna oprema, releji i motori. Klasičan primer jednog SCADA sistema prikazan je na Slici 1.



Slika 1. SCADA sistem

Softver koji upravlja Smart Grid sistemima je kritičan, jer njegov ispad može da izazove značajne posledice po snabdevanje električnom energijom. S obzirom da se radi od kompleksnom sistemu koji je prostorno distribuiran, potrebno je sve detektovane anomalije sakupiti na jednom mestu. U ovu svrhu se koriste tzv. sistemi za upravljanje logovima (npr. syslog server), ili u zadnje vreme i Security Information and Event Management (SIEM) sistemi.

2. ANALIZA PROBLEMA

Kompleksni Advanced Distribution Management System (ADMS) se integriše sa SCADA-om i zajedno rukuju sa ogromnim brojem promenljivih. Pošto se radi o kompleksnom softveru postoji mogućnost pojave nenamernih softverskih grešaka (eng. bug). Neke od ovakvih grešaka je relativno teško otkriti prilikom razvoja samog softvera, pa takve greške kasnije u toku rada mogu da izazovu različite probleme. Neki od problema se ogledaju u usporavanju rada samog softvera, drugi mogu da izazovu sigurnosne propuste i omoguće neovlašćen pristup sistemu, dok treće mogu da dovedu do padanja servisa i nemogućnosti daljeg funkcionisanja softvera.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Imre Lendak, docent.

Pored softverskih grešaka mogu da se jave i različiti problemi sa hardverom. U zavisnosti koja hardverska komponenta otkaze mogu se javiti različiti problemi. Prilikom otkaza mrežnih kartica računar ostaje bez konekcije ka ostatku sistema. Ukoliko na primer otkazu procesor ili matična ploča ceo računar postaje neupotrebljiv.

Pošto su SCADA i ADMS prostorno distribuirani sistemi, mogu se očekivati različiti problemi sa gubitkom poruka između udaljenih delova sistema. Razlozi mogu biti različiti, od fizičkog prekida komunikacionih kanala, pa do različitih problema u mreži.

Praćenje svih važnih događaja u softverskim sistemima se vrši pomoću logova. Logovi su tekstualni zapisi svih interesantnih događaja u sistemu. Svaka mašina bila klijentska ili serverska, mora da loguje sve svoje akcije koje su od značaja. U velikim distribuiranim softverskim sistemima vrši se obrada velikog broja podataka, samim tim se sakuplja i velika količina logova.

Zbog visokog nivoa kritičnosti SCADA i ADMS je poželjno da ti sistemi budu u stanju da detektuju gore navedene tipove grešaka i da ih zabeleže u log fajlovima. Log fajlovi se čuvaju na više različitih načina, npr. na fajl sistemu, na Windows operativnom sistemu u Event Log-u, ili u bazama podataka.

Krajnji korisnici softvera za Smart Grid bi trebalo da svaki problem u radu softvera prijave proizvođaču softvera. Prilikom prijave problema osim samog opisa poželjno je da se dostave i logovi u trenutku javljanja problema i neposredno pre pojave problema. Prikupljeni log fajlovi mogu biti od velike koristi prilikom analize problema. Za njihovo prikupljanje se koristi ERT. Prilikom pokretanja ERT-a potrebno je uneti naziv datoteke u kojoj će biti snimljeni prikupljeni podaci. Pored osnovnog naziva fajla moguće je dodati i opis. Kada se popune ova polja potrebno je još kliknuti na dugme „Kreiranje izveštaja“ i program će prekopirati sve log fajlove od datog softvera i kompresovati ih. Kompresovani logovi se nalaze na putanji koja je predefinisana u podešavanjima samog ERT-a. Prednost ovog alata se ogleda u tome što korisnik softvera ne mora da zna gde se nalaze log fajlovi nego samo da pokrene ERT. Kao mana ovog programa može se navesti da on prikuplja logove samo lokalno, tj. samo na mašini na kojoj je instaliran. Kao moguće poboljšanje moglo bi se razmišljati o razvoju alata koji bi prikupljao logove i sa svih odabranih računara u okviru jednog sistema, npr. sa jednog klijenta i svih servera.

Da bismo sve ove probleme na vreme detektovali i preuzeli odgovarajuće akcije potrebni su nam alati za obradu logova. Grupa alata koji služe za upravljanje sigurnosnim informacijama i događajima se naziva Security Information and Event Management (eng. SIEM). Neki od poznatijih SIEM alata su HP ArcSight Enterprise Security Manager (ESM), IBM Security QRadar SIEM, McAfee Enterprise Security Manager i Splunk Enterprise.

Da bi sve ove logove imali na jednom mestu, nameće se potreba za centralizovanim prikupljanjem logova. U tu svrhu, se može razviti algoritam za detekciju i analizu

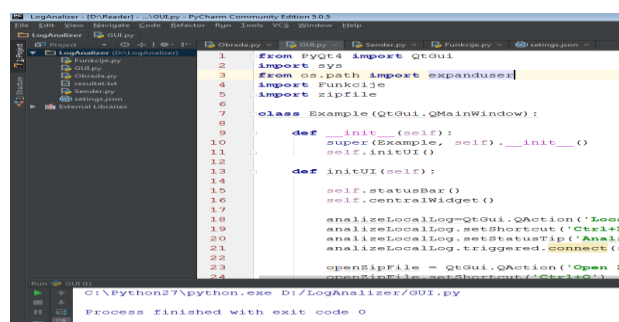
značajnih događaja u informacionom sistemu koji se primenjuje u pametnim elektroenergetskim sistemima.

U okviru ovog rada će biti opisano rešenje za parsiranje logova i ekstrakciju interesantnih događaja iz njih i smeštanje opisa detektovanih događaja u Syslog server, koji služi kao centralno mesto skladištenja logove iz celog sistema. Prednost ovakvog rešenja je što može da se koristi u distribuiranim sistemima.

3. REŠENJE

Zadatak koji je potrebno ispuniti je razvoj alata za obradu logova. Pošto su logovi u stvari tekstualni fajlovi, najlakše ćemo ih obraditi uz pomoć računara. Potrebno je implementirati algoritam koji prolazi kroz logove i pronalazi događaje od interesa. U zavisnosti od konfiguracije sistema i broja korisnika koji koriste sistem, logovi mogu biti raspoređeni u više foldera. Potrebno je izvršiti navigaciju kroz svaki folder i izanalizirati svaku liniju log fajla da bi se pronašli važni događaji koje je sistem zabeležio. U tu svrhu razvijen je alat LogAnalizer. Pored parsiranja logova potrebno je detektovane anomalije prikazati u čitljivom obliku, i obezbediti da se detektovani događaji sakupljaju na jednom mestu.

Da bismo ostvarili prikazivanje i sakupljanje interesantnih događaja, koristićemo sistem za prikupljanje i prikazivanje poruka syslog.



Slika 2. Razvoj LogAnalizer rešenja

LogAnalizer je napisan u Python programskom jeziku koji služi za računarsku obradu logova. Da komande za upravljanje LogAnalizerom ne bi zadavali iz komandnog prozora, alat poseduje grafičko korisničko okruženje (Graphical User Interface – GUI) koji omogućava lakše zadavanje komandi.

Postoje tri režima rada samog alata.

1. Analiza logova na lokalnoj masini
2. Analiza kompresovanih logova, npr. Zip
3. Analiza logova sa zadate lokacije

Prvi režim je analiza logova na lokalnoj mašini. Ulazni parametar za ovaj mod je lokacija log fajlova koji se nalaze na samoj mašini gde je postavljen LogAnalizer. Prilikom postavljanja LogAnalizera definiše se lokacija na kojoj se nalaze logovi.

Drugi režim rada podržava analizu kompresovanih fajlova koji su dopremljeni sa udaljenog računara. Na njihovom izvoru su ti fajlovi prikupljeni alatom za prikupljanje i kompresiju log fajlova u datom trenutku, i dopremljeni su proizvođaču softvera u sklopu prijave problema. U ovom režimu rada aplikacija traži od korisnika da zada putanju do lokacije na kojoj se nalaze .zip fajlovi. Drugi

parametar koji je potrebno zadati je određeni folder u koji će log fajlovi biti raspakovani. U drugom režimu kao ulaz su nam potrebni kompresovani logovi. U tu svrhu korišćen je alat Error Reporting Tool-a (ERT).

Treći režim rada je analiza log fajlova sa zadate lokacije na računaru. Ovaj mod služi prevashodno za analizu logova koji su raspakovani, a u cilju ponovne analize. Najčešće se koristi kod prijave bug-ova kada inženjer na osnovu analize logova detektuju potencijalnu grešku. Program od korisnika traži da odabere lokaciju na kojoj se nalaze log fajlovi.

Pomoću jednog od ova tri režima rada, dobijamo putanju do foldera gde se nalaze log fajlovi. Potrebno je pribaviti sve putanje od određenog foldera ka svim logovima koji se nalaze u folderu. Metoda pomoću koje se pribavlja lista svih putanja prikazana je na Slici 7. Ulazni parametar je putanja do željenog foldera, a izlaz je lista svih putanja do log fajlova.

```
def get_filepath(directory):
    file_paths = [] # list which will store all of the full filepaths.

    # Walk the tree.
    for root, directories, files in os.walk(directory):
        for filename in files:
            # Join the two strings in order to form the full filepath.
            filepath = os.path.join(root, filename)
            file_paths.append(filepath) # Add it to the list.

    return file_paths
```

Slika 3. Pribavljanje putanja do log fajlova

Sada je potrebno proći kroz sve fajlove i pronaći interesantne događaje. Da bi sam algoritam pretrage bio podesiv, koristi se konfiguracioni fajl. Pošto je alat razvijen u Python programskom jeziku, za format konfiguracionog fajla odabran je JavaScript Object Notation (JSON) format, koji se lako obrađuje u Python-u sa jedne strane, i čitljiviji i razumniji je za ljudsko oko sa druge strane. U JSON fajlu navedena su imena svih log fajlova i ključnih reči, na osnovu kojih se vade interesantni događaji tokom parsiranja logova. Po potrebi je moguće menjati konfiguraciju kako se sistem dograđuje i menja. Izgled konfiguracionog fajla prikazan je na Slici 5.

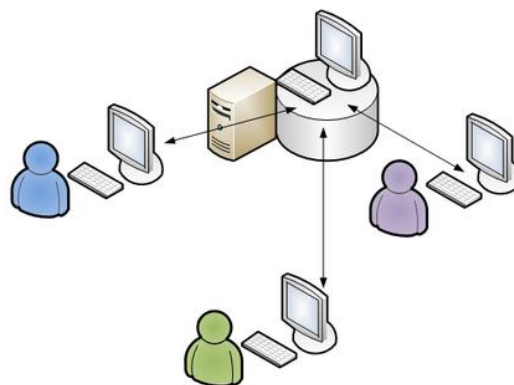
```
config.json
{
  "App1.log": [
    "FATAL",
    "Access violation"
  ],
  "Server.log": [
    "FATAL",
    "ERROR"
  ],
  "App_SystemMonitor": [
    "Access",
    "FATAL"
  ],
  "App_Security.log": [
    "Access",
    "FATAL"
  ]
}
```

Slika 4. LogAnalyzer konfiguracioni fajl

LogAnalyzer prvo upoređuje ime fajla sa imenima fajlova koji su navedeni u konfiguracionom fajlu. Kada se imena fajlova podudaraju, iz JSON fajla se učitavaju ključne reči za dati fajl. Algoritam prolazi kroz svaku liniju fajla i traži ključne reči, npr. „FATAL“. Ukoliko se u nekoj liniji log fajla nađe zadata reč, LogAnalyzer tu liniju zapisuje u tekstualni fajl koji je jedan od izlaza aplikacije. Pored zapisivanja u tekstualni fajl, ta linija se šalje syslog serveru. Syslog je odabran kao određeni format jer je standardizovan standardom RFC 5424. Komunikacija između LogAnalyzer i Syslog servera treba da bude u

skladu sa standardom RFC 5424. LogAnalyzer pre slanja formata poruku u skladu sa datim standardom.

Da bi iskoristili prednost LogAnalyzer i Sysloga potrebno je prostorno distribuirati LogAnalyzer. Pošto je Syslog centralizovan sistem za prikupljanje i prikazivanje logova on ima mogućnost da prima poruke sa više različitih izvora. Jedan od primera distribuiranosti LogAnalyzer dat je na slici 6. Na jednoj mašini je instaliran Syslog server i preko mreže je povezan sa više računara. Svaki računar može lokalno kod sebe da obrađuje log fajlove, a detektovane anomalije šalje ka Syslog serveru.



Slika 5. Prikaz distribuiranosti LogAnalyzer

Za slanje poruka ka syslog serveru LogAnalyzer koristi Python-ovu ugrađenu funkciju koja služi za komunikaciju sa syslog serverom. Metoda koja služi za slanje poruka ka serveru je prikazana na slici 7.

```
handler = logging.handlers.SysLogHandler(address = ('localhost',514), facility=16)
```

Slika 6. Metoda za slanje poruka ka serveru

4. REZULTATI I DISKUSIJA

LogAnalyzer kao rešenje je proistekao iz potrebe da se pređe sa ručne obrade logova na mašinsku. Prijave problema koje stižu od krajnjih korisnika nisu uvek detaljne. Uglavnom prijava sadrži kratak opis problema i u većini slučajeva i ERT. Da bi inženjeri koji analiziraju problem mogli da detektuju koja komponenta je logovala grešku moraju prvo da otpakuju ERT i da počnu pretragu logova. Obično se prvo gledaju logovi komponente koja je navedena u prijavi, a zatim i ostalih komponenti od procesa. Uz pomoć LogAnalyzer ovaj proces se umnogome ubrzava. Svi traženi događaji koji budu pronađeni u logovima biće prikazani na jednom mestu, gde će inženjer lakše moći da zaključi šta je dovelo do greške u softveru.

Pored brzine, ključna stvar je i preciznost. Koliko god bili fokusirani pri manuelnoj obradi, uvek neke stvari mogu da nam promaknu. Ti problemi ne moraju biti odmah vidljivi, već se mogu odnositi i na one probleme na koje i nismo pretpostavili da se mogu javiti. Uz pomoć LogAnalyzer, sve ključne događaje koje smo naveli u konfiguracionom fajlu, pronalazimo u logovima bez izuzetaka.

Za demonstraciju rada samog alata korišćeni su podaci jedne velike kompanije. Radi se o podacima prikupljenim uz pomoć Error Reporting Tool-a (ERT). ERT pristupa lokaciji gde se nalaze log fajlovi i u momentu pokretanja pravi kopiju svih logova i njihovo kompresovanje. Testni podaci koji su korišćeni su veličine 403MB. Prikupljeni su na jednoj radnoj stanici gde je instaliran softver čije logove smo analizirali. Na toj radnoj stanici bilo je ulogovano 7 različitih korisnika i ukupan broj linija log fajlova je iznosio 226,255. Pri ručnom pretraživanju logova za pronalazak problema bi nam bilo potrebno oko 15 minuta, dok smo uz pomoć LogAnalizera taj posao obavili za manje od jednog minuta. Pored povećanja brzine i produktivnosti povećava se i preciznost, a samim tim se i smanjuje mogućnost greške. U tabeli 1 su dati rezultati eksperimenta na još par uzoraka ERT-a.

Tabela 1. Uparedni prikaz rezultata

Redni broj	Veličina ERT-a	Broj linija loga	Broj detektovanih događaja	Vreme
1.	403MB	226255	5	15s
2.	50MB	86927	4	8s
3.	325MB	305264	4	14s
4.	201MB	229171	0	12s

Kao jedan od izlaza LogAnalizera je tekstualni fajl u kome su navedeni detektovani događaji. U fajlu se nalaze sve linije loga koje sadrže zadate anomalije pronađene u logovima. Ispred svake pojedinačne linije stoji i naziv fajla iz kojeg je data linija izvučena. Upoređivanjem dobijenih rezultata inženjer koji vrši analizu lakše će da odredi na koju komponentu da prijavi problem, koji će potom razvojni inženjeri da rešavaju.

Da bi se unapredilo izloženo rešenje, možda bi se moglo razmišljati o integraciji sa nekim od SIEM alata. Na tržištu postoji veći broj proizvođača SIEM alata i gotovo svako proizvođač ima svoj standard za ulazne podatke. Neki od poznatijih standarda su Common Event Format (CEF) i Open Log Format (OLF) oba ova standarda koriste syslog transportni protokol za komunikaciju. Trenutno nam je bilo prihvatljivije rešenje sa korišćenjem syslog servera, pre svega zbog toga što nam je bio dostupan besplatno.

5. ZAKLJUČAK

Cilj ovog rada je bio upoznavanje sa radom softvera za nadzor elektroenergetskih mreža i detekciju anomalija u njegovom radu. Fokus je stavljen na tekstualne logove i analizu zabeleženih anomalija u njima.

Analizirani softver ne beleži logove prema nekom od zvaničnih standarda, te je bilo potrebno implementirati rešenje za konverziju logova u standardni format. Veći problem je predstavljalo uočavanje gde se desio problem, tj. vreme potrebno da se dođe do dela loga koji opisuje razloge ne funkcionisanja softvera.

Kao rezultat ovog rada je razvijena LogAnalizer aplikacija za mašinsku obradu logova. Ona detektuje anomalije u logovima na osnovu zadate konfiguracije. Kao rezultat rada aplikacije generiše se tekstualni fajl u

kome su navedene sve identifikovane anomalije. Pored tekstualnog fajla kao izlaz se generišu i poruke ka syslog serveru. Syslog server koristimo da bismo sve anomalije mogli da sakupimo na jednom mestu radi lakše analize. Prednost syslog servera se ogleda i u tome što možemo da primamo poruke sa više klijentskih mašina na kojima je postavljena LogAnalizer aplikacija.

Autor planiran da prikazano rešenje proširi dodatnim analizama sa akcentom na bezbednost. U planu je i integracija sa nekim od *Security Information and Event Management (SIEM)* alata, koji omogućavaju kontinualno praćenje rada sistema i detektovanje anomalija primenom naprednih algoritama.

6. LITERATURA

- [1] Welcome to Python.org, <https://www.python.org/>, Jun 2016
- [2] Preporuke bezbednosti mrežnih servisa https://www.amres.ac.rs/dokumenti/institucije/csirt/csi-rt-dokumenti/preporuke_bezbednosti_mreznih_servisa.pdf, Septembar 2016
- [3] Understanding Syslog: Servers, Messages & Security <http://www.networkmanagementsoftware.com/what-is-syslog/>, Avgust 2016
- [4] Branislav Atlagić, Softver sa kritičnim odzivom u elektroenergetskim sistemima, Novi Sad 2014
- [5] Max Belkov blog, <http://maxbelkov.github.io/visualsyslog>, Jul 2016
- [6] Technology roadmap, https://www.iea.org/publications/freepublications/publication/smartgrids_roadmap.pdf, Septembar 2016

Kratka biografija:



Nikola Vidović rođen 08.12.1987. godine u Bečeju. Nakon završetka osnovne škole "Đura Jakšić" u Čurugu, upisuje srednju Elektrotehničku školu "Mihajlo Pupin" u Novim Sadu, smer Elektrotehničar automatike. 2006. godine upisuje Fakultet tehničkih nauka u Novom Sadu smer Računarstvo i automatika, a kasnije se usmerava na Automatsko upravljanje. 2015. godine upisuje Master studije na smeru Primenjeno softversko inženjerstvo.

REKONFIGURACIJA DISTRIBUTIVNIH MREŽA U REALNOM VREMENU**REAL TIME RECONFIGURATION IN DISTRIBUTION SYSTEMS**

Stefan Živković, Fakultet Tehničkih Nauka, Novi Sad

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je obrađen problem optimalne rekonfiguracije distributivnih mreža. Proračun je primenjen na uravnotežene distributivne mreže sa simetričnim režimom. Takođe, obrađena je rekonfiguracija u realnom vremenu kako bi se istakle razlike u gubicima aktivne snage za sistem čija se topologija ne menja u odnosu na sistem čije se topologija optimalno menja. U radu je korišćena metodologija izmene grana.

Abstract – This paper describes reconfiguration in distribution systems. Calculations are applied on balanced distribution networks. Method branch exchange appears as employed methodology. Also, real time reconfiguration is done to point out to energy losses difference for the system with fixed network state for a specified time period and for the system where real time reconfiguration is introduced.

Ključne reči: rekonfiguracija, izmena grana, minimum gubitaka aktivne snage, realno vreme, benefit, kriterijumska funkcija, zavisne konture, putanja

1. UVOD

Gubici električne energije u elektroenergetskom sistemu (EES) su posledica isporuke električne energije do potrošača. Oni predstavljaju značajan faktor u EES-u, i ne mogu se zanemariti ili eliminisati. Povećanje gubitaka utiče na povećanu emisiju štetnih gasova i imaju uticaj na kapacitete proizvodnje. Generalno, gubici električne energije se dele na tehničke i komercijalne gubitke. Komercijalni ili netehnički gubici su vezani pre svega za greške u merenju i krađu električne energije.

Tehnički gubici su vezani za fiziku prenosa električne energije i kao takvi oni se ne mogu u potpunosti otkloniti, ali se ulaganjem u mrežu i optimalnim upravljanjem mogu značajno smanjiti. Ako se tome doda da su distributivna preduzeća direktno stimulisana da smanje gubitke (ako su gubici veći od očekivanih plaćaju penale, a u suprotnom ostvaruju profit) jasno je da redukcija gubitaka predstavlja jedan od osnovnih ciljeva za svako distributivno preduzeće. Jedan od veoma efikasnih načina da to urade jeste optimalna rekonfiguracija distributivnih mreža (ORDM).

U literaturi može da se nađe velik broj radova na temu ORDM. Jedan od prvih postupaka, zasnovan na heuristici, objavili su Merlin i Back [1].

NAPOMENA:

Ovaj rad je proistekao iz master rada čiji je mentor dr Goran Švenda, red.prof.

Početno stanje mreže jeste upetljana topologija, nakon čega se prekidači otvaraju jedan po jedan dok se ne dobije radijalna topologija. Ovaj metod je kasnije modifikovan od strane Shirmohammadi i Hong [2] koji su upotreбили koncept najmanjih struja – za upetljane mreže bi sukcesivnom upotrebom proračuna tokova snaga dolazili do radijalne konfiguracije. Cinvalar i Greinger [3] su iskoristili aproksimativne relacije za procenu smanjenja gubitaka. Roytelman i Melnik [4] su predstavili rekonfiguraciju kao višekriterijumski problem koji su rešili kombinacijom proračuna tokova snaga i tehnike izmene grana. Broadwater [5] je upotrebio estimaciju potrošnje kao metod za optimalnu rekonfiguraciju. Chen [6] je prikazao benefite usled satne rekonfiguracije bazirane na kratkotrajnim i dugotrajnim smanjenjima gubitaka. Lopez [7] je prikazao primenu algoritma rekonfiguracije u realnom vremenu uvažavajući promenu potrošnje koristeći dnevne dijagrame potrošnje.

Ideja ovog rada jeste da se uradi kako analiza gubitaka aktivne snage koji se dobijaju usled ne optimalnog upravljanja distributivnom mrežom tako i mogućnost njihovog smanjenja usled primene ORDM.

Nakon uvoda u drugom delu rada izvršena je postavka problema ORDM. U trećem delu je opisan algoritam rekonfiguracije. U četvrtom delu urađena je analiza primene rekonfiguracije u realnom vremenu. Nakon zaključka koji je dat u petom delu, u delu šest referentno je navedena literatura koja je korišćena za realizaciju ovog rada.

2. POSTAVKA PROBLEMA

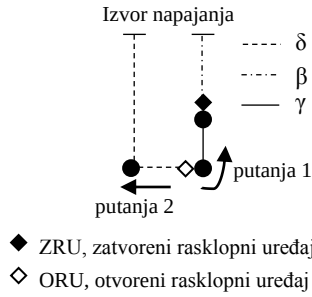
Problem ORDM se sastoji od pravilne definicije kriterijumske funkcije i pogonskih ograničenja. Kao najatraktivniji kriterijum koji se pominje u različitoj literaturi i različitim radovima, jeste minimizacija gubitaka aktivne snage u distributivnim mrežama [2,3,4].

2.1 Kriterijumska funkcija

Za formulaciju aproksimativnih relacija koje se koriste u KF za procenu smanjenja gubitaka nakon rekonfiguracije, od interesa jesu grane na čijim krajevima se nalaze ORU i ZRU, odnosno struje koje se imaju u tim granama. Na slici 2.1 bitno je uočiti ORU i ZRU, putanje 1 i 2, i skupove grana β , γ i δ . Na slici 2.1 prikazani su osnovni pojmovi koji su potrebni za postavljanje problema ORDM. Kriterijumska funkcija (KF) se zasniva na :

Def. 2.1: *Kontura* – putanje sa obe strane ORU prema izvoru napajanja.

Def. 2.2: *Putanja 1* – skup elemenata od čvora ORU prema izvoru napajanja, dok *putanja 2* predstavlja skup elemenata od grane ORU prema izvoru napajanja.



Slika 2.1 – Pojašnjenje po kom pravilu se koriste aproksimativne relacije za proračun struja u konturi

KF za procenu smanjenja gubitaka aktivne snage posle rekonfiguracije, može se predstaviti u jednostavnoj formi kao u (2.01):

$$KF = \max(\Delta P),$$

U produžetku, skup jednačina (2.02)-(2.11), su formulisane aproksimativne jednačine koje su neophodne za proračun KF:

$$\hat{I}_{l, \text{posle}} = \hat{I}_l - \hat{I}_{ZRU}, \quad l \in \beta, \quad (2.02)$$

$$\hat{I}_{l, \text{posle}} = \hat{I}_{ZRU} - \hat{I}_l, \quad l \in \gamma, \quad (2.03)$$

$$\hat{I}_{l, \text{posle}} = \hat{I}_{ZRU} + \hat{I}_l, \quad l \in \delta, \quad (2.04)$$

$$I_l = |\hat{I}_l|, \quad (2.05)$$

$$I_{l, \text{posle}} = |\hat{I}_{l, \text{posle}}|, \quad (2.06)$$

$$P_l = 3 \cdot R_l \cdot I_l^2, \quad (2.07)$$

$$P_{l, \text{posle}} = 3 \cdot R_l \cdot I_{l, \text{posle}}^2, \quad (2.08)$$

$$P = \sum_{l \in \alpha} P_l, \quad (2.09)$$

$$P_{\text{posle}} = \sum_{l \in \alpha} P_{l, \text{posle}}, \quad (2.10)$$

$$\Delta P = P - P_{\text{posle}}, \quad (2.11)$$

gde je:

- l – redni broj grane,
- $\alpha, \beta, \gamma, \delta$ – skup svih grana u konturi, grana koje se topološki nalaze iznad ZRU (prema izvoru napajanja), između ORU i ZRU, skup grana koje se topološki nalaze iznad ORU (prema izvoru napajanja), respektivno, videti sliku 2.1

$\hat{I}_l, \hat{I}_{l, \text{posle}}$ – kompleksne vrednosti struja grane (l) pre i posle rekonfiguracije, respektivno,

$\hat{I}_{ZRU}$ – kompleksna vrednost struje grane sa ZRU,

$I_l, I_{l, \text{posle}}$ – moduli struja grane (l) pre i posle rekonfiguracije, respektivno,

R_l – rezistansa grane (l),

$P_l, P_{l, \text{posle}}$ – gubici aktivne snage grane (l) pre i posle rekonfiguracije, respektivno,

P, P_{posle} – gubici aktivne snage konture pre i posle rekonfiguracije, respektivno,

ΔP – razlika gubitaka pre i posle rekonfiguracije. Što je vrednost veća, to su gubici posle rekonfiguracijemani.

2.2 Pogonska ograničenja

Dok kriterijumska funkcija služi kao sredstvo optimizacije mreže po određenom kriterijumu, ograničenja postoje da bi se proverilo da li se primenom rešenja ORDM DM dovodi u režim sa narušenim ograničenjima. Ukoliko ograničenja nisu narušena, predloženo optimalno stanje mreže je ispravno i dozvoljeno.

Ograničenja koja se sreću u literaturi su sledeća [2,3,4]:

1. Granice naponski svih čvorova nakon rekonfiguracije:

$$V_{\min} < V_k < V_{\max}, \quad k=1, \dots, n, \quad (2.12)$$

gde je: (2.01)

$V_{\min}$ – minimalna dozvoljena vrednost napona, 95(90) % od nominalne vrednosti napona;

$V_{\max}$ – maksimalna dozvoljena vrednost napona, 105(110) % od nominalne vrednosti;

V_k – vrednost napona nakon ORDM u čvoru k ,

n – ukupan broj čvorova DM;

2. Trajno dozvoljena, radna struja provodnika grane [2,3,4]:

$$I_l < I_{nl}, \quad l=1, \dots, n_{gr}, \quad (2.13)$$

I_l – vrednost struje nakon rekonfiguracije u provodniku grane (l),

I_{nl} – trajna vrednost dozvoljene radne struje provodnika grane (l),

n_{gr} – ukupan broj grana;

3. Isporuka električne energije svim potrošačima DM [2,3,4].

4. Zadržavanje radijalne strukture DM [2,3,4].

U ovom radu u obzir su uzeta strujna ograničenja, i ona se eksplicitno proveravaju, dok se za ograničenja radijalnosti i napajanja svih potrošača podrazumeva da nisu narušena s obzirom da se koristi metod izmene grana, u kojem se u jedan rasklopni uređaj otvara a drugi zatvara.

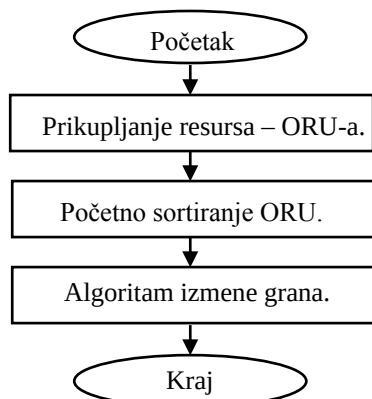
3. ALGORITAM REKONFIGURACIJE

Algoritam rekonfiguracije obrađen u ovom radu, bazira se na principu izmene grana – promena statusa uključenosti para rasklopnih uređaja, od kojih je jedan otvoren a drugi zatvoren [8]. Na slici 3.1 dat je algoritam izmene grana.

1. Sakupljanje svih otvorenih rasklopnih uređaja vrši se po izvodima. Sa pretragom se kreće od početne sekcije izvoda i kreće se topološki „na dole“, po lejerima.
2. Početno sortiranje ORU se vrši prema benefitu koji jedan ORU može da obezbedi u paru sa nekim od ZRU. Ključni momenat je upotreba aproksimativne relacije za procenu poboljšanja KF kada bi se zatvorio ORU i otvorio ZRU. Potrebno je napomenuti da se prilikom provere benefita za početno sortiranje ORU ne menja konfiguracija mreže, već se samo na osnovu aproksimativnih relacija simulira prebacivanje

opterećenja (potrošnje) sa jednog izvoda na drugi izvod.

3. Na kraju, algoritam izmene grana je momentat u algoritmu rekonfiguracije gde se vrši menjanje konfiguracije u modelu mreže. U algoritmu izmene grana zapravo dolazi do promene topologije i postizanja optimalne topologije.



Slika 3.1 – Algoritam rekonfiguracije

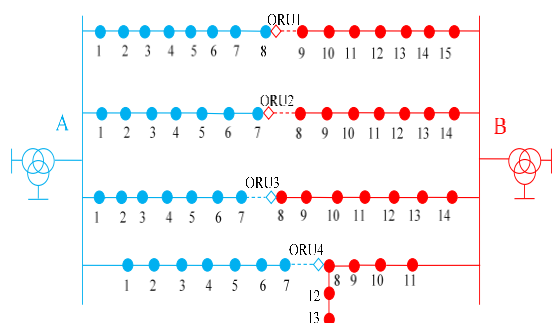
4. ANALIZA PRIMENE REKONFIGURACIJE U REALNOM VREMENU

Ideja je da se isprati ponašanje test DM usled primene ORDM i uradi analiza dobijenih rezultata. ORDM se izvršava na svaki sat sa početnim trenutkom u 0:00 h. Na slici 4.2 date su konfiguracije koje su primenom ORDM dobijene u toku celog dana. Na slikama je sa uokvirenim slovom R naznačeno da se rekonfiguracija primenjuje za deo mreže levo od znaka.

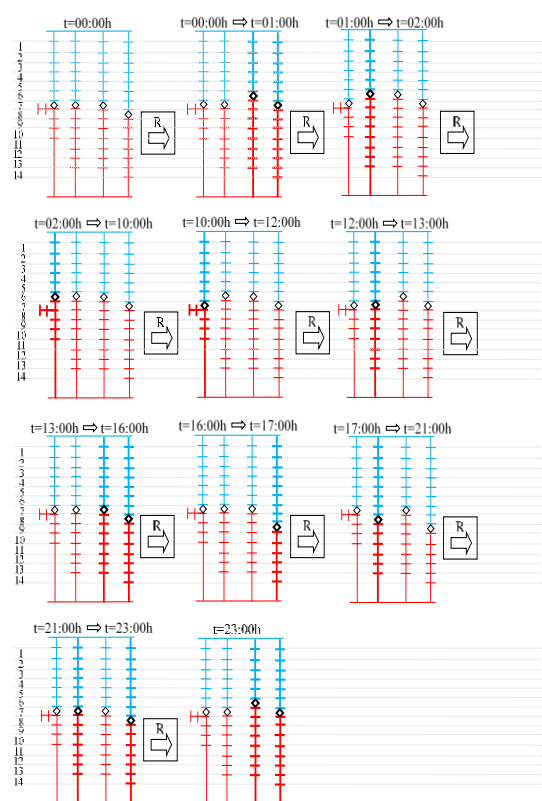
Desno od znaka je novo stanje, za trenutak u kome se rekonfiguracija izvršila. Znak implikacije između dva vremenska trenutka ukazuje na sledeće: pomeriti vreme izvršenja rekonfiguracije na trenutak kada će biti rezultata. Tako npr., za trenutak 0:00 radi se rekonfiguracija i prelazi u novo stanje. Sledeći trenutak za izvršenje jeste 1:00, pa 2:00.

Nakon rekonfiguracije u 2:00, stanje mreže se ne menja sve do 10:00, i tek u 10:00 se izvršava sledeća rekonfiguracija. Po identičnom modelu, dobijena su i ostala stanja mreže koja su prikazana na slici.

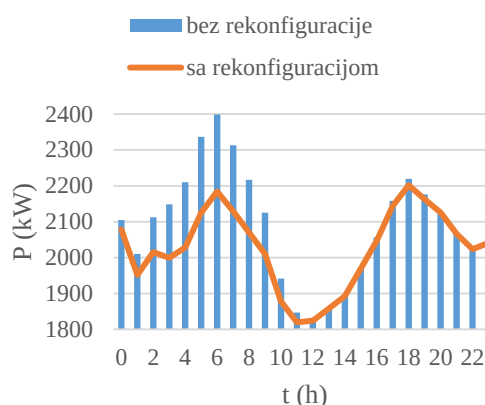
Rezultati u tabeli 4.1 predstavljen je benefit dobijene primenom funkcije ORM u odnosu na početno stanje sa slike 4.1. Za svaki sat računat je benefit u odnosu na početno stanje.



Slika 4.1 – Test primer



Slika 4.2 – Pregled svih konfiguracija u danu, primenom rekonfiguracije



Slika 4.3 – Grafički prikaz gubitaka sa svaki sat

Ovako koncipirana tabela daje uvid u vrednosti gubitaka aktivne snage kada se topologija DM menja, odnosno ne menja tokom razmatranog vremenskog perioda. Benefit od 1540,15 kWh predstavlja uštedu električne energije samo u jednom danu.

Dakle, očigledno je da prostora za smanjenje gubitaka električne energije ima.

Međutim, treba naglasiti da ovom analizom nisu uvaženi troškovi koji se imaju prilikom svake manipulacije kojom se benefit ostvaruju.

Tabela 4.1 – Benefit dobijen rekonfiguracijom

T [h]	$\Delta P_{\text{BEZ REK.}}$ [kWh]	$\Delta P_{\text{SA REK.}}$ [kWh]	Benefit [kW]	Benefit [%]
00:00	2104,60	2078,16	26,44	1,25
01:00	2010,37	1951,68	58,69	2,92
02:00	2112,59	2015,57	97,02	4,59
03:00	2148,59	1999,18	149,41	6,95
04:00	2210,01	2028,38	181,64	8,22
05:00	2336,65	2125,72	210,93	9,03
06:00	2398,83	2184,60	214,23	8,93
07:00	2312,93	2129,65	183,28	7,92
08:00	2216,68	2069,84	146,84	6,62
09:00	2125,32	2010,00	115,32	5,46
10:00	1941,68	1877,77	63,92	3,29
11:00	1846,95	1819,46	27,49	1,49
12:00	1831,30	1824,57	6,73	0,37
13:00	1857,84	1857,84	0,00	0,00
14:00	1892,49	1892,49	0,00	0,00
15:00	1969,34	1969,34	0,00	0,00
16:00	2056,00	2047,27	8,71	0,42
17:00	2158,47	2143,95	14,52	0,67
18:00	2219,34	2201,82	17,52	0,79
19:00	2175,95	2162,77	13,22	0,60
20:00	2126,73	2125,57	1,16	0,05
21:00	2066,45	2066,45	0,00	0,00
22:00	2023,20	2023,20	0,00	0,00
23:00	2044,50	2041,38	3,12	0,15
ΔE [kWh]	50 186,81	48 646,66	1540,15	

5. ZAKLJUČAK

Pored opisa kriterijumske funkcije, najvažnijih pojmova, detaljnog opisa metodologije izmene grana, fokus rada jeste analiza i primena optimalne rekonfiguracije DM u realnom vremenu. Njenom primenom moguće je smanjiti gubitke aktivne snage.

Rekonfiguracija u realnom vremenu spada u skup naprednih funkcionalnosti za upravljanje distributivnih mreža. Jedna od ideja rekonfiguracije u realnom vremenu jeste da se razvije mehanizam kojim će rekonfiguracija redovno aktivirati po unapred određenom vremenskom okviru (1h, 2h, 4h, 6h, 12h), i da na osnovu aktuelnih režima u mreži generiše ili ne generiše upravljačke akcije. Ispod su navedeni neki od izazova vezani za primenu rekonfiguracije:

1. Uticaj rekonfiguracije na koordinaciju relejne zaštite.
2. Uticaj rekonfiguracije na podešenje kondenzatorskih baterija kada se deo izvoda koji sadrži baterije prebaci na susedni izvod. Da li su pozicije kondenzatorskih baterija i dalje na pravi način podešene?
3. Kakav uticaj imaju DG-i nakon rekonfiguracije? Da li je operacija prebacivanja distributivnih generatora sa jednog izvoda na drugi dozvoljena? Šta će se desiti sa tokovima snaga, da li će se uspostaviti inverzni tok snage, kako to utiče na režim?
4. Integracija sa sistemima za prognozu potrošnje, koji će rekonfiguraciji obezbediti sledeće:
 - 4.1 informaciju o bitnoj promeni potrošnje koja će uslediti u toku dana,
 - 4.2 informaciju kada se dostiže vršno opterećenje na izvodu.

5. Robusno IT rešenje, brz prenos informacija, paralelan rad.

Ovim radom, rekonfiguracija DM u realnom vremenu, kao veoma kompleksna tema, samo je otvoren. Veliki broj, pre svega problema praktične primene rezultata optimalne rekonfiguracije DM, prevazilaze okvire ovog rada. Mrežu njih svakako spadaju: detaljna cost/benefit analiza; tehnička i vremenska ograničenja rasklopne opreme i upravljanja sa njom; itd.

6. LITERATURA

1. A. Merlin and H. Back, Search for a Minimal-Loss Operating Spanning Tree Configuration in Urban Power Distribution Systems, *Roc. of 5th Power Systems Com. Con., Cambridge, U. K.*, Sept. 1 - 5, 1975.
2. Dariush Shirmohammadi, H.Wayne Hong: Reconfiguration of electric distribution networks for resistive line losses reduction, *IEEE Transactions on Power Delivery*, Vol. 4, No. 2, April 1989, pp 1492-1498
3. S.Civanlar, J.J.Grainger, H.Yin, S.S.H.Lee: Distribution feeder reconfiguration for loss reduction, *IEEE Transactions on Power Delivery*, Vol. 3, No. 3, July 1988, pp 1217-1223
4. I.Roytelman, V.Melnik, S.S.H.Lee, R.L.Lugtu: Multi-objective feeder reconfiguration by distribution management system, *Siemens Energy & Automation, Inc. Empros Power Systems Control, Brooklyn Park, MN* 55428, pp 517-522
5. R. Broadwater, A. Khan, H. Shalan and R. Lee. Time varying Load Analysis to Reduce Distribution Losses Though Reconfiguration. *IEEE Trans. on Power Delivery*. Vol. 8 No. 1, pp. 294-300. 1993
6. C. S. Chen and M. Y. Cho. Energy Loss Reduction by Critical Switches. *IEEE Transactions on Power Delivery*. Vol. 8 No. 3, 1993., pp. 1246-1253.
7. E. López, H. Opazo, L. García and P. Bastard. On-Line reconfiguration considering variability demand: Applications to real networks. *IEEE Transactions on Power Systems*. Vol. 19 No. 1, pp., 2004.549-553
8. D. Popović, D. Bekut, V. Dabić: *Specializovani DMS Algoritmi*

Kratka biografija:



Stefan Živković je rođen u Beogradu 1991. godine. Osnovne akademske studije završio je 2014. godine, na fakultetu tehničkih nauka u Novom Sadu, odsek Elektrotehnika i računarstvo, usmerenje elektroenergetski sistemi. Master akademske studije je upisao 2014. godine, takođe na fakultetu tehničkih nauka, usmerenje elektroenergetski sistemi.

ANALIZA POUZDANOSTI DISTRIBUTIVNE MPEŽE PRIMENOM DIJAGRAMA STABLA OTKAZA**DISTRIBUTION NETWORK RELIABILITY ANALYSIS USING FAULT TREE DIAGRAM**

Nikola Žilić, Savo Đukić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad opisuje jednu metodu analize pouzdanosti distributivnog sistema – dijagram stabla otkaza. Metoda startuje identifikacijom svih glavnih otkaza komponenata i/ili podсистема sistema koji se analizira. Zatim se različiti slučajevi mogućih otkaza povezuju odgovarajućim logičkim kolima. Predložena metoda je verifikovana na modifikovanoj IEEE test mreži.

Abstract – The paper describes a method for distribution system reliability analysis – fault tree diagram. The method starts with the identification of all major failures of system components and/or subsystems of the system under analysis. Then the various cases of possible failures are mutually connected with the appropriate logic circuits. The proposed method is verified on a modified IEEE test distribution network.

Ključne reči: analiza pouzdanosti, dijagram stabla otkaza

1. UVOD

Pouzdanost predstavlja apstraktan termin koji označava izdržljivost, zavisnost i dobre performanse. Predstavlja nešto što se može proračunati, oceniti, izmeriti, planirati i dizajnirati u deo opreme ili ceo sistem. Pouzdanost označava sposobnost sistema da vrši funkciju za koju je dizajniran tokom radnog veka i pod uslovima za koje je projektovan.

Posmatrano sa stanovišta potrošača, idealna pouzdanost elektroenergetskog sistema bi predstavljala neprekidno napajanje električnom energijom istih. U praksi, ključni indikatori pouzdanosti elektroenergetskih sistema za potrošače su učestanost i trajanje prekida napajanja električnom energijom.

Primena koncepta pouzdanosti na distributivne sisteme se razlikuje od primene na proizvodne i prenosne, u smislu da je više orijentisana na krajnje potrošače nego na sam sistem. Pouzdanost proizvodnje i prenosa pre svega ukazuje na njihov kapacitet i verovatnoću gubitka napajanja.

Veoma je važno napomenuti da distributivni sistem predstavlja važnu vezu između elektroenergetskog sistema i korisnika. U mnogim slučajevima ove veze su radijalne što napajanje potrošača čini sklonim prekidima usled ispada samo jednog elementa.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Savo Đukić, docent.

U prošlosti, distributivnom delu elektroenergetskog sistema se posvećivalo manje pažnje po pitanju planiranja pouzdanosti u odnosu na proizvodne i prenosne sisteme. Glavni razlog za to leži u činjenici da su proizvodni i prenosni sistemi kapitalno intenzivniji i da ispadi u ovim delovima elektroenergetskog sistema mogu prouzrokovati katastrofalne ekonomske posledice po društvo.

Sa druge strane, više od 80% prekida napajanja potrošača dešava se zbog ispada u distributivnom sistemu. Distributivni deo elektroenergetskog sistema predstavlja najslabiju kariku između krajnjih potrošača i izvora napajanja.

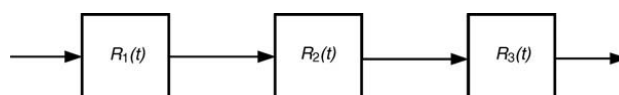
Postoji mnoštvo prilika za elektrodistributivna preduzeća da pametno investiraju u proširenje distributivnog sistema kako bi ispratili rast potrošnje u budućnosti. Koristeći se metodama verovatnoće pouzdanosti eliminišu se rizici od neadekvatnih investicija i obezbeđuje se optimalna pouzdanost napajanja potrošača.

U nastavku rada opisan je način proračuna osnovnih indeksa pouzdanosti serijskih, paralelnih i parcijalno redundantnih sistema (Glava 2). U Glavi 3 opisana je metoda dijagrama stabla otkaza. Opisana metodologija je u Glavi 4 primenjena na test distributivnu mrežu. Osnovni zaključci rada dati su u Glavi 5.

2. POUZDANOST KOMPLEKSNIH SISTEMA**2.1. Serijski sistemi**

Većina sistema se sastoji od podсистема i komponenti. Funkcionalno, ovi podсистemi i komponente su uobičajeno povezani serijskim i paralelnim vezama ili kombinacijom ove dve veze.

Izraz „serijska“ se odnosi na funkcionalan odnos komponenti, a ne na fizičku povezanost. Serijska povezanost označava da je potrebno da svaka komponenta u serijskoj vezi mora da bude raspoloživa da bi sistem funkcionisao. Na Slici 1 prikazan je šematski izgled serijske veze.



Slika 1. Serijska veza

Pouzdanost i stopa otkaza sistema zavise od stope otkaza i pouzdanosti pojedinačnih komponenti sistema. Stoga, da bi mogla da se odredi pouzdanost sistema kao celine potrebni su podaci o stopama otkaza pojedinačnih komponenti. Pouzdanost predstavlja verovatnoću

funkcionisanja sistema u datom vremenskom intervalu, tako da kod sistema sa serijskom povezanošću komponenti ona predstavlja verovatnoću da će svaka komponenta funkcionisati u datom vremenskom intervalu. Ako su otkazi komponenti nezavisni jedni od drugih, tada se verovatnoća da će sistem serijske veze funkcionisati računa kao proizvod verovatnoća da će svaka pojedinačna komponenta funkcionisati. Matematički se to može zapisati na sledeći način:

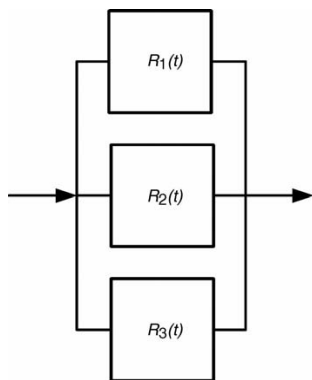
$$R_{sistem}(t) = R_1(t)R_2(t)R_3(t) \cdots R_n(t). \quad (1)$$

Ako komponente u sistemu serijske veze imaju eksponencijalnu verovatnoću otkaza sa odgovarajućim stopama otkaza $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$, onda je pouzdanost sistema jednaka:

$$R_{sistem}(t) = e^{-(\lambda_1 + \lambda_2 + \lambda_3 + \dots + \lambda_n)t}. \quad (2)$$

2.2. Paralelni sistemi

Paralelni sistemi u smislu pouzdanosti označavaju sisteme u kojima je potrebno da samo jedna komponenta u paralelnoj vezi funkcionise, da bi sistem funkcionisao. Na Slici 2 prikazan je šematski izgled paralelne veze komponenti.



Slika 2. Paralelna veza

Pošto je potrebno da funkcionise samo jedan element u paralelnoj vezi da bi sistem funkcionisao, ostali elementi su redundantni. Redundantnost ovakvog sistema ga čini visoko pouzdanim, jer je potrebno da svi elementi u paralelnoj vezi otkazu da bi sistem otkazao. Ako su otkazi međusobno nezavisni, onda je verovatnoća otkaza sistema jednaka proizvodu pojedinačnih verovatnoća otkaza svakog elementa. Ako je $Q(t)$ verovatnoća otkaza u datom periodu, onda je verovatnoća otkaza sistema paralelne veze jednaka:

$$Q(t)_{sistem} = Q_1(t)Q_2(t)Q_3(t) \cdots Q_n(t). \quad (3)$$

Pouzdanost ovakvog sistema se preko verovatnoće otkaza može izraziti na sledeći način:

$$R(t)_{sistem} = 1 - Q(t)_{sistem}. \quad (4)$$

Pouzdanost sistema paralelne veze, ukoliko svi pripadajući elementi imaju konstantnu stopu otkaza i strogo eksponencijalnu funkciju pouzdanosti, može se izraziti na sledeći način:

$$\begin{aligned} R(t)_{sistem} &= 1 - (1 - e^{-\lambda_1 t})(1 - e^{-\lambda_2 t}) = \\ &= (1 - e^{-\lambda_3 t}) \cdots (1 - e^{-\lambda_n t}). \end{aligned} \quad (5)$$

Pouzdanost paralelnog sistema se ne može opisati eksponencijalnom funkcijom kao što je to slučaj kod

serijskog sistema, što je posledica variranja stope otkaza paralelnih sistema tokom vremena.

2.3. Parcijalno redundantni sistemi

Ukoliko sistem, da bi funkcionisao, zahteva da x od ukupno n komponenti radi, pri čemu sve komponente sistema imaju istu stopu otkaza, pouzdanost sistema se može odrediti korišćenjem binomne raspodele verovatnoće. Za analizu takvih sistema koristi se sledeća jednakost:

$$\begin{aligned} (R(t) + Q(t))^n &= R(t)^n + nR(t)^{n-1}Q(t) + \\ &+ nR(t)^{n-2}Q(t)^2 + \cdots + nR(t)Q(t)^{n-1} + Q(t)^n. \end{aligned} \quad (6)$$

Binomno proširenje $(R(t)+Q(t))^n$ predstavlja najopštiji izraz koji pokriva i serijske i paralelne sisteme. Ako je potrebno da samo jedna od n komponenti funkcionise, kako bi ceo sistem funkcionisao (paralelni sistem), pouzdanost sistema je jednaka:

$$\begin{aligned} R_{sistem}(t) &= R(t)^n + nR(t)^{n-1}Q(t) + \cdots \\ &\cdots + nR(t)Q(t)^{n-1} = 1 - Q(t)^n. \end{aligned} \quad (7)$$

Ako je potrebno da svih n komponenti funkcionise, kako bi ceo sistem funkcionisao (serijski sistem), pouzdanost sistema je jednaka:

$$R(t)_{sistem} = R(t)^n \quad (8)$$

Generalno, ako je potrebno da od n komponenti x radi, kako bi sistem funkcionisao, pouzdanost sistema je jednaka:

$$\begin{aligned} R(t)_{sistem} &= R(t)^n + nR(t)^{n-1}Q(t) + \\ &+ nR(t)^{n-2}Q(t)^2 + \cdots + nR(t)^{n-x}Q(t)^x. \end{aligned} \quad (9)$$

3. METODA DIJAGRAMA STABLA OTKAZA

U okviru metode dijagrama stabla otkaza prvo se identifikuju svi glavni otkazi komponenti i/ili podsistema sistema ili dela sistema koji se analizira. Nakon toga se različiti slučajevi mogućih otkaza povezuju prikladnim I ili ILI logičkim kolima.

Neka se pretpostavi da su A i B dva otkaza koja su nezavisna jedan od drugog. Različite logičke kombinacije ova dva slučaja otkaza na ulazu logičkog kola mogu prouzrokovati otkaz na izlazu logičkog kola. Ako su poznate verovatnoće da se na ulazima logičkog kola nađu slučajevi sa otkazom, onda se može proceniti kolika je verovatnoća stanja otkaza na izlazu. Stanje na izlazu logičkog kola zavisiće od logike kojom su ulazi međusobno povezani.

3.1. Logičko I kolo

Ako je $P(A)$ jednako verovatnoći otkaza A , a $P(B)$ verovatnoći otkaza B , za logičko I kolo važi:

$$P_{izlaza} = P(A \cap B) = P(A)P(B). \quad (10)$$

U slučaju seta otkaza A_1 do A_n dovedenih na ulaze logičkog I kola sa n ulaza, verovatnoća otkaza na izlazu računa se na sledeći način:

$$\begin{aligned} P_{izlaza} &= P(A_1 \cap A_2 \cap A_3 \cdots \cap A_n) = \\ &= P(A_1)P(A_2)P(A_3) \cdots P(A_n). \end{aligned} \quad (11)$$

Na osnovu prethodnog izraza može se zaključiti da je kod logičkog I kola verovatnoća otkaza na izlazu jednaka proizvodu verovatnoća otkaza na ulazu.

3.1.2. Logičko ILI kolo

Za logičko ILI kolo važi:

$$P_{izlaza} = P(A \cup B) = P(A) + P(B) - P(A)P(B). \quad (12)$$

U zavisnosti od veličine verovatnoće otkaza na ulazima logičkog ILI kola, jednačina za P_{izlaza} se može pojednostaviti. Ovo posebno važi ako imamo tri ili više ulaza. Proizvod $P(A)$ i $P(B)$ se može zanemariti u slučaju kada obe verovatnoće imaju vrednosti manje od 0,1. U tom slučaju izraz (12) se može pojednostaviti:

$$P_{izlaza} = P(A \cup B) = P(A) + P(B). \quad (13)$$

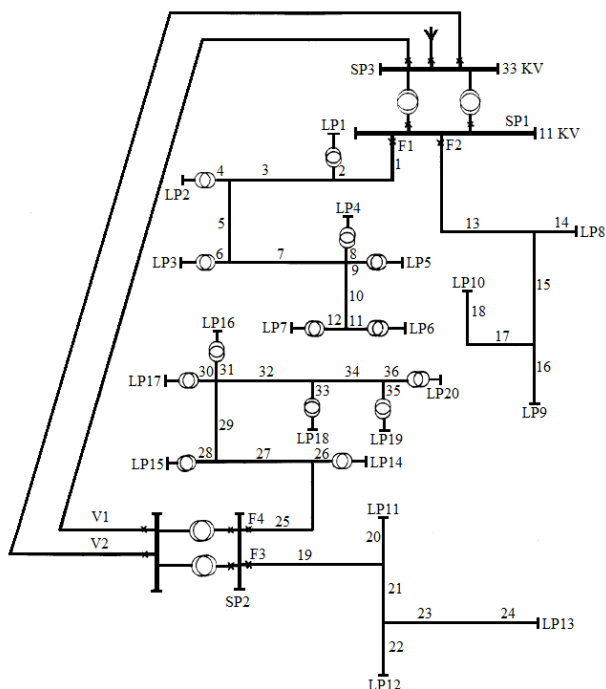
Generalno, ako se set slučajeva otkaza A_1 do A_n dovede na logičko ILI kolo sa n ulaza, onda se verovatnoća otkaza na izlazu računa na sledeći način:

$$P_{izlaza} = P(A_1 \cup A_2 \cup A_3 \dots \cup A_n) = P(A_1) + P(A_2) + P(A_3) + \dots + P(A_n). \quad (14)$$

Na osnovu prethodnog izraza može se zaključiti da je kod logičkog ILI kola verovatnoća otkaza jednaka sumi verovatnoća otkaza na ulazu.

4. PRIMENA

Metoda dijagrama stabla otkaza je primenjena na test distributivnu mrežu sa Slike 3. Odgovarajući parametri i podaci su dati u [2].



Slika 3. Test distributivna mreža

Rezultati analize su dobijeni uz uvažavanje sledećih pretpostavki:

- Struktura mreže je radijalna.
- Osigurači i rastavljajući su 100% pouzdani, odnosno njihova nepouzdanost je zanemarljiva.
- Zanemaruje se nepouzdanost napajanja sabirnica SP3.
- Na početku svake sekcije se nalazi rastavljaj.

Da bi se odredila pouzdanost napajanja pojedinačnih potrošača, izračunate su stope otkaza (λ) svake komponente distributivne mreže. Na osnovu tih vrednosti,

izvršen je proračun pouzdanosti napajanja (verovatnoće otkaza) pojedinačnih potrošača razmatrane test distributivne mreže. Rezultati su prikazani u Tabeli 1.

Tabela 1. Verovatnoća otkaza pojedinačnih potrošačkih čvorova razmatrane test mreže

potrošači	sekcije na putanji	P
LP1	1, 2	0,076539
LP2	1,3,4	0,113872
LP3	1, 3, 5, 6	0,139526
LP4	1,3,5,7, 8	0,176859
LP5	1,3,5,7, 9	0,174920
LP6	1,3,5,7, 10, 11	0,200573
LP7	1,3,5,7, 10, 12	0,198635
LP8	13, 14	0,063588
LP9	13, 15, 16	0,100921
LP10	13, 15, 17, 18	0,126575
LP11	19, 20	0,098249
LP12	19, 21, 22	0,135582
LP13	19, 21, 23, 24	0,161236
LP14	25, 26	0,120918
LP15	25, 27, 28	0,144632
LP16	25, 27, 29, 31	0,166408
LP17	25, 27, 29, 30	0,174187
LP18	25, 27, 29, 32, 33	0,195962
LP19	25, 27, 29, 32, 34, 35	0,233296
LP20	25, 27, 29, 32, 34, 36	0,225516

Razmatrana test distributivna mreža se sastoji od komponenti koje su povezane serijski (radijalni sistem napajanja), tako da se pouzdanost napajanja svakog potrošača smanjuje sa povećanjem broja komponenti na putanji do tačke napajanja.

5. ZAKLJUČAK

U radu je opisan način proračuna pouzdanosti napajanja potrošača priključenih na distributivnu mrežu metodom dijagrama stabla otkaza. Za primenu ove metodologije neophodno je potpuno poznavanje sistema koji se analizira, njegovih sastavnih elemenata, njihove povezanosti, kao i vrednosti odgovarajućih parametara.

Dijagram stabla otkaza pruža uvid u uticaj svake komponente distributivne mreže na pouzdanost napajanja potrošača. Razmatrana metoda omogućava jednostavnu analizu pouzdanosti složenih sistema sa radijalnom strukturom. Zasniva se na računanju ukupne stope pouzdanosti putanje napajanja potrošača do izvora napajanja. Pouzdanost sistema se povećava dodavanjem paralelnih komponenti, ugradnjom komponenti sa većom stopom pouzdanosti (kvalitetnijih komponenti), kao i obezbeđenjem alternativne putanje napajanja.

Metoda opisana u ovom radu omogućuje donošenje objektivnih odluka u analizi pouzdanosti, čime se izbegava prosuđivanje na osnovu intuicije, koje je ponekad neopravdano prisutno u praksi. Pored toga, analizirana metoda omogućava utvrđivanje osnovnih faktora koji doprinose lošoj pouzdanosti, kao i alternativnih rešenja, što može biti od velikog praktičnog značaja.

6. LITERATURA

- [1] Ali A. Chowdhury, Don O. Koval: *Power Distribution System Reliability, Practical Methods and Applications*, John Wiley and Sons, Hoboken, New Jersey, USA, 2009.
- [2] R.N.Allan, R.Billinton, I.Sjarief, L.Goel, K.S.So: "*IEEE Transactions on Power Systems*", Vo1.6, No. 2, May 1991, pp. 813-820.

Kratka biografija:



Nikola Žilić rođen je u Sremskoj Mitrovici 1988. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – smer Elektroenergetski sistemi odbranio je 2016. godine.



Savo Đukić rođen je u Novom Sadu 1983. godine. Diplomirao je i doktorirao na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – smer Elektroenergetski sistemi 2007. i 2014. godine, respektivno.

UPRAVLJAČKI SISTEM FORMIRAČICE KARTONSKIH KUTIJA CONTROL SYSTEM OF THE TRAYFORMING MACHINE

Dušan Stanković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj - Predmet ovog rada jeste analiza upravljačkog sistema mašine koja formira kartonsku ambalažu za industriju mleka zasnovanom na Siemens S7 300 CPU 315 2PN/DP programabilno logičkom kontroleru.

Abstract – Subject of this work is analysis of the control system for trayforming machine in milk industry, based on Siemens S7 300 CPU 315 2PN/DP programmable logic controller.

Cljučne reči: Upravljanje.

1. UVOD

Industrijska automatizacija inkorporira programabilno logičke kontrolere u proces proizvodnje. Programabilno logički kontroleri koriste procesne sisteme koji dozvoljavaju varijacije kontrole ulaza i izlaza koristeći jednostavno programiranje. Koristeći jezike zasnovane na logici, programabilno logički kontroler može da primi veliki broj ulaza i isto tako da vrati niz logičnih izlaza, ulazni uređaji mogu biti senzori a izlazni motori, ventili itd. Programabilno logički kontroleri su slični kompjuterima, međutim dok su kompjuteri optimizovani za kalkulacije, kontroleri su optimizovani za zadatke upravljanja u industrijskom okruženju. Najveća prednost programabilno logičkih kontrolera je njihova fleksibilnost, sa istim kontrolerom moguće je upravljati sa nizom upravljačkih sistema, naravno u domenu njegovih mogućnosti. Ova fleksibilnost vodi ka isplativom sistemu za složena i raznovrsna upravljanja [1].

2. MEHANIČKI OPIS SISTEMA

Mašina za formiranje kartonskih ambalaža dizajnirana je da formira kartonske kutije, koje se koriste za transport i izlaganje u policama, raznih tipova plastičnih posuda u industriji hrane (čase od jogurta, čase od želea, čase od krem sira itd). Na slici 1 je prikazana mašina.

Sastoji se od sledećih stanica:

1. Magacin kartona.
2. Lančani transporter
3. Stanica za formiranje kutije
4. Stanica za savijanje unutrašnjih krilaca
5. Stanica za savijanje spoljašnjih krilaca
6. Stanica za krajnje formiranje kutije
7. Među stanice za nanošenje lepka.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Rajs, docent.



Slika 1. Mašina za formiranje kartonskih kutija

2.1 Opis rada mašine

Ravan karton prethodno presečen, perforisan na odgovarajućim pozicijama i složen se izvlači iz magacina kartona i transportuje pomoću lančanog transportera do elementa za oblikovanje, koji gura karton kroz alat za formiranje.

Tokom transportovanja kartona od magacina do alata za formiranje nanosi se vruć topljivi lepak na krajeve kartona da bi se obezbedilo formiranje kutije. Nakon formiranja na stanici 3 i postavljanja osnovnog oblika kutije, ona se transportuje na sledeću stanicu koja obezbeđuje savijanje unutrašnjih krilaca. Na sledećoj stanici se vrši savijanje spoljašnjih krilaca kutije.

Tokom transportovanja do poslednje stanice nanosi se lepak za krajnje formiranje kutije. Na stanici za krajnje formiranje vrši se fiksiranje unutrašnjih i spoljašnjih krilaca za dno kutije.

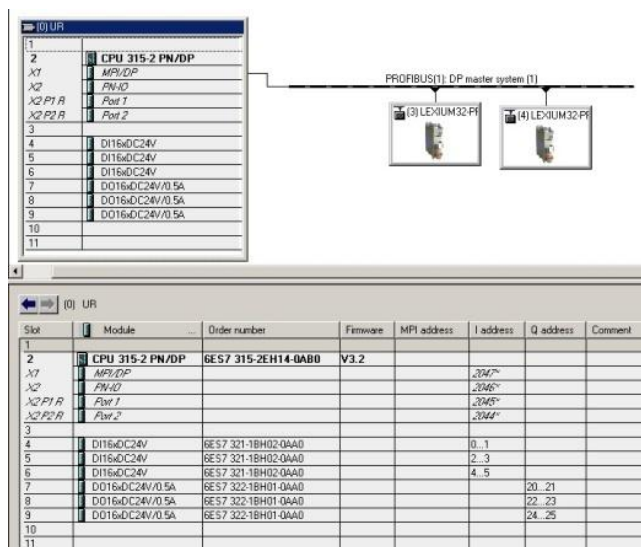
Potom se kutija transportuje na izlazni transporter gde se odvodi na sledeći proces u proizvodnji (stavljanje posuda u nju, obmotavanje folijom u termo tunelu itd). Projektovano ciklusno vreme je 3 sekunde ili ti 20 kutija u minuti.

3. HARDVER

Upravljački sistem je zasnovan na programabilno logičkom kontroleru S7-300 CPU 315-2-PN/DP proizvođača Siemens. Za upravljanje sinhronim servo motorima koriste se dva servo pretvarača Lexium 32 proizvođača Schneider Electric. Kao komunikacioni protokol između programabilno logičkog kontrolera i servo pretvarača koristi se Profibus-DP. Pneumatski cilindri su upravljani standardnim pneumatskim razvodnicima 5/3 proizvođača SMC. Razvodnici poseduju dva solenoida koji se aktiviraju standardnim digitalnim izlazima (PNP 24V DC). Izlaznih kartica ima tri od kojih

svaka ima 16 izlaza. Za dobijanje informacija sa senzora koristi se standardne ulazne kartice kojih takođe ima tri sa po 16 ulaza na sebi.

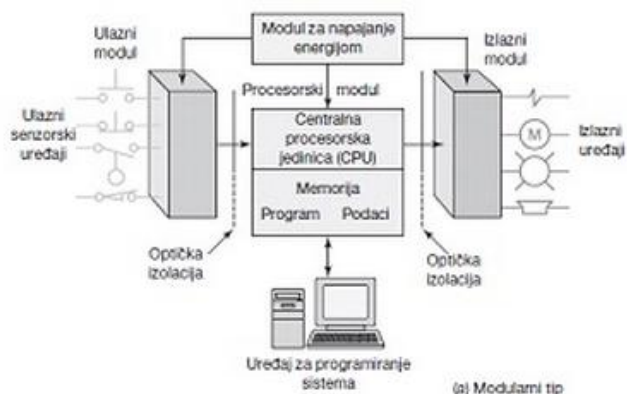
Ekran osjetljiv na dodir Magelis Schneider Electric se koristi kao operatorski panel. Komunikacioni protokol između ekrana i programabilno logičkog kontrolera je Profinet. Na slici 2 je prikazana hardverska konfiguracija upravljačkog sistema.



Slika 2. Hardverska konfiguracija

3.1 Arhitektura PLK

Tipičan PLK sistem sastoji se od više delova, kao što je prikazano na slici 3. To su: centralna procesorska jedinica (CPJ), ulazne/izlazne komponente (U/I) i napajanja. Izraz arhitektura može se odnositi na hardver PLK, na softver PLK ili na neku njihovu kombinaciju [2].



Slika 3. PLK sistem

3.2 Programabilno logički kontroler S7-300 CPU 315-PN/DP Siemens

Programabilno logički kontroler S7-300 se sastoji od centralno procesorske jedinice CPU 315-PN/DP modularnih ulazno, izlaznih kartica koje se dodaju na rek kontrolera. Sam CPU je procesor sa srednjom veličinom programske memorije.

Koristi se u postrojenjima koja sadrže distribuirane strukture i centralizovane I/O. Može biti implementiran kao Profinet I/O kontroler i kao standardni Profibus DP master. Takođe se može koristiti kao inteligentni

distribuirani modul (Profibus DP ili Profinet slave). Idealna je platforma za upravljanje kretanjem [3].

3.2.1 Ulazni i izlazni moduli

Kao ulazna kartica koristi se 6ES7 321-1BA02-0AA0 proizvođača Siemens koja na sebi ima 16 ulaza koji su optički izolovani.

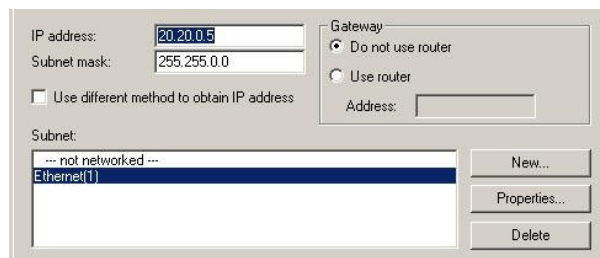
Kao izlazna kartica koristi se 6ES7 322-1BH01-0AA0 proizvođača Siemens. Na sebi ima 16 izlaza koji su optički izolovani izlazne struje 0,5A. Na slici 2 je prikazana i topologija mreže.

3.2.2 Profibus DP

Kao što je rečeno, na Profibus mreži se nalaze dva servo pretvarača. Baud rejt mreže je podešen na 1,5Mbps, adresa PLK je 1 a pretvarača 3 i 4.

3.2.3 Profinet

Svaki uređaj sa Profinet portom na sebi poseduje IP i MAC adresu. Za alokaciju IP adrese i Subnet maske dostupna su dva metoda DCP i DHCP. Na slici 4 je prikazana konfiguracija IP adrese Profinet porta.



Slika 4. Konfiguracija IP adrese

3.3 Operatorski panel

Operatorski panel koji je korišćen na ovoj mašini je HMI GTO 3510 proizvođača Schneider Electric. Ima dijagonalu ekrana od 7", od interfejsa poseduje USB port koji je servisnog tipa, RS485 i Ethernet port za komunikaciju sa programabilno logičkim kontrolerom preko više vrsta protokola.

3.4 Odabir sinhronih servo motora i pretvarača

Sinhroni servo motori sa permanentnim rotorskim magnetom dizajnirani su za aplikacije sa visokim zahtevima po pitanju brzine i preciznosti.

Upravljanje servo motorom vrši servo pretvarač (drajver). Pretvarač može biti kontrolisan digitalnim i analognim signalima, ali je već dosta rasprostranjena praksa da pretvarač bude vođen preko nekog komunikacionog protokola (Profibus DP, CanOpen, Ethernet IP)

Najkritičniji faktori u odabiru motora mogu biti:

- maksimalni moment
- brzina
- odnos inercija
- Srednji moment

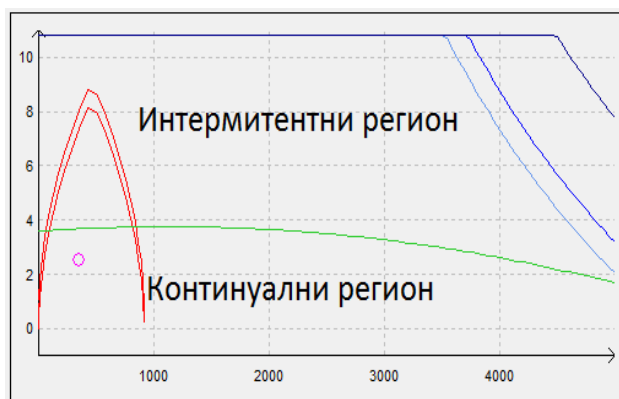
3.4.1 Maksimalni moment

Moment na početku trapeznog profila je maksimalni moment jer pokriva mehanička trenja i ubrzanje masa. Ova tačka profila je nazvana maksimalnim momentom.

3.4.2 Brzina

Maksimalna brzina je data kao krajnja desna tačka krive moment brzina, a maksimalni moment kao vrh krive. Kriva ima dva regiona kontinualni i intermitentni. Ako je kombinacija moment brzina zahtevana od aplikacije, mora da se uvek nađe u kontinualnom području.

Ako je kombinacija u intermitentnom regionu motor može raditi samo u ograničenom vremenu. U suprotnom počeo bi da se pregreva. (Slika 5)



Slika 5. Moment-brzina kriva

3.4.3 Profil kretanja

Zahtevi aplikacije su najbolje opisani profilom kretanja i profilom momenta.

3.4.4 Odnos inercija

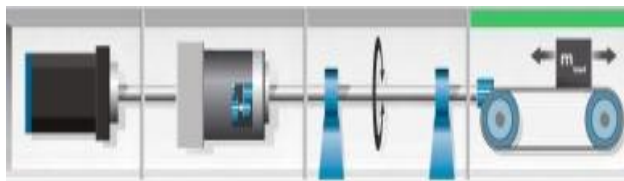
Moment inercije sistema može biti podeljen u dva dela: moment inercije motora i moment inercije opterećenja. Odnos inercije je tipično 5:1, sistem se poboljšava ukoliko je odnos manji.

3.4.5 Srednji moment

Ovaj moment predstavlja sredinu momenta kompletnog mašinskog ciklusa, ili ekvivalent momentu stalnog stanja.

3.4.6 Dimenzionisanje motora i pretvarača

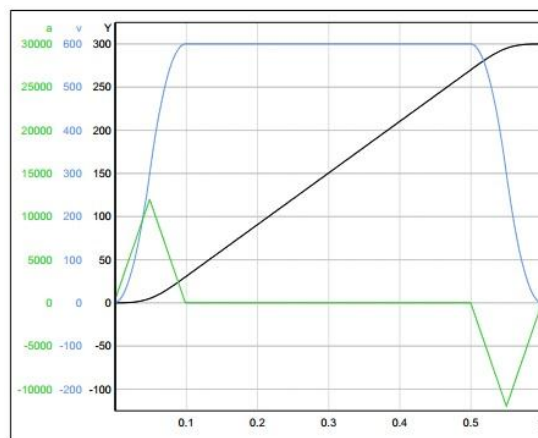
Metod odabira se objašnjava na primeru motora lančanog transportera. Na osnovu zadate mehanike postavljamo model sistema. Na slici 6 je prikazan model.



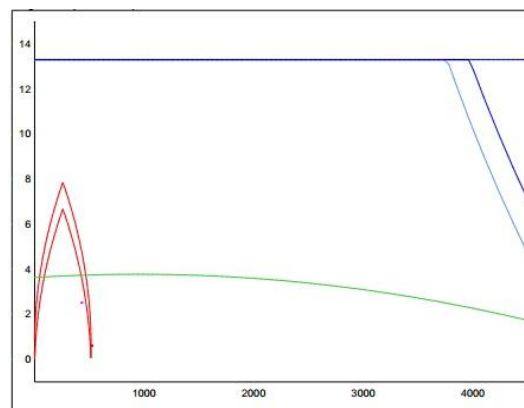
Slika 6. Mehanički model

Potrebno je bilo ostvariti pomeranje lanca od 300mm za vreme manje od 1s, brzinom ne većom od 500mm/s. Na slici 7 prikazan je s profil kretanja gde se za vreme manje od 1s dostiže pređeni put lanca od 300mm.

Uzimajući u obzir cenu, odnos momenata inercija, nominalne struje pretvarača, veličine motora i reduktora izabran je sistem čiji je moment brzina dijagram prikazan na slici 8.



Slika 7. Profil kretanja



Slika 8. Moment brzina dijagram

Sa slike se može zaključiti da je maksimalni moment u intermitentnom režimu ali u ograničenom periodu, a srednji moment u kontinualnom, što čini ovu konfiguraciju prihvatljivom. Na osnovu ovog modelovanja odabran je motor BMH1002 reduktor GBX080009 i servo pretvarač LXM32AD12N4 proizvođača Schneider electric.

4. Softver

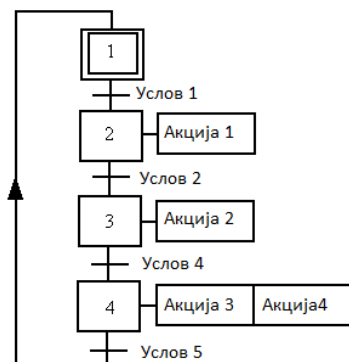
Za programiranje Siemens S7-300 kontrolera koristi se Simatic Step 7 programsko okruženje. Programski paket Simatic Step 7, omogućava korisnicima više programskih editora za pisanje programa. Kao što su LD (Ladder diagram-lestvičasti dijagram), IL (Instruction list- lista instrukcija) i FBD (Function block diagram- funkcijski blok dijagram) sve u skladu sa standardom IEC 61311. [4]

4.1 Organizacija programa

Step 7 softver dozvoljava struktuiranje programa, drugim rečima podela programa u individualne blokove. Skladno ovome program je podeljen na funkcije koje upravljaju delovima mašine kao što su transporter (Transporter), magacin (MagazineInfeed), formirač kutija (Erector), nanošenje lepka (HotMelt), krajnje formiranje (Presser) i upravljačkim funkcijama kao što su I/O, Ctrl i AlarmFC [5].

4.2 Upravljački algoritam

Upravljački algoritam je baziran na GRAFCET-u (graf stanja i prelaza). Za prelazak iz svakog stanja u sledeće potrebno je imati uslov, za svako stanje se vezuje neka akcija (Slika 9).



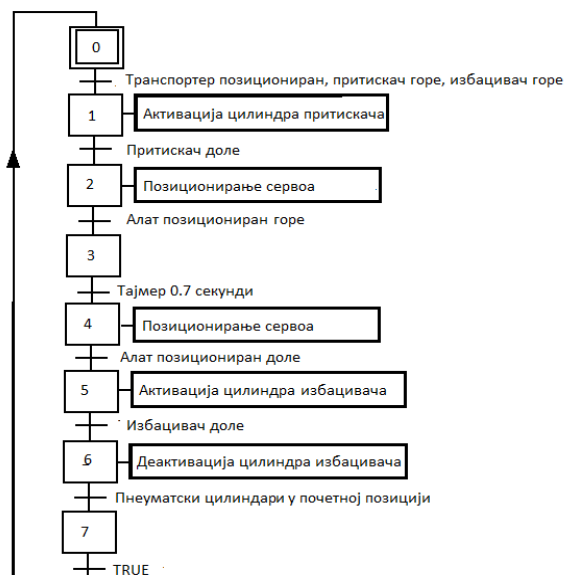
Slika 9. primer GRAFCET-a

4.2.1 Implementacija GRAFCET-a

U LD dijagramu je GRAFCET podeljen na tri zone, zonu uslova, zonu prelaza i zonu izvršenja. U zoni uslova se dovode uslovi za prelazak u sledeće stanje, u zoni prelaza se prelazi na sledeće stanje ukoliko je ispunjen odgovarajući uslov. Zona izvršenja služi za izvršenje odgovarajućih akcija koja su za dato stanje određena.

4.2.2 GRAFCET alata za formiranje kutije

Primer GRAFCETA dela mašine za formiranje kutije (Slika 10)



Slika 10. Algoritam rada alata za formiranje

4.2.3 Upravljanje servo pretvaračima, PLCopen koncept

PLCopen motion upravljanje nudi standardne funkcijske blokove za jednostavno i napredno upravljanje, bazirane na IEC 61131 jezicima. Sledeći blokovi su implementirani kao upravljanje: MC Power, MC Home, MC Jog, MC MoveAbsolute, MC MoveRelative itd.

4.3 Operatorski panel, opis ekrana

Svaki deo mehanike je prezentovan kroz jedan ekran (magacin, alat za formiranje, alati za savijanje, transporter, među stanice za lepljenje, alat za krajnje formiranje itd) na kojima se nalaze parametri za te mehaničke sklopove i ručne komande. Pored njih postoje ekrani za podešavanje režima rada, receptura i alarma.

5. Testiranje i puštanje u rad

Pre stavljanja pod napon testira se omksi svaka veza na panelu. Nakon provere šemiranja pristupa se postepenom stavljanju pod napon. Testiranje u polju mašine se svodi na testiranje svih servo motora, senzorskih razvoda, kao i veze sa pneumatskim razvodima. Proverava se stanje svakog digitalnog ulaza i izlaza. Nakon kreiranja I/O liste i testiranju ručnih komandi pristupa se parametrisanju servo pretvarača i njihovom testiranju i tjuniranju. Testiraju se ručne komande servo motora, pa proveravaju sekvence korak po korak bez proizvoda. Testiranje sa proizvodom i korigovanja mehanike kao i sekvence. Uspostavljanje stabilnog ciklusa od 3s po kutiji.

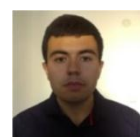
6. Zaključak

U ovom radu analiziran je jedan industrijski upravljački sistem. Prednosti jednog ovakvog sistema su brojne. Održavanje i dijagnostika kvara su pojednostavljeni. Upravljanje servo pretvaračima u zatvorenoj sprezi pokazalo se kao pravo rešenje za ovaj sistem. Jednostavnu integraciju servo pretvarača u sistem upravljanja treba posebno istaći. Iako različitih proizvođača nije bilo problema u njihovoj konfiguraciji. Vrlo često takva rešenja znaju da prave ogromne probleme prilikom puštanja u rad. Prostor za unapređenje je u dodavanju više senzora za različite detekcije ispadanja kartona. Mašina je integrisana u pogon mlekare u Ljubljani. Postignuto je ciklusno vreme od 3s po kutiji tj 20 kutija u minuti.

7. LITERATURA

- [1]https://en.wikipedia.org/wiki/Industrial_control_system, (datum pristupa: 2016-09-10)
- [2]http://www.mikroknjiga.rs/Knjige/PLC/01_PLC.pdf, (datum pristupa: 2016-09-10)
- [3]<https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10026479>, (datum pristupa: 2013-09-10)
- [4]<https://en.wikipedia.org/wiki/Profibus>, (datum pristupa: 2013-09-10)
- [5]http://www.automation.com/pdf_articles/IEC_Programming_Thayer_L.pdf, (datum pristupa: 2016-10-10)

Kratka biografija:



Dušan Stanković rođen je u Beloj Crkvi 1990. godine. Osnovne akademske studije na smeru Mehatronika - Mehatronika, robotika i automatizacija završio je 2013. godine sa prosečnom ocenom 9,51. Master rad na Fakultetu tehničkih nauka iz oblasti Mehatronika - Mehatronika, robotika i automatizacija odbranio je 2016.

AUTOMATIZACIJA POSTROJENJA MINI PIVARA**AUTOMATION OF MINI BREWERY PLANT**Nikola Bukvić, *Fakultet tehničkih nauka, Novi Sad***Oblast – MEHATRONIKA**

Kratak sadržaj – Ovaj rad se bavi problematikom automatizacije postrojenja za proizvodnju piva u minimiziranim uslovima (mini pivara) a što se ubraja u proizvodni pogon procesnog tipa. Prikazane su sličnosti i razlike između uobičajenog načina proizvodnje piva - u velikim postrojenjima (pivarama) i proizvodnje u minimiziranim uslovima (mini pivara), kako sa stanovišta tehnološkog postupka, tako i sa stanovišta razlike u stepenu automatizacije. Ukratko je opisan tehnološki postupak dobijanja piva kao prehrambenog proizvoda kroz pojedinačne faze, kako u velikim pivarama tako i u malim postrojenjima. U radu je opisano i jedno postrojenje mini pivare sa svim svojim elementima i mestom ugradnje u Apatinskoj pivari sa elementima visokog stepena automatizacije.

Abstract – This paper is related to the issue of automation of plants for the production of beer in minimal conditions (mini brewery) which is one of the manufacturing plant process type. Displayed are the similarities and differences between the traditional ways of brewing - in large plants (breweries) and production in minimal conditions (mini-breweries), both from the aspect of technological process, as well as from the aspect of differences referring to the degree of automation. Briefly it describes the technological process for the preparation of beer as a food product through the individual phases, as in the large breweries and in small plants. This paper describes one mini brewery plant with all its elements and place of installation in Apatin brewery with elements of high degree of automation.

Ključne reči: pivara, mini pivara, procesna oprema, automatizacija

1. UVOD

Prema drevnim iskopinama i zapisima još iz vremena kada se pisalo klinastim pismom na kamenu i glini, arheolozi su utvrdili da su stari Vavilonci proizvodili pivo u domaćinstvu još 7000 godina pre nove ere. Imali su šesnaest sorti piva a za kuvanje su koristili ječmeni slad i pšenicu. Tragovi zakonskih propisa, međutim, upućuju na zaključak da su još tada normama regulisali sadržaj ekstrakta i cenu piva. Oдавde se prenela proizvodnja piva u drevni Egipat, Persiju, Grčku i druge zemlje.

Egipćani su znali da prave pivo već 2000 godina pre nove ere.

U početku je pivo proizvedeno samo za kućnu upotrebu, ali sa razvojem trgovine ono postaje predmet razmene i proizvodi se u količinama koje premašuju potrebe jedne porodice. Započinje izdvajanja pivarstva kao samostalnog zanimanja. Dakle, tokom nekoliko milenijuma, putem neprekidnih zapažanja i istraživanja ljudi su naučili da ovladaju teškim procesima koji se odigravaju kod klijanja, kod enzimatskog oštećenja skroba (ukomljavanja) i kod fermentacije.

Ipak, bez obzira na to, i bez obzira što je pivo proizvedeno u velikim količinama, uspeh proizvođenja zavisi je od slučaja ili od veštine majstora, ili njegovog višegodišnjeg iskustva. Uloga mikroorganizama kod transformacija organskih materija, postala je poznata tek u devetnaestom veku. Tek naučnim istraživanjima je dokazano, da su pobuđivači fermentacije sladovine određene kvasne gljivice i da fermentacija predstavlja biokatalitički proces, koji se odigrava unutar ćelija kvasca. Započelo je proučavanje problema fermentacije na širokom frontu, što je uskoro dalo vrlo plodotvorne rezultate, kako u nauci, tako i u tehnici.

Visok nivo savremene tehnologije dostignut je zahvaljujući razviku kako tehničke mikrobiologije i biohemije, tako i energetike. Usavršavanje parne mašine veoma se povoljno odrazilo na razvoj pivarstva. Najvažniji tehnički pronalazak za pivarstvo predstavlja pronalazak mašine za hlađenje. Sve ovo dovodi do stvaranja uslova za pretvaranje usitnjenih preduzeća u velike mehanizovane fabrike.

Analizirajući razvoj pivarstva u svetu i kod nas, može se konstatovati da je ova grupacija sa relativno velikom tradicijom i iskusnim kadrovima, te da je od zanatske primitivne proizvodnje izrasla u savremenu industriju.

Kada se obuhvati sav potreban repromaterijal i sirovine potrebne za proizvodnju i pakovanje piva vidi se da ova industrija indirektno zapošljava velik broj stanovništva - od poljoprivrednika, trgovaca, ugostitelja, inženjera tehnologije, inženjera mašinstva, elektrotehnike, ekonomista, itd. [1]

Danas, nakon višedecenijske dominacije velikih proizvođača piva, pojavljuje se sve više zanatskih pivara koje za cilj imaju povratak na tradicionalan način proizvodnje piva (u malim šaržama uz potpuno pridržavanje pravila propisanih *Reinheitsgebote*¹-om) uz zadržavanje savremenih elemenata tehnike i tehnologije (automatizacija procesa rada).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Stevan Stankovski, red.prof.

¹ Reinheitsgebote – „Nemački zakon o čistoći piva“ je regulativa nastala u Nemačkoj 1516.godine kojom se propisuje da se pivo pravi od isključivo četiri osnovna sastojka – vode, ječmenog slada, hmelja i kvasca.

2. PROIZVODNJA PIVA

2.1 Osnovne karakteristike proizvodnje piva

U srednjoj Evropi, pod nazivom pivo podrazumeva se slabo alkoholno piće, koje se proizvodi u procesu alkoholnog vrenja iz slada, hmelja, vode i pivskog kvasca. Međutim, u Srbiji, kao i u većini drugih zemalja Evrope i sveta, pored upotrebe ovih sirovina dozvoljena je delimična zamena slada sa drugim nesladovanim sirovinama, kao što su: kukuruzna krupica, pirinač, šećer i sl. Količina nesladovanih žitarica ili njihovih prerađevina ograničena je procesom i željenim karakteristikama piva.

Kada se govori o osnovnim sirovinama za proizvodnju piva, onda je nužno objasniti njihovu pojedinačnu namenu. Voda je osnovni sastojak svih napitaka. Ona služi kao rastvarač. Slad daje pivu sastojke ekstrakta od kojih zavisi punoća ukusa i koncentracija osnovnog ekstrakta piva. Hmelj mu daje ugodan miris i gorak ukus a ima efekta na mikrobiološku stabilnost piva. Pivski kvasac izaziva alkoholno vrenje, u kojem šećer prelazi u alkohol i ugljendioksid.

Proizvodnja piva je složen i dugotrajan proces, koji se deli na tehnologiju slada i tehnologiju piva.

U radu će biti razmatrana samo tehnologija proizvodnje piva.

Tehnologija piva, ili pivarstvo, u užem smislu reči odlikuje se većim brojem tehnoloških faza, koje se mogu grupisati u sledeće: 1) proizvodnja sladovine; 2) glavno i naknadno vrenje; 3) filtracija i punjenje piva;

2.2. Kratak opis pojedinih faza tehnološkog procesa proizvodnje piva

Na osnovu prethodno navedene osnovne tri faze u proizvodnji piva, vitalnu opremu svake pivare čine: mlin za slad, komovnjak, kotao za kominu, bistrenik, prihvatna posuda za sladovinu (peti sud), kotao za kuvanje sladovine, virpul, hladnjak sladovine, aerator sladovine, fermentori (cilindrično-konusni sudovi, otvoreni sudovi itd.), sudovi za odležavanje, filter, tankovi bistrog piva, punjač boca, etiketirka, upakivač, paletizer [2].

1) Mlin za slad - ječmeni slad je pre upotrebe u varionici potrebno usitniti na mlinu za sladovinu. Slad se drobi (usitnjava) kako bi se stvorili bolji uslovi za ubrzavanje fizičkih i biohemijskih procesa rastvaranja sadržaja zrna slada u toku ukomljavaanja u cilju maksimalno mogućeg prevođenja ekstraktivnih materija u vodeni rastvor tj. sladovinu (sledeća faza u proizvodnji piva).

2) Komovnjak - služi za mešanje drobljenog slada sa toplom vodom pri čemu se dobija komina. Sastojci iz slada se enzimatskim reakcijama transformišu u rastvorne oblike. Ovo je cilindrična posuda sa ravnim ili sfernim dnom i sfernim poklopcem, na čijem se centru nalazi cev za odvod vodene pare. Takođe, da bi mešanje bilo bolje na dno komovnjaka se ugrađuje mešalica u vidu propelera sa pogonom ispod kotla. U komovnjaku se odvija proces transformacije složenih ugljenih hidrata, tj. proces prelaska skroba u prostije šećere (glukoza, fruktoza, saharoza, maltoza i maltotrioza) uz pomoć aktivacije i inaktivacije određenih enzima koji se pobuđuju na određenim temperaturama. Iz tog razloga je regulacija temperature u komovnjaku od ključalnog značaja. Provera razgradnje se

proverava reakcijom ošećerenja (proverava se da li ima nerazgrađenih polisaharida koji sa jodom daju plavu boju).

3) Kotao za kominu - ima sličnu namenu i konstrukciju kao i komovnjak sa tom razlikom što se u kotlu za kominu prvo vrši ošećerenje nesladovanih sirovina (kukuruzna krupica, ječam, pirinač, itd.), a potom dolazi do mešanja sa kominom nastalom od ošećerenog ječmenog slada u komovnjaku.

4) Bistrenik - služi za odvajanje sladovine i ispiranje ekstrakta iz tropa (čvrstog dela zrna) od koga na kraju ostaje isprani trop (treber) [3].

Bistrenik je posuda cilindričnog oblika sa ravnim dnom. Iznad glavnog dna postavlja se od segmenata filtraciono dno (sito). Ovo dno je na odstojanju od 10 do 15 mm od osnovnog dna i služi za filtraciju. Filtraciona sita postavljaju se na osnovno dno i oslanjaju se na nožice visine 12 do 15 mm, a svojim rubovima naležu na rubove posude. Za odvod sladovine sa dna bistrenika služe odvodne cevi koje su na kraju povezane na centralnu laternu². U bistreniku je na vertikalnoj osnovi postavljen uređaj za sečenje trebera sa noževima. Ispiranje tropa od zaostalog ekstrakta se vrši vodom koja se u bistrenik dovodi preko cevi do rasprsnih glava, čime se postiže veća kontaktna površina materijala (tropa i vode).

Da bi se pospešilo ceđenje sladovine i izlučivanje ekstrakta, uključuju se rotirajući noževi koji rahle(rasecaju) čitavu masu trebera. Na kraju ceđenja se automatski spušta letva za isterivanje trebera iz suda tako što se rotiranjem odguruje do otvora za prihvatanje trebera.

5) Prihvatna posuda za sladovinu - takozvani "peti sud" - služi kao akumulacioni rezervoar sladovine jer se ceđenje sladovine u bistreniku vrši postepeno u fazama, a zbog prirode narednog koraka (kuvanje sladovine u kotlu sladovine) potrebno je uvarak u celosti transportovati iz bistrenika u kotao sladovine.

6) Kotao sladovine - izrađen je na način da se obezbedi intenzivno kuvanje sladovine. Zadatak kuvanja sladovine je između ostalog i rastvaranje gorkih sastojaka hmelja koji se dodaju u ovoj fazi proizvodnje. U novije vreme se vrši kuvanje pod pritiskom što dovodi do skraćivanja vremena trajanja samog kuvanja. Da bi se dobilo intenzivno kuvanje u centru cilindričnog suda se postavlja parni plašt ili se ugrađuje interni bojler. Na ovaj način, pošto je u sredini kotla temperatura viša nego na periferiji dolazi do konvencionalnih strujanja [5].

7) Virpul - predstavlja jednostavan cilindrični sud u koji se kuvana sladovina doprema tangencionalno dovedenom cevi iz kotla sladovine. Na ovaj način se postiže takozvani "virpul efekat" (efekat vira) gde se usled centripetalnih sila u centru suda u vidu kupe talože preostale čvrste čestice sladovine.

8) Hladnjak sladovine - najčešće je izrađen u vidu pločastog izmenjivača toplote. Pločasti izmenjivači toplote se sastoje od dve sekcije - u prvoj sekciji se sladovina hladi sa bunarskom vodom, dok se u drugoj sekciji hlađenje vrši sa smešom glikol-voda. Cilj rashlađivanja sladovine jeste prilagođavanje temperature koja je adekvatna za tzv. zasađivanje kvasca tj. za alkoholno vrenje, odnosno početak alkoholne fermentacije.

² Laterna – kupola (tank) cilindričnog oblika sa revizionim staklenim otvorom na vrhu.

9) Aerator sladovine - predstavlja uređaj za obogaćivanje sladovine, a može se vršiti ubacivanjem sterilnog vazduha ili kiseonika. Konstrukcija ovog uređaja je jednostavna - dovod vazduha klase 4.4.4.³ uronjen u vod sladovine a koji je višeg pritiska od pritiska sladovine u posmatranom vodu. Aeracijom sladovine se stvaraju povoljniji uslovi za rast i razvoj kvasca u sladovini pa samim tim se i poboljšava aktivnost kvasca tokom alkoholnog vrenja.

10) Fermentori (cilindrično-konusni sudovi) - služe za alkoholno vrenje. U njih se sladovina doprema iz virpula preko hladnjaka sladovine, a potom se u sladovinu dodaju prethodno odmerene količine kvasca koje su potrebne za uspešno alkoholno vrenje. U procesu fermentacije tj. alkoholnog vrenja dolazi do oslobađanja ugljen-dioksida usled hranjenja kvasca prostim šećerima iz sladovine a koji se dalje rekuperiše. Čelije kvasca se nakon procesa fermentacije talože na dno koje je u vidu konusa kako bi se lakše odstranili iz suda. Pivo se nakon obavele fermentacije hladi, i priprema za narednu fazu - maturaciju.

11) Nakon fermentacije, pivo odlazi u tankove za maturaciju tj. odležavanje gde se vrši dalji proces finalizacije piva (biohemijske reakcije se i dalje odvijaju ali sporije).

12) Filter - nakon odležavanja piva, potrebno ga je filtrirati kako bi se izbistrilo. Filtracija se vrši propuštanjem piva kroz razne vrste medijuma. Najčešće je u upotrebi takozvani kiselgur filter, tj. medijum sačinjen od dijatomejske zemlje⁴. Na ovaj način se iz piva izdvajaju zaostale čelije kvasca, kao i ostaci hladnog taloga nastalog u procesu odležavanja.

13) Tankovi za bistro pivo - služe kao akumulacioni tankovi filtriranog piva spremnog za otakanje u boce.

14) Punjač - uređaj za istakanje piva iz sudova za skladištenje filtriranog (bistrog) piva u određenu ambalažu (staklenu, PET⁵, limenke, burad itd.).

U ovom radu će se autor baviti temama samo vezanim za prethodno navedene procese rada u tehnološkom postupku dobijanja piva, kako u velikim postrojenjima tako i u minimiziranim postrojenjima (mini pivarama), dok će uređaji za etiketiranje, pakovanje i skladištenje gotovog proizvoda biti zanemareni.

2.3. Osnovne razlike između velikih postrojenja i minimiziranih postrojenja (mini pivara)

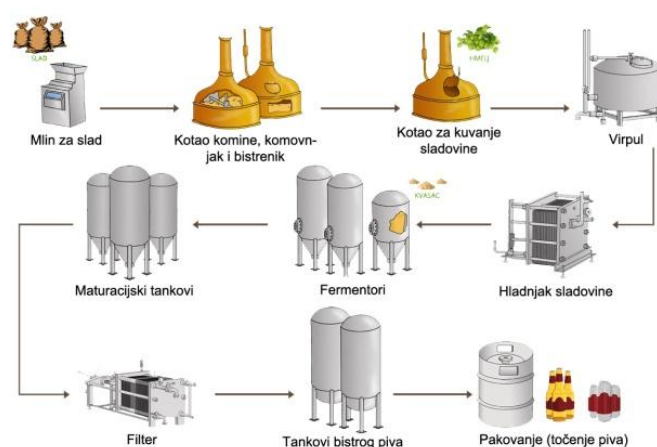
Kako je gore naveden osnovni tehnološki postupak proizvodnje piva, može se zaključiti da se radi o jednom veoma složenom procesu i da pojedine operacije nije moguće zaobići. Iz tog razloga, zbog prirode procesa dobijanja piva, minimizirana postrojenja pivara (mini pivara) se ne razlikuje suštinski od velikih postrojenja. Velike količine proizvodnje u velikim postrojenjima dovodi do potreba za izradom i korišćenjem sudova

velikih zapremina, većeg broja glavnih i pomoćnih sudova kao i pomoćnih uređaja za transport, odvajanje i doziranje sirovina i poluproizvoda namenjenih za velike količine sa kojima se vrše određene operacije (spiralni transportni puževi, elevatori, mlinovi slada za mokro mlevenje, pneumatski transportni sistemi, itd.).

Međutim, u malim postrojenjima se radi uštede prostora, a usled manjeg vremenskog ograničenja, praktikuje upotreba manjeg broja sudova i jednostavnijih komponenti. Ali, kao i u velikim postrojenjima, potrebno je obezbediti energiju za potrebe grejanja i hlađenja, električnu energiju potrebnu za napajanje mernih i regulacionih komponenti, kao i tehničku vodu i osnovne sirovine za potrebe pravljenja piva.

Ovim se potvrđuje gore pomenuta potreba za prolaskom kroz sve faze proizvodnje velikih postrojenja u uslovima minimiziranog postrojenja (mini pivare).

Na slici 2.1. prikazan je osnovna tehnološka shema proizvodnog procesa u pivarskoj industriji.



Slika 2.1. Shematski prikaz proizvodnog procesa piva

3. IZRADA IDEJNOG REŠENJA AUTOMATIZOVANE MINI PIVARE

Pod pojmom "mini pivara" se u Srbiji podrazumevaju i pivare koje se u drugim zemljama klasifikuju i kao „mikro“ ili „nano“ pivare a što predstavlja postrojenja koja na godišnjem nivou proizvode od 1500 hl (hektolitara) do 5000 hl piva.

3.1. Idejno rešenje automatizovane mini pivare

Pred autora rada je postavljen zadatak konstruisanja i izrade automatizovane mini pivare što obuhvata detaljnu analizu proizvodnje piva u velikim proizvodnim sistemima, a potom primena stečenih znanja i iskustava u minimizaciji tog procesa u cilju izrade didaktičkog modela.

Rešenjem postavljenog zadatka su prikazani: odabir godišnjih kapaciteta, proračun zapremina sudova za ispunjenje godišnjih kapaciteta, odabir mernih, regulacionih i upravljačkih komponenti (merača temperature, protoka, konduktiviteta, PLC-a⁶, invertora, elektromotora, reduktora, pumpi, ventila), izrada procesnih sudova i ugradnja armatura, povezivanje, programiranje i puštanje u rad sistema.

Na slici 3.1. prikazan je 3D model idejnog rešenja automatizovane mini pivare.

³ 4.4.4. kvalitet komprimovanog vazduha – maksimalne veličine od 15μm čvrstih čestica, maksimalne koncentracije od 8 mg/m³, koncentracije ulja do 5 mg/m³ [6].

⁴ Kiselgur (dijatomejska zemlja) - Dijatomejska zemlja (poznata i kao kremena zemlja) je sitni pijesak napravljen od silikatne sedimente stijene koja je nastala taloženjem ostataka dijatomeja. Dijatomeje su mikroskopske jednostanične vodene biljke poznate još i kao alge kremenjašice.

⁵ PET - Polyethylene terephthalate.

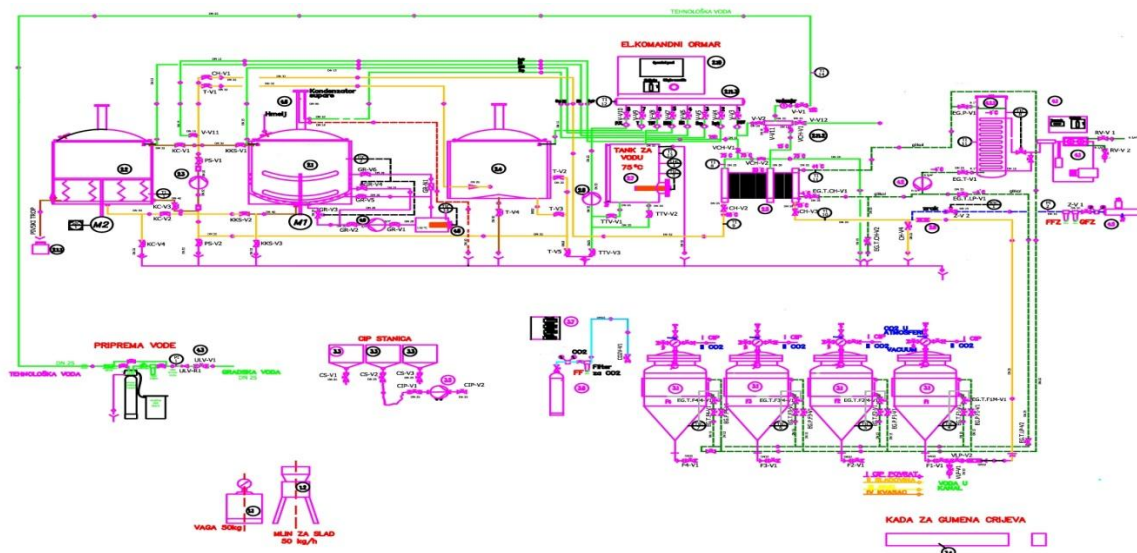
⁶ PLC - Programmable Logic Controller



Slika 3.1. 3D model idejnog rešenja mini pivare

3.2 Automatizacija mini pivare

Pod pojmom automatizacije mini pivare podrazumeva se automatizovan prelazak iz jedne u drugu fazu procesa proizvodnje piva a na osnovu parametara koje beleže merni uređaji upoređujući ih sa prethodno zadatim vrednostima na osnovu kreiranog i odabranog recepta piva.



Slika 3.2 - Shematski prikaz idejnog rešenja automatizovane mini pivare

5. ZAKLJUČAK

Uopšteno govoreći o proizvodnji piva, primećuje se da se radi o izuzetno komplikovanom tehnološkom procesu. U velikim proizvodnim sistemima (pivarama) automatizacija ovih procesa predstavlja veliki izazov i za iskusne inženjere. U malim postrojenjima o kakvim se govori u ovom radu stepen automatizacije je značajno niži ali takođe predstavlja velik izazov s obzirom da se sama tehnologija dobijanja piva ne može značajno izmeniti i minimizovati. Minimizacija se vrši u gabaritima osnovnih i pomoćnih sudova kao i korišćenjem manjeg broja procesnih komponenti (merača protoka, pritiska, temperature, konduktiviteta tj. provodljivosti, regulacionih ventila, itd.), dok koraci tj. redosled operacija i broj operacija u tehnološkom postupku ostaju nepromenjeni. Autor je radom obuhvatio konstrukciju, odabir komponenti i izradu kompletnog postrojenja mini pivare a potom u saradnji sa tehnologom iz oblasti proizvodnje piva automatizovao veći deo operacija.

Međutim, postoji niz operacija koje je ili neizvodljivo automatizovati u trenutnim uslovima ili ne postoji finansijska opravdanost za taj poduhvat.

Automatizacijom u ovom radu su obuhvaćene sledeće aktivnosti: automatsko odmeravanje masene količine ječmenog slada za jedan usipak, odmeravanje količine vode za potrebe ukomljavanja, regulacija temperatura i vremena pauza u fazi ukomljavanja, bistrenje piva u bistreniku, ceđenje, regulacija temperature i praćenje vremena trajanja kuvanja sladovine u kotlu sladovine, prebacivanje sladovine preko izmenjivača toplote u fermentor, aeracija sladovine, regulacija temperature u fermentorima i maturacijskim tankovima, CIP sistema.

Automatizacijom nisu obuhvaćene sledeće aktivnosti: vađenje trebera (troja) iz bistrenika, dodavanje hmelja u kotao sladovine, odmeravanje i dodavanje ćelija kvasca u fermentore, točenje piva iz maturacijskog suda u boce.

Pored svega navedenog može se primetiti da se radi o minimiziranom postrojenju sa visokim stepenom automatizacije.

6. LITERATURA

- [1] Semir dr ing. Mahmud (1979), "Tehnologija piva",
- [2] K. Stefanić, V.Marić (1990), "Pivarski priručnik"
- [3] Dipl.piv.inž. Tibor Koša (1974), "Pivarski priručnik"
- [4] Wolfgang Kunze (2014), "Technology Brewing & Malting"
- [5] Dragan Šešlija (2012), Materijal za pripremu ispita iz predmeta "Automatizacija procesa rada"

Kratka biografija:



Nikola Bukvić rođen je u Somboru 1990 god. Upisuje osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu 2009. godine, na smeru Mehatronika. Master akademske studije na Fakultetu tehničkih nauka upisuje 2013. godine, na smeru Mehatronika, robotika i automatizacija Master rad odbranio je 2016 god.

**KINESTETIČKO OBUČAVANJE ČETVORONOŽNOG ROBOTA KOJI SE UPRAVLJA
NA OSNOVU MOMENATA U ZGLOBOVIMA****KINESTHETIC TEACHING FOR A TORQUE CONTROLLABLE QUADRUPEDAL
ROBOT**

Hana Gostimir, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – Ovaj rad se bavi kinestetičkim obučavanjem četvoronožnog robota ANYmal koji se upravlja na osnovu momenata u zglobovima, gde se treba pobrinuti za kontrolu i stabilnost tela robota. Razvijena je aplikacija, koja u simulaciji opisuje spoljašnju silu primenjenu na telo robota, a dobijenu od strane korisnika (preko 3D miša ili 3D interaktivnog markera). Kontroler koji vrši kompenzaciju gravitacije primenjen je u fazi obučavanja, da bi se omogućilo vođenje noge koju želimo obučiti, bez napona u zglobovima, kao i kontrola baze tela robota. Obučavanje se obavlja putem snimanja keyframe-ova, koji mogu biti realizovani pomoću već postojećih kontrolera. Rezultat rada pokazan je eksperimentima u simulatoru, kao i jednostavnim eksperimentom na pravom robotu.

Abstract – This paper presents the use of kinesthetic teaching for a torque controllable quadrupedal robot ANYmal, where the control and stability of the floating base has to be taken care of. In simulation, external force is applied on the robot's body by a user with 3D haptic mouse or 3D visual marker. Robot is controlled by gravity compensation controller, to achieve effortless guidance of the teaching leg and the control of the floating base. Teaching is done by capturing keyframes, which can be executed by existing controllers. The work is concluded with a simulation experiment and a simple experiment on the real robot.

Ključne reči: Kinestetičko obučavanje, Četvoronožni robot, Upravljanje na osnovu momenata u zglobovima, Kompenzacija gravitacije, Keyframe-ovi

Key Words: Kinesthetic teaching, Quadrupedal robot, Torque control, Gravity compensation, Keyframes

1. UVOD**1.1. Motivacija**

Sva živa bića u prirodi svakodnevno uče nešto novo, da bi to kasnije mogla koristiti. Stečeno znanje uglavnom potiče iz učenja putem imitacije [1]. Robot, kao mašina, takodje ima privilegiju razvoja i usavršavanja putem učenja. Cilj da robote načinimo što je moguće sličnijim živim bićima, otvara dilemu: obučavanje nasuprot programiranju.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji je mentor dr Branislav Borovac, red. prof.

Kinestetičko obučavanje robota, što je srž ovog rada, ima sledeće prednosti:

- Korisnik dobija neposredan i intuitivan, vizuelni odziv od strane robota.
- Čovek može da oseti ograničenja tela robota.
- Korisnici bez veštine programiranja mogu lakše proširiti sposobnosti robota i prilagoditi ih novim situacijama.
- Robot se mnogo lakše može osposobiti za izvođenje specijalnih manevara ili manipulativnih zadataka.

Iako je u [2] pomenut u budućem radu, prema našim saznanjima kinestetičko obučavanje do sada nije primenjeno na četvoronožne robote.

1.2. Cilj

Cilj ovog rada je obučavanje četvoronožnog robota ANYmal da ponovi naučeno kretanje. Ovaj cilj može biti ispunjen kroz realizaciju sledećih zadataka:

- razvijanje kontrolera koji omogućava vođenje noge bez napona u zglobovima,
- nalaženje različitih načina za dobijanje korisničkog ulaznog signala, koji se u simulaciji koristi kako bi imitirao primenu spoljašnje sile na bilo koju tačku tela robota,
- snimanje kretanja i čuvanje podataka (kojima je opisano kretanje) u odgovarajućoj formi,
- realizacija naučenog kretanja,
- realizacija eksperimenata u simulaciji i na pravom robotu.

**2. OPIS SISTEMA – PREGLED KINESTETIČKOG
OBUČAVANJA****2.1. Hardver**

ANYmal [3] je četvoronožni robot, dimenzija psa srednje veličine. Svaka od njegove 4 noge ima 3 stepena slobode – dva u kukovima i jedan u kolenu. Dizajniran je za autonomne operacije u izazovnim sredinama. Pogon je ostvaren kroz specijalne aktuatore upravljane obrtnim momentom. Sposoban je za dinamičko trčanje, kao i za penjanje.



Slika 1. ANYmal, RSL, ETH Zurich [3]

2.2. Softver

Softverski deo ovog sistema je realizovan u ROS (Robot Operating System) operativnom sistemu. Simulacije su izvedene u Gazebo okruženju. Model robota u simulaciji je povezan sa sistemom preko dva *plugin*-a. Prvi je zadužen za primenu spoljašnje sile na telo robota, dok je drugi veza između robota i kontrolera. Kontroler, kao centar rada, bio je povezan sa dva GUI-a – jedan koji služi za primenu spoljašnje sile, i drugi za proces obučavanja i realizacije kretanja.

2.3. Pregled kinestetičkog obučavanja

Prateći ideju kinestetičkog obučavanja, sistem smo podelili u četiri glavne jedinice: kontroler, primena spoljašnje sile, obučavanje i realizacija kretanja.

3. KONTROLER

Za potrebe ovog rada impementiran je *Guided teaching controller* kao nadgradnja već postojećeg kontrolera zaduženog za kretanje robota (*Locomotion controller* [4]). On je odgovoran za ispitivanje da li noga služi kao oslonac, upravljanje noge, primenu kompenzacije gravitacije na nogu koja se obučava, kao i za adaptaciju poze baze tela robota.

3.1. Priprema – oslonac na tri noge

Nakon odluke, koju nogu želimo da obučimo, sledi priprema za oslonac na preostale tri noge. Optimizator poze tela robota računa poligon za oslonac i kao rezultat daje željenu poziciju i orijentaciju baze tela robota. U odnosu na željeni položaj i orijentaciju tela robota, izračunavaju se virtualni momenti i sile primenjene na telo robota. Oni se koriste za rešavanje problema optimizacije [5], koja kao rezultat daje željene sile kontakta. Konačno, vektor referentnih momenata primenjenih u zglobovima nogu koje služe za oslonac, dobijen je tako da odgovara silama kontakta.

3.2. Kompenzacija gravitacije

Noga robota, izabrana za obučavanje, nakon pripreme oslonca, više nije u kontaktu sa podlogom i upravlja se na osnovu momenata u zglobovima. Na osnovu jednačine kretanja noge robota:

$$M(q) * \ddot{q} + C(q, \dot{q}) * \dot{q} + g(q) = \tau$$

gde je M matrica mase, q pozicije zglobova, C Koriolis i centrifugalna matrica, g uticaj gravitacije i τ pogonski obrtni momenti, vidimo da se u stanju mirovanja uticaj gravitacije izjednačava sa momentima u zglobovima.

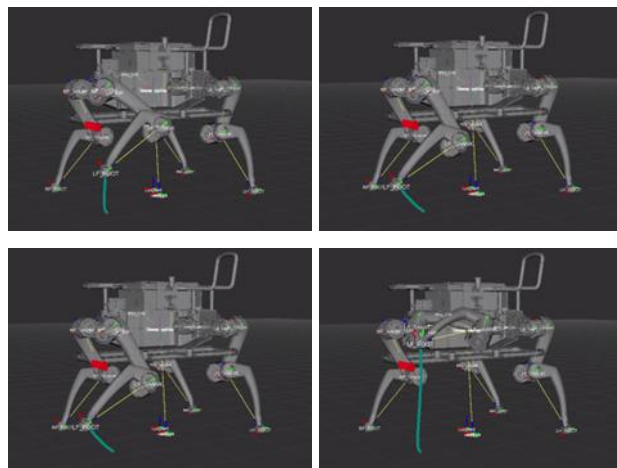
$$\tau = g(q)$$

Stoga zaključujemo da je primena kompenzacije gravitacije pogodna za obučavanje, kako bi vođenje noge bilo bez napona u zglobovima.

S obzirom da izračunati uticaj gravitacije u svakom zglobu neće biti isti kao i mereni, pretpostavljamo da će se noga u stanju mirovanja polako pomerati iz poslednje ostvarene pozicije tokom obučavanja, pa dodajemo prigušenje, kako bi noga ostala u istom položaju i nakon prestanka primene sile na nju.

$$\tau = g(q) - k_d * \dot{q}$$

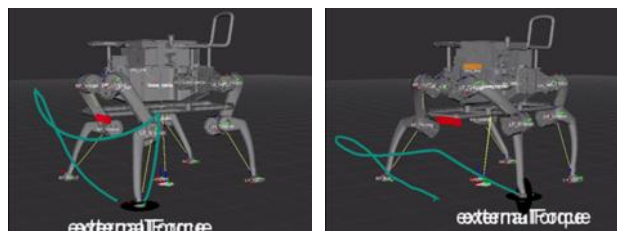
Rezultati eksperimenata primene kompenzacije gravitacije sa i bez dodatog prigušenja prikazani su na slici 2.



Slika 2. Kompenzacija gravitacije bez (levo) i sa (desno) dodatim prigušenjem

3.3. Adaptacija položaja i orijentacije baze robota

Optimizator poze [6] glavnog tela robota, osim izračunavanja poligona za oslonac, vrši i adaptaciju tela u odnosu na poziciju stopala obučavane noge. Optimizacija pozicije i orijentacije baze tela, vrši se kroz kriterijum minimizacije korišćenja kinematike noge. Radni prostor robota veći je u slučaju kada optimizator podešava položaj i orijentaciju tela robota, ali kada nam je neophodna visoka preciznost za izvršavanje zadatka, treba isključiti upotrebu optimizatora, zbog bolje kontrole nad stopalom obučavane noge robota (Slika 3.).



Slika 3. Kretanje sa (desno) i bez (levo) optimizacije poze

3.4. Prenosnje oslonca sa jedne na drugu nogu

Kako bi omogućili obučavanje više nogu, tokom procesa obučavanja, neophodno je bilo realizovati prenošenje oslonca sa jedne na drugu nogu. Ovaj zadatak se sastojao od ostvarivanja kontakta sa podlogom prethodno obučavanom nogom robota. Ovde je kretanje zadato krajnjom pozicijom stopala, a noga se upravlja na osnovu momenata u zglobovima. Nakon toga sledi faza pripreme za oslonac na tri noge, pa je sledeća noga spremna za obučavanje.

4. PRIMENA SPOLJAŠNJE SILE

Za vođenje noge robota potrebno je na nju primeniti spoljašnju silu. U radu sa pravim robotom to je vrlo jednostavno, jer primenjujemo silu sopstvenim rukama. Međutim, u simulaciji moramo da pronadjemo drugačiji način. Interakciju između modela robota i korisničkog ulaznog signala u *Gazebo* simulatoru, postizemo implementacijom *gazebo model plugin-a*. Korisnički ulazni signal dobijen je putem 3D miša (sila i moment) i *RViz* interaktivnog markera (samo sila). (Slika 4.)

4.1. 3D miš

Korišćen je Space Navigator (3D Connexion) koji ima 6 stepeni slobode pa kao informaciju od korisnika dobijamo pravac i smer željene sile i momenta. Za primenu spoljašnje sile postoje dodatni parametri koji se moraju definisati, pa je napravljen i grafički korisnički interfejs. Dodatnim parametrima opisujemo deo tela robota na koji treba da se primeni sila, referentna tačka na tom delu tela, referentni koordinatni sistem u kome je definisana spoljašnja sila, kao i amplituda sile. Eksperimentalno smo zaključili da je 3D miš preosetljiv za ovakvu vrstu sistema. Korisniku je potrebno mnogo vremena da se privikne, pa čak i nakon toga, precizno vođenje noge je zaista teško.



Slika 4. Space Navigator (levo), Rviz interaktivni marker (desno)

4.2 RViz interaktivni marker

Sa ciljem da omogućimo precizno vođenje noge implementirali smo aplikaciju za primenu spoljašnje sile na robota u simulaciji putem *RViz* interaktivnog markera. Ideja ovog pristupa je da korisnik odabere krajnju poziciju stopala noge koja se obučava. Zatim se generiše spoljašnja sila, na to stopalo, u pravcu i smeru željenog položaja. Generisana sila je opisana jednačinom

$$F = K_p * \Delta x - K_d * \dot{\Delta x}$$

gde je Δx je razdaljina između trenutne i željene pozicije stopala, a $\dot{\Delta x}$ linearna brzina stopala. Parametri K_p i K_d su dobijeni manuelnim podešavanjem.

U odnosu na interaktivni marker, 3D miš je mnogo brži u vođenju noge, dok je sa druge strane, za zadatke koji zahtevaju visoku preciznost, interaktivni marker daleko efikasniji.

5. OBUČAVANJE I REALIZACIJA KRETANJA

5.1. Obučavanje noge robota

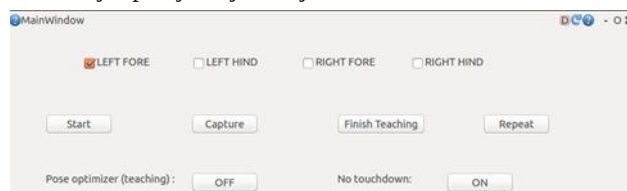
Prema [7], najbolji rezultati procesa obučavanja, dolaze iz hibridnog pristupa koji obuhvata snimanje *keyframe*-ova zajedno sa praćenjem trajektorije. Međutim, putanja noge tokom obučavanja u simulaciji se dosta razlikuje od željene, te smo se opredelili za snimanje *keyframe*-ova.

5.1.1 Keyframe-ovi

Keyframe je koordinatni sistem snimljen u određenom vremenskom trenutku. Tokom ovog rada, ovaj termin je korišćen za poziciju stopala noge koja se obučava, u određenom trenutku vremena. Snimanje *keyframe*-ova je omogućeno putem ROS paketa *tf*, koji daje poziciju stopala definisanu u osnovnom koordinatnom sistemu. Zbog toga možemo pretpostaviti da je – uz dobru lokalizaciju i mapiranje – robot robustan u odnosu na početnu poziciju.

5.1.2 GUI za obučavanje

Interakcija između korisnika i sistema omogućena je preko *Guided teaching* GUI-a (Slika 5.). Prekidačem *No touchdown* korisnik daje informaciju o tome da li obučavana noga treba da očekuje kontakt sa podlogom tokom realizacije kretanja. Snimanje *keyframe*-a obavlja dugme *Capture*, koje uzima ključne informacije kao što su: ime noge koja se obučava, da li se očekuje kontakt tokom kretanja, pozicija stopala u osnovnom koordinatnom sistemu; i pritiskom na dugme *Finish* sačuva ih u dokument koji opisuje trajektoriju.



Slika 5. GUI za obučavanje i realizaciju kretanja

5.1.3 Dokument koji opisuje trajektoriju

Dokument koji sadrži informacije o naučenoj trajektoriji čuva se u *YAML Ain't Markup Language* (YAML) formatu. Ovaj dokument je napravljen u skladu sa potrebama *Free Gait* kontrolera [6], koji se koristi tokom realizacije kretanja. Podaci o kretanju noge nose najvažnije informacije: ime noge koja je obučavana, da li se očekuje kontakt, da li se uključuje optimizacija položaja i orijentacije baze robota, i trajektoriju stopala definisanu pozicijama u određenim trenucima u vremenu.

5.2. Realizacija kretanja

Nakon što smo odabrali i snimili sve željene *keyframe*-ove, trajektorija stopala je dobijena *spline* interpolacijom i upravljanje robota se prebacuje na *Free Gait* kontroler [6], da bi se izvelo kretanje sačuvano u dokumentu. Ovaj kontroler se bazira na *Locomotion* kontroleru, kao i *Guided Teaching* kontroler, pa su suštinski delovi isti. Međutim, umesto kompenzacije gravitacije, tokom realizacije, noga robota je u stanju kretanja i upravljana je na osnovu pozicije zglobova. Pritiskom na dugme *Repeat*, ovaj kontroler postaje aktivan, a realizacija kretanja se vrši čitanjem i izvršavanjem trajektorije opisane u dokumentu.

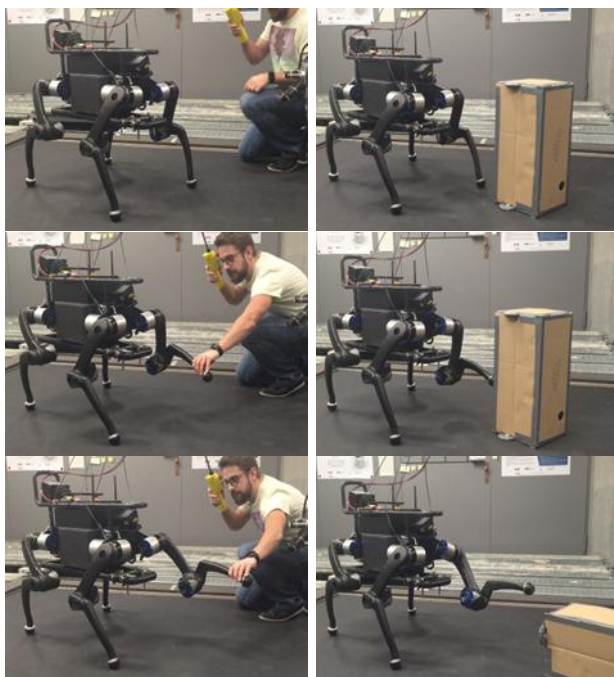
6. REZULTATI

Ishodi implementacije kinestetičkog obučavanja i realizacije kretanja testirani su u simulaciji, kao i na pravom robotu. Bilo je moguće manipulirati samo objektima jako male težine zbog nedostatka mogućnosti upravljanja na osnovu sile u zglobovima.

Na robotu *ANYmal* prvo je testirana kompenzacija gravitacije. Mogao se osetiti uticaj prigušenja, a usled neprecizne kalibracije noga se polako pomera i nakon prestanka delovanja spoljašnje sile (Slika 6.). Zadatak manipulacije guranja kartonske kutije uspešno je obavljen i prikazan je na slici 7.



Slika 6. *ANYmal* kompenzacija gravitacije



Slika 7. *ANYmal* obučavanje i realizacija kretanja – guranje kutije

5. ZAKLJUČAK I BUDUĆI RAD

5.2. Zaključak

Glavni cilj ovog rada - obučiti četvoronožnog robota da ponovi pokret - je ispunjen. Primenili smo kontroler koji vrši kompenzaciju gravitacije. Zatim smo, uz pomoć dodatog prigušenja, održavali nogu "mirnom" u vazduhu. Adaptacijom poze baze robota u odnosu na poziciju stopala obučavane noge, dobijen je veći radni prostor, a adaptaciju ne treba koristiti u slučajevima zadataka visoke preciznosti. Primena spoljašnje sile, na robota u simulaciji, putem GUI-a zajedno sa 3D mišem daje najbrže rezultate, međutim, ukoliko se robot obučava za zadatak u kome je potrebna preciznost, bolje je koristiti *RViz* interaktivni marker. *Keyframe*-ovi su korišćeni za snimanje kretanja, a podaci su sačuvani u *YAML* formatu. Dokument je napravljen u skladu sa potrebama *Free Gait* kontrolera, koji se koristi tokom realizacije kretanja. Obučavanje i realizacija kretanja su testirani u različitim scenarijima u simulatoru, a rad zaključujemo uspešnim eksperimentima na pravom robotu.

5.2. Budući rad

Učenje zadataka u kojima se robot upravlja na osnovu sile u zglobovima [8], bilo bi od velikog značaja za ovaj rad, jer bi zadaci manipulacije dobili stvarnu dimenziju. Takođe, bilo bi zanimljivo uzeti u obzir elastičnost zglobova. Posebnu pažnju bi trebalo posvetiti obučavanju poze tela robota radi boljih mogućnosti penjanja i specijalnih manevara. Konačno, ako je moguće obezbediti treću tačku oslonca na glavnom telu (bazi) robota, bilo bi interesantno razmotriti obučavanje robota za manipulisane sa više ekstremiteta.

6. LITERATURA

- [1] A. Billard and D. Grollman, "Robot learning by demonstration," vol. 8, no. 12, p. 3824, 2013, revision 138061.
- [2] D. Grollman and O. Jenkins, "Learning robot soccer skills from demonstration," in *Development and Learning*, 2007. ICDL 2007. IEEE 6th International Conference on, July 2007, pp. 276–281.
- [3] RSL, "Anymal," <http://www.rsl.ethz.ch>.
- [4] M. Hutter, C. Remy, M. Hoepflinger, and R. Siegwart, "Efficient and versatile locomotion with highly compliant legs," *Mechatronics*, IEEE/ASME Transactions on, vol. 18, no. 2, pp. 449–458, April 2013.
- [5] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger, and R. Siegwart, "Control of dynamic gaits for a quadrupedal robot."
- [6] P. Fankhauser, "Free gait – concept and usage," 2016.
- [7] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz, "Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '12. New York, NY, USA: ACM, 2012, pp. 391–398.
- [8] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

Kratka biografija:



Hana Gostimir rođena je u Beogradu 1990. godine. III beogradsku gimnaziju završila je 2009. s odličnim uspehom. Na Elektrotehničkom fakultetu, Univerziteta u Beogradu – odsek Signali i sistemi – diplomirala je 2013., a master akademske studije završila 2016. na istom Fakultetu iz oblasti Robotike. Master rad na Fakultetu tehničkih nauka, Univerziteta u Novom Sadu, iz oblasti Mehatronika, robotika i automatizacija odbranila je 2016. godine.

RAZVOJ SISTEMA ZA DALJINSKO UPRAVLJANJE PNEUMATSKIM IZVRŠNIM ORGANIMA PREKO INTERNETA**DEVELOPMENT OF SYSTEM FOR CONTROLLING PNEUMATIC ACTUATOR OVER THE INTERNET**

Miloš Tanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – U radu je predstavljen način upravljanja pneumatskim izvršnim organima putem interneta.. Pored toga, opisan je i sam proces testiranja rada ovakvog sistema sa prezentovanjem rezultata.

Abstract – The paper presents one way of controlling pneumatic actuator over the Internet. Besides, testing process of the system is described with results presented.

Glavne reči: Internet, pneumatika, daljinsko upravljanje, povratna informacija..

1. UVOD

Tema ovog rada predstavlja funkcionisanje takvog sistema koji objedinjavanjem više tehnologija čini da korisnik svojim mobilnim uređajem upravlja udaljenim aktuatorom.

Cilj je da aktivnost koja može biti potencijalno opasna uz pomoć daljinskog upravljanja drastično umanjí rizik jer korisnik ne mora biti u neposrednoj blizini i imati bliski kontakt sa izvršnim organima kao što je to bilo ranije.

2. PNEUMATIKA

Pneumatika je naučna i tehnička disciplina koja izučava primenu gasova pod pritiskom različitim od atmosferskog, koji uzrokuju neko kretanje ili mehanički rad. Pneumatski sistemi umeju često da omoguće pokretačku energiju na jeftiniji, bezbedniji, fleksibilniji i pouzdaniji način nego većina električnih motora i aktuatora [1].

2.1. Pneumatski sistem

Više elementarnih pneumatskih jedinica čini pneumatski sistem. To su: kompresor, komponente za distribuciju vazduha pod pritiskom, razvodni ventili i pneumatski aktuatori.

Dobro konstruisan i upravljan sistem za proizvodnju, pripremu i distribuciju vazduha pod pritiskom može znatno da uštedi energiju, smanji troškove održavanja i poveća produktivnost fabrika.

3. DALJINSKO UPRAVLJANJE

Jedna od glavnih prednosti bežičnog upravljanja je nepostojanje vodova kojim upravljamo.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Šešlija, red. prof.

To ume da bude od velikog značaja u velikim industrijskim sistemima gde ponekad nesmotrenom reakcijom ili nepažljivim rukovanjem može doći i do značajnijih negativnih posledica. One mogu biti razne, od prestanka rada postrojenja do gubitka kontrole nad aktuatorom.

3.1. Bežični prenos podataka

Osnovna karakteristika bežičnih mreža jeste rad bez korišćenja komunikacionih kanala u vidu kablova.

Tako se bežične mreže mogu podeliti na:

- 1) Bežične mreže kratkog dometa – (bluetooth)
- 2) Bežične mreže srednjeg dometa – (IEEE 802.11)
- 3) Bežične mreže velikog dometa – (satelitske mreže, mobilna telefonija)

Kod računarskih mreža se najčešće koristi IEEE 802.11 tehnologija.

Prenos podataka pomoću snopa svetlosnih signala se najčešće ostvaruje preko infracrvene i laserske tehnologije prenosa podataka. **Prenos podataka pomoću radio talasa** se vrši radio talasima frekvencijama širokog spektra. Najčešće korišćene tehnologije koje koriste ovakav vid prenosa podataka su Globalni sistem za mobilnu komunikaciju (GSM), Wireless Fidelity (WiFi), Bluetooth, ZigBee i RFID.[2]

4. POVEZIVANJE MOBILNIH UREĐAJA I MIKROKONTROLERA PREKO INTERNETA

Ovakvo rešenje je izabrano zbog mogućnosti upravljanja izvršnim organima sa bilo koje lokacije, dok god je mikrokontroler povezan na internet i dok mobilni uređaj ima pristup globalnoj mreži. Da bi mikrokontroleru omogućili pristup internetu, korišćen je Arduino Yun, mikrokontroler koji ima ugrađen WiFi modul. Aplikacija za mobilni uređaj je izrađena za Android operativni sistem zbog svoje otvorene arhitekture.

Metode kojima se uspostavlja komandna veza između mobilnog uređaja koji ima pristup kućnoj bežičnoj mreži ili pristup mobilnoj mreži koju omogućava mobilni operater sa jedne strane, i mikrokontrolera povezanog na poslovnu bežičnu mrežu sa druge strane, mogu biti:

- Mapiranje portova
- Anketiranje (polling)
- Korišćenje obrnutog protokola transfera HTML dokumenata (Reverse HTTP)

4.1. Mapiranje portova

Ova metoda jeste primena prevođenja mrežnih adresa gde se preusmeravanjem komunikacionih zahteva uspostavlja veza između jedne internet adrese i njenog porta do druge dok komunikacioni paketi prolaze kroz mrežnu kapiju ili mrežni zaštitni zid (firewall).

4.2. Anketiranje (Polling)

Anketiranje predstavlja proces gde kompjuter ili kontrolni uređaj čeka na eksterni uređaj da proveri svoj status ili spremnost, uglavnom sa hardverom niže kompleksnosti.

4.3. Korišćenje obrnutog protokola transfera HTML

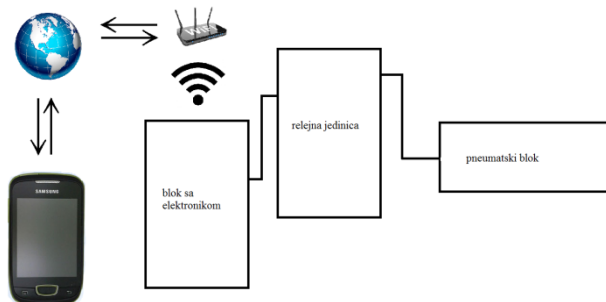
Ovaj metod predstavlja upotrebu aplikacije kao posrednika između dva uređaja. Ta aplikacija, server ili servis će istovremeno kontrolisati stranu koja šalje zahteve i automatski ih obrađivati i prosledivati dalje. Jedan od takvih servisa jeste Yaler.

5. EKSPERIMENTALNO ISPITIVANJE

Naredno poglavlje biće posvećeno detaljnijem objašnjenju komponenti sistema, pojašnjenju rada i razvoja aplikacije kako u Android tako i u Arduino okruženju. Na kraju je detaljno opisan sam proces testiranja sistema kroz koji se vide rezultati testiranja.

5.1. Komponente sistema

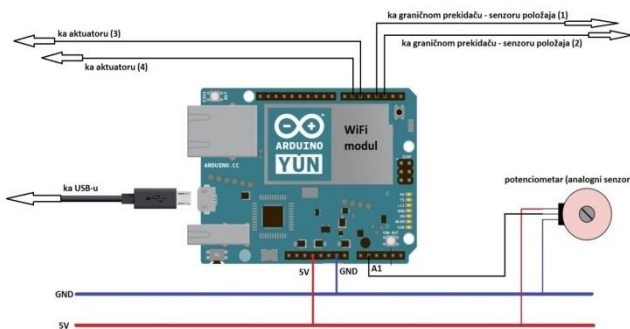
Na slici 1 predstavljen je put komunikacije između komponenti sistema, od mobilnog uređaja do pneumatskog bloka.



Slika 1. Eksperimentalno ispitivanje

Prvi blok, predstavljen je na slici 2 u vidu šeme povezivanja Arduino mikrokontrolera sa ostalim komponentama sistema.

Da bi Arduino Yun povezali na bežičnu mrežu, potrebno je uraditi reset mikrokontrolera preko dostupnih hardverskih prekidača na samom uređaju, a zatim preko korisničkog interfejsa izabrati željenu bežičnu mrežu i povezati se.



Slika 2. Blok šema elektronskog bloka

Drugi blok čini relejna jedinica koja je postavljena zbog razlike u naponskim nivoima između elektronskog i pneumatskog bloka.

Treći blok čini celina sa pneumatikom koja se sastoji od dva 3/2 jednostrano upravljana razvodnika i dva cilindra jednosmernog dejstva.

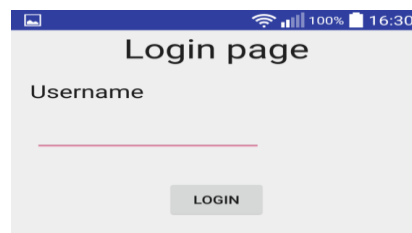
5.2. Razvoj i aplikacija u Android okruženju

Korisnički interfejs ovog sistema je baziran na direktnoj manipulaciji koja koristi interakciju na dodir uređaja koja oponaša pokrete ruke i prstiju čoveka.

Aplikacija je pisana i izrađena u programskom okruženju Android Studio, u programskom jeziku Java. Zamisao je bila da aplikacija sadrži mogućnost:

- upravljanja nad dva aktuatora,
- da postoji povratna informacija o stanju aktuatora koja se bazira na očitavanju senzora (hardverska povratna informacija); na očitavanju stanja poslatog signala (softverska povratna informacija),
- da postoji očitavanje senzora analogne vrednosti,
- zaštita upravljanja u zavisnosti od ovlašćenja korisnika [3]

Deo aplikacije za prijavu korisnika (slika 3.) osmišljen je kao zaštita sistema upravljanja od nedozvoljenog pristupa aplikaciji od strane osoba koje nemaju ovlašćenja za takve akcije.



Slika 3. Izgled ekrana za prijavu korisnika

Sledeći segment aplikacije jeste deo za kontrolu pneumatskih aktuatora i očitavanje stanja senzora (slika 4).



Slika 4. Izgled ekrana za kontrolu i čitanje stanja senzora

Segment za kontrolu i očitavanje stanja senzora sastoji se od:

- dva on/off dugmeta za aktivaciju pneumatskih aktuatora (Aktuator 1 i Aktuator 2),
- radio dugmeta za očitavanje uvučenog stanja aktuatora 1 (STANJE 1 - UVUČENO) i (STANJE 2 - IZVUČENO)
- labele za prikaz stanja analognog senzora (Očitavanje senzora),
- labele za prikaz stanja aktuatora 2 (Stanje 2: ON/OFF)

5.3. Razvoj i aplikacija u Arduino okruženju

Arduino je fizičko-računarska platforma (razvojni sistem) otvorenog koda i predstavlja savršen alat kada je neophodno dizajnirati povezivanje uređaja. Arduino YUN je mikrokontrolerska tabla zasnovana na ATmega32u4 i Atheros AR9331 koji služi za bežičnu komunikaciju. Atheros procesor podržava Linux distribuciju koja je bazirana na OpenWrt poznatom i kao OpenWrt-Yun.

Tabla ima ugrađenu Ethernet i WiFi podršku, USB ulaz, ulaz za microSD karticu, 20 digitalnih pinova za ulaz/izlaz, kristalni katalizator od 16MHz, mikro USB konekciju, ICSP header i 3 dugmeta za reset [4].

5.4. Testiranje sistema

U sledećem delu aplikacije biće prikazano:

- ON/OFF dugme za kontrolu Aktuatora 1,
- ON/OFF dugme za kontrolu Aktuatora 2,
- radio dugme za prikaz uvučenog/izvučenog stanja Aktuatora 1,
- labela za prikaz očitavanja vrednosti analognog senzora,
- labela za prikaz položaja Aktuatora 2

Aplikacija će po startovanju odmah ispitati stanja Aktuatora (da li su u izvučenom ili uvučenom stanju) i podesiti stanja senzora položaja (slika 5). Takođe će u zavisnosti od tih položaja Aktuatora, podesiti vrednosti ON/OFF tastera u odgovarajuće.

Prevlačenjem dugmeta "Aktuator 1" u položaj ON, u pozadini aplikacije biće pozvan link:

<http://try.yaler.io/gsiot-0a0v-nrfa/arduino/digital/5/1>,

što znači da će se na Yaler servis proslediti zahtev da se na globalnoj mreži pronade uređaj koji u sebi sadrži ključnu reč "gsiot-0a0v-nrfa".

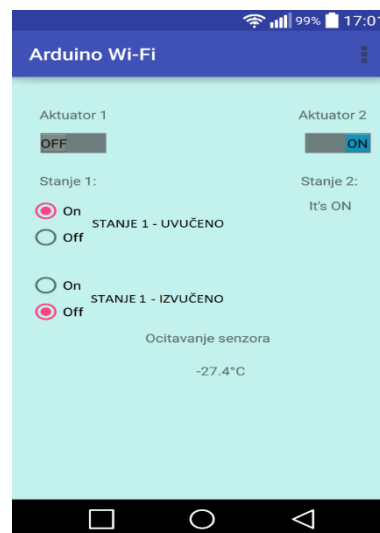
Yaler servis će prihvatiti taj zahtev i proslediti komandu na željeni uređaj.

Mikrokontroler sa imenom Arduino će prihvatiti taj zahtev, obraditi ga i na njegovom digitalnom izlazu 5 će postaviti logičku jedinicu.

Digitalni izlaz 5 je preko relejne jedinice povezan sa pneumatskim razvodnikom koji kontroliše "Aktuator 1". Aplikacija će po pozivu linka dobiti odgovor:

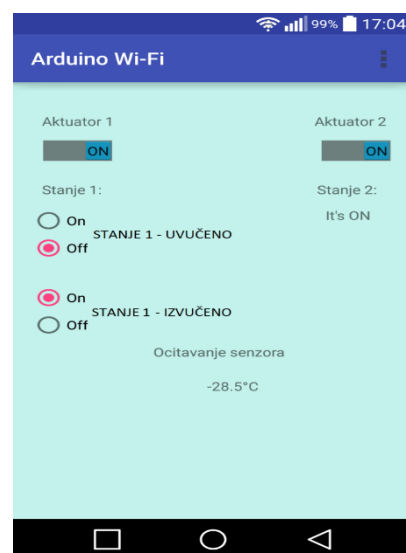
```
{"command":"digital","pin":5,"value":1,"action":"write"}
```

iz kog će se videti da li je poziv bio uspešan ili ne. Aplikacija neće samo jednom poslati poziv na link nego će to raditi sve vreme dok ne dobije pozitivan odgovor od Arduina.



Slika 5. Prikaz ekrana aplikacije gde je Aktuator 1 deaktiviran, Aktuator 2 aktiviran

Po dobijanju visokog napona, tj. logičke jedinice na digitalnom izlazu broj 5 aktiviraće se relej koji je povezan na ovaj digitalni izlaz, te će on proslediti 24V na špulnu pneumatskog razvodnika koji će aktivirati pneumatski aktuator, u našem slučaju cilindar jednosmernog dejstva.



Slika 6. Prikaz ekrana aplikacije gde je Aktuator 1 aktiviran, Aktuator 2 aktiviran

Izvlačenjem pneumatskog cilindra, aktiviraće se granični prekidač 2. Ovaj prekidač je takođe povezan na relej, koji će 24V sa graničnog prekidača prevesti u 5V i sprovesti ih na digitalni ulaz broj 3 koji je podešen u aplikaciji da čita stanje graničnog prekidača 2. Aplikacija sve vreme osluškuje stanja na digitalnim ulazima broj 2 i 3 i u ovom slučaju će dobijanjem visokog napona na digitalnom ulazu broj 3 proslediti informaciju o pristigloj logičkoj jedinici na Yaler servis, koja će to automatski proslediti na mobilni uređaj. Radio dugme koje prikazuje stanje - IZVUČENO će preći u ON položaj (slika 6).

Granični prekidač 1 je prvobitno bio aktiviran zbog uvučenog položaja Aktuatora 1. Izvlačenjem cilindra, on se deaktivira i taj signal šalje preko relejne jedinice na digitalni ulaz broj 2, koji je takođe podešen u aplikaciji da čita stanje graničnog prekidača 1. Dobijanjem niskog

napona na ulazu 2, Yaler servis će proslediti ovu informaciju na mobilni uređaj sa kog stiže zahtev o osluškivanju stanja na digitalnom izlazu broj 2 koji će biti u obliku:

```
{"command": "digital", "pin": 2, "value": 0, "action": "write"}
```

i informaciju o logičkoj nuli na datom ulazu proslediti na ekran aplikacije te će radio dugme koje prikazuje stanje graničnog prekidača 1 - UVUČENO preći u OFF položaj. Prevlačenjem "Aktuator 1" dugmeta u OFF položaj dešava se obrnuta situacija ali na istovetan način. Aplikacija okida poziv na link:

<http://try.yaler.io/gsiot-0a0v-nrfa/arduino/digital/5/0>

koji Yaler preuzima i prosleđuje na Arduino mikrokontroler koji setuje stanje logičke nule na digitalni izlaz broj 5. Logička nula na ovom izlazu preko relejne jedinice ukida 24V na špulni jednostrano upravljanog pneumatskog razvodnika i pneumatski cilindar se vraća u prvobitan položaj. Vraćanjem u ovo stanje, cilindar aktivira granični prekidač 1 i on šalje signal preko relejne jedinice na digitalni ulaz broj 2 koji preko Yaler servisa dostavlja informaciju o logičkoj jedinici do aplikacije, te radio dugme za stanje 1 - UVUČENO prelazi u stanje ON. Paralelno sa svim ovim akcijama, granični prekidač 2 se deaktivira i signal o njegovom stanju preko releja dolazi do digitalnog ulaza broj 3 na mikrokontroleru. Arduino informaciju o logičkoj nuli na digitalnom ulazu broj 3 prosleđuje aplikaciji preko Yaler servisa koja radio dugme za stanje 1 - IZVUČENO prebacuje u stanje OFF. Informacije o stanju analognog signala koji je na aplikaciji predstavljen labelom "Očitavanje senzora" dobijamo pozivom linka:

<http://try.yaler.io/gsiot-0a0v-nrfa/arduino/analog/1>

što znači da se na mikrokontroleru ispituje vrednost analognog ulaza A1 na koji smo postavili potencijometar. On u našem eksperimentu simulira temperaturu, npr. prostorije u kojoj se izvodi upravljanje pneumatskih aktuatora. Na ovakav analogni ulaz se može priključiti bilo koji analogni senzor.

6. ZAKLJUČAK

Prednost ovakvog sistema u odnosu na klasične sisteme su ogromne i višestruke. Ukoliko zadovoljava uslove o povezanosti mikrokontrolera na bežičnu i mobilnog uređaja na globalnu mrežu, kao i uslove o jačini i vrsti signala, sistem može biti aktiviran sa bilo koje lokacije na svetu, koja može biti u sredini zaštićenoj kako od opasnosti od neželjenih situacija prouzrokovanih radom aktuatora tako i od buke i prljavštine.

7. LITERATURA

- [1] Dragan Šešlija: *Osnove pneumatskog upravljanja* [skripta], Fakultet tehničkih nauka, Novi Sad, 2011.
- [2] Miloš Tanović: *Razvoj didaktičkog sistema upravljanja pneumatskim aktuatorom pomoću bežične tehnologije* [diplomski rad], Fakultet tehničkih nauka, Novi Sad, 2013.
- [3] <https://developer.android.com/studio/intro/index.html>, pristupljeno septembra 2016.
- [4] <https://www.arduino.cc/en/Main/ArduinoBoardYun>, pristupljeno avgusta 2016.

Kratka biografija:



Miloš Tanović je rođen 1987. godine u Novom Sadu, gde je i završio srednju elektrotehničku školu. Nakon diplomiranja na Fakultetu tehničkih nauka iz oblasti Mehatronika, robotika i automatizacija 2013., zaposlio se u kompaniji "SE DMS" u Novom Sadu. Od februara 2016. je zaposlen u kompaniji "Levi9". Master rad odbranio je novembra 2016. godine.

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2016. godine učestvovali su sledeći recenzenti:

Aco Antić	Dušan Dobromirov	Milan Simeunović	Rastislav Šostakov
Aleksandar	Dušan Gvozdenac	Milan Trifković	Slavica Mitrović
Erdeljan	Dušan Kovačević	Milan Trivunić	Slavko Đurić
Aleksandar Ristić	Dušan Uzelac	Milan Vidaković	Slobodan Dudić
Bato Kamberović	Duško Bekut	Milena Krklješ	Slobodan Krnjetin
Biljana Njegovan	Đorđe Ćosić	Milica Kostreš	Slobodan Morača
Bogdan	Đorđe Lađinović	Milica Miličić	Sonja Ristić
Kuzmanović	Đorđe Obradović	Milinko Vasić	Srđan Kolaković
Bojan Batinić	Đorđe Vukelić	Miloš Slankamenac	Srđan Popov
Bojan Lalić	Đura Oros	Milovan Lazarević	Srđan Vukmirović
Bojan Tepavčević	Đurđica Stojanović	Miodrag	Staniša Dautović
Bojana Beronja	Filip Kulić	Hadžistević	Stevan Milisavljević
Branislav Atlagić	Goran Sladić	Miodrag Zuković	Stevan Stankovski
Branislav Nerandžić	Goran Švenda	Mirjana	Strahil Gušavac
Branislav Veselinov	Gordana	Damnjanović	Svetlana Nikoličić
Branislava Kostić	Milosavljević	Mirjana Malešev	Tanja Kočetov
Branislava	Gordana Ostojić	Mirjana Radeka	Tatjana Lončar
Novaković	Igor Budak	Mirko Borisov	Turukalo
Branka Nakomčić	Igor Dejanović	Miro Govedarica	Todor Bačkalić
Branko	Igor Karlović	Miroslav	Toša Ninkov
Milosavljević	Ivan Beker	Hajduković	Uroš Nedeljković
Branko Škorić	Ivana Katić	Miroslav Plančak	Valentina Basarić
Cvijan Krsmanović	Ivana Kovačić	Miroslav Popović	Velimir Čongradec
Damir Đaković	Jasmina Dražić	Mitar Jcanović	Velimir Todić
Danijela Lalić	Jelena Atanacković	Mladen Kovačević	Veljko Malbaša
Darko Čapko	Jeličić	Mladen Radišić	Veran Vasić
Darko Marčetić	Jelena Borocki	Momčilo Kujačić	Veselin Avdalović
Darko Reba	Jelena Kiurski	Nađa Kurtović	Veselin Perović
Dejan Ubavin	Jelena Radonić	Nebojša Pjevalica	Vladimir Katić
Dragan Ivanović	Jovan Petrović	Neda Pekarić Nađ	Vladimir
Dragan Ivetić	Jovan Vladić	Nemanja	Radenković
Dragan Jovanović	Jovanka Pantović	Stanisavljević	Vladimir Strezoski
Dragan Kukolj	Karl Mičkei	Nenad Katić	Vladimir Škiljajica
Dragan Mrkšić	Katarina Gerić	Nikola Brkljač	Vlado Delić
Dragan Pejić	Ksenija Hiel	Nikola Đurić	Vlastimir
Dragan Šešlija	Laslo Nađ	Nikola	Radonjanin
Dragana Bajić	Leposava Grubić	Jorgovanović	Vuk Bogdanović
Dragana	Nešić	Nikola Radaković	Zdravko Tešić
Konstantinović	Livija Cvetičanin	Ninoslav Zuber	Zoran Anišić
Dragana Šarac	Ljiljana Vukajlov	Ognjen Lužanin	Zoran Brujic
Dragana Štrbac	Ljiljana Cvetković	Pavel Kovač	Zoran Jeličić
Dragi Radomirović	Ljubica Duđak	Peđa Atanasković	Zoran Mijatović
Dragiša Vilotić	Maja Turk Sekulić	Petar Malešev	Zoran Milojević
Dragoljub	Maša Bukurov	Predrag Šiđanin	Zoran Mitrović
Novaković	Matija Stipić	Radivoje Rinulović	Zoran Papić
Dragoljub Šević	Milan Rackov	Rado Maksimović	Željen Trpovski
Dubravka Bojanić	Milan Rapajić	Radovan Štulić	Željko Jakšić