



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXX

Број: 14/2015

Нови Сад

Едиција: „Техничке науке – Зборници“
Година: XXX Свеска: 14

Издавач: Факултет техничких наука Нови Сад
Главни и одговорни уредник: проф. др Раде Дорословачки, декан Факултета
техничких Наука у Новом Саду

Уређивачки одбор:

Проф. др Раде Дорословачки
Проф. др Владимир Катић
Проф. др Драгиша Вилотић
Проф. др Филип Кулић
Проф. др Срђан Колаковић
Проф. др Владимир Црнојевић
Проф. др Дарко Реба
Проф. др Драган Јовановић
Проф. др Мила Стојаковић

Проф. др Драган Спасић
Проф. др Драгољуб Новаковић
Проф. др Миодраг Хаџистевић
Проф. др Растислав Шостаков
Проф. др Војин Грковић
Проф. др Стеван Станковски
Проф. др Иван Луковић
Проф. др Ђорђе Лађиновић
Доц. др Милан Мартинов

Редакција:

Проф. др Владимир Катић, уредник
Проф. др Жељен Трповски, технички
уредник
Проф. др Зора Коњовић

Проф. др Драгољуб Новаковић
Мр Мирослав Зарић
Бисерка Милетић

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

Техничка обрада: Графички центар ГРИД

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

Председник Савета: проф. др Радош Радивојевић

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 четрнаеста 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 28.10.2015. до 20.11.2015. год., а који се промовишу 15.02.2016. год. То су оригинални прилози студената са главним резултатима њихових мастер радова. Део радова већ раније је објављен на некој од домаћих научних конференција или у неком од часописа.

У Зборнику су ови радови дати као репринт уз мање визуелне корекције.

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

У овој свесци, са редним бројем 14. објављени су радови из области:

- машинства,
- електротехнике и рачунарства,
- грађевинарства и
- мехатронике.

У свесци са редним бројем 15. објављени су радови из области:

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

У свесци са редним бројем 16. објављени су радови из области:

- инжењерског менаџмента,
- математике у техници,
- геодезије и геоматике и
- управљања ризиком од катастрофалних догађаја и пожара.

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. Miroslav Gluhajić, Jovan Dorić, SIMULACIJA I POREĐENJE OPTEREĆENJA LETEĆEG RUKAVCA KOLENA KOLENASTOG VRATILA KOD USISNOG I PREHRANJIVANOG MOTORA	2593
2. Nikola Medić, Mladomir Milutinović, KOSTRUKCIONO REŠENJE ALATA ZA IZRADU MASKE ZA TROFAZNU UTIČNICU INJEKCIONIM PRESOVANJEM	2597
Radovi iz oblasti: Elektrotehnika i računarstvo	
1. Dušan Mandić, MOGUĆNOSTI VINDOVS 10 JUNIVERSAL APS TEHNOLOGIJE ZA VIZUELIZACIJU ELEKTROENERGETSKIH MREŽA	2601
2. Ivan Morar, JEDNO REŠENJE SPREŽNOG PODSISTEMA	2605
3. Bojan Šibar, Vojin Šenk, PRIMENA TEORIJE INFORMACIJA NA OPTIMIZACIJU POSTUPKA RECENZIRANJA	2608
4. Milan Stojkov, PYTHON MODUL ZA RAD SA ACTIVITI SISTEMOM ZA UPRAVLJANJE POSLOVNIM TOKOVIMA	2612
5. Uroš Stepanović, KLASIFIKACIJA NEZAVISNIH KOMPONENTI BOLD SIGNALA NA OSNOVU SPEKTRALNE GUSTINE SNAGE	2616
6. Andrija Mitrović, Predrag Vidović, PRORAČUN TOKOVA SNAGA MIKSOVANE DISTRIBUTIVNE MREŽE U DOMENU SIMETRIČNIH KOMPONENTI	2619
7. Mirjana Klimenta, Predrag Vidović, PRORAČUN TOKOVA SNAGA MIKSOVANE DISTRIBUTIVNE MREŽE U FAZNOM DOMENU ..	2623
8. Радосав Белоица, ДИСТРИБУИРАНИ СИСТЕМ ЗА ПОВЕЗИВАЊЕ C# ПРОЈЕКТАТА	2627
9. Danilo Banjanin, PRIMENA INTELIGENTNIH PID REGULATORA U SISTEMIMA ZA KLIMU, GREJANJE I HLAĐENJE	2631

10.	Vladimir Benčić, INTEGRISANA UPOTREBA WII I KINECT KONTROLERA U FIZIKALNOJ TERAPIJI	2635
11.	Владо Оравец, ПРИМЕНА СОЛАРНИХ СИСТЕМА ЗА ЗАГРЕВАЊЕ ТОПЛЕ ВОДЕ У ДОМАЋИНСТВУ	2639
12.	Mladen Korica, PARTICIONISANJE MODELA PODATAKA DISTRIBUTIVNE MREŽE U CLOUD OKRUŽENJU ...	2643
13.	Darko Čolak, IMPLEMENTACIJA SISTEMA ZA KONTROLU VERZIJA U OKVIRU FENEK ALATA	2647
14.	Miloš Savić, SIMULACIONI MODEL FOTONAPONSKOG SOLARNOG SISTEMA	2651
15.	Dušan Stanivuković, ANALIZA STATIČKE SIGURNOSTI ELEKTROENERGETSKIH SISTEMA SA PRIMENOM KOREKTIVNIH MERA	2655
16.	Radomir Marković, IDENTIFIKACIJA I AUTORIZACIJA BAZIRANA NA CLAIM-OVIMA U SMART GRID SISTEMIMA	2659
17.	Natasa Samardžić, Tijana Kojić, Jelena Vukmirović, Djordjije Tripković, Branimir Bajac, Vladimir Srdić, Goran Stojanović, PERFORMANCE ANALYSIS OF RESISTIVE SWITCHING DEVICES BASED ON BaTiO ₃ THIN FILMS, konferencija IC-MAST, Mikonos, Grčka, septembar 2015.	2663
18.	Dušan Grče, Vladimir Katić, IDEJNO REŠENJE MIKROMREŽE U LABORATORIJI FTN	2667
19.	Иван Иванчић, Милош Живанов, ПРОЈЕКТОВАЊЕ И РАЗВОЈ УСБ ЗИГБИ УРЕЂАЈА ЗА РАД СА АНДРОИД ОПЕРАТИВНИМ СИСТЕМОМ	2671
20.	Žolt Kanjo, PROGRAMSKO OKRUŽENJE ZA KONVERZIJU RAZLIČITIH IZVORA PODATAKA U CIM/XML	2675
21.	Nikola Pap, SOFTVERSKO RASPOREĐIVANJE POSLOVA PO IZVRŠIOCIMA	2679
22.	Marijana Rackov, IMPLEMENTACIJA IZVEDENIH POLJA KORIŠĆENJEM OCL IZRAZA	2683
23.	Nemanja Jovičić, ANALIZA UTICAJA VISOKONAPONSKIH ELEKTROENERGETSKIH SISTEMA NA GASOVODE ...	2687
24.	Milan Meljanac, Željko Trpovski, KOMPARATIVNA ANALIZA STANDARDA H-264 I H-265	2691
25.	Драган Млађеновић, ЈЕДНО РЈЕШЕЊЕ КОМПОНЕНТЕ ЗА УПРАВЉАЊЕ ДИСТРИБУИРАНИМ АКВИЗИЦИОНО УПРАВЉАЧКИМ СЕРВЕРИМА	2695
26.	Зоран Тркуља, ИМПЛЕМЕНТАЦИЈА ТЕКСТУАЛНЕ СИНТАКСЕ ЗА EUIS DSL	2699
27.	Miodrag Popović, Željko Trpovski, TEHNIKE OPTIMIZACIJE WEB STRANICA, MODERNI ALGORITMI ZA RANGIRANJE WEB SAJTOVA	2703
28.	Đorđe Jarac, Željko Trpovski, SIMULACIJA SISTEMA ZA KODOVANJE I MODULACIJU DVB-S2 NA PREDAJNOJ STRANI ...	2707
29.	Milana Antin, FORMIRANJE MAPE DUBINE IZ JEDNE SLIKE	2711
30.	Stefan Pijetlović, Ilija Bašičević, Đorđe Kovačević, Nikola Teslić, A SOLUTION OF AN ONTOLOGY USED AS A DATA MODEL FOR AN INTERNET CONTENT PROVIDER, konferencija TELFOR, Beograd, novembar 2013.	2715
31.	Dunja Torbić, UGRADNJA ALATA ZA KONTINUALNU INTEGRACIJU U OKRUŽENJE VISUAL STUDIO	2719

32. Nebojša Bobić,	
SINHRONIZACIJA PODATAKA IZMEĐU SQL SERVERA I SQL COMPACT SERVERA	2723
33. Ilija Čordašić,	
ROBOKID	2727

Radovi iz oblasti: Građevinarstvo

1. Olinka Franić, Jasmina Dražić,	
IZBOR FASADNE OBLOGE STAMBENO-POSLOVNOG OBJEKTA U NOVOM SADU NA	
BAZI ANALIZE TROŠKOVA	2731
2. Bojan Poletanović,	
PROJEKAT DELIMIČNO UKOPANOG REZERVOARA, ČELIČNE NADSTREŠNICE I ANALIZA	
SEIZMIKE NA DELIMIČNO UKOPANI REZERVOAR	2735
3. Jelena Joksović,	
PROCENA STANJA, KONTROLNI PRORAČUN I SANACIJA KONSTRUKCIJE SUŠARE U	
MESTU GAJDOBRA	2739
4. Александра Ђурић,	
ДОГРАДЊА, КОНСТРУКЦИЈСКА И ЕНЕРГЕТСКА САНАЦИЈА ЗГРАДЕ ОСНОВНЕ	
ШКОЛЕ У НАСЕЉУ ГРБАВИЦА У БРЧКОМ У БРЧКОМ	2743
5. Sanela Tešić,	
ANALIZA UTICAJA RIZIKA GRADILIŠTA NA POVREDE NA RADU U GRAĐEVINARSTVU	2747
6. Dejan Ankić,	
ISPITIVANJE EFEKATA SUDARANJA NA SUSEDNE KONSTRUKCIJE U TOKU ZEMLJOTRESA ...	2751
7. Моника Сантоши Томић,	
ЗАМЕНА РАВНОГ КРОВА КОСИМ КРОВОМ НА ПОСЛОВНОМ ОБЈЕКТУ РАДИ	
ДОБИЈАЊА НОВОГ КОРИСНОГ ПРОСТОРА	2755
8. Исидора Сремачки,	
PROCENA STANJA I SANACIJA VIŠESPRATNE ZGRADE U HOPOVSKOJ ULICI U NOVOM	
SADU NAKON DOGRADNJE	2759
9. Radovan Marinković, Matija Stipić,	
HIDRAULIČKA ANALIZA DOGRADNJE GRADSKOG VODOVODNOG SISTEMA U	
SOMBORU - ZAPADNI SISTEM	2763

Radovi iz oblasti: Mehatronika

1. Saša Rebović,	
RAZVOJ WEB APLIKACIJE ZA ONLINE PRODAJU MOBILNIH TELEFONA	2767
2. Marko Tikvić,	
PRIMENA HERMITOVOG SPLAJNA ZA PLANIRANJE I PRAĆENJE PUTANJE MOBILNOG	
ROBOTA	2771
3. Slobodan Mijailović,	
RAZVOJ UPRAVLJAČKOG SISTEMA ZA UPRAVLJANJE STANJIMA U STAMBENIM OBJEKTIMA.....	2775
4. Bojan Lazarevski,	
УЛТРАЗВУЧНО МЕРЕЊЕ НИВОА ТЕЧНОСТИ	2779
5. Tobias Kovačić,	
BIBLIOTEKA ZA RS485 FULL DUPLEX KOMUNIKACIJU IZMEĐU KONTROLERA KOJI SU	
DEO UPRAVLJAČKOG SISTEMA ROBOTA MARKO	2783
6. Danijela Birmančević,	
RAZVOJ UDALJENE LABORATORIJE ZA UPRAVLJANJE PNEUMATSKIH CILINDARA	2786
7. Лазар Живковић, Мирко Раковић,	
ТЕЛЕМЕТРИЈСКИ СИСТЕМ ЗА ОБРАДУ И АКВИЗИЦИЈУ ПОДАТАКА О МЕРЕЊУ	
ПОТРОШЊЕ ПРИРОДНОГ ГАСА ПРИМЕНОМ „М-BUS“ ПРОТОКОЛА	2790

SIMULACIJA I POREĐENJE OPTEREĆENJA LETEĆEG RUKAVCA KOLENA KOLENASTOG VRATILA KOD USISNOG I PREHRANJIVANOG MOTORA**SIMULATION AND COMPARISON OF LOAD FLYING SLEEVE CRANKSHAFT, AT INTAKE AND TURBO ENGINES.**

Miroslav Gluhajić, Jovan Dorić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – U ovom radu dat je prikaz svih elemenata klipnog mehanizma i uz pomoć softvera MSC Adams izvršena simulacija i poredjenje opterećenja na letećem rukavcu kolena, kolenastog vratila kod usisnog i prehranjivanog motora sa unutrašnjim sagorevanjem.

Ključne reči: Klipni mehanizam, opterećenje letećeg rukavca kolena, kolenastog vratila, usisni i prehranjivani motor.

Abstract – The task of this work was to displaying all elements of reciprocating mechanism and with the help of software MSC Adams performed the simulation and comparison of the load on the flying knee sleeve, crank shafts of inlet and turbo combustion engines.

Key words: The piston mechanism, load a flying knee sleeve, crankshaft, intake and turbo engine.

1. UVOD

Motori sa unutrašnjim sagorevanjem se sastoje od pokretnih i nepokretnih elemenata, mehanizama i sistema. Klipni mehanizam ima zadatak da primi silu pritiska gasova i pretvori je u obrtno kretanje kolenastog vratila. Klipni mehanizam mora biti tako dimenzionisan da uz što manju masu pokretnih delova može da primi sva opterećenja koja se javljaju pri radu motora, a uz što manje mehaničke gubitke i habanje nalegajućih površina.

Ovi zahtevi postaju sve zahtevniji, jer se od motora sa unutrašnjim sagorevanjem u poslednje vreme zahteva povećanje snage uz istovremeno smanjene potrošnje goriva, što je ključno za povećanje efikasnosti motora, ovaj trend razvoja motora je poznat kao Downsizing.

Motori koji su produkt ove filozofije imaju manji broj cilindara i manju radnu zapreminom ali im je litarska snaga povećana u odnosu na motore starije generacije, a samim tim i elementi klipnog mehanizma trpe veća opterećenja.

2. OSNOVI ELEMENTI KLIPNOG MEHANIZMA**2.1. Klip**

Klip je jedan od opterećenijih delova klipnog mehanizma motora sa unutrašnjim sagorevanjem, jer pored mehaničkih tripi i toplotna opterećenje koja potiču od procesa sagorevanja goriva unutar motora. Za pravilan rad klipa neophodno ga je pravilno oblikovati, dimenzionisati i odabrati materijal, pa se iz tog razloga konstruisanje i proizvodnja klipova poverava specijalizovanim proizvođačima koji nastupaju kako partneri proizvođača motora.



Slika 1. Osnovni delovi klip motora sa unutrašnjim sagorevanjem

Klip motora je jedan od glavnih razloga koji ograničavaju povećanje broja obrtaja klipnih motora, zbog povećanja inercijalnih sila. Pri radu motora klip je izložen velikim mehaničkim i toplotnim opterećenjima i pri tim uslovima on mora da izvrši određene zadatke:

- da prenosi sile pritiska gasova preko klipnjače na kolenasto vratilo motora
- da prenese normalnu (bočnu) silu na zidove cilindra koje nastaju pri radu motora
- da uz pomoć zaptivnih elemenata (klipnih prstenova) vrši zaptivanje prostora za sagorevanje pri kretanju klipa unutar cilindra motora.
- da toplotu koji primi od vrelih produkata sagoravanja što efikasnije prenese na zidove cilindra kako bi se uz pomoć rashladne tečnosti vršilo hladjenje motora sa unutrašnjim sagorevanjem.
- kod dvotaktnih motora vrši i razvod radne smeše pomoću usisnih i izduvnih kanala.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Jovan Dorić, red. prof.

2.2. Osovinica klipa

Osovinica klip ima zadatak da zgloбно spoji klip sa klipnjačom klipnog mehanizma i na taj način omogući prenos sile pritiska gasova sa klipa na klipnjaču i dalje na kolenasto vratilo.



Slika 2. Osovinica klip motora sa uskočnim prstenovima

Uskočni prstenovi imaju zadatak da ograniče aksijalno pomeranje osovinice u klipu i spreči njen eventualni kontakt sa zidom cilindra. Osovinica klipa je opterećenja na savijanje usled dejstva sile pritiska gasova i mase klipa.

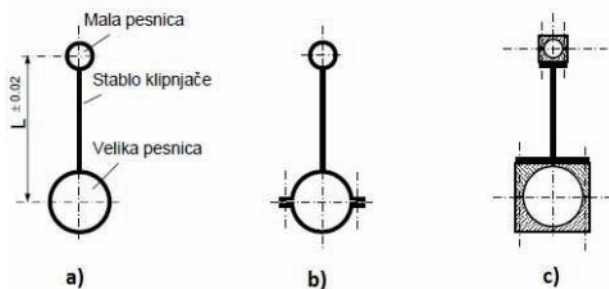
2.3. Klipnjača

Klipnjača je spojni element klipnog mehanizma koji povezuje klip sa kolenastim vratiolom motora i ima zadatak da translatorno-oscilatorno kretanje klipa u cilindru pretvori u obrtno kretanje kolenastog vratila, tj. da primi i prenese sile pritiska gasova i inercijalne sile koje deluju na klip, na kolenasto vratilo motora. U radnim uslovima klipnjača je izložena dejstvu intenzivno promenljivog opterećenja (po veličini i smeru) od sila pritiska gasova i inercijalnih sila. Nije retka pojava da pri radu motora klipnjača bude izložena i udarnom karakteru opterećenja.



Slika 3. Klipnjača klipnog motora sa unutrašnjim sagorevanjem

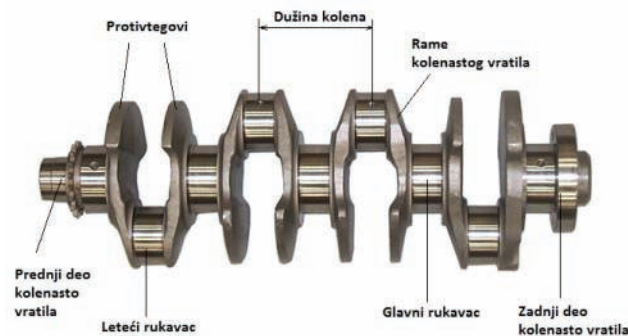
Klipnjača se sastoji iz tri osnovna dela, a to su: mala pesnica, telo ili stablo klipnjače i velika pesnica. Mala pesnica klipnjače služi za vezu sa osovinicom klipa, a velika pesnica za ostvarivanje veze sa letećim rukavcem kolenastog vratila. Klipnjača može biti iz jednog, dva ili više delova. Kod savremenih motora klipnjače se najčešće izrađuju iz dva dela, radi jednostavnije montaže na kolenasto vratilo. Pri radu motora klipnjača ima vrlo soženo kretanje koje je praćeno složenim opterećenjem. S obzirom da i kod ovog elementa masa igra značajnu ulogu, dosta pažnje se poklanja oblikovanju i odabiru materijala kako bi se smanjile inercijalne sile koje.



Slika 4. Osnovni delovi klipnjače

2.4. Kolenasto vratilo

Kolenasto vratilo je najodgovorniji, najopterećeniji i za izradu najkomplikovaniji pokretni element klipnog mehanizma, koje ima zadatak da primi sile pritiska gasova i inercijalne sile translatorno-oscilatornih i obrtnih masa pokretnih delova klipnog mehanizma. Sile pritiska gasova i inercijalne sile kolenasto vratilo u obliku obrtnog momenta predaje transmisiji radne mašine, osim toga kolenasto vratilo obezbeđuje pogon pomoćnih mehanizama i sistema motora.



Slika 4. Osnovni delovi kolenastog vratila

Položaj kolena kolenastog vratila zavisi od taktnosti motora, broja i položaja cilindara i redosleda paljenja. Kolena se postavljaju tako da se obezbedi:

- što bolja ravnomernost pri obranju (štoravnomerniji obrtni moment motora)
- što bolja uravnoteženost motora
- što manja dopunska naprezanja usled ugaonih, poprečnih i aksijalnih oscilacija vratila
- što ravnomernija opterećenost vratila
- što potpunija iskorišćenost energije izduvnih gasova ako motor poseduje turbokompresor.

Glavni rukavci kolenastog vratila služe kao oslonci i njihov broj može biti različit.

Ako se ispred i iza svakog letećeg rukavca nalazi glavni rukavac tada je broj glavnih rukavaca n_{gr} za jedan veći od broja letećih rukavaca n_{lr} .

Takva kolenasta vratila se nazivaju potpuno oslonjena kolenasta vratila. Kolenasta vratila koja imaju dva ili tri ledeća, izmđu dva glavna rukavca, nazivaju se nepotpuno oslonjena kolenasta vratila.

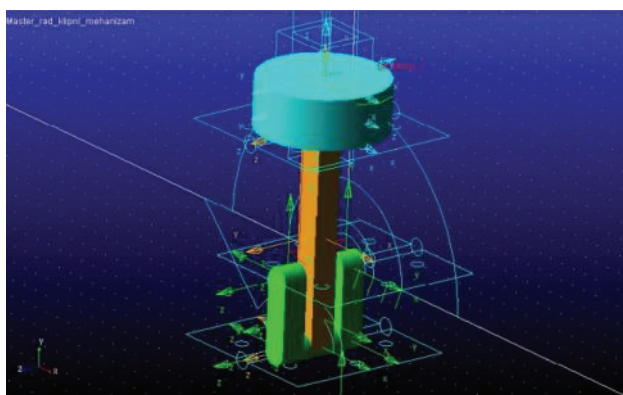
3. SIMULACIJA I POREĐENJE OPREREĆENJA

Simulacija i poredjenje opterećenja je izvedeno uz pomoć softvera MSC Adams u kome je izvršeno modelovanje klipnog mehanizma sa realnim parametrima (prečnik klipa, hod klipa, dužina klipnjače, masa elemenata klipnog mehanizma) već izvedenog motora sa unutrašnjim sagorevanjem.

Reč je o benzinskom motoru PSA/BMW grupacije od 1600 ccm³, sa prečnikom klipa 77mm i hodom 85,8mm, a pošto se ovaj motor proizvodi u dve varijante, kao usisni i prehranjivani, identičan klipni mehanizam trpi različita opterećenja u ova dva slučaja.

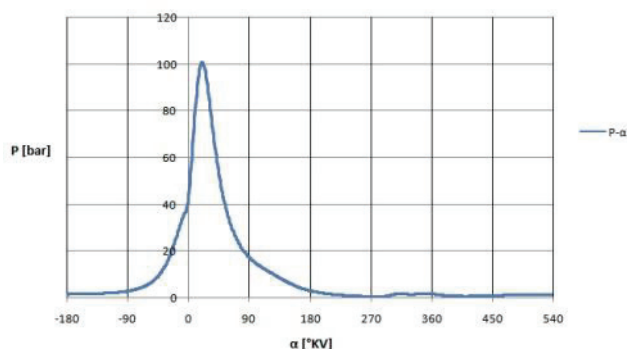
Iz softvera Ricardo Wave su izvučena dva P- α dijagrama za usisni i prehranjivani motor koji su ujedno i ulazni parametri za sprovođenje simulacije opterećenja.

P- α dijagram predstavlja promenu pritiska u zavisnosti od ugla obrtanja kolenastog vratila.



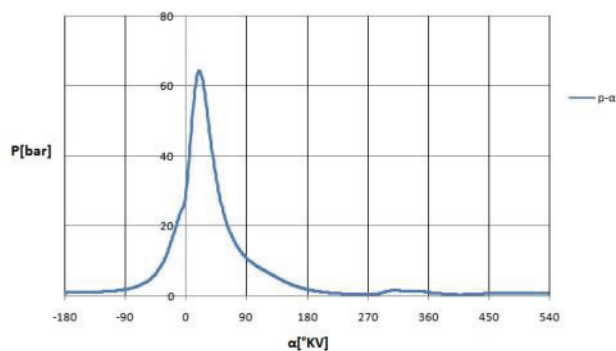
Slika 5. Model klipnog mehanizma modeliran u softveru MSC Adams.

Nakon što je model klipnog mehanizma završen i definisani svi ključni parametri, izvršeno je ubacivanje P- α dijagrama radi definisanje opterećenja čiji će se uticaj meriti na letećem rukavcu kolena kolenastog vratila. Pošto je reč o 4-taktnom motoru, radni ciklus traje 720°KV, tj. dva puna kruga, pa su i p- α dijagrami prilagođeni tom uslovu.

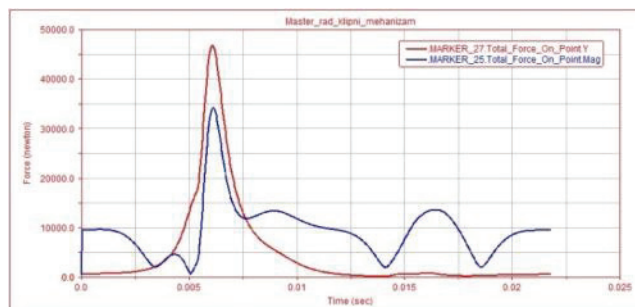


Slika 6. P- α dijagram prehranjivanog motora

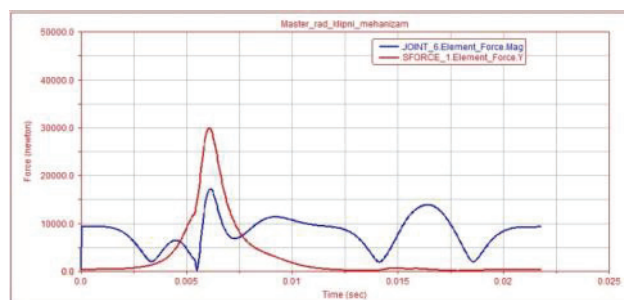
Pošto je prečnik klipa konstantna veličina a menja se samo pritisak unutar cilindra, moguće izračunati silu koja deluje na klip, a samim tim i na leteći rukavac u zavisnosti od ugla obrtanja kolenastog što je dovoljno za početak simulacije i poredjenje opterećenja.



Slika 7. P- α dijagram usisnog motra.



Slika 8. Sila pritiska gasova koja deluje na čelo klipa (crvena kriva) i ukupna rezultujuća sila na letećem rukavcu (plava kriva) kod prehranjivanog motra



Slika 9. Sila pritiska gasova koja deluje na čelo klipa (crvena) i ukupna rezultujuća sila na letećem rukavcu (plava), kod usisnog motra.

4. ZAKLJUČAK

Nakon izvršene simulacije opterećenja letećeg rukavca kolena, kolenastog vratila i analize rezultata zaključuje se da leteći rukavac trpi znatno veća opterećenja kod prehranjivanog nego kod usisnog motora. S obzirom da je simulacija bazirana na identičnom klipnom mehanizmu inercijalne sile koje figurišu u opterećenju letećeg rukavca su identična u oba slučaja.

Ako se uporede dijagrami opterećenja, može se uočiti da je ukupna rezultujuća sila pri taktu sagoravanje (segment dijagrama gde je sila pritiska gasova najveća) znatno veća kod prehranjivanog, nego kod usisnog motora sa unutrašnjim sagorevanje, zaključuje se su da je lteći rukavac znatno više opterećen kod prehranjivanog nego kod usisnog motora.

Pošto u poslednje vreme većina proizvođača motora u svojoj ponudi ima motore identičnih konstrukcionih karakteristika klipnog mehanizma, ali koji se znatno razlikuju po krivama sange i obrtnog momenta, mora se

voditi računa o pravilnom dimenzionisanju elemenata klipnog mehanizma i odabiru materijala.

Motore koji se u startu osmišljeni kao usisni nije preporučljivo kasnije modifikovati i transformisati u prehranjivane, jer tada dolazi do većeg opterećenja i habanja pojedinih elemenata klipnog mehanizma, koji nisu projektovani da izdrže veća opterećenja, pa je samim tim i vek motora kraći.

5. LITERATURA

- [1] - www.motorna-vozila.com
- [2] - dipl. ing. Predrag Djukić. Nove generacije automobilskih motora. Novi Sad 2003.
- [3] - Ivan Mihalec, Darko Kozarac, Zoran Lulić. Konstrukcije motora. Zagreb 2015.
- [4] – dr. Ivan Klinar. Motori sa unutrašnjim sagorevanjem. Novi Sad 2008.

Kratka biografija:



Miroslav Gluhajić je rođen u 1989 godine u Novom Sadu, gde je 2008. godine završio srednju Elektrotehničku školu. Diplomski – master rad iz oblasti simulacija i projektovanje motora sa unutrašnjim sagorevanjem je odbranio 2015. godine na Fakultetu Tehničkih Nauka u Novom Sadu.



Dr Jovan Dorić je redovan profesor na Fakultetu Tehničkih Nauka u Novom Sadu. Oblast rada su: Motori sa unutrašnjim sagorevanje, simulacija i projektovanje.

KOSTRUKCIONO REŠENJE ALATA ZA IZRADU MASKE ZA TROFAZNU UTIČNICU INJEKCIJONIM PRESOVANJEM

MOLD DESIGN FOR MANUFACTURING THE MASK OF THREE-PHASE SOCKET MASK BY INJECTION MOLDING

Nikola Medić, Mladomir Milutinović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Rad se bavi problematikom konstruisanja alata za injekciono brizganje plastike. U radu je detaljno opisana procedura modelovanja alata za izradu maske za trofaznu utičnicu, a čiji dizajn je većim delom sopstveno dizajnersko rešenje.

Abstract – This paper deals with mold injection process and mold design. In the paper detailed procedure of modeling and design of the mold for producing a masks for the three-phase socket is given. Design of the mask is mostly an original design solution.

Ključne reči: injekciono brizganje, dizajn alata, maska za trofaznu utičnicu, ABS

Keywords: injection molding, mold design, three-phase socket mask, ABS.

1. UVOD

Učešće delova od plastike postaje sve značajnije u mnogim oblastima tehnike, kao što su: autoindustrija, prehrambena industrija, avioindustrija, elektrotehnika, elektronika, telekomunikacije, IT sektor, proizvodnja kućne tehnike, hemijska indurstija i dr.

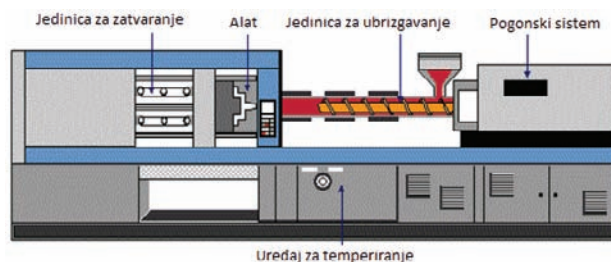
Prednost delova izrađenih od plastičnih masa u odnosu na delove izrađene od metala i ostalih nepolimernih materijala su [1]:

- Mala težina – gustina,
- Visoka čvrstoća u odnosu na količinu materijala,
- Ekstremna postojanost i otpornost na razne hemikalije i vremenske uslove,
- Izrada vrlo složenih delova se odvija po principu proizvodnje ugrabeno gotovih komponenti,
- Veliki asortiman u pogledu boja,
- Dobre elastično i toplotno izolaicone osobine,
- Niski zahtevi u pogledu temperature i pritiska u odnosu na metal,
- Relativno niska cena proizvodnje i dr.

Brizganje ili injekciono presovanje plastike je najvažniji ciklični postupak prerade plastičnih masa. U ovom ciklusu pripremak nema određenu formu i najčešće je u obliku granulata, praha, peleta i komadića. Po količini prerađenog polimera, brizganje je odmah iza ekstrudiranja. Postupak se izvodi na specijalnim mašinama, koje se popularno zovu brizgalice (ili brizgaljke). Osnovni delovi ovakvih mašina prikazani su na slici 1.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Mladomir Milutinović.



Slika 1. Mašina za brizganje plastike [2]

Sam proces brizganja plastičnih masa se odvija u nekoliko faza koje su prikazane na slici 2.



Slika 2. Ciklus brizganja termoplasta [3]

U ovom radu dato je jedno konstruktivno rešenje alata za izradu maske trofazne utičnice za slučajeve maloserijske i srednjeserijske proizvodnje. Alat ima jednu podeonu ravan i dve kalupne šupljine što znači da je moguće izraditi dva otpreska u jednom hodu mašina. Prilikom konstruisanja alata težilo se ka maksimalnoj standardizaciji, unifikaciji i tipizaciji delova alata, tako da bi proces brizganja bio što efikasniji i ekonomičniji.

Procedura dizajniranja maske uključivala je analizu postojećih rešenja na bazi čega se i uz poštovanje različitih kriterijuma došlo do modela čija geometrija u najvećoj meri predstavlja sopstveno dizajnersko rešenje.

2. DIZAJN MASKE ZA TROFAZNU UTIČNICU

Maska za trofaznu utičnicu je jedan od osnovnih delova svake utičnice a njena konstrukcija veoma je važna kako zbog bezbednosti, tako i zbog estetskog momenta, jer je ona za većinu kupaca nosilac kvaliteta čitave utičnice.

Sama konstrukcija ove maske mora da ispuni nekoiko kriterijuma:

- Kriterijume kompatibilnosti sa telom utičnice i sa utikačem električnog uređaja,
- Eksploatacione kriterijume,
- Bezbedonosne kriterijume i
- Estetske kriterijume.

Kada je reč o kompatibilnosti maske sa telom utičnice, tu se pre svega misli na lako uklapanje maske sa telom putem vijčane veze i uklapanje otvora maske sa delom tela koji dolazi u kontakt sa utikačem električnog uređaja. Kompatibilnost sa utikačem električnog uređaja ogleda se u tome da se kontaktne površine utičnice i utikača dimenzionišu tako da se utikač lako utakne i istakne iz strujne utičnice, a isto tako da utičnica omogućući čvrsto držanje, odnosno čvrst kontakt sa utikačem.

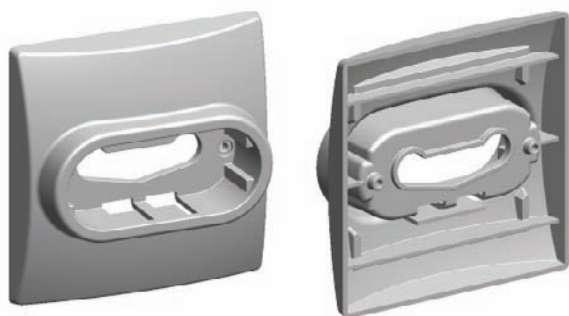
Pod bezbednosnim zahtevima se podrazumeva da maska prekrije sve provodnike u zidu i kontakte sa utikačem. Takođe je obavezna kompatibilnost sa elementima za zaštitu od domašaja dece.

Pod funkcionalnim zahtevima se podrazumeva što bolje iskorišćenje utičnice uz istovremeno neometano korišćenje drugih elemenata u blizini utičnice. Na primer, prevelike dimenzije utičnice nisu pogodne za često mali prostor između uređaja i zida prostorije u kojoj se uređaj nalazi.

Eksplotacioni zahtevi maske za utičnicu podrazumevaju njenu upotrebu, vek trajanja i lakoću održavanja.

Estetski zahtevi su veoma važni za masku utičnice, jer je ona prividni nosilac kvaliteta čitave utičnice. Upravo estetski zahtevi maske su odlučujući za kupovinu trofazne utičnice.

Maska utičnice u najvećem delu predstavlja originalno dizajnersko rešenje jer se konstrukcija i dimenzije bitno razlikuju od postojećih izvedbi, a čije su pozitivne i negativne karakteristike uzete u obzir prilikom modelovanja. Usvojena varijanta maske prikazana je na slici 3. Maska je modelovana u programskom paketu SolidWorks 2015.



Slika 3. Maska za trofaznu utičnicu

3. MATERIJAL ZA IZRADU MASKE

Izabrani materijal za izradu maske za trofaznu utičnicu je ABS (akrilonitril butadien stiren). Ovaj materijal je mešavina termoplastičnih polimera i dobija se mešanjem, odnosno polimerizacijom monomera diena i akrilonitrila uz prisustvo polibudiena.

ABS vrlo malo apsorbuje vlagu, ali relativno mali sadržaj vlage otežava preradu, uzrokuje greške na površini proizvoda i otežano doziranje, pa se ABS pre prerade suši.

ABS odlikuje velika tvrdoća, krutost, urdarna žilavost, dimenziona stabilnost i postojanost na niskim i visokim temperaturama.

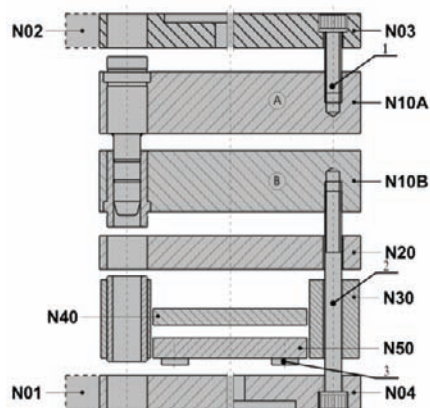
Svoju primenu ima u: automobilske industriji, hemijskoj industriji, industriji nameštaja, elektroindustriji, industriji računara i igračkih konzola, industriji igračaka i dr.

4. PROJEKTOVANJE I KONSTRUKCIJA ALATA ZA IZRADU ZADATOG DELA

Alat ili kalup za brizganje plastike je namenski deo sistema za brizganje polimera. Osnovni zadaci alata su: prihvatanje rastopa iz mlaznice jedinice za ubrizgavanje, njegovo hlađenje do postizanja forme otpreska, kao i potiskivanje (izbacivanje) otpreska iz kalupne šupljine. Osnovni elementi alata za brizganje su:

- Ulivni sistem,
- Kalupnu šupljinu,
- Sistem za odbacivanje otpreska,
- Sistem za vođenje alata,
- Kućište i
- Elemente za temperiranje alata.

Svaki alat za brizganje sastoji se od nepokretno i pokretnog deo koji su spojeni za nepokretni, odnosno pokretni deo mašine. Po pravilu pokretni deo alata ima složeniju konfiguraciju i u njemu se, u najvećem broju slučajeva, nalazi sistem za izbacivanje otpreska iz kalupne šupljine. Prema obliku alatnih ploča, kućišta alata mogu biti pravougaona ili kružna. U praksi se mnogo češće primenjuju pravougane ploče. Prilikom projektovanja ovih alata uvek se teži ka tome da što više elemenata alata, a time i samog kućišta bude standardno. Na slici 6.1 prikazano je pravougaono kućište sa pripadajućim standardnim elementima i elementima za vezu (1, 2 i 3) prema nemačkom D-M-E standardu.



Slika 4. Kućište alata za injeksiono presovanje plastike sa elementima za vezu [4]

Prema slici 4 standardni nazivi i oznake (prema D-M-E standardu) su:

- Pokretna stezna ploča N01 i N04,
- Nepokretna stezna ploča N02 i N03,
- Nepokretna kalupna ploča N10A,
- Pokretna kalupna ploča N10V,
- Među-ploča (ili osnovna ploča) N20,
- Distantna letva (ili odstojni segment) N30,
- Izbacivačka ploča N40 i
- Nosač izbacivačke ploče N50.

Projektovanja alata za injeksiono presovanje maske za trofaznu utičnicu obuhvatalo je sledeće aktivnosti:

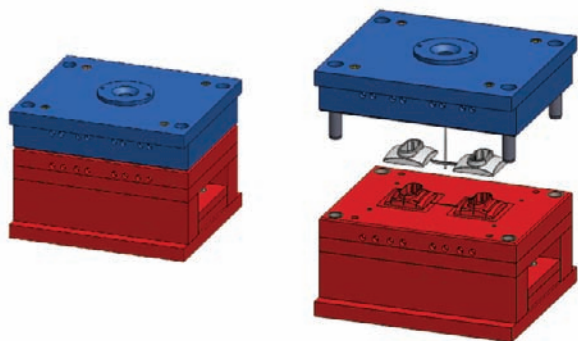
- Usvajanje idejnog rešenja alata za injeksiono presovanje,
- Projektovanje formi kalupne šupljine,
- Izbor standardnih i nestandardnih komponenti alata,
- Izbor materijala ploča alata i usvajanje njihovih dimenzija.

4.1. Idejno rešenje alata za injeksiono presovanje

Usvojen kalup za izradu maske za trofaznu utičnicu odlikuju sledeće karakteristike:

- Pravougaoni oblik,
- Jedna podeona ravan,
- Dve kalupne šupljine,
- Standardne dimenzije ploča alata,
- Standardni izbacivački mehanizam i mehanizam za vođenje,
- Hladan (otvrdnut) ulivni sistem,
- Linijski sistem hlađenja i dr.

Na slici 5 prikazano je idejno rešenje usvojenog alata za injeksiono presovanje.



Slika 5. Alat u zatvorenom (levo) i otvorenom položaju (desno), pokretni deo kalupa (crvena boja), nepokretni deo kalupa (plava boja) i otpresak (siva boja)

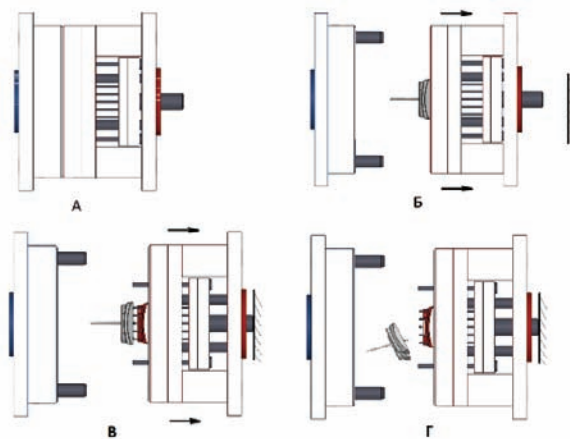
Na slici 6 prikazan je princip rada usvojenog alata. Na početku ciklusa alat je u zatvorenom položaju (slika 6A). Po završetku hlađenja otpreska alat se otvara (slika 6B), zatim sledi aktiviranje izbacivačkog sistema (slika 6V) i na kraju se otpresak izbacuje iz alata (slika 6G).

Alat se sastoji od pokretne i nepokretne stezne ploče, kalupnih ploča koje nose po dve pokretne i nepokretne forme, osnovne ploče, odstoynog bloka, prstena za centriranje kalupa, sistema za izbacivanje otpreska, koji se sastoji od izbacivačkih ploča, igala, povratnika, izvlakača i udarača. Prikaz rastavljenog alata za brizganje maske za trofaznu utičnicu dat je na slici 7.

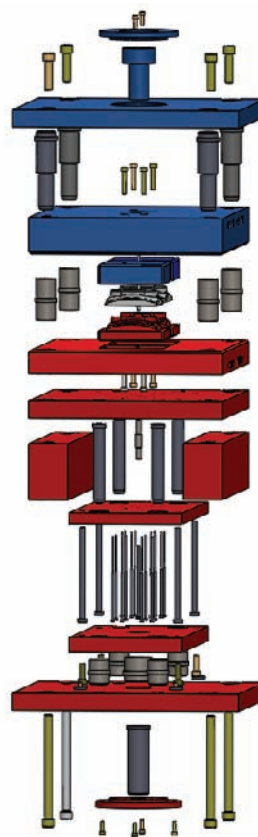
Kao što je već rečeno, alat ima dve kalupne šupljine, što stvara potrebu za složenijim ulivnim sistemom nego kada je to slučaj sa alatima koji izrađuju jedan otpresak po ciklusu.

Usvojeni ulivni sistem pripada grupi otvrdnutih ulivnih sistema, jer se oni koriste pri manjim serijama iz razloga što ne postoji potreba za takozvanim toplim diznama, čija je cena isplativa samo u slučaju masovne proizvodnje. Izgled i konstrukcija ulivnog sistema dati su na slici 8. Njegovi osnovni elementi su:

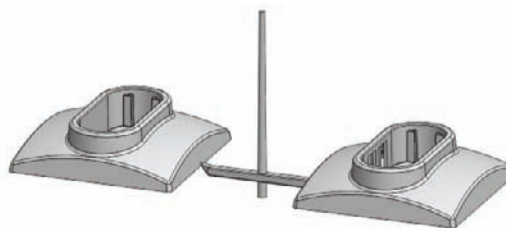
- Dovodni kanal, kružnog poprečnog preseka, prečnika od 4 mm na mestu spajanja sa razvodnim kanalima i nagibom od 1,5 stepeni.
- Razvodni kanali, kvadratnog poprečnog preseka dimenzija 4x4 mm i
- Ulazni kanali, istog poprečnog preseka kao i razvodni kanali na mestu njihovog spajanja, odnosno 1,5x1,5mm na mestu ulivanja u kalupnu šupljinu.



Slika 6. Princip rada alata za izradu maske trofazne utičnice



Slika 7. Alat za brizganje maske za trofaznu utičnicu



Slika 8. Otpresci maske za trofaznu utičnicu sa ulivnim sistemom

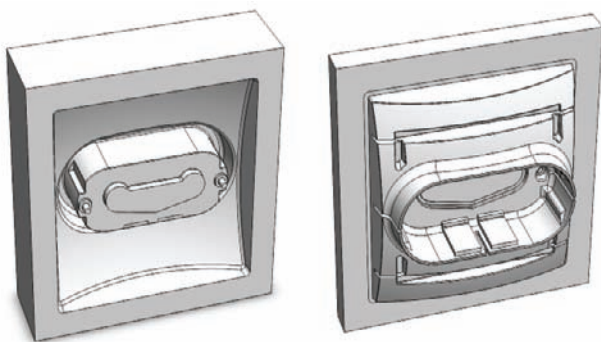
4.2. Projektovanje formi kalupne šupljine

Pod projektovanjem formi kalupne šupljine se podrazumeva konstruisanje jezgra koje se spaja sa pokretnom kalupnom pločom (pločom jezgra) i gnezda koje se spaja sa nepokretnom kalupnom pločom (pločom gnezda). Projektovanje formi kalupne šupljine za izradu maske za trofaznu utičnicu je izvršeno u programskom paketu SolidWorks 2015, preko njegove alatke za projektovanje alata za injeksiono presovanje koja se zove Mold Tools.

Projektovanje formi za navedeni deo izveden je prema sledećim koracima:

- Korekcija dimenzija dela,
- Analiza nagiba i definisanje podeone linije,
- Definisanje podeone ravni,
- Zatvaranje otvora na modelu i
- Generisanje formi za zadati plastični deo.

Generisano jezgro (pokretna forma) i gnezdo (nepokretna forma) za izradu maske za trofaznu utičnicu prikazani su na slici 9.



Slika 9. Nepokretna (levo) i pokretna forma (desno)

4.3. Izbor nestandardnih i standardnih delova kućišta alata

Pri projektovanju i izboru elemenata alata posebno se vodilo računa o ugradnji standardnih elemenata, kako bi cena alata bila što niža i troškovi održavanja bili što niži. Svi delovi alata osim pokretne i nepokretne forme, ulivne čaure, ploče za centriranje kalupa i udarača su standardni.. Standardni elementi se poručuju kod proizvođača ovih delova i prilagođavaju se naknadnim obradama za potrebe usvojenog alata. Namenski se izrađuju delovi koji nisu standardni.

4.4. Izbor materijala i dimenzija ploča alata

Usvojene standardne ploče alata proizvodi kompanija „Meusburger,“ nemački proizvođač delova alata za injeksiono presovanje plastomera [7]. Podaci o dimenzijama usvojenih ploča, materijal od koga se izrađuju kao i sva tehnička dokumentacija vezana za ovaj alat mogu se naći u master radu [8].

5. ZAKLJUČAK

U radu je opisan postupak projektovanja i dato je konstruktivno rešenje alata za injeksiono brizganje maske trofazne utičnice. Izvedena konstrukcija najvećim delom sastoji se od standardnih elemenata i prilagođena je tehnokonomskim kriterijumima malo-serijske i srednjeserijske proizvodnje. Primenom CAD softverskog paketa SolidWorks 2015, odnosno njegovog modula Mold Tools, proces projektovanja u znatnoj meri bio je automatizovan, posebno u segmentu generisanja pokretne i nepokretne forme (jezgara). Ovakvim pristupom ujedno se postiže veća tačnost i bolji kvalitet otpreska.

S obzirom na sve veću prisutnost proizvoda od plastičnih masa u svakodnevnom životu, može se reći da je potreba za ovim proizvodima konstantna, a samim tim i za alatima za injeksiono presovanje. Zbog toga svaka inovacija u polju injeksionog presovanja je dobrodošla, svako novo rešenje alat omogućava dalje unapređenje ovog postupka. U tom smislu može se posmatrati i ovaj rad.

6. LITERATURA

- [1] Čatić I.: „Proizvodnja polimernih tvorevina,“ Biblioteka Polimerstvo, zelena serija, Zagreb 2006.
- [2] www.harbec.com
- [3] Vilotić D.: „Mašine za injeksiono presovanje,“ Fakultet tehničkih nauka, Novi Sad, 2013.
- [4] Matin I.: „Programski sistem za projektovanje alata za injeksiono presovanje plastike – dokumentacija tehničkog rešenja,“ Fakultet tehničkih nauka, Novi Sad 2015.
- [5] Perošević B.: „Kalupi za injeksiono presovanje plastomera,“ Naučna knjiga, Beograd, 1995.
- [6] www.arburg.com
- [7] www.meusburger.com
- [8] Medić N. Modelovanje i konstrukcija alata za injeksiono brizganje maske trofazne utičnice, Master rad, FTN, 2015.

Kratka biografija:



Nikola Medić, rođen je u Novom Sadu 5. jula 1989. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Tehnologija obrade rezanjem je odbranio 2014. godine.



Mladomir Milutinović, rođen je u Arilju 1967. god. Doktorirao je na Fakultetu tehničkih nauka u Novom Sadu 2014. godine. Zaposlen je kao docent na Katedri za tehnologije oblikovanjem i inženjerstvo površina na Fakultetu tehničkih nauka.

MOGUĆNOSTI WINDOWS 10 UNIVERSAL APPS TEHNOLOGIJE ZA VIZUELIZACIJU ELEKTROENERGETSKIH MREŽA**WINDOWS 10 UNIVERSAL APPS FOR VISUALIZATION OF POWER NETWORKS**

Dušan Mandić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je opisan način funkcionisanja i mogućnosti Windows 10 Universal apps tehnologije. Stečeno znanje je primenjeno na izradi aplikacije za vizuelizaciju elektroenergetske mreže. Aplikacija je razvijana u C# programskom jeziku i Microsoft Visual Studio 2015 programskom okruženju.

Abstract- The paper describes features and possibilities of Windows 10 Universal apps and application for vizualization of power network. Application was developed using C# programming language and Microsoft Visual Studio 2015 environment.

Ključne reči: Platforma, universal apps, Windows 10

1. UVOD

Razvoj tehnologije, pogotovo pametnih telefona i tableta, donosi nove i veće hardverske mogućnosti čime se polako smanjuje razlika u performansama između klasičnih personalnih računara i mobilnih uređaja. Nove hardverske mogućnosti omogućavaju dodatne softverske opcije, kao i brži protok i obradu podataka. Ovakav tempo razvoja i širok spektar mogućnosti dovode do toga da korisnici žele isto iskustvo pri korišćenju aplikacija na različitim platformama i uređajima što znači i optimizaciju korisničkog interfejsa na osnovu operativnog sistema, vrste interakcije sa korisnikom i veličine i rezolucije ekrana, uključujući i različite kontekste upotrebe aplikacije. Zbog ovih uslova, sve veći značaj se daje softverskoj praksi razvijanja multiplatformskih aplikacija koje će zadovoljiti zahteve i potrebe većine korisnika. Inicijalni pristup razvoju nativnih aplikacija sa posebnim izvornim kodom za svaku platformu i vrstu uređaja zahteva mnogo resursa, pažnje i vremena, a može otežati dalji razvoj i održavanje softvera. U današnjem okruženju, razvijanje iste aplikacije za svaku platformu od interesa nije održivo rešenje za kompanije koje žele da budu konkurentne na tržištu. Zbog toga se sav napor ulaže u razvijanje alata i tehnologija koji će omogućiti kreiranje multiplatformskih aplikacija sa jednim izvornim kodom portovanim na različite operativne sisteme i vrste uređaja.

U ovom radu će biti opisan način funkcionisanja i mogućnosti Universal apps tehnologije koja se pojavila sa Windows 10 operativnim sistemom i omogućuje razvoj multiplatformskih aplikacija za sve uređaje koji podržavaju navedeni operativni sistem.

Pored ovoga, opisana je i implementacija univerzalne aplikacije za vizuelizaciju elektroenergetske mreže gde su

primenjeni određeni principi koji su preporučeni za razvoj multiplatformskih aplikacija, kao što je MVVM šablon koji će biti objašnjen u kasnijim poglavljima.

Rad je podeljen u tri poglavlja. U prvom poglavlju opisan je pojam platforme, multiplatformskog softvera i njegove vrste, arhitektura Windows 10 operativnog sistema, potrebni alati i okruženje za razvoj univerzalnih aplikacija, kao i vrste prezentacionog dizajn šablona. Drugo poglavlje se bavi mogućnostima Universal apps tehnologije, novitetima u XAML deklarativnom jeziku, kao i različitim stilovima interakcija sa korisnicima. U trećem poglavlju je opisana implementacija i funkcionalnost univerzalne aplikacije za vizuelizaciju elektroenergetske mreže.

2. MULTIPLATFORMSKI SOFTVER I WINDOWS 10

Platforma predstavlja kombinaciju hardvera i softvera koja omogućava rad softverskih aplikacija. Platformama se mogu smatrati:

- *Hardverske platforme*, odnosno računarska arhitektura i arhitektura procesora kao što su x86 ili x86 64 CPU platforme.
- *Softverske platforme*, odnosno operativni sistemi ili programsko okruženje.

Platforme predstavljaju sistem ili skup podсистема koji obezbeđuju određenu funkcionalnost. One stvaraju podlogu za rad drugih softverskih sistema ili aplikacija. Operativni sistemi su takođe softverske platforme koje se izvršavaju na većem broju hardverskih platformi. Posredstvom softverskih platformi ostvaruje se neutralnost softverskih aplikacija u odnosu na hardversku platformu [6].

2.1 Multiplatformski softver

Kada se govori o multiplatformskom softveru odnosno platformski neutralnom softveru (*Cross-platform*), zapravo se govori o aplikacijama koje mogu da se izvršavaju na više vrsta operativnih sistema ili hardverskih platformi. Pojam platformski nezavisnog softvera (*Platform Independent*) ima slično značenje, međutim platformski nezavisan softver se može izvršavati na svakoj platformi, dok se multiplatformski softver izvršava bar na dve platforme. Pojam multiplatformski može da se koristi ka svim tipovima softvera, uključujući programske jezike, operativne sisteme, aplikacijske programe i tipove podataka.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivetić, red.prof.

Postoji nekoliko pristupa pri razvoju multiplatformskog softvera i ovi pristupi se baziraju na različitim mehanizmima realizacije. Osnovni mehanizmi realizacije su: virtuelne mašine/izvršni sistemi (*Java* i *.NET*), *web* mehanizmi i mehanizmi delovanja (*UML – Unified Modeling Language*, *MDA – Model Driven Architecture*).

Multiplatformski softver je prvobitno počeo da se širi razvijanjem aplikacija za više operativnih sistema u sklopu jedne vrste uređaja. Tako su nastale multiplatformske aplikacije za *PC* računare, mobilne telefone, a posebnu grupu su činile igrice. Vremenom su se igrice portovale na više uređaja čime se razvila praksa da se ista igrice razvijaju za računare i igračke konzole čime bi se obuhvatio veći broj korisnika.

2.2 Windows 10 Universal apps

Windows 10 Universal apps predstavlja tehnologiju izrade aplikacija koja dolazi uz pomenuti operativni sistem i omogućuje implementaciju multiplatformskih aplikacija koje mogu da se izvršavaju na svim uređajima koji podržavaju *Windows 10* familiju operativnih sistema (*Windows 10*, *Windows 10 Mobile*, *Xbox...*). Pod uređajima koji podržavaju navedeni operativni sistem spadaju *PC* računari, laptop računari, tableti raznih veličina, mobilni uređaji, konzole, holografski računari itd. U poređenju sa *web* baziranim aplikacijama, univerzalne aplikacije su ograničene operativnim sistemom što dovodi do manjeg stepena multiplatformske kompatibilnosti, a samim tim i manjeg opsega korisnika. Izbor tehnologije i vrste aplikacije treba da zavise od svrhe i zahteva korisnika softvera koji se razvija. *Universal apps* pristup razvijanja softvera ima za cilj da se sve aplikacije namenjene korišćenju na više uređaja kodiraju jednom, sa jednim setom poslovne logike i korisničkog interfejsa koji će se adaptirati na sve *Windows 10* uređaje u zavisnosti od veličine ekrana, rezolucije i načina interakcije sa korisnicima. Osnovu *Universal apps*-a čine: *Windows Core*, *Universal Windows Platform* i *Windows App*.

Windows Core predstavlja zajedničko jezgro za sve *Windows 10* verzije operativnog sistema. Pod tim se podrazumeva redukovani podskup zajedničkog programskog koda koji dostavlja osnovne mogućnosti funkcionisanja *Windows*-a na svim uređajima. Jedna od mogućnosti koje se nalaze u *Windows Core*-u je univerzalni hardverski drajver koji može da se instalira i radi na svim verzijama operativnog sistema.

Universal Windows Platform predstavlja jedinstvenu *API* kolekciju namenjenu lakšem razvijanju *Windows 10* univerzalnih aplikacija i u suštini je evoluirana i proširena verzija *Windows Runtime* aplikacionog modela. Jedinstvena *API* kolekcija je zajednička za sve uređaje čime se postiže visok nivo fleksibilnosti pri razvoju aplikacija. Refaktorisani aplikacioni model je podeljen u *API* kontrakte (*Contracts*). *API* kontrakt predstavlja javni *API* komplet za određenu familiju uređaja čime izlaže sve *API*-e unutar kontrakta i obezbeđuje funkcionalnosti koje su u njima implementirane. *UWP* je definisana *API* kontraktima koje sadrži, a kako se menjaju verzije kontrakta, tako se menja i verzija *UWP*-a.

Windows App predstavlja konkretnu *Windows 10* univerzalnu aplikaciju. Osnovna stvar koju treba napomenuti je da kada se razvijaju ovakve aplikacije, one ne ciljaju određeni operativni sistem već *UWP*, tj. određene familije uređaja. Prednost ovakvog načina rada je što kompajliranjem aplikacije dobijamo samo jedan binarni kod koji može da se pokreće na svim uređajima. Kada se razvija *Windows App*, biraju se familije uređaja koje će aplikacija ciljati u zavisnosti od toga koji *API* komplet će biti neophodni za njenu implementaciju. Univerzalna familija uređaja (*Universal Device Family*) se razlikuje od ostalih. Ona sadrži *API* komplet koji je zajednički za familije uređaja koji je nasleđuju. Kada aplikacija cilja univerzalnu familiju, to znači da će ona imati maksimalan domet i da će se pokretati na svim uređajima različitih verzija *Windows 10* sistema. Familije uređaja koje nasleđuju univerzalnu familiju dodaju svoj *API* komplet i time garantuju da će se takva aplikacija moći izvršavati na svim uređajima iz njihove familije. Ovakvi *API* kompleti nazivaju se platformske ekstenzije ili dodaci.

Korisnički interfejs često sadrži dosta pretrpanog programskog koda, uglavnom zbog kompleksne logike kojom mora da rukuje. Prezantacioni šabloni su dizajnirani da bi redukovali kompleksnost programskog koda u prezentacionom sloju i učinili korisnički interfejs čistim i lako podesivim. Osnovna podela prezentacionih šablona: *MVC*, *MVP* i *Presentation Model*. *Model View ViewModel* (*MVVM*) je šablon koji u svojoj osnovi ima svrhu da razgraničava funkcionalnosti u različitim delovima aplikacije definisanjem nezavisnih slojeva. Ovaj šablon je *Microsoft*-ova varijacija *Presentation Model* dizajn šablona. Svaki sloj ima svoju ulogu i može da se proširuje uz minimalne posledice na ostale slojeve. *Model* predstavlja stvarne podatke i informacije. *View* je prezentacioni sloj, definiše se u *xaml* jeziku i koristi podatke iz modela da bi ih prezentovao. *ViewModel* sadrži logiku (metode, svojstva i komande), predstavlja apstrakciju *View*-a i nije svestan njegovog postojanja.

3. MOGUĆNOSTI TEHNOLOGIJE

3.1 Adaptivni dizajn

Univerzalne aplikacije mogu da se prilagode različitim verzijama platforme, različitim vrstama uređaja i različitim dimenzijama ekrana. Jedan od glavnih problema, ali i izazova koji je bio pred *Microsoft* kompanijom u vezi univerzalnih aplikacija je da se njihov dizajn korisničkog interfejsa prilagođava širokom spektru uređaja na kojem mogu da se izvršavaju. Ovde se prvenstveno misli na dimenzije i rezoluciju ekrana.

Kada se aplikacija pokrene na određenom uređaju, sistem u pozadini koristi algoritam skaliranja da bi normalizovao prikaz kontrola, fontova, slika i ostalih *UI* elemenata na ekranu. Algoritam skaliranja uzima u obzir pretpostavljenu udaljenost korisnika od ekrana i gustinu, tj. *PPI* vrednost (*Pixels Per Inch*) kako bi optimizovao veličinu elemenata. Algoritam se u pozadini izvršava kompletno automatski i ne mora se posebno implementirati. Način na koji algoritam skaliranja funkcioniše je zbog toga što se dizajn *UWP* aplikacija i veličina njenih grafičkih elemenata obavlja u logičkim

pikselima, a ne u stvarnim fizičkim pikselima. Za razliku od fizičkog piksela koji predstavlja najmanju jedinicu ekrana, logički piksel je nezavisna jedinica ekrana u odnosu na uređaj i po prostoru koji zauzima na ekranu može da sadrži više fizičkih piksela. Logički pikseli omogućavaju da se pri dizajniranju fokus prebaci na unapređenje doživljaja korisnika, bez brige o *PPI* vrednosti, veličini ekrana ili udaljenosti. Faktori skaliranja su predefinisane vrednosti u odnosu na logičku rezoluciju i dijagonalu ekrana i navode se u procentima. Tako se postiže da se uvek u pravom odnosu dobija stvarna veličina logičkog piksela na ekranu uređaja.

3.2 XAML mogućnosti

XAML je deklarativni jezik koji se koristi za inicijalizaciju objekata i vrednosti svojstva tih objekata. To se postiže upotrebom hijerarhijskom strukturom i definisanjem veza između više elemenata. U deklarativnom programskom jeziku, ponašanje i integracija komponenti se opisuju bez korišćenja proceduralnog programiranja. *XAML* je primarni format za deklarisanje korisničkog interfejsa aplikacije (među koje spadaju i univerzalne aplikacije) i svih elemenata koji se nalaze unutar njega, kada se koriste programski jezici *C#*, *Visual Basic* ili *C++/CX*.

Kada se dizajnira korisnički interfejs, uglavnom se koriste predefinisane kontrole, mada postoji i mogućnost pravljenja *custom* kontrola. Najčešće korišćene kontrole su one za interakciju sa korisnikom i za organizovanje elemenata u određenom prostoru. Tu spadaju *Button*, *Check box*, *Combo box*, *Text block*, *Grid*, *Stack panel*, *Canvas* itd. Nove opcije koje dolaze sa *Windows 10* i *Universal apps* tehnologijom su kontrole *Relative panel* i *SplitView*, kompajlirano povezivanje i fazno renderovanje.

3.3 Interakcija sa korisnicima

Kada se razvija univerzalna aplikacija, jedna od ključnih stvari koja mora biti zadovoljena je da aplikacija dobro radi i interaktuje. Interakcija sa korisnikom u velikoj meri se definiše pomoću događaja. Događaji (*Events*) predstavljaju poruke koje šalju objekti prilikom izvršavanja određene akcije. *Universal apps* podržava interakciju sa korisnikom uz pomoć miša, tastature, olovke ili na osnovu dodira ekrana (*Touch input*). Dolaskom univerzalnih aplikacija koje se izvršavaju na različitim uređajima, težilo se pravljenju zajedničke baze funkcionalnosti koja će moći da podržava više familija interaktivnih uređaja pa s tim i više predefinisanih događaja u kontrolama. To je postignuto definisanjem događaja pokazivača (*Pointer Events*) koji spadaju u grupu događaja na osnovu dodira (*Touch Events*). *Touch events* su podeljeni u tri kategorije: pokazivači (*pointers*), *gesture* i manipulacije.

Pointer predstavlja apstrakciju tri različite vrste interakcije: *touch-a*, miša i olovke (*pen*), čime se omogućuje da aplikacija koristi zajednički *API* komplet za stvari koje su zajedničke navedenim vrstama interakcije.

Gesture predstavljaju fizičke pokrete izvršene na ili od strane uređaja za interakciju (jedan ili više prstiju ruke, miš, olovka itd.). One služe za aktiviranje operacija na

korisničkom interfejsu, a način na koji se to postiže zavisi od samog uređaja.

Manipulacije predstavljaju klasu kompleksnih gestura koje mogu biti detektovane od strane *Windows* mehanizma za prepoznavanje gestura. Manipulacije služe za pomeranje (*panning*), rotiranje i zumiranje elemenata. Manipulacije nastaju primenom različitih vrsta transformacija (*TranslateTransform*, *RotateTransform*, *ScaleTransform*) na osnovu informacija dobijenih od pojedinačnih gestura.

4. IMPLEMENTACIJA UNIVERZALNE APLIKACIJE

U aplikaciji za vizuelizaciju elektroenergetske mreže mora da postoji brz i efikasan način za razmenu podataka i njihov prikaz. Pod ovim se misli da se u svakom trenutku mora znati koju vrstu podataka treba prikazati u zavisnosti od zahteva i interakcije sa korisnikom.

4.1 Struktura elemenata i priprema podataka

Elementi elektroenergetske mreže koja se vizualizuje definisani su u *xml* datoteci. U ovoj datoteci definisana su četiri tipa elementa: *SubstationEntity*, *NodeEntity*, *SwitchEntity* i *LineEntity*. Svaki tip elementa ima određene atribute sa zadatim vrednostima, a atributi *Name*, *X* i *Y* su zajednički za sve elemente. *X* i *Y* atributi predstavljaju vrednosti za geografsku širinu i dužinu određenog elementa elektroenergetske mreže. Geografska pozicija *Substation*, *Node* i *Switch* entiteta u mreži je definisana jednim parom *X* i *Y* atributa. *LineEntity* se razlikuje od prethodno navedenih tipova elemenata i predstavlja sekciju kojom se povezuju ostali elementi. Stoga je sekcija definisana nizom *Point* struktura. Svaka *Point* struktura u sebi sadrži jedan par *X* i *Y* atributa.

Pri pokretanju aplikacije, učitava se *xml* datoteka sa definisanim mrežom koja se parsira na osnovu definisanog modela elemenata. Parser prolazi kroz *xml* datoteku i na osnovu modela uzima vrednosti iz učitane datoteke, mapira ih na osnovu definisanih svojstva i smešta u internu strukturu podataka. Krajnji rezultat parsiranja je objekat *Repository* klase koji sadrži podatke za sve tipove elemenata definisanih u listama za svaki tip podataka posebno.

4.2 Iscrtavanje elemenata

Elementi se iscrtavaju na osnovu vrednosti geografske pozicije iz učitane datoteke. Iscrtavanje se izvršava pomoću nekoliko biblioteka u sklopu .NET tehnologije, a najvažnije su *Windows.UI.Xaml.Media* i *Windows.UI.Xaml.Shapes*. U sklopu prve navedene koristi se apstraktna *Geometry* klasa i konkretne klase koje je nasleđuju, a u sklopu *Shapes* biblioteke koristi se *Path* klasa. U sklopu aplikacije, proste geometrije služe za definisanje *Substation*, *Node* i *Switch* tipa elemenata. *Node* entitet se definiše uz pomoć *EllipseGeometry* klase. *Substation* i *Switch* entiteti se definišu preko *RectangleGeometry* klase. S obzirom da *Geometry* klase ne definišu sopstveno renderovanje na ekranu, jer nisu definisane kao *UIElement*-i, one se nakon definisanja navode kao svojstvo vrednost nekog *UIElement*-a da bi izazvali efekat renderovanja. Obično se geometrije

definišu kao *Data* properti u sklopu *Path* klase, koje pripada *Windows.UI.Xaml.Shapes* biblioteci. *LineEntity* se definiše uz pomoć *PathGeometry* klase. Ova klasa spada u kompleksne geometrije. Definisani elementi se iscrtavaju na *Canvasu*. *Canvas* pripada *Windows.Controls* biblioteci i predstavlja najprostiji tip panela.

Glavni problem korišćenih biblioteka za iscrtavanje elemenata je što nemaju podršku za renderovanje 2D grafike uz pomoć *GPU* akceleracije, koja bi u velikoj meri smanjila vreme startovanja i odziva aplikacije.

4.3 Izgled i struktura aplikacije

Aplikacija je testirana na dvema familijama uređaja: *Desktop* koja pokreće *Windows 10* operativni sistem i *Mobile* koja pokreće *Windows 10 Mobile*. Aplikacija cilja *Universal Device Family API* komplet, tako da može biti pokrenuta i na ostalim uređajima. Korisnički interfejs aplikacije na *Desktop* i *Mobile* platformi se u određenoj meri razlikuje. Adaptivnost korisničkog interfejsa aplikacije može da se postigne na dva načina: pomoću *StateTrigger*-a i kreiranjem višestrukih pogleda za različite familije uređaja.

Prikaz elektroenergetske šeme nije statičan, tako da se korisniku dozvoljava pomeranje i zumiranje sadržaja. Pomeranje sadržaja se vrši pomeranjem miša dok je aktiviran levi klik ili povlačenjem prsta na ekranu. Definisana su tri nivoa *zoom*-a i u zavisnosti od toga koji je trenutna vrednost faktora *zoom*-a, aktivira se određeni nivo detaljnosti. Nivoi detaljnosti se definišu jer:

- Pri inicijalnom prikazu šeme, korisniku se ne prikazuju svi elementi jer je vrednost faktora *zoom*-a mala. Ukoliko bi se odmah iscrtili svi elementi pri niskom nivou *zoom*-a, prikaz šeme bi bio nejasan.
- Renderovanje elemenata koji se iscrtavaju na višim nivoima *zoom*-a može da se vrši u pozadini dok je aplikacija startovana i dok se ne dostigne određen nivo detaljnosti, što dovodi do poboljšanja performansi.

5. ZAKLJUČAK

U ovom radu opisana je *Universal apps* tehnologija i njene mogućnosti u okviru razvoja multiplatformskih aplikacija. Data je kratka istorija postepenog razvoja multiplatformskog softvera i univerzalnih aplikacija počevši sa *Windows 8* operativnim sistemom, kao i arhitektura *Windows 10* operativnog sistema i razlike u odnosu na prethodne verzije. Takođe je objašnjen i način prilagođavanja prikaza sadržaja aplikacije na raznim uređajima i veličinama ekrana uz pomoć algoritma skaliranja i logičkih piksela. Poseban deo poglavlja se bavio načinom rukovanja različitim vrsta interakcije sa korisnikom, gde centralno mesto zauzima koncept pokazivača (*pointer*).

U implementaciji su objašnjene korišćene biblioteke i načini njihove primene koji su omogućili vizuelizaciju elektroenergetske mreže. Takođe su prikazane i prednosti razvoja upotrebom definisanog koncepta kao što je *MVVM* šablon. Upotrebom ovog šablona, aplikacija se odlikuje visokim stepenom modularnosti što u kasnijim

fazama razvoja olakšava izmene i dodavanja novih funkcionalnosti.

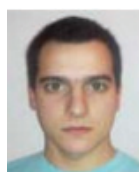
Za unapređenje i poboljšanje aplikacije treba spomenuti problem performansi kod uređaja sa manjom hardverskom moći, a tu se pre svega misli na mobilne uređaje. Pošto ovakve šeme sadrže nekoliko hiljada ili desetine hiljada elemenata, pri učitavanju dolazi do pada performansi kada se koriste grafičke biblioteke navedene ranije u tekstu. Ovo se posebno može primetiti pri korisničkoj interakciji, kada se pri većem nivou *zoom*-a učitava nivo detaljnosti gde se iscrtavaju novi elementi. Što se tiče upotrebe na računaru, pad performansi se uočava pri inicijalnom iscrtavanju mreža sa većim brojem elemenata.

Pravci daljeg razvoja idu ka povećanju funkcionalnosti aplikacije i ubacivanju novog sloja pri iscrtavanju mreže u vidu geografske mape, što bi omogućilo tačnu poziciju elemenata i lakšu orijentaciju korisnika. Ovo bi posebno bilo korisno za mobilne telefone koji poseduju *GPS* i *API* za određivanje trenutne pozicije korisnika, čime bi se učitavao samo deo mreže koji upada u radijus oko lokacije, a to bi dovelo do poboljšanja performansi.

6. LITERATURA

- [1] Guide to Universal Windows Platform apps, <https://msdn.microsoft.com/library/windows/apps/dn894631.aspx>
- [2] A developer's guide to Windows 10, <https://channel9.msdn.com/Series/A-Developers-Guide-to-Windows-10>
- [3] Windows 10 samples, <https://github.com/Microsoft/Windows-universal-samples>
- [4] Comparison of Architecture presentation patterns, <http://www.codeproject.com/Articles/66585/Comparison-of-Architecture-presentation-patterns-M>
- [5] Cross-platform definition, <http://www.linfo.org/cross-platform.html>
- [6] Marija Vidaković, Platformski neutralne arhitekture, FON, Beograd, 2008.
- [7] Andreas Sommer, Comparison and evaluation of cross-platform frameworks for the development of mobile business applications, Faculty of Informatics, Munich, 2012.

Kratka biografija:



Dušan Mandić je rođen 25. septembra 1990. godine u Somboru, Republika Srbija. Školske 2009/2010 je upisao Fakultet Tehničkih Nauka u Novom Sadu, odsek Elektrotehnika i računarstvo, smer Računarstvo i Automatika, usmerenje. Bečelov rad iz oblasti Primenjene računarske nauke i informatika odbranio 2013. god.

JEDNO REŠENJE SPREŽNOG PODSISTEMA ONE SOLUTION OF THE COUPLING SUBSYSTEM

Ivan Morar, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Tema ovog rada jeste projektovanje i izrada sprežnog podsistema čija je uloga povezivanje dve ili više razvojne ploče(modula) kompanije Xilinx korišćenjem brzih komunikacionih linija. Kako bi se rešio problem sinhronizacije između modula implementirano je sinhronizaciono kolo čija je uloga da obezbedi precizno podešavanje frekvencije takta. Ovaj sprežni podsistem poseduje i podršku za USB komunikaciju i konektor za dodatni IO interfejs.

Abstract – The subject of this paper is development and production of the coupling subsystem whose purpose is to make connection between a two or more Xilinx development boards (modules) using fast transmission lines. In order to solve synchronization problem between modules, synchronization circuit has been developed which has purpose to enable fine adjustment of clock frequency. This coupling subsystem has support for the USB communication and connector for additional IO interface.

Ključne reči: Modularni sistemi, Brza serijska komunikacija, Sinhronizacija.

1. UVOD

Ugrađeni računarski sistemi predstavljaju deo nekog većeg sistema u kojem vrše upravljanje fizičkim procesima. Iz tog razloga ugrađeni sistemi se još nazivaju i upravljački sistemi. Iako su projektovani da rešavaju jednostavnije zadatke, vremenom zahtevi koji se postavljaju ispred ovih uređaja postaju sve veći. Od njih se zahteva sve veći broj funkcija koje mogu obaviti, obično za što kraće vreme. Kao posledica ugrađeni uređaji postaju sve složeniji i samim tim proces proizvodnje postaje sve kompleksniji. Jedan od načina koji obezbeđuje uvećanje performansi je paralelizacija.

Korišćenjem paralelizacije moguće je ostvariti modularni dizajn, tj. napraviti sistem koji se sastoji od više nezavisnih funkcionalnih jedinica koji se nazivaju moduli. Ovaj pristup omogućava lako skaliranje sistema prema veličini problema koji treba rešiti. Pored uštede u ceni sistema i fleksibilnosti dizajna, modularan sistem nudi pogodnosti kao što su lako proširivanje sistema i laka zamena neispravnih modula. Pored prednosti modularnog dizajna postoje i ograničenja. Moduli unutar sistema moraju biti u mogućnosti da ostvare međusobnu komunikaciju što podrazumeva vezu preko komunikacione linije.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dušan Majstorović, docent.

Povećanjem broja modula raste broj poruka koji se razmenjuje što rezultuje potrebom za većom propusnom moći komunikacione linije.

Dodatni problem koji se javlja u modularnim sistemima je problem međusobne sinhronizacije modula. Ovaj problem se javlja kod sistema koji rade usaglašeno, tj postoji zahtev da se ponašaju kao jedan. To su obično sistemi koji rade u realnom vremenu. Stoga uloga sinhronizacije je da omogući sistemu izvršavanje operacija u pravilnom redosledu.

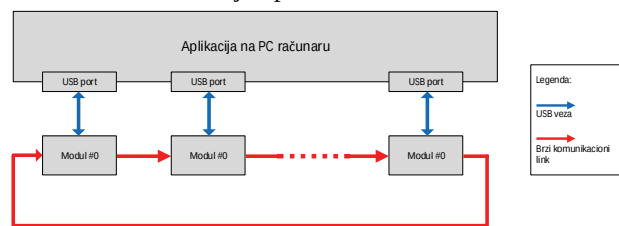
2. TEORIJSKA RAZMATRANJA

2.1. Modularnost sistema

Ideja modularnog dizajna sistema je da se dizajnira takav sistem koji bi omogućio lako dodavanje nezavisnih jedinica – modula. Kako bi to bilo moguće potrebno je ostvariti interfejs koji bi spajao module u jednu celinu. Uloga interfejsa je da omogući međusobnu komunikaciju između modula.

Za ostvarivanje brze komunikacije iskorišćeni su brzi serijski primopredajnici koji su implementirani u Xilinx čipovima. Dodatno povećanje propusne moći ostvareno je paralelizacijom serijskih primopredajnika. Maksimalno je podržano 8 primopredajnih linija.

Za povezivanje modula iskorišćena je *Daisy Chain* konfiguracija. U ovoj konfiguraciji podaci putuju u jednom smeru tako da svaki modul mora imati konektor za prijem i za predaju podataka. Pored brzog serijskog interfejsa, sprežni podsistem poseduje USB komunikacioni modul. Njegova uloga je povezivanje aplikacije na PC računaru i modula - Xilinx razvojne ploče.

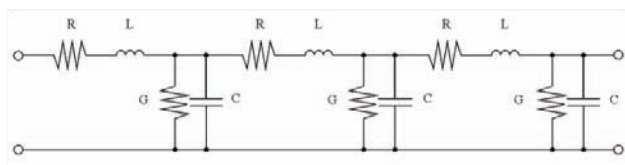


Slika 1. Daisy Chain komunikaciona topologija sa USB komunikacijom.

2.1. Brza serijska komunikacija

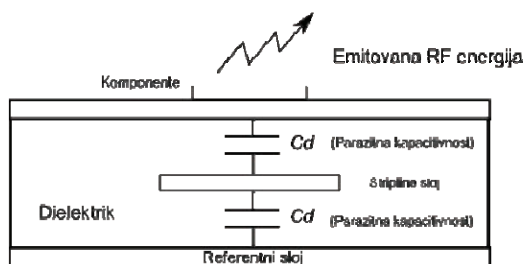
Za serijsku komunikaciju korišćeni su signali frekvencije reda GHz. Na ovim frekvencijama provodnici se posmatraju kao transmisioni vodovi kod kojih se moraju uzeti u obzir parazitne komponente provodnika. Ukoliko se ne obraća pažnja o realizaciji prenosnih vodova prilikom izrade štampane ploče, može doći do narušavanja tajminga signala ili do narušavanja integriteta signala što dovodi do grešaka prilikom prenosa podataka i time sistem postaje neupotrebljiv.

Vodovi tokom svoje dužine moraju imati uniformnu geometriju. To znači da im je kapacitivnost i induktivnost bilo kog segmenta ista. Ove parazitne induktivnosti i kapacitivnosti ne zavise od dužine provodnika i mogu se predstaviti kao beskonačni niz induktivnosti i kapacitivnosti [1].



Slika 2. Ekvivalentna šema realnog voda.

Prilikom projektovanja komunikacionih vodova korišćena je *stripline* tehnologija. Prednost ove tehnologije jeste zaštita od spoljašnjih smetnji i sprečavanje emitovanja RF signala u okolinu [2].



Slika 3. Presek voda u stripline tehnologiji.

U cilju smanjenja uticaja elektromagnetnih smetnji, signali na visokim frekvencijama se prenose pomoću komplementarnih električnih signala – diferencijalnog para. Još jedna prednost diferencijalnih vodova jeste minimizacija elektromagnetnih smetnji koje generiše signalni par. Međutim na štampanoj ploči nije moguće upresti provodnike tako da se signali vode paralelno. Efikasnost poništavanja smetnji upredenih parica može ići čak i preko 99%, dok je kod paralelnih parica efikasnost daleko manja, obično manje od 50% [3].

Tipična impedansa usamljenog voda koja se koristi za prenos signala je 50Ω , tako da je tipična vrednost diferencijalnog voda duplo veća i iznosi oko 100Ω . Impedansa vodova u *stripline* tehnologiji je računato po formuli:

$$Z_{diff} \approx 2 * Z_0 \left(1 - 0,347e^{-2,9\frac{D}{B}}\right) \Omega \quad (1)$$

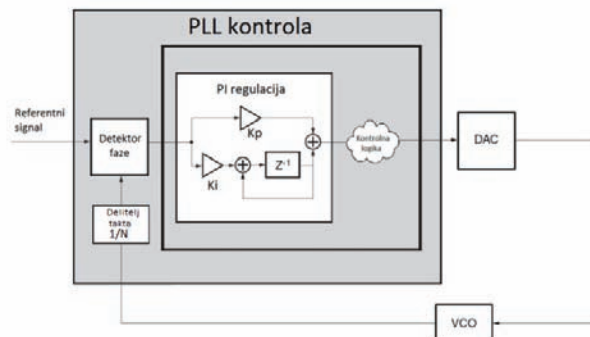
Ukoliko se ne vodi računa o impedansi duž transmissionog voda dolazi do pojave refleksije signala. U tom slučaju integritet signala se narušava i dolazi do grešaka u prenosu. Koeficijent refleksije se može predstaviti kao:

$$\rho = \frac{Z_{term} - Z_0}{Z_{term} + Z_0} \quad (2)$$

2.2. Koncept sinhronizacije sistema

Sinhronizacija sistema se vrši prosleđivanjem podatka o taktu kroz komunikacionu liniju. Ovaj podatak o sinhronizaciji prosleđuje se nekoliko puta u sekundi. Dobijeni podatak o sinhronizaciji sadrži informaciju o fazi signala glavnog modula koja se upoređuje sa fazom lokalnog oscilatora. Tu razliku detektuje detektor faze i informaciju o grešci prosleđuje u PI regulator.

Pošto je PI regulator realizovan u digitalnoj logici, kao izlaz se dobija digitalni podatak. Taj podatak je transformisan u odgovarajući format koji se šalje DA konverteru koji generiše napon za upravljanje VCO-om. Na ovaj način petlja za PLL je zatvorena. PLL kontrola je realizovana u FPGA logici na razvojnim pločama, dok su DAC i VCO sastavni deo spreznog interfejsa.

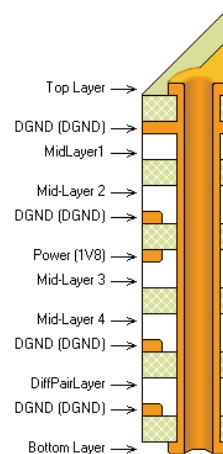


Slika 4. Blok dijagram PLL kontrole.

Cilj sinhronizacije je da eventualnu razliku između takta dva oscilatora iste frekvencije svede na minimum. Iako su oscilatori iste osnovne frekvencije oni nikad ne mogu biti u potpunosti isti. Njihovo tipično odstupanje je oko ± 100 ppm-a. U kratkim vremenskim intervalima ova greška je zanemarljiva, međutim ukoliko uređaji rade više sati, greška se akumulira.

3. REALIZACIJA SPREŽNOG PODSISTEMA

Zbog velikog broja komunikacionih linija između konektora za IO ploču, FPGA ploču i brzih komunikacionih linija, ploča je morala da bude projektovana u velikom broju slojeva. Korišćeno je 7 signalnih slojeva i 5 slojeva mase i napajanja.



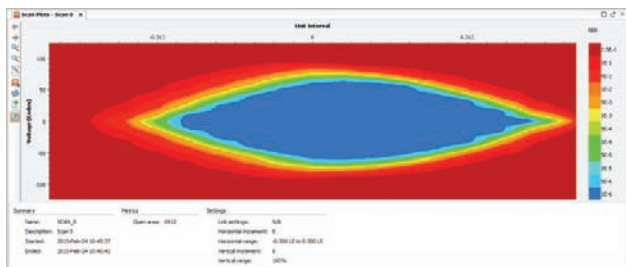
Slika 5. Raspored slojeva na sprežnom podsystemu.

Prilikom određivanja slojeva posebna pažnja je data sloju u kome se nalaze diferencijalni parovi za brze signale. U tom sloju je bilo neophodno kontrolisati impedansu vodova. Kao materijal od kojeg je napravljena štampana ploča je FR4 koji nema preterano dobre karakteristike, ali zbog male dužine brzih komunikacionih linija procenjeno je da neće biti prevelikog uticaja na integritet signala.

Projektovanje štampane ploče urađeno je u softverskom paketu Altium Designer. Ovaj softverski paket nudi alate koji omogućavaju jednostavno rutiranje diferencijalnih signala.

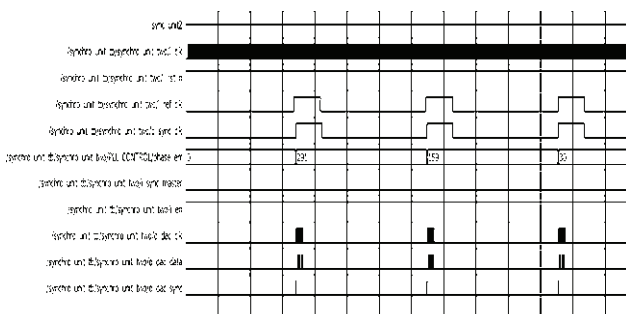
4. REZULTATI

Nakon fizičke realizacije sprežnog podsistema izvedena su testiranja funkcionalnosti zadate specifikacijom. Testiranje brzih komunikacionih linija urađeno je uz pomoć *LogiCORE™ IP Integrated Bit Error Ratio Test (IBERT)* IP jezgra [4]. Prilikom testiranja generisane su bit sekvence uz pomoć PRBS31 pseudo slučajnog algoritma. Komunikacija je uspešno ostvarena na svim komunik. linijama. Izgled dijagrama oka dat je na slici:



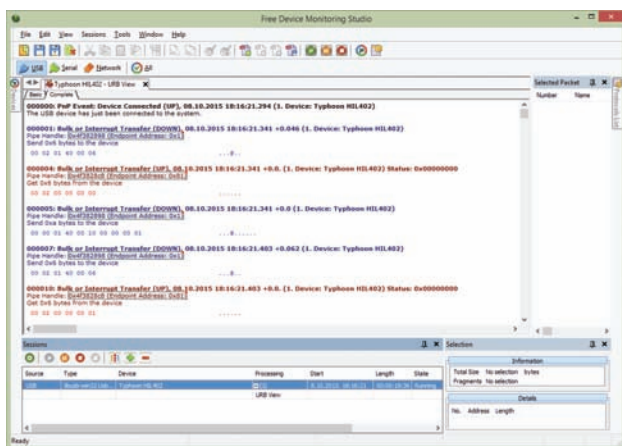
Slika 6. Dijagram oka jedne komunikacione linije.

Verifikacija modula za sinhronizaciju urađena je uz pomoć simulacije. Na slici 7 prikazani su signali simulacije modula za sinhronizaciju. Signal *phase_err* predstavlja faznu grešku. Može se videti kako se vrednost fazne greške smanjuje svakim sledećim sinhronizacionim paketom koji se šalje.



Slika 7. Signali sinhronizacionog modula.

USB komunikacija je testirana uz pomoć softverskog alata za koja omogućuje nadgledanje USB saobraćaja između razvojne ploče i aplikacije na računaru. S obzirom da se uređaj uspešno prijavio u *Device Manager-u* i da se komunikacioni paketi nesmetano razmenjuju može se smatrati da je USB komunikacija uspešno ostvarena.



Slika 8. Razmena komunikacionih paketa na USB-u.

5. ZAKLJUČAK

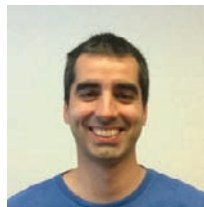
U ovom radu je projektovan i izrađen sprežni podsistem za povezivanje više FPGA razvojnih ploča kompanije Xilinx. Za realizaciju ovog rada bilo je neophodno sticanje i usavršavanje znanja iz oblasti transmisionih vodova i tehnike sinhronizacije uređaja. Svi zadaci postavljeni specifikacijom uspešno su rešeni i dobijen je u potpunosti funkcionalan uređaj.

Dodatna poboljšanja koja bi se mogla primeniti su povećanje broja komunikacionih linija sa dosadašnjih 8 na 16 čime bi se omogućilo povećanje propusnog opsega i ugradnja dodatnog oscilatora čija je uloga generisanje takta za brze komunikacione linije koji bi olakšao proces rutiranja takt signala kroz FPGA čip.

6. LITERATURA

- [1] Stephen H. Hall, Garrett W. Hall, James A. McCall, "High-Speed Digital System Design - A Handbook of Interconnect Theory and Design Practices", New York, Wiley, 2000.
- [2] Mark I. Montrose, "Printed Circuit Board Design Techniques for EMC Compliance: A Handbook for Designers", IEEE the Institute of Electrical and Electronics Engineers, Inc., New York, 2000.
- [3] „Differential Pair Routing“, <http://techdocs.altium.com/display/ADOH/Rule+Definition+for+Differential+Pair+Routing>, (20.3.2015.)
- [4] Harry Fu, Romi Mayder, „Serial Link Signal Integrity Analysis with IBIS-AMI Simulation and On-Chip Eye Scan for Low-Cost, High-Volume FPGA Transceivers“, Xilinx, 2012.

Kratka biografija:



Ivan Morar rođen je u Pančevu 1984. god. Osnovne akademske – bečelor studije završio je na Fakultetu tehničkih nauka u Novom Sadu iz oblasti Elektrotehnike i računarstva – na smeru za mikroracunarsku elektroniku 2012. godine.

**PRIMENA TEORIJE INFORMACIJA NA OPTIMIZACIJU POSTUPKA
RECENZIRANJA****APPLICATION OF THEORY OF INFORMATION AND COMMUNICATION ON
EVALUATION PROCES**Bojan Šibar, Vojin Šenk, *Fakultet tehničkih nauka, Novi Sad***Oblast - ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu opisan je proces recenziranja predstavljen kao problem optimizacije određivanja redosleda aplikacija na nekom konkursu. Optimizovan je postojeći algoritam za određivanje redosleda aplikacija. Predložen je i postupak za određivanje kvaliteta recenzenata. Algoritam je upoređen sa aktuelnim stanjem u toj oblasti

Abstract – In this paper, the evaluation process is described as the problem of determining the optimal sequence of applications to a contest. Existing evaluation algorithm is optimized for the purpose of determining the right order of applications. A new procedure for determining the quality of evaluators is developed. The algorithm is compared with existing algorithms for the purpose.

Cljučne reči: Rangiranje alternativa, Rangiranje recenzenata, Potiskivanje uticaja manjka kompetencije recenzenata

1. UVOD

Problem optimizacije određivanja redosleda potencijalnih aplikacija na nekom konkursu aktuelan je u mnogim oblastima i nedovoljno dobro rešen. Postoje različiti algoritmi za ovu svrhu, kao što su MCDA [1], poređenje parova [2], PAPRIKA [3].

Nerešen problem je traženje algoritma koji će na osnovu menjanja ulaznih parametara odrediti preciznost konačnog redosleda. Takav algoritam treba da pokaže vezu između evaluatora – osoba zaduženih za pregledanje i poređenje entiteta i njihovih stručnosti.

**2. DOSADAŠNJI PRISTUPI PROBLEMU
RECENZIRANJA**

Multi criteria decision analysis, ili MCDA, možemo primeniti pri donošenju mnogih složenih odluka. To se najviše odnosi na rešavanje problema izbor između više alternativnih rešenja. MCDA nam pomaže da se fokusiramo na ono što je nam je u datom momentu važno; logičan je i dosledan, i jednostavan za korišćenje. U suštini, MCDA je koristan za

podelu odluka u manje i razumljivije delove,
analizu svakog dela posebno, i
integraciju delova za proizvodnju konačnog rešenja.

Kada se koristi za grupno donošenje odluka, MCDA pomaže učesnicima da razmotre vrednosti koje svaki

od njih vidi kao važan parametar. Ona takođe omogućava ljudima da dođu do kompromisa, kao i da testiraju svoju konačnu odluku [1].

MCDA se sastoji od sedam komponenti:

1. zacrtani cilj,
2. pojedinac ili grupa donosioca odluka (evaluatori),
3. identifikacija opcija i alternativa pri postizanju cilja,
4. kriterijumi za evaluaciju (interesi) i njihova analiza,
5. ishodi ili posledice u vezi sa alternativnim izborom,
6. donošenje odluke,
7. povratne informacije.

Upoređivanje parova je proces upoređivanja entiteta sa ciljem da se pronađe bolji u odnosu na kriterijume koji sami zadamo, tj. da odredimo koji ima veću količinu nekih kvantitativnih osobina, ili da proverimo da li su dva entiteta identična. Metod poravnavanja parova, kako se još naziva, koristi se u naučnim istraživanjima preferencija, stavova, izbornih sistema, društvenog izbora, itd [2]. Način poređenja parova je eksplicitno dizajniran da zadovolji Kondorseov kriterijum koji se bavi pravičnošću proglašenja kandidata za pobednika iako je neki drugi kandidat dobio sve moguće mečeve jedan na jedan (head-to-head). Poređenje parova obezbeđuje da kandidat koji je dobio sve moguće mečeve uvek ima veći broj bodova ukupno nego bilo koji drugi kandidat [2].

Metod PAPRIKA se odnosi na modele vrednosti za rangiranje unapred određenih alternativa za koje se zna koji je zadatak donosioca odluka (npr. kao kada treba uporediti sve kandidate za posao) i modela za rangiranje potencijalno svih hipotetički mogućih alternativa koje se menjaju tokom vremena (npr. pacijenti koji apliciraju za zdravstvenu negu). Kao analogiju, pretpostavimo da želimo da rangiramo ljude od najmlađih do najstarijih. Ukoliko znamo kako je svaka osoba rangirana u odnosu na sve ostale (upoređivanje parova) u vezi sa njihovim godinama - odnosno za svaki mogući par pojedinaca smo identifikovali ko je mlađi od dva lica ili da li su istih godina - onda možemo da proizvedemo ukupno rangiranje takmičara od najmlađih do najstarijih.

Međutim, u zavisnosti od broja kriterijuma i kategorija, broj rangiranja parova svih mogućih alternativa potencijalno se meri u milionima ili čak milijardama. Mnoga od ovih rangiranja parova su automatski rešena zbog jednog para koji ima višu kategoriju za najmanje jedan kriterijum i nijednu znatno manju kategoriju za druge kriterijume nego druge alternative - poznat kao "dominantan par". Ovo ipak ostavlja potencijalno milione ili milijarde "nedominantnih parova".

NAPOMENA:

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

3 ALGORITAM KORIŠĆEN NA TAKMIČENJU ZA NAJBOLJU TEHNOLOŠKU INOVACIJU 2005-2015

Ovaj master rad ima za zadatak da simulira proces određivanja redosleda N potencijalnih alternativa – ideja. Svaka ideja prolazi kroz proces recenziranja u S sekcija, gde svaka sekcija ima Q_s pitanja na koje svaka ideja dobija određen odgovor koji se preslikava u broj bodova. Proces recenziranja obavljaju evaluatori. Evaluatori su nezavisni od ideja i njihov zadatak je da ocenjuju ideje po unapred zadanom modelu. Potencijalni odgovori su takođe unapred poznati i dati su u modelu recenzije, tako da evaluator samo bira jedan od njih (npr. daje odgovor da ili ne, ili bira između više ponuđenih opcija). Numerička vrednost odgovora određuje se kasnije. Recenziranje se odvija u dve etape. U obe se zapravo određuje rang lista ideja po svakoj sekciji, pa se onda te rang liste ponderišu i tako dobijeni poeni koriste za formiranje konačne rang liste. Prva etapa se teorijski gledano oslanja na MCDA. Ideja može biti evaluirana više puta od strane više evaluatora. Ideje se nasumice dodeljuju evaluatorima koji ih zatim ocenjuju, a svaki odgovor ima gradaciono veću vrednost. Ideje se zatim linearizuju i rangiraju, a procesi linearizacije i rangiranja će biti detaljnije opisani. U drugoj etapi, ideje se zavisno od predhodnog ranga uparuju u trojke – triplete. Ovaj deo se razlikuje od prvog po tome što se ideje više ne ocenjuju po nekom unapred zadanom modelu nego se takmiče međusobno, slično kao u PAPRIKA metodu.

Kod MCDA prvo utvrđujemo koji kriterijumi su od veće a koji od manje važnosti, pa tako radimo i ovde. Bodovi za svako pitanje ponderišu se tako da ukupan zbir pondera bude 1. Takođe, nisu sve sekcije iste važnosti, pa tako svaka od njih nosi određenu težinu a zbir pondera je opet jednak 1. Jedna ideja može biti evaluirana više puta i to od strane različitih evaluatora. Ovim želimo da postignemo da jedna ideja ne zavisi samo od jedne evaluacije tj. od ocena samo jednog evaluatora, nego želimo da istu ideju pregleda više evaluatora pa ćemo uprosečiti njihove odgovore i iz toga izvući rang ideje.

Evaluator je entitet koji je nezavisan od ideje i od recenzije. Evaluator je zamišljen da predstavlja osobu koja će pregledati tj. recenzirati ideju. Svakog evaluatara određuju unapred poznati koeficijenti stručnosti. Kako smo napomenuli ranije da svaki odgovor ima određene težine za svako polje koje figurira u odgovoru, tako i evaluator ima određene koeficijente stručnosti za svako od datih polja. Koeficijenti stručnosti su celi brojevi koji se kreću u vrednostima od 1 do 7. Evaluator ne može imati iste koeficijente stručnosti za dva polja, a zbir koeficijenata stručnosti mora biti u opsegu od 12 do 20.

Ideja je entitet nad kojim vršimo sva recenziranja. Iz liste evaluatora biramo jednog na slučaj, ali tako da istoj ideji ne možemo dodeliti dva evaluatora sa istim najboljim poljem stručnosti. Simulirani evaluator zavisno od mesta koje bi ideja trebalo da zauzme generiše svoj odgovor (viša pozicija, viša ocena). Kako evaluator predstavlja osobu koja pregleda alternative, ljudski faktor je predstavljen varijansom koju simuliramo Gausovim generatorom slučajnih promenljivih. Ulazni parametri za svaki odgovor su očekivana vrednost odgovora i varijansa evaluatora. Srednja vrednost odgovora se računa tako što se podeli inverzni predviđeni rang ideje sa ukupnim

brojem ideja i pomnoži brojem mogućih odgovora. Npr. ukoliko postoji 100 ideja za recenziju i evaluator treba da odgovori na pitanje koje ima 4 moguća odgovora, i ako ideja koju pregleda treba da zauzme 40 mesto, srednja vrednost za taj odgovor će biti 2.4. Odgovor se zatim zaokružuje na ceo broj i to predstavlja konačni odgovor.

Kako je svaka ideja pregledana od strane više evaluatora, neophodno je izračunati srednju ocenu za svako pitanje, ponderišući svaki odgovor sa stručnošću recenzenta za dato pitanje. Nakon ovog pristupa se linearizaciji. Linearizacija je proces u kom se formira nova lista, za svako pitanje, tako da prvi tim ima broj poena koji je jednak ukupnom broju ideja, a poslednji tim ima 1 poen, i tako formirana lista se množi ponderom za određeno pitanje. Proces ponavljamo u okviru sekcije, sabiranjem ponderisanih poena sa svakog pitanja unutar nje. Opet se formira nova linearizovana lista za svaku sekciju, i tako formirana lista se množi ponderom za određenu sekciju i određuje konačan rang prvog kruga.

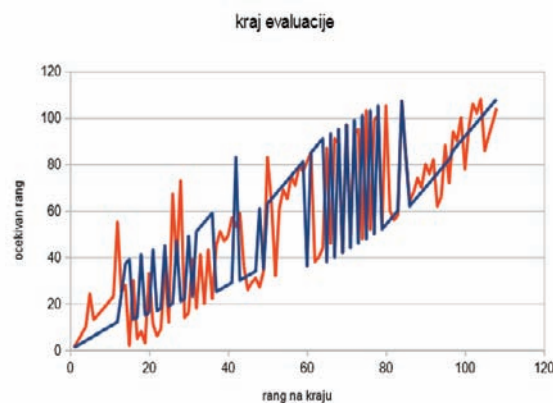
Ulaz u drugu fazu je rang lista iz prvog kruga. Recenziranje u drugoj fazi se odvija u više podfaza gde kraj jedne podfaze označava početak druge. Podfaze se razlikuju po načinu grupisanja ideja, tako da je u svakoj sledećoj podfazi rastojanje ideja na dotadašnjoj rang listi sve manje. Formiraju se tripleti i dodeljuju evaluatorima koji pristupaju ocenjivanju po istim sekcijama kao i u prvom krugu. Evaluator poredi 3 prijave po svakoj sekciji i za svaku sekciju bira najslabiju prijavu. Za svaku ideju se saberu ponderi sekcija u kojima je bila najlošija, dodeli im se negativni predznak i pomnože se sa brojem ideja koje se ocenjuju. Tom proizvodu se dodaje trećina od broja ideja koje se ocenjuju i dodaje početnom stanju sa rang liste (odnosno stanju poena sa rang liste iz prethodne faze), i lista se presortira i linearizuje.

Primer: 60 ideja. Evaluator dobije da ocenjuje ideje A, B i C. Ideja A ima zbir pondera od 0.15, ideja B ima 0.35 i ideja C ima 0.50.

Ideja A: $-0.15 \times 60 + 60/3 = 11$ Ideja B: $-0.35 \times 60 + 60/3 = -1$ Ideja C: $-0.5 \times 60 + 60/3 = -10$

4. REZULTATI I ANALIZA ALGORITMA ZA IZBOR NAJBOLJE TEHNOLOŠKE INOVACIJE

Rezultati su bili u okvirima očekivanih. Preciznost algoritma smo merili tako što smo izračunali apsolutnu razliku mesta koje je jedna ideja zauzela na kraju u odnosu na njen očekivani rang. Korišćeno je 99 ideja, a rezultat je prikazan na slici 1.



Slika 1. Prikaz rezultata prvog načina rada

5. UOČENI PROBLEMI

Posmatrajući drugu fazu postavilo se pitanje: šta bi se desilo u idealnom slučaju, da su na ulazu u drugu fazu ideje poređane u perfektnom redu i da se daju na evaluaciju perfektnom evaluatoru. Primer za ovo u slučaju 9 ideja dat je na slici 2.

9 test	9	12 test	9	14.25 test	9	15.75	0
test 8	8	11 test	8	13.25 test	8	14.75	0
test 7	7	10 test	6	11.25 test	6	8.25	-1 faza 1: 2 pobede, poraz
test 6	6	9 test	4	9.25 test	7	7	1 faza 2: 2 pobede, poraz
test 5	4	7 test	7	5.5 test	4	6.25	-1 faza 3: 2 pobede, poraz
test 4	2	5 test	5	1.25 test	5	2.75	1 najljaci i 2 losija
test 3	5	-1 test	2	0.5 test	2	2	-1
test 2	3	-3 test	3	-0.75 test	3	0.75	1
test 1	1	-5 test	1	-9.5 test	1	-12.5	0

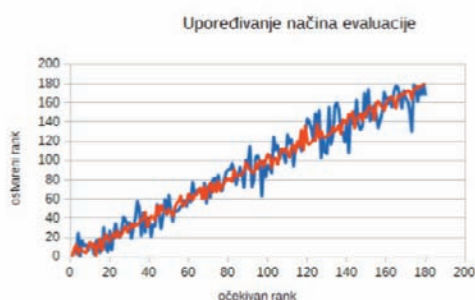
Slika 1. Uočeni problemi druge faze

Kao što se vidi sa slike 2, perfektan evaluator je na kraju malo izmešao ulaznu sekvencu. Radi lakšeg praćenja ideje su obojene pa da se mogu lakše pratiti njihovi skokovi i padovi tokom faza.

6. UNAPREĐIVANJE FAZE 2

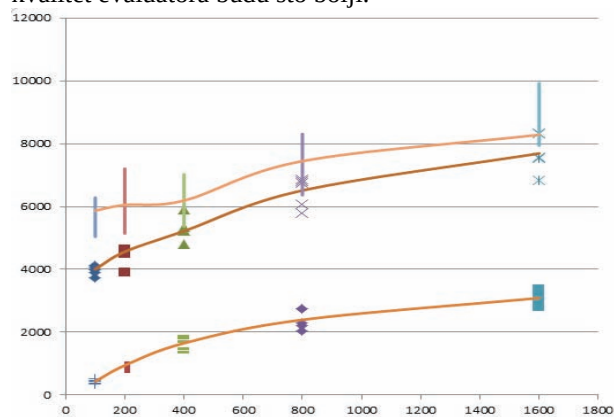
Umesto dosadašnjeg načina rada, ocenjivanje u drugoj fazi će biti malo promenjeno. Ceo koncept faze dva će ostati manje-više isti. Nakon prve faze ideje se sortiraju i uparuju se u triplete kao što je i ranije rađeno ali sa jednom manjom izmenom. Naime, evaluator kome je triplet dodeljen ideje rangira od najbolje ka najlošijoj, odnosno odabire koja od ideja je najbolja a koja najlošija za svaku sekciju, ili čak i pitanje unutar sekcije. Pošto sada nemamo promene u bodovima, sa novim pristupom, potencijalni savršeni evaluator bi sa ovim sistemom ocenjivanja samo razmestio ideje za svaku od sekcija, tako što bi one jednostavno zamenile mesta. Pošto više evaluatora pregleda istu ideju u različitim tripletima, novo mesto se određuje uprosečavanjem (uzimajući u obzir i kompetenciju recenzenta za datu sekciju odnosno pitanje), što kad se linarizuje daje novo stanje rang liste. Takođe, ovaj sistem za savršen raspored, za savršenog evaluatora ne unosi nikakve promene u konačnom redosledu.

Kada ponovimo ceo proces ispočetka i unesemo ova poboljšanja, rezultati ukupnog simuliranog rangiranja su se veoma popravili. Greška koja se sada javlja na kraju iznosi u nekim slučajevima i manje od 5%. Ovo znači da je na uzorku od 180 ideja za slučaj sa uobičajenim varijansama evaluatorska svaka ideja pomerena gore ili dole za najviše devet mesta. Rezultati se mogu videti na slici 3.



Slika 3. Smanjenje greške novim načinom recenziranja u drugoj fazi

Nakon ovog, ispitano je kako se preciznost rang liste iz prve faze preslikava na preciznost rang liste nakon druge faze. Na slici 4. upoređeno je ukupno apsolutno odstupanje konačne rang liste u zavisnosti od prosečne varijanse evaluatorske i preciznosti početne rang liste. Najniža kriva data je za slučaj da je početna rang lista ujedno i tačna. Srednja kriva odnosi se na rang listu dobijenu nakon prvog kruga opisanog ranije. I, konačno, najviša kriva pokazuje zavisnost konačne rang liste od prosečne varijanse evaluatorske za potpuno slučajno početno redosled ideja. Vidi se da je važno da i početna rang lista i kvalitet evaluatorske budu što bolji.



Slika 4. Zavisnost apsolutnog odstupanja konačne rang liste od varijanse i uređenosti inicijalne rang liste dobijene nakon prvog kruga recenziranja

7. RANGIRANJE EVALUATORA

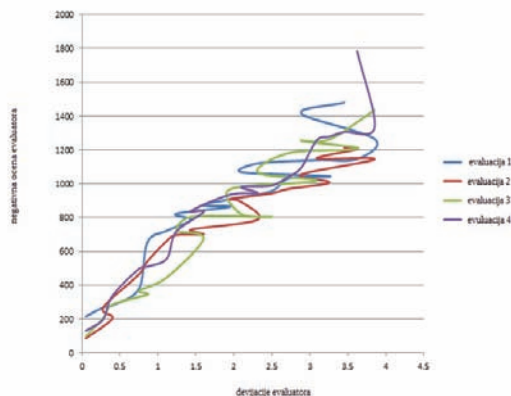
Kako prethodni metodi rangiranja evaluatorske nisu pokazali ono što smo očekivali, pristup je ponovo promenjen.

Ukoliko pogledamo koji konačan plasman ima svaka ideja i zatim pogledamo, koje mesto je ista ideja zauzela posmatrajući svako pitanje posebno vidimo da su na neka pitanja ideje znatno odstupale od svojih konačnih rasporeda. Pretpostavka je da su baš u odgovorima na ova pitanja, evaluatorske davali odgovore koji veoma odstupaju. Ako se podsetimo načina na koji evaluatorske daju ocene: evaluator će uzeti redni broj koje bi ideja trebalo da zauzme, oduzme ga od broja ideja, zatim podeli sa brojem ideja i dobija broj u intervalu od [0,1]. Taj broj se onda množi sa brojem odgovora na dato pitanje i dobijamo prvi parametar za Gausov generator slučajnih brojeva, dok je drugi parametar evaluatorske varijansa.

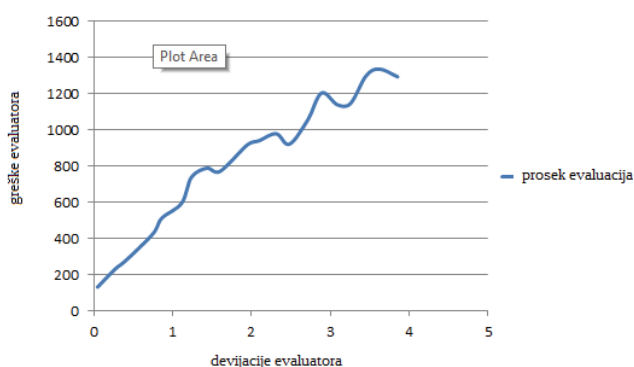
Za očekivati je da najbolja ideja dobija samo najviše ocene. Sve ostalo možemo posmatrati kao grešku evaluatorske. Međutim, ovo važi za slučaj kada znamo krajnju rang listu. Zato ćemo uzeti da je rezultat na dobijen na kraju u granicama od po npr. 10%. Ovo znači da je ideja 1, koja je npr. zauzela 14. mesto od 180 mogućih, mogla da zauzme bilo koje mesto od 14 ± 18 , tj. bilo koje mesto u opsegu od 1 do 32. S tim u vezi, za očekivati je da ta ideja zauzme iste pozicije posmatrajuci rang listu za svako pitanje posebno. Ukoliko je ideja van okvira u kojem smo projektovali tj. pretpostavili njen rang, uzećemo evaluatorske za tu ideju i dodeliti negativnu ocenu evaluatorske.

Negativne ocene ćemo dodeljivati po sledecem principu. Ukoliko je ideja van okvira koji smo pretpostavili, za

određeno pitanje, na osnovu okvira koji smo izračunali, izračunamo i odgovore koje ideja treba da dobije. Pa tako za ideju 1, okviran odgovor bi bio u opsegu: broj ideja umanjeno za poziciju podeljen ukupnim brojem ideja, što za okvir (1,32) iznosi [1, 0.822] ukoliko bi ocene išle od 1 do 4. Okviran odgovor se zatim množi brojem mogućih odgovora, što je u našem primeru 4, i to daje odgovor u opsegu [4, 3.28].



Slika 5. Određivanje ranga evaluatora na osnovu uticaja na konačnu rang listu (stari metod)



Slika 6. Određivanje ranga evaluatora na osnovu toga što je dao ocenu koja izlazi iz predviđenog opsega (novi metod)

Posmatrajući slike 5. i 6. vidi se da je nov, nelinearni postupak bitno unapredio proces rangiranja evaluatora, što je bio i konačni cilj ovog master rada. Uspešno su izolovani evaluatori koji su uneli najveće poremećaje u konačan plasman.

Grafik sa slike 6. bi za dovoljan broj ponavljanja konvergirao ka linearnoj funkciji. Logično je i za očekivati da lista ima mala talasanja. To je posledica Gausovog generatora slučajnih brojeva, zato što imamo konačan broj ponavljanja i pojedini evaluatori nisu stigli da iskažu svoju pravu prirodu (neki su dali bolje, a neki lošije ocene od onih koje bi u proseku dali).

Sada kada imamo podatke koji od evaluatora je koliko grešio tj kolika mu je devijacija možemo da pokušamo da na neki način njegove ocene tj. njegov uticaj na konačnu listu u nekoj meri ublažimo. Rezultati pojedinih evaluatora se mogu posmatrati kao manje ili više validni za konačan plasman.

7. ZAKLJUČAK

Postojeći algoritam koji je do sada korišćen za pregledanje alternativa za potrebe pronalaženja najboljih iz skupa alternativa je vrlo dobar i pošto je i do sada bio korišćen u praksi nije do sada davao loše rezultate, naprotiv. Kroz ovaj rad, međutim, uočeni su potencijalni propusti i pronađena su mesta gde bi ovaj algoritam mogao biti dalje unapređen.

Unapređena verzija je kroz ovaj istraživački rad testirana i izvučeni su neki korisni zaključci. Sa dodatim unapređenjima već postojećem algoritmu je povećana preciznost i ostaje samo da se isproba u praksi, za šta i jeste namenjen. Takođe, ova poboljšanja su otvorila vrata, da se i sami evaluatori rangiraju, odnosno da se i njihove stručnosti uporede.

Poređenjem i rangiranjem stručnosti i samih evaluatora, ostavlja se mogućnost da se ovaj algoritam i dalje unapredi. Smanjivanjem, odnosno prigušivanjem koeficijenata, postoji mogućnost da se uticaj evaluacije recenzenata sa velikom varijansom na konačni rezultat umanjuje, i tako poboljša konačna rang lista, ali ovo nije bio zadatak ovog rada, i ostaje kao zadatak za dalja istraživanja.

Takođe, jedna od ideja jeste da ovaj rad posluži kao osnova za pravljenje procesa recenziranja gde bi sami predlagači ideja pregledali konkurentne aplikacije. Time bi bili uštedeni ogromni resursi, pošto angažman evaluatora košta. Delimičnim ili potpunim izostavljanjem evaluatora ideje bi deo poena dobijale kao i do sada iz plasmana na kraju, a deo iz toga koliko dobro njihovi predlagači pregledaju druge ideje, što opet na neki način govori o stručnosti samih osoba koje stoje iza određene alternative.

U poređenju sa ostalim algoritmima koji se koriste za ovu svrhu i dosadašnja verzija je bila unikatna, a nova bi predstavljala značajan pomak zato što bi se sa mnogo manje resursa dobijao gotovo isti kvalitet rang liste na kraju.

8. LITERATURA

- [1] https://en.wikipedia.org/wiki/Multiple-criteria_decision_analysis version of 28 September 2015, 13:07.
- [2] https://en.wikipedia.org/wiki/Pairwise_comparison version of 14 September 2015, 22:30.
- [3] https://en.wikipedia.org/wiki/Potentially_all_pairwise_rankings_of_all_possible_alternatives version of 21 October 2015, 00:16.

Kratka biografija:



Bojan Šibar rođen je u Novom Sadu 1989. god. Diplomski-master rad odbranio je 2015. godine na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacije i obrada signala.

PYTHON MODUL ZA RAD SA ACTIVITI SISTEMOM ZA UPRAVLJANJE POSLOVNIM TOKOVIMA**PYTHON MODULE FOR ACTIVITI WORKFLOW MANAGEMENT SYSTEM**

Milan Stojkov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je prikazana implementacija Python wrapper-a za Activiti sistem za upravljanje poslovnim tokovima. Opisana je specifikacija i implementacija modula ovog sistema koji omogućuju komunikaciju sa Activiti sistemom koristeći REST API. Takođe, prikazana je praktična primena ovog modula.

Abstract – This paper presents the implementation of the Python module for the Activiti Workflow Management System. The specification and implementation of the modules that communicate with Activiti using the REST API is described. A practical usage of the system is also demonstrated.

Ključne reči: Python, Django, Activiti, REST, veb aplikacija, poslovni procesi

1. UVOD

Tradicionalni informacioni sistemi uglavnom koriste *data-driven* pristup prilikom realizacije, ali vremenom se pokazalo da se pažnja mora posvetiti i samim poslovnim procesima koje ti informacioni sistemi realizuju. Poslovni proces sastoji se od skupa aktivnosti koje se koordinisano izvršavaju u odgovarajućem organizacionom i tehnološkom okruženju. Ove aktivnosti zajedno dovode do poslovnog cilja. BPM (*Business Process Management*) privlači pažnju stručnjaka različitog obrazovanja i interesnih sfera. Menadžerima je u cilju da organizacija radi što bolje u oblasti kojom se bavi, što podrazumeva razumevanje vrednosti koje organizacija treba da dostavi krajnjem korisniku i kako to da postigne, dok ljudi iz IT sektora sagledavaju tehničku stranu poslovnih procesa. Takođe, u razvoju informacionih sistema uočena je potreba da se na apstraktan način prikažu poslovni proces i zatim realizuje siguran, robustan i skalabilan sistem za podršku upravljanju poslovnim procesima [1][2]. Kako bi se predstavili modeli poslovnih procesa postoje različite grafičke notacije, a u ovom radu je korišćena BPMN 2.0 notacija [3]. Razlog zbog kojeg se koristi ta notacija je što je to standard koji se primenjuje u Activiti sistemu za upravljanje poslovnim tokovima [2]. Activiti je BPMN 2.0 sistem otvorenog koda koji obezbeđuje okruženje za izvršavanje poslovnih procesa. Iako je Activiti sistem napisan u programskom jeziku Java i prirodno se može ugraditi u Java (veb) aplikaciju, poseduje i Activiti REST [4] komponentu. Ta komponenta obezbeđuje API [5] za komunikaciju sa Activiti sistemom koji mogu koristiti aplikacije koje ne moraju biti napisane na Java programskom jeziku.

NAPOMENA:

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

Jedan pristup za korišćenje REST API-a jeste da se realizuju omotači (wrappers) u izabranom programskom jeziku oko definisanih REST poziva. Do trenutka pisanja ovog rada, potpuno funkcionalni moduli napisani za programski jezik Python nisu postojali ili bar nisu bili javno dostupni. Kako je Python jedan od najpopularnijih programskih jezika u svetu, odlučeno je da se obezbedi podrška upravo za njega.

2. ALATI ACTIVITI SISTEMA

Glavna komponenta Activiti sistema je *process engine* koji pruža osnovne mogućnosti za izvršavanje BPMN 2.0 procesa i stvaranje novih zadataka u procesnim tokovima. Sa *Activiti Modeler* alatom korisnici su u stanju da modeluju BPMN 2.0 poslovne procese u veb čitaču. *Activiti Designer* je Eclipse [6] plugin koji omogućava nadogradnju izmodelovanog poslovnog procesa, dodavanje poslovne logike implementirane na Java programskom jeziku, pokretanje unit testova, itd. Dodatni alat koji Activiti pruža je *Activiti Explorer* kojim se mogu pregledati pokrenuti i završeni procesi, kao i tabele u bazi podataka koja je konfigurisana da radi sa Activiti sistemom. Poslednja, ali najznačajnija komponenta za ovaj rad, je Activiti REST komponenta koja obezbeđuje veb aplikaciju koja pokreće Activiti *process engine* kada se veb aplikacija startuje. Obezbeđuje REST API koji omogućuje udaljenu komunikaciju sa Activiti sistemom [7].

2.1 Activiti Engine i BPMN 2.0

Activiti Engine sistem može da smešta definicije procesa, startuje nove instance procesa, izvršava korisničke zadatke i sl. U suštini, *Activiti Engine* je mašina stanja. BPMN 2.0 definicija procesa sastoji se iz elemenata poput događaja (*events*), zadataka (*tasks*) i prolaza za usmeravanje toka (*gateways*) koji su povezani strelicama (*sequence flows*). Kada se takav proces postavi u engine i pokrene procesna instanca, elementi se izvršavaju jedan po jedan. Svako stanje, tj. BPMN 2.0 element može imati neku logiku koja će se izvršiti kada procesna instanca uđe u to stanje [7].

2.2 Activiti REST

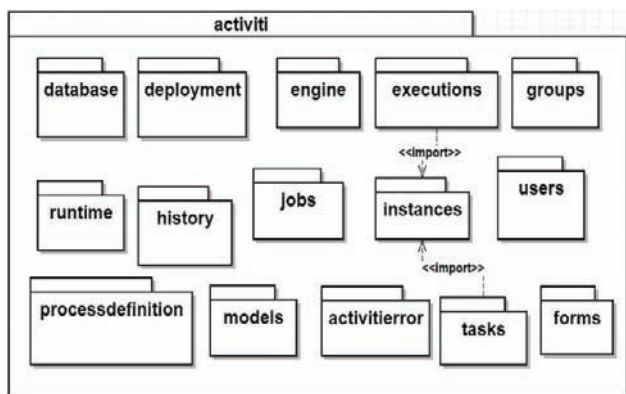
REST predstavlja jedan pristup za izgradnju distribuiranih sistema [8]. Ideja je da se umesto kompleksnih mehanizama kao što su RPC [9] ili SOAP [10] za uspostavljanje veze i razmenu poruka koristi HTTP protokol [11]. Glavna apstrakcija informacija po REST-u je resurs. Svaki entitet na vebu je resurs i identifikovan je svojim URI-jem [12]. Activiti poseduje svoju REST komponentu i svaki REST servis ima ekvivalent u određenom Java kodu (obrnuto ne važi). Činjenica je da je REST API siromašniji što se mogućnosti tiče od Activiti

engine-a koji se može ugraditi u Java aplikacije. Svi Activiti REST servisi rade sa JSON [13] objektima ukoliko se ne zahteva binarni sadržaj. Svi REST resursi zahtevaju da postoji validan Activiti korisnik koji će biti autentifikovan. Podržane HTTP metode koje su se i koristile prilikom realizacije Python modula su GET, POST, PUT i DELETE.

3. SPECIFIKACIJA I IMPLEMENTACIJA MODULA

Activiti paket sadrži 15 modula koji su prikazani UML [14] modelom na slici 3.1.

Prikazani moduli sadrže klase i funkcije za dodavanje, ažuriranje i brisanje korisnika, grupa, definicija procesa, procesnih instanci, poslova, izvršavanja, *deployment*-a, modela, dobavljanje i izvršavanje zadataka, elemenata formi, kao i pregled obavljenih zadataka i završenih procesa, zatim informacija o tabelama u Activiti bazi podataka i podataka o Activiti sistemu i klasu za opis grešaka.



Slika 3.1 Activiti paket sa pripadajućim modulima

Zbog preglednosti dijagrama na slici 3.1 nije prikazano da *activitierror* modul koriste svi ostali moduli, već se to podrazumeva. Pre predstavljanja pojedinačnih modula potrebno je naglasiti da je *requests* modul [15] korišćen kako bi se jednostavno moglo upravljati HTTP zahtevima. Funkcije u modulima su implementirane tako da im se prosleđuje URL na kojem se nalazi *activiti-rest* veb aplikacija i kredencijali za HTTP autentifikaciju kao obavezni parametri, dok ostali parametri zavise od radnje koju funkcija obavlja (npr. identifikator entiteta koji se briše ili JSON objekat koji sadrži podatke o entitetu koji se kreira).

Activiti REST aplikaciji se šalje jedan od četiri zahteva koji su propisani zvaničnom dokumentacijom (GET, POST, PUT, DELETE), zajedno sa traženim parametrima. Od REST odgovora koji se dobija, formiraju se objekti odgovarajućih klasa koje se nalaze u modulima sa slike 3.1. U slučaju odgovora čiji status kod ne pripada grupi 2xx kreira se objekat klase *ActivitiError*.

3.1 Deployment modul

Deployment je jedinica pakovanja unutar sistema. Kada se *deployment* smesti u sistem, proverava se njegova korektnost i smešta u bazu. Nakon toga se svaki proces unutar *deployment*-a može pokrenuti. Ovaj modul pokriva deo funkcionalnosti koji se odnosi na rukovanje *deployment*-ima. Tu se misli na kreiranje novog *deployment*-a, dobavljanje svih ili pojedinačnog

deployment-a, brisanje *deployment*-a i izlistavanje ili dobavljanje pojedinačnih resursa *deployment*-a.

3.2 Process definition modul

Modul pokriva deo funkcionalnosti koji se odnosi na rukovanje definicijama procesa. Tu se misli na dobavljanje svih ili pojedinačnog procesa, aktiviranje ili suspendovanje procesa, ažuriranje kategorije procesa i rukovanje kandidatima koji mogu pokrenuti proces.

3.3 Models modul

Models modul se bavi funkcionalnostima koje se odnose na rukovanje modelima. To podrazumeva dobavljanje svih modela ili pojedinačnog modela, kreiranje, ažuriranje ili brisanje modela.

3.4 Instances modul

Ovaj modul pokriva funkcionalnosti koje se odnose na rukovanje procesnim instancama. To podrazumeva pokretanje novih instanci definicija procesa. Za svaku definiciju, može biti više pokrenutih instanci koje se izvršavaju u isto vreme. Služi i za smeštanje i preuzimanje varijabli procesa, pretragu instanci, itd.

3.5 Executions modul

Executions modul se bavi funkcionalnostima koje se odnose na rukovanje izvršavanjima. Razlika između procesnih instanci i izvršavanja je u tome što je procesna instanca ono što se logički izvršava, a izvršavanje ono što se stvarno izvršava u tom trenutku. Izvršavanje bez roditelja je u stvari „jednako“ procesnoj instanci (čak i po identifikatoru), ali kad se naiđe na paralelni *gateway* [16], izvršavanje dobija potomke koji su izvršavanja za sebe, dok se ne spoje nazad u prvobitno izvršavanje.

3.6 Tasks modul

Modul omogućava pronalaženje svih zadataka dodeljenih nekoj osobi ili grupi. Omogućava kreiranje zadataka koji nisu vezani za neku instancu procesa, manipulaciju dodavanja zadataka korisniku, preuzimanje i završavanje zadataka.

3.7 History modul

Ovaj modul pokriva funkcionalnosti koje se odnose na rukovanje istorijskim informacijama. To podrazumeva pretragu informacija koje su prikupljene od strane sistema pre svega o završenim instancama procesa, zadacima, promenljivama.

3.8 Forms modul

U modulu su implementirane funkcije koje se bave formama za unos podataka. Kako zadaci mogu počinjati tako što će se popunjavati određeni podaci, Activiti je uveo koncept početne forme koja je vezana za startni zadatak. Takođe, postoje forme koje se vezuju za obične zadatke unutar procesa. Forme ne moraju biti ugrađene u definiciju procesa, ali ako je to slučaj, ovaj modul omogućava jednostavan pristup informacijama o njima.

3.9 Database modul

Funkcije ovog modula omogućavaju pribavljanje informacija o tabelama i kolonama u bazi. U većini slučajeva, nema potrebe da se koristi tokom razvoja aplikacije, ali su realizovane kako bi se upotpunio Activiti paket.

3.10 Engine modul

Engine modul omogućavaju pribavljanje informacija o sistemu i njegovim atributima. U većini slučajeva, nema

potrebe da se koristi tokom razvoja aplikacije, ali su realizovane kako bi se upotpunio Activiti paket.

3.11 Runtime modul

U ovom modulu postoji samo jedna funkcija koja obaveštava *engine* da je signalni događaj primljen i nije eksplicitno vezan sa specifično izvršavanje.

3.12 Jobs modul

Namena ovog modula je da obezbedi funkcije koje barataju naprednijim konceptima Activiti sistema – poslovima. Poslovi se koriste za izvršavanje asinhronne logike koja je uglavnom potrebna za izvršavanje kontinualnih procesa. Ovaj koncept je objašnjen na primeru *ServiceTask*-ova [7] za koje postoji podrška samo u programskom jeziku Java, ali zbog mogućnosti da se ova prepreka prevaziđe u budućnosti implementirani su i ovi pozivi. Ideja je da postoji određeni broj niti koje će izvršavati asinhronne poslove. Baza podataka čuva podatke o asinhronim poslovima i u jednoj niti kreće traganje za određenim poslom i ako se pronađe, posao se zaključava dok se ne obavi i posle toga se signalizira nastavak izvršavanja kontinualnog procesa [7].

3.13 Users modul

U ovom modulu obrađene su funkcije koje se tiču administracije korisnika. To podrazumeva kreiranje, ažuriranje i brisanje korisnika i njihovih informacija kao i dobavljanje liste ili pojedinačnog korisnika.

3.14 Groups modul

Groups modul obrađuje funkcije koje se tiču administracije grupa. To podrazumeva kreiranje, ažuriranje i brisanje grupa, dobavljanje liste ili pojedinačne grupe kao i dodavanje i brisanje korisnika iz grupe.

3.14 ActivitiError modul

Za sve module je zajedničko da imaju isti JSON odgovor za greške. Kada se desi greška odgovor sadrži JSON objekat koji opisuje grešku koja se desila. Pošto JSON objekat sadrži samo atribut *message* i *exception*, kako bi se upotpunio odgovor i koristio u projektima dodat je i atribut *statusCode* i za takav objekat je napravljena klasa.

4. PRIMENA IMPLEMENTIRANIH MODULA U PYTHON DJANGO APLIKACIJI

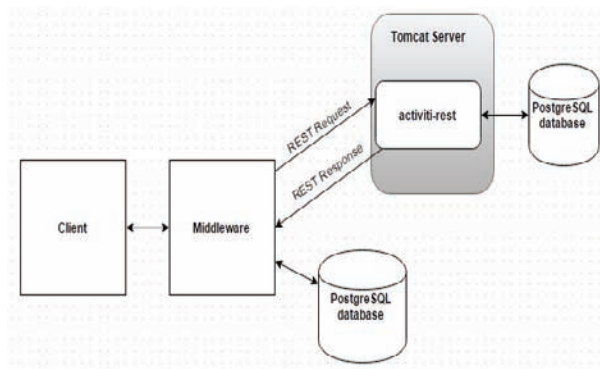
Kako bi se proverila validnost implementacije modula urađena je aplikacija na Python programskom jeziku uz oslonac na Django okruženje [17] za razvoj veb baziranih aplikacija. Aplikacija se bavi pojednostavljenim procesom servisiranja računara.

4.1 Arhitektura sistema

Servisiranje računara je poslovni proces koji je realizovan kao veb aplikacija, a celokupna arhitektura sistema predstavljena je na slici 4.1. Klijent je veb čitač koji renderuje HTML/CSS [18][19] stranice proširene sa Bootstrap [20] CSS bibliotekom za jednostavnije i efikasnije formiranje sadržaja stranica kao i njegovog rasporeda na stranicama.

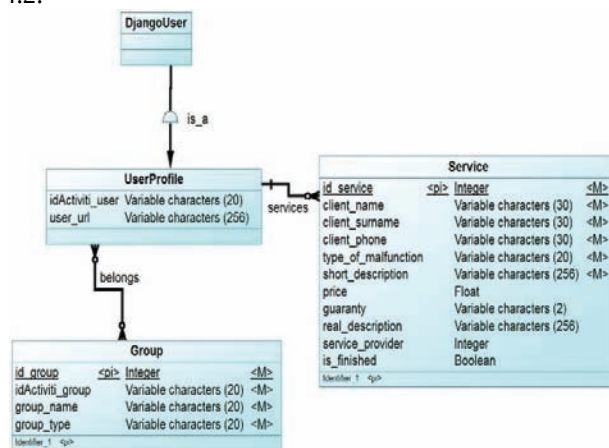
Middleware je Django aplikacija kojom se upravljalo kroz ugrađeni server tokom razvoja. Baza podataka sa kojom radi aplikacija je PostgreSQL [21].

Django aplikacija komunicira sa *Aactiviti REST* aplikacijom (koja ima svoju PostgreSQL bazu podataka) koja se nalazi na Tomcat serveru [22].



Slika 4.1 Arhitektura sistema

Na osnovu podataka koji se zahtevaju u BPMN 2.0 procesu napravljen je model podataka za bazu sa kojom će Django aplikacija komunicirati i prikazan je na slici 4.2.



Slika 4.2 Model podataka za Django bazu podataka

Entiteti koji figurišu u ovom modelu su *Group* koji predstavlja grupe kojima pripadaju članovi servisa, *UserProfile* koji se oslanja na *User* tabelu kojom je opisan korisnik u Django sistemu i proširen je sa nekoliko novih polja i *Service* koji predstavlja obavljanu uslugu.

4.2 Implementacija aplikacije

Posle startovanja aplikacije i logovanja, korisnik može započeti novi proces tako što će popuniti odgovarajuću formu prikazanu na slici 4.3 čiji elementi odgovaraju elementima startnog događaja BPMN procesa.

Dodaj novu uslugu

Ime klijenta:

Prezime klijenta:

Broj telefona:

Tip usluge:

Kratak opis akcije:

Slika 4.3 Dodavanje nove usluge

Kada korisnik unese tražene podatke i doda uslugu podaci forme se na serverskoj strani preuzimaju i validiraju. Pre čuvanja u bazu, pravi se Python rečnik koji sadrži identifikator definicije procesa računara i podatke iz forme. Rečnik se prosleđuje funkciji za slanje podataka forme Activiti sistemu, što ujedno znači i startovanje instance procesa, i posle provere povratne vrednosti te funkcije, podaci o servisu se čuvaju u Django PostgreSQL bazu. Kod sa delom opisane funkcije gde se koristi implementirani REST API dat je u listingu 4.1.

```

form = ServiceForm(request.POST)
if form.is_valid():
    #pronalazi korisnika iz zahteva
    #kreira se request_for_activiti rečnik
    #za slanje podataka Activiti sistemu
    ...
    f_data =
    activiti.forms.submit_task_form_data
    (url, request_for_activiti, username,
    password)
    if isinstance(f_data,
    activitierror.ActivitiError):
        ...
    else:
        #čuvanje podataka
        ...

```

Listing 4.1 Slanje podataka forme Activiti sistemu

Kada korisnik želi da preuzme zadatak kako bi ga obavio, bira ga iz tabele (liste) zadataka koja se nalazi na posebnoj stranici. Izgled stranice za preuzimanje zadataka dat je na slici 4.4, a segment koda za komunikaciju sa Activiti sistemom pri preuzimanju zadataka dat je na listingu 4.2.



Slika 4.4 Stranica za preuzimanje novih zadataka

```

...
claims = {
    "action" : "claim",
    "assignee" : usr
}
tsk = tasks.take_task_action(url, k,
claims, username, password)
...

```

Listing 4.2 Preuzimanje zadataka – Activiti sistem

Na sličan način implementirani su i ostali pozivi kako bi se ispratio proces do kraja.

5. ZAKLJUČAK

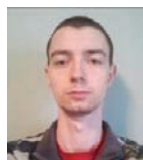
U ovom radu prikazana je realizacija Python *wrapper*-a za Activiti REST sistem za upravljanje poslovnim tokovima. Opisana je specifikacija i implementacija modula koji pokrivaju sve REST pozive propisane zvaničnom specifikacijom. Dato rešenje je verifikovano na poslovnom procesu kojim je opisano servisiranje računara. Prikazana je kompleksna arhitektura razvijanog sistema i opisan je model baze podataka koji je Django aplikacija koristila u implementaciji. Activiti je projekat

koji se stalno razvija i nove verzije izlaze na svega nekoliko meseci, što implicira da će i prikazano rešenje biti podložno promenama i unapređenjima. Drugi pravac razvoja bi mogao biti razvoj novog sistema gde bi celokupan postojeći Activiti sistem bio razvijen na programskom jeziku Python.

6. LITERATURA

- [1] Weske M, Business Process Management: Concepts, Languages, Architectures (2nd edition), Springer, 2012.
- [2] Activiti, <http://www.activiti.org/>
- [3] BPMN 2.0, <http://www.bpmn.org/>
- [4] REST, <http://rest.elkstein.org/>
- [5] API, https://en.wikipedia.org/wiki/Application_programming_interface
- [6] Eclipse, <https://eclipse.org/>
- [7] Tijs Rademakers, Activiti in Action, Manning Publications, 2012.
- [8] Roy Thomas Fielding, Architectural Styles and the Design of Network-based Software Architectures, University of California, Irvine, 2000.
- [9] RPC, <http://www.cs.cf.ac.uk/Dave/C/node33.html>
- [10] SOAP, <http://www.w3.org/TR/soap/>
- [11] HTTP, <http://www.w3.org/Protocols/>
- [12] URI, <http://www.w3.org/Addressing/>
- [13] JSON, <http://json.org/>
- [14] UML, <https://www.uml.org/>
- [15] Requests: HTTP for Humans, <http://docs.python-requests.org/en/latest/>
- [16] BPMN 2.0 Parallel Gateway, http://training-course-material.com/training/BPMN_2.0_Gateways
- [17] Django Web Framework, <https://www.djangoproject.com/>
- [18] CSS, <http://www.w3.org/standards/webdesign/htmlcss#whatcss>
- [19] HTML, <http://www.w3schools.com/html/>
- [20] Twitter, Bootstrap, <http://getbootstrap.com/>
- [21] PostgreSQL database, <http://www.postgresql.org/>
- [22] Apache, Tomcat, <http://tomcat.apache.org/>

Kratka biografija:



Milan Stojkov rođen je u Zrenjaninu 1991. godine. Upisao je smer Računarstvo i automatika na Fakultetu tehničkih nauka Univerziteta u Novom Sadu 2010. godine. Diplomirao je 2014. godine i iste godine upisao master studije, takođe na Fakultetu tehničkih nauka.

**KLASIFIKACIJA NEZAVISNIH KOMPONENTI BOLD SIGNALA NA OSNOVU
SPEKTRALNE GUSTINE SNAGE****CLASSIFICATION OF INDEPENDENT BOLD COMPONENTS ACCORDING TO
POWER SPECTRAL DENSITY**Uroš Stepanović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Cilj rada bio je da se ispita mogućnost klasifikacije prostorno nezavisnih BOLD (Blood Oxygenation Level Dependant) komponenti na osnovu spektralne gustine snage.

Klasifikaciju smo radili na MR BOLD slikama 11 subjekata snimljenih u Centru za imaging Dijagnostiku, Instituta za onkologiju Vojvodine. Za izdvajanje prostorno nezavisnih BOLD komponenti korišćena je nezavisna analiza komponenti kako je implementirana u FSL (FMRIB-Functional Magnetic Resonance Imaging of the Brain, Software Library). Za određivanje spektralne gustine snage ovako dobijenih komponenti korišćena je pwelch MATLAB funkcija. Komponente kod kojih je maksimalna gustina snage bila veća od 1 i nalazila se u najnižem frekventnom opsegu smatrane su korisnim komponentama dok su ostale komponente klasifikovane kao šum. Ovim postupkom sve korisne komponente klasifikovane su ispravno (100%) dok je procenat uspešno klasifikovanih komponenti šuma bio 65%. Posmatrajući sve dostupne signale uspešnost klasifikacije bila je 77%.

Abstract –The aim of this study was to investigate possibilities of spatially independent BOLD (Blood Oxygenation Level Dependant) component classification according to power spectral density (PSD). MR BOLD images of 11 subjects were obtained at Diagnostic Imaging Centre, Oncology Institute of Vojvodina. For identification of spatially independent components, independent component analysis (ICA) as implemented in FSL (FMRIB-Functional Magnetic Resonance Imaging of the Brain, Software Library) has been used. PSD was estimated using pwelch MATLAB function. Components that have PSD maximum greater than 1 in lowest frequency range were considered as useful while other were classified as noise. Using this approach all useful components (100%) were successfully classified while 65% of noise components were classified correctly. Taking into account all components 77% was correctly classified.

Ključne reči: ICA, fMR, PSD, classification

1. UVOD

Funkcionalna magnetna rezonanca (fMR) je tehnika koja omogućava neinvazivne studije moždane aktivnosti i od ranih 90-ih jedan je od najvažnijih alata u neuronaukama.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila doc. dr Olivera Šveljo.

Ova tehnika našla je svoju primenu kako u kliničkim tako i u istraživačkim aplikacijama. Najpopularnija fMR tehnika zasniva se na BOLD (Blood Oxygenation Level Dependant) kontrastu, koji se bazira na različitim magnetnim svojstvima oksigenizovane (dijamagnetske) i dezoksioksigenizovane (paramagnetske) krvi. Kada se neuroni aktiviraju, lokalno dolazi do promene u protoku i stepenu oksigenizacije krvi što dovodi do promene MR signala [1]. Ovako dobijen BOLD signal međutim ima relativno nizak odnos kontrasta i šuma (CNR) te pomeraji i uticaj neželjenih fizioloških izvora poput disanja ili rada srca otežavaju detektovanje signala koji potiču isključivo od aktivacije neurona. Postoji nekoliko vrsta signala koji se mogu razlikovati unutar BOLD signala a koji su identifikovani već prilikom prvih implementacija nezavisne analize komponenti u fMR eksperimentima. Izdvojeni signali su klasifikovani kao signali vezani za zadatak, prolazno vezani za zadatak i oni koji su posledica pomeraja. U novije vreme koristi se podela na korisne signale tj. signale koji su od interesa i signale koji nisu od interesa. U signale od interesa spadaju signali koji su vezani za zadatak, one koji su vezani za određenu funkciju i signali koji su delimično (prolazno) vezani za zadatak, dok se u signale koji nisu od interesa odnosno šum ubrajaju komponente koje potiču od fizioloških procesa, pomeraja, kao i komponente koje su posledica samog snimanja odnosno predstavljaju sistemski šum [2].

Nezavisna analiza komponenti (ICA - Independent Component Analysis) pokazala se pogodnom za analizu BOLD signala iz više razloga a posebno treba istaći da prostorna ICA pronalazi sistematski vremenski koherentne moždane regije koje se ne preklapaju a da pri tome u vremenskom domenu nema nikakvih ograničenja.

Prednost ICA u odnosu na druge tehnike analize BOLD signala je njena sposobnost da utvrdi dinamiku procesa za koje ne postoji unapred utvrđen model i to je jedan od razloga što ova tehnika postaje sve popularnija za analizu fMR podataka. Danas se ICA analiza uspešno koristi u brojnim fMR studijama za identifikaciju moždanih signala bilo da se radi o fiziološkim signalima ili specifičnim vezanim za određeni kognitivni zadatak [2].

edan od problema koji se javlja pri analizi rezultata koji se dobijaju ICA analizom je izdvajanje signala od interesa od onih koji nisu od interesa odnosno korisnih signala i signala šuma. U ovom radu ispitivali smo mogućnosti klasifikacije ovih signala na osnovu njihove spektralne gustine snage.

2. IZDVAJANJE NEZAVISNIH BOLD KOMPONENTI

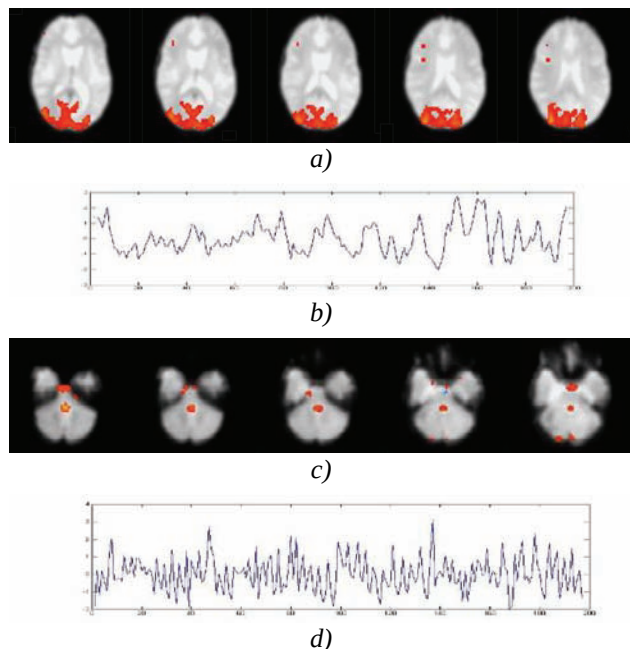
Bile su dostupne BOLD MR slike 11 subjekata snimljenih na MR uređaju MAGNETOM TRO 3T (Siemens, Erlangen, Germany) u Centru za imaging dijagnostiku na Institutu za onkologiju Vojvodine u Sremskoj Kamenici.

Nezavisne BOLD komponente izdvojene su prostorno nezavisnom analizom komponenti kako je implementirana u FSL (FMRIB Software Library). Ovo podrazumeva modelovanje podataka kao linearan zbir različitih procesa, čije su prostorne distribucije vremenski i statistički nezavisne (osim ukupne modulacije amplituda kao posledice udruženih vremenskih procesa). U ovom slučaju ICA je modelovana sledećim izrazom:

$$X=AM+E \quad (1)$$

Gde je X TxV matrica BOLD signala sa T vremenskih odmeraka i V voksel; M je KxV matrica prostornih komponenti nezavisnih izvora (koji se svaki sastoji iz V voksel), pri čemu je $K \ll T$. A je TxK matrica K odgovarajućih vremenskih signala koji se sastoje iz T uzoraka svaki. E je ostatak u probabilističkom ICA modelu [3] i podrazumeva se da sadrži nestruktuiran šum koji kontroliše najslabiji jedinični vektor inicijalne generalne analize komponenti (PCA) koja je primenjena pre implementacije ICA algoritma. Kada su konačno određene matrice A i M tek potom se ručno procenjuje koji od ovih signala je koristan signal a koji signal šuma.

Primenom ovog postupka na dostupan set BOLD MR slika identifikovano je ukupno 226 nezavisnih BOLD procesa od kojih je 76 ručno klasifikovano kao koristan signal a 150 kao šum. Na slici 1 dati su primeri nezavisnih komponenti BOLD signala, prostorno i vremenski, koji su ručno klasifikovani kao koristan signal i signal šuma.



Slika 1. Primer korisne BOLD komponente identifikovan ICA postupkom a) u prostornom i b) vremenskom domenu. Primer šuma identifikovan ICA postupkom c) u prostornom i d) u vremenskom domenu

3. ODREĐIVANJE SPEKTRALNE GUSTINE SNAGE

Spektralna gustina snage (Power Spectral Density – PSD) prikazuje kako je snaga nekog vremenskog signala raspoređena po frekvencijama i definisna je izrazom:

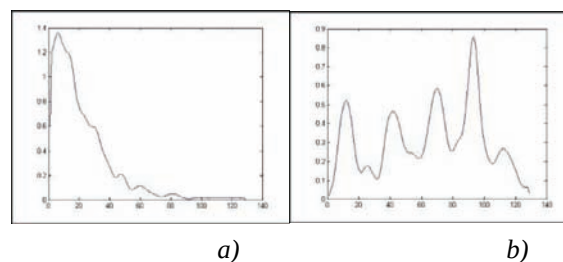
$$PSD(\omega) = F_T(\omega)F_T^*(\omega) \quad (2)$$

Gde je $F_T(\omega)$ normalizovana Furijeova transformacija signala $f(t)$.

Za izračunavanje spektralne gustine snage korišćena je MATLAB funkcija *pwelch()* u okviru koje je primenjen Velšov (Welch) postupak:

1. signal se podeli na L segmenata dužine M koji se preklapaju u D tačaka (ako je $D=M/2$ to podrazumeva preklapanje od 50%)
2. Nakon što se signal podeli na L segmenata primenjuje se prozor (u vremenskom domenu). Ovim postupkom na krajnji rezultat veći uticaj imaju središnji odmerci u segmentu u odnosu na periferne odmerke te zbog toga može doći do gubitka bitnih informacija pa je to razlog što se koriste preklapajući segmenti.
3. Na svaki deo tako dobijenih prozora primenjuje se Furijeova transformacija a potom se spektralna gustina snage izračunava prema izrazu 2.

Kao napogodnije odabrano je preklapanje od 50% uz primenu Hamingovog (Hamming) filtra u 2. koraku. Na slici 2 date su PSD funkcije korisnog signala i signala šuma sa slike 1.



Slika 2. Spektralna gustina snage a) korisnog signala i b) signala šuma sa slike 1

4. POSTUPAK KLASIFIKACIJE

Za klasifikaciju korisnih signala i signala šuma korišćen je sledeći postupak:

- Za posmatrani signal određuje se PSD korišćenjem Velš metode (MATLAB *pwelch()*)
- Identifikuje se maksimalna vrednost PSD
- Slede 2 koraka klasifikacije:
 - o Ukoliko je max vrednost manja ili jednaka 1, signal se automatski odbacuje kao šum
 - o Ukoliko je max vrednost veća od 1, proveravamo da li se ona nalazi u prvih 20 elemenata:
 - Ne nalazi se u prvih 20 elemenata – signal se proglašava za šum
 - Nalazi se u prvih 20 elemenata – signal se klasifikuje kao dobar

5. REZULTATI KLASIFIKACIJE

Predloženim postupkom ispravno je klasifikovana 65% signala šuma dok su svi korisni signali (100%) ispravno klasifikovani (Slika 3).



Slika 3. Grafički prikaz uspešnosti klasifikacije signala šuma

Posmatrajući sve dostupne signale opisanim postupkom 77% signala uspešno je klasifikovano (Slika 4)



Slika 4. Grafički prikaz uspešnosti klasifikacije svih dostupnih signala

U tabeli 1 uporedno su dati rezultati ručne klasifikacije i klasifikacije predloženim postupkom na osnovu spektralne gustine snage po subjektima.

Tabela 1. Uporedni rezultati ručne klasifikacije i PSD klasifikacije po subjektima

subjekat	Ručna klasifikacija		PSD klasifikacija	
	korisni	šum	Korisni	šum
SUB_1	8	10	10	8
SUB_2	6	18	11	13
SUB_3	6	10	7	9
SUB_4	11	19	24	6
SUB_5	6	14	9	11
SUB_6	7	25	18	14
SUB_7	7	11	11	7
SUB_8	5	15	10	10
SUB_9	8	9	13	4
SUB_10	6	8	9	5
SUB_11	6	11	7	10
Ukupno:	76	150	129	97

6. DISKUSIJA

Nezavisnom analizom komponenti BOLD signala moguće je razdvojiti šum (sistemske fluktuacije, fluktuacije usled fizioloških procesa ...) i komponente koje nisu posledica šuma (fluktuacije od interesa) te se na osnovu toga mogu izdvojiti komponente od interesa za dalju analizu. Kada se ovom problemu pristupa ručno, to zahteva mnogo vremena i podrazumeva duboko poznavanje (idealno svih mogućih) prostorno vremenskih karakteristika korisnog BOLD signala i signala šuma u fMR eksperimentu. Upravo je to bio jedan od osnovnih motiva da se ispita mogućnost automatske klasifikacije na osnovu spektralne gustine snage izložen u ovom radu. Uočeno je da se glavna snaga korisnih signala nalazi u najnižem frekventnom opsegu te je ova činjenica iskorišćena za formiranje jednostavnog algoritma za klasifikaciju

korisnih signala i signala šuma. Iako, opisani postupak nije bio u potpunosti uspešan tj. nije bilo moguće u potpunosti razdvojiti korisne signale od signala šuma, klasifikacija samo na osnovu jednog parametra (maksimalna vrednost PSD) dala je relativno visok procenat (77%) uspešno klasifikovanih signala. Treba napomenuti da su ovim postupkom svi korisni signali ispravno klasifikovani te nije bilo gubitaka korisnih informacija. Iako postupak nije dao potpunu klasifikaciju ipak može biti koristan za selekciju tj. smanjenje broja signala koje je potrebno klasifikovati ručno što značajno doprinosi smanjenju vremena potrebnog za izdvajanje korisnih komponenti dobijenih nezavisnom analizom BOLD signala. Naime ICA postupkom identifikovano je 226 nezavisnih komponenti od čega su 76 ručno identifikovane kao korisne a 150 kao signali šuma. Nakon klasifikacije na osnovu spektralne gustine snage od 226 komponenti 129 je klasifikovano kao korisno dok je 97 komponenti odbačeno kao šum. Tako je za dalju obradu preostalo 129 signala što je nešto više od polovine ukupnog broja signala pre klasifikacije te bi ovaj postupak mogao da prepolovi vreme neophodno za ručnu klasifikaciju signala.

7. ZAKLJUČAK

Predloženi postupak nije bio dovoljan za potpuno uspešnu klasifikaciju signala, ali može da bude dobra osnova za dalje unapređenje postupka klasifikacije uvođenjem više parametara poput snage signala u pojedinim opsezima, koeficijenta spljoštenosti, koeficijenta asimetričnosti... Dalja unaprđenja moguća su i primenom postupaka mašinskog učenja.

8. REFERENCE

- [1] Šveljo O. Detekcija elokventnih govornih moždanih zona funkcionalnom magnetnom rezonancom. Univerzitet u Beogradu 2012, doktorska disertacija
- [2] Calhoun VD, Adali T, Hansen LK, Larsen J, Pekar JJ. ICA of Functional MRI Data: An Overview. 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA 2003), April 2003, Nara, Japan, Conference Proceedings, pp 281-287
- [3] Beckman CF, Smith SM. Probabilistic Independent Component Analysis for Functional Magnetic Resonance Imaging. FMRIB Technical Report TR02CB1

Kratka biografija:



Uroš Stepanović rođen je u Sremskoj Mitrovici 26.01.1991. godine. Studije je upisao 2010. godine u Novom Sadu na Fakultetu tehničkih nauka, smer Energetika, elektronika i telekomunikacije. Studije je nastavio na Katedri za telekomunikacije i obradu signala, da bi se potom usmerio na telekomunikacione sisteme. Master studije upisuje 2014. godine na Katedri za telekomunikacije i obradu signala, grupa Telekomunikacioni sistemi.

PRORAČUN TOKOVA SNAGA MIKSOVANE DISTRIBUTIVNE MREŽE U DOMENU SIMETRIČNIH KOMPONENTI

LOAD FLOW CALCULATION FOR A MIXED DISTRIBUTION SYSTEM IN SEQUENCE DOMAIN

Andrija Mitrović, Predrag Vidović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je obrađen problem tokova snaga miksovanih distributivnih mreža u domenu simetričnih komponenti. Takođe su obrađeni modeli elemenata, kao i sam model miksovane distributivne mreže i dat je algoritam za proračun tokova snaga miksovanih distributivnih mreža. Njegova verifikacija za različite tipove potrošača urađena je na primeru miksovane mreže. Takođe je izvršena analiza rezultata dobijenih ovim algoritmom.

Abstract – In this paper Load Flow calculation in sequence domain for mixed distribution networks is described. Here are also presented mathematical models of network elements and network itself. An algorithm is given for Load Flow calculation. Verification of mathematical model is given for different consumer types and a thorough analysis is given for the results.

Ključne reči: Tokovi snaga, domen simetričnih komponenti, miksovane distributivne mreže.

1. UVOD

Proračun tokova snaga jeste jedan od bazičnih proračuna u elektroenergetici. Tokovi snaga predstavljaju proračun promenljivih stanja, odnosno kompletnog režima mreže, na bazi poznatog napona izvora napajanja mreže i poznatih potrošnji u svim čvorovima mreže. Režim EES, pored statusa rasklopne opreme i podešenja regulatora regulacionih resursa, karakteriše bilans snaga (aktivnih i reaktivnih) koji u svakom trenutku treba da je zadovoljen, što predstavlja glavni motiv za unapređenje i razvoj proračuna tokova snaga [1]. Zbog različitih osobina distributivnih (velike dimenzije, mali broj petlji, prisustvo veoma kratkih i veoma dugih deonica i velika vrednost odnosa R/X) i prenosnih mreža (upetljani pogon, umerena dimenzionalnost, mala vrednost odnosa R/X i veliki broj merenja) razvili su se različiti postupci za proračun tokova snaga ovih mreža [2]. Najpre su se pojavila rešenja za proračune simetričnih tokova snaga trofaznih radijalnih distributivnih mreža i distributivnih mreža s malim brojem petlji. Kada su u pitanju radijalne mreže, tada su ti postupci zasnovani na slaganju mreže po slojevima i procedurama sumiranja struja i korekcija napona. Proračun tokova snaga se daleko usložnjava kada se uvažavaju realni režimi distributivnih mreža, koji su posledica nesimetričnih potrošnji uravnoteženih ili mreža koje

nisu uravnotežene [3]. U okviru ovog rada je objašnjen postupak za rešavanje problema tokova snaga miksovanih distributivnih mreža u domenu simetričnih komponenti koristeći se procedurama sumiranja struja i korekcija napona.

2. EKVIVALENTNE ŠEME ELEMENATA MIKSOVANE DISTRIBUTIVNE MREŽE I MODEL MREŽE

Ovde su obrađeni matematički modeli (ne)uravnoteženih potrošača, sekcija (vodova) i uravnoteženih transformatora sa osnovnim spregama: Yy, Dy, Yd i Dd (zvezdišta mogu i ne moraju da budu uzemljena).

2.1. Model potrošača

U ovom delu su obrađeni modeli potrošača. Model potrošača za razliku od modela ostalih elemenata, koji su dati u domenu simetričnih komponenti, je dat u faznom domenu zbog toga što se potrošači predstavljaju preko svojih faznih snaga. Snaga potrošnje potrošača, $\hat{S}_x(U_x)$, $x \in \{a, b, c\}$, gde su sa a, b i c indeksirane faze mreže, data je funkcijom:

$$\hat{S}_x(U_x) = k_{spx} P_x^{spec} - j k_{sqx} Q_x^{spec} + k_{ipx} \frac{U_x}{U_n} P_x^{spec} - j k_{iqx} \frac{U_x}{U_n} Q_x^{spec} + k_{ypx} \left(\frac{U_x}{U_n} \right)^2 P_x^{spec} - j k_{yqx} \left(\frac{U_x}{U_n} \right)^2 Q_x^{spec}, \quad x \in \{a, b, c\}, \quad (1)$$

pri čemu su:

P_x^{spec} , Q_x^{spec} – normalizovana specificirana fazna aktivna i reaktivna snaga potrošnje potrošača;

k_{spx} , k_{sqx} – koeficijenti učešća delova fazne aktivne i reaktivne snage potrošnje potrošača koje nisu zavisne od napona;

k_{ipx} , k_{iqx} – koeficijenti učešća delova fazne aktivne i reaktivne snage potrošnje potrošača koje su linearno zavisne od napona;

k_{ypx} , k_{yqx} – koeficijenti učešća delova fazne aktivne i reaktivne snage potrošnje potrošača koje su zavisne od kvadrata napona.

Za koeficijente učešća važi:

$$k_{spx} + k_{ipx} + k_{ypx} = 1.0, \quad x \in \{a, b, c\}, \quad (2)$$

$$k_{sqx} + k_{iqx} + k_{yqx} = 1.0, \quad x \in \{a, b, c\}. \quad (3)$$

Tri karakteristične grupe potrošača su:

- 1) konstantne snage ($k_{spx} = k_{sqx} = 1$, a ostali koeficijenti su nule);
- 2) konstantne struje ($k_{ipx} = k_{iqx} = 1$, a ostali koeficijenti su nule);
- 3) konstantne admitanse ($k_{ypx} = k_{yqx} = 1$, a ostali koeficijenti su nule);

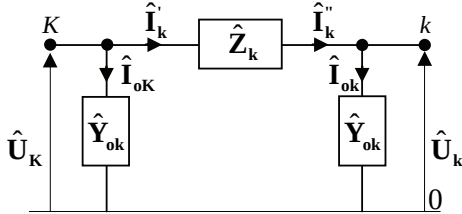
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Predrag Vidović, doc.

Potrošači, koji se napajaju sa transformatora kod kojih je na sekundaru neuzemljena zvezda ili trougao, se predstavljaju preko konstantne admitanse (impedanse), kako bi se omogućilo ažuriranje nulte komponente napona na sekundaru ovih transformatora [3].

2.2. Model sekcije voda

Vodovi su elementi elektroenergetskog sistema namenjeni za prenos i distribuciju električne energije. Postoje dve vrste vodova: nadzemni (vazdušni) i kablovski [4]. Načelna šema sekcije n -faznog voda prikazana je na slici 1. Čvorovi K i k jesu početak i kraj sekcije.



Slika 1. Načelna šema n -fazne sekcije (voda)

Matematički model sekcije sa slike 1 glasi:

$$\hat{\mathbf{I}}_k' = \hat{\mathbf{I}}_k'' \quad (4)$$

$$\hat{\mathbf{I}}_{0K} = \hat{\mathbf{Y}}_{0K} \hat{\mathbf{U}}_K, \quad \hat{\mathbf{I}}_{0k} = \hat{\mathbf{Y}}_{0k} \hat{\mathbf{U}}_k \quad (5)$$

$$\hat{\mathbf{U}}_k = \hat{\mathbf{U}}_K - \hat{\mathbf{Z}}_k \hat{\mathbf{I}}_k' \quad (6)$$

pri čemu su korišćene sledeće oznake:

$\hat{\mathbf{U}}_K, \hat{\mathbf{U}}_k$ – vektori napona u domenu simetričnih komponenti na početku i kraju sekcije, dimenzija $n \times 1$;

$\hat{\mathbf{I}}_k', \hat{\mathbf{I}}_k''$ – međusobno jednaki vektori struja u domenu simetričnih komponenti na početku i kraju sekcije, dimenzija $n \times 1$;

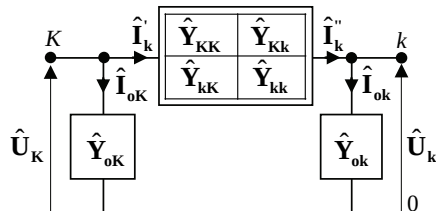
$\hat{\mathbf{I}}_{0K}, \hat{\mathbf{I}}_{0k}$ – vektori struja u domenu simetričnih komponenti otopnih parametara na početku i kraju sekcije, dimenzija $n \times 1$;

$\hat{\mathbf{Z}}_k, \hat{\mathbf{Y}}_{0k}$ – matrični reprezentanti rednih i otopnih parametara u domenu simetričnih komponenti sekcije (jednaki na oba kraja sekcije), dimenzija $n \times n$; ako je sekcija uravnotežena, onda su te matrice dijagonalne;

2.3. Modeli transformatora

Elektroenergetski trofazni transformatori su statički uređaji pošto nemaju obrtne delove. Oni su namenjeni unutrašnjoj transformaciji električne energije (električna energija u električnu energiju) [4].

Načelna šema trofaznog transformatora prikazana je na slici 2.



Slika 2. Načelna šema trofaznog transformatora

Matematički model transformatora sa slike 2 glasi:

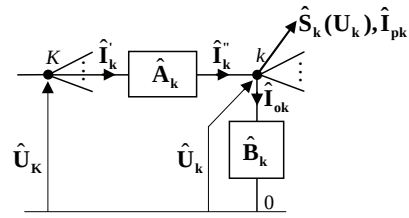
$$\hat{\mathbf{I}}_{0K} = \hat{\mathbf{Y}}_{0K} \hat{\mathbf{U}}_K, \quad \hat{\mathbf{I}}_{0k} = \hat{\mathbf{Y}}_{0k} \hat{\mathbf{U}}_k \quad (7)$$

$$\hat{\mathbf{I}}_k' = \hat{\mathbf{Y}}_{KK} \hat{\mathbf{U}}_K + \hat{\mathbf{Y}}_{Kk} \hat{\mathbf{U}}_k \quad (8)$$

$$-\hat{\mathbf{I}}_k'' = \hat{\mathbf{Y}}_{kK} \hat{\mathbf{U}}_K + \hat{\mathbf{Y}}_{kk} \hat{\mathbf{U}}_k \quad (9)$$

2.4. Model mreže

Saglasno s modelima potrošača, sekcija (vodova) i transformatora, model miksovane distributivne mreže može da se konstituiše koristeći se isključivo Γ segmentima prikazanim na slici 3 [3], složenim po slojevima. Svaki Γ segment je asociran jednoj grani mreže (sekciji voda ili transformatoru).



Slika 3. Segment Γ_k trofazne radijalne distributivne mreže

Sa $\hat{\mathbf{A}}_k$ označen je matrični reprezent rednih parametara – redna grana segmenta Γ_k .

Sa $\hat{\mathbf{B}}_k$ označen je admitantni matrični reprezent otopnih parametara – otočna grana segmenta Γ_k . $\hat{\mathbf{B}}_k$ ovde predstavlja sumu admitantnih matričnih reprezenata otopnih parametara kraja sekcije ili transformatora kojoj ili kojem je asociran razmatrani segment Γ_k , otopnih parametara na počecima sekcija ili transformatora koji se napajaju preko razmatranog segmenta i matričnih reprezenata elemenata direktno priključenih u čvoru k [3].

3. PRORAČUN TOKOVA SNAGA MIKSOVANIH DISTRIBUTIVNIH MREŽA U DOMENU SIMETRIČNIH KOMONENTI

Ovde je opisan algoritam za proračun tokova snaga miksovanih distributivnih mreža. On se sastoji od inicijalizacije postupka, iterativnog dela i provere konvergencije. Iterativni deo se sastoji od tri koraka (proračun struja otopnih elemenata, procedura sumiranja struja i procedura korekcija napona) koji se ponavljaju sve dok se ne ispune uslovi konvergencije.

3.1. Proračun struja otopnih elemenata

Proračun struja otopnih elemenata se radi za svaki čvor u mreži osim za balansni ($k = 1, 2, 3, \dots, n$) i za svaku fazu čvora. Taj proračun se radi u faznom domenu pošto je snaga potrošača data u faznom domenu, tako da se u okviru svake iteracije pre ovog proračuna prelazi u fazni domen da bi se proračunala struja otopnih elemenata, te se nakon toga vraća u domen simetričnih komponenti i nastavlja sa iterativnim delom proračuna. Proračun struja otopnih elemenata radi se na sledeći način:

$$\hat{\mathbf{I}}_k = \hat{\mathbf{I}}_{pk} + \hat{\mathbf{I}}_{0k} \quad (10)$$

odakle se vidi da se struja otopnih elemenata sastoji od dva dela:

- Vektora struje potrošnje $\hat{\mathbf{I}}_{pk}$, čiji se elementi (struje po fazama potrošača) računaju na sledeći način:

$$\hat{I}_{pkx} = \left(\frac{\hat{S}_{kx}(\hat{U}_{kx})}{\hat{U}_{kx}} \right)^*, \quad x \in \{a, b, c\}. \quad (11)$$

- Vektora otočne struje $\hat{\mathbf{I}}_{0k}$ koji se računa na sledeći način:

$$\hat{\mathbf{I}}_{0k} = \hat{\mathbf{B}}_k \hat{\mathbf{U}}_k. \quad (12)$$

3.2. Procedura sumiranja struja

Procedura sumiranja struja počinje od poslednjeg ka prvom segmentu ($k = n, \dots, 3, 2, 1$). Pošto se miksovana distributivna mreža sastoji od sekcija različitih faznosti, onda treba voditi računa o tome da se prilikom sumiranja struja može desiti da su vektori struja različitih dimenzija. Te, pošto se proračun radi u domenu simetričnih komponenti, prilikom sumiranja struja sekcija koje su različitih faznosti prvo se pomoću odgovarajućih matrica transformacija prelazi u fazni domen, zatim se vektori struja u faznom domenu proširuju nulama na mestima gde su one potrebne, te se u faznom domenu sabiraju i onda vraćaju u domen simetričnih komponenti odgovarajućim matricama. Kod sumiranja struja sekcija koje su iste faznosti ovo se ne radi nego se samo vektori struja saberu. Sumiranje struja se obavlja pomoću sledeće relacije:

$$\hat{\mathbf{I}}_k'' = \hat{\mathbf{I}}_k + \sum_{j \in \mathcal{A}_k} \hat{\mathbf{I}}_j', \quad (13)$$

gde je:

α_k – skup indeksa segmenata koji se napajaju sa k -tog segmenta.

U slučaju da je Γ_k segment sekcija važi:

$$\hat{\mathbf{I}}_k' = \hat{\mathbf{I}}_k'', \quad (14)$$

a u slučaju da je taj segment transformator, struja $\hat{\mathbf{I}}_k$ predstavlja struju na sekundarnu transformatora, te je potrebno da se ta struja iznese na primarnu stranu transformatora.

Direktna i inverzna komponenta se iznose na primarnu stranu transformatora pomoću sledećih relacija:

$$\hat{I}^D = e^{jk^d/6} \hat{I}^d, \quad (15)$$

$$\hat{I}^I = e^{jk^I/6} \hat{I}^i. \quad (16)$$

Nulta komponenta se na analogan način kao i direktna i inverzna komponenta iznosi samo u slučaju sprege $Y_{N_y N_n}$ (uzemljena zvezda i na primarnoj i na sekundarnoj strani):

$$\hat{I}^O = e^{jk^o/6} \hat{I}^o, \quad (17)$$

a u slučaju trougla sa jedne od strana transformatora nulta komponenta je jednaka nuli sa te strane, a sa strane gde se nalazi uzemljena zvezda:

$$\hat{I}^0 = (3\hat{Y}_o + \hat{Y}_k)\hat{U}^0, \quad (18)$$

gde je sa $\hat{Y}_o$ označena admitansa uzemljenja, a sa $\hat{Y}_k$ admitansa kratkog spoja.

(Napomena: Realni odnosi transformacije transformatora eliminisani su primenom sistema relativnih jedinica.)

3.3. Procedura korekcija napona

U okviru ove procedure radi se korekcija napona čvorova mreže na osnovu prethodno izračunatih struja grana. Proračun se radi od prvog ka poslednjem čvoru ($k = 1, 2, 3, \dots, n$). Naponi čvorova se dobijaju uvažavajući padove napona na segmentima.

Korekcija napona na sekcijama se radi pomoću relacije:

$$\hat{\mathbf{U}}_k = \hat{\mathbf{U}}_K - \hat{\mathbf{A}}_k \hat{\mathbf{I}}_k'. \quad (19)$$

U slučaju transformatora korekcija napona zavisi od sprege kao i kod procedure sumiranja struja. Tako se

korekcija napona za direktnu i inverznu komponentu uvek obavlja pomoću sledećih relacija:

$$\hat{U}^d = e^{-jk^d \pi/6} (\hat{U}^D - \hat{Z}_k \hat{I}^D), \quad (20)$$

$$\hat{U}^i = e^{-jk^i \pi/6} (\hat{U}^I - \hat{Z}_k \hat{I}^I). \quad (21)$$

Što se tiče korekcije nulte komponente napona, ona se na analogan način korekciji direktne i inverzne komponente radi samo u slučaju sprege Y_{NYn} :

$$\hat{U}^o = e^{-jk^o\pi/6} \left(\hat{U}^O - \left(\hat{Z}_k + 3\hat{Z}_o \right) \hat{I}^O \right). \quad (22)$$

U slučaju trougla na primarnoj strani, a uzemljene zvezde na sekundaru transformatora korekcija nulte komponente se radi pomoću relacije:

$$\hat{U}^0 = -(3\hat{Z}_\rho + \hat{Z}_k)\hat{I}^0. \quad (23)$$

U slučaju trougla ili neuzemljene zvezde na sekundaru transformatora nulta komponenta napona se koriguje (koristeći konstantne admitanse koje predstavljaju potrošač) zbog toga što je nulta komponenta struje na sekundaru tih transformatora jednaka nuli:

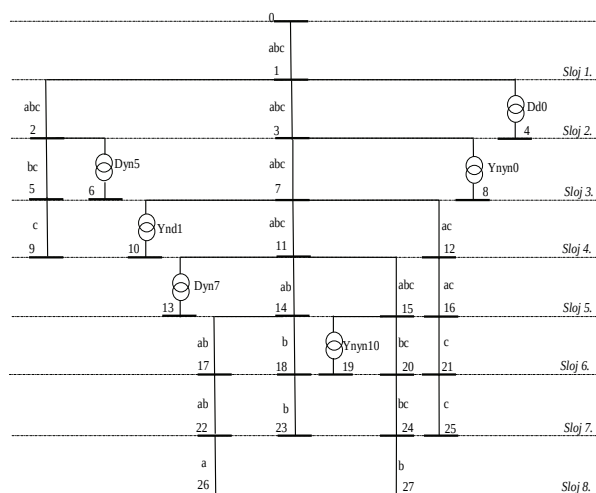
$$\hat{U}^o = -\left(\frac{1}{\hat{\mathbf{Y}}^{oo}}\right)\left(\hat{\mathbf{Y}}^{od}\hat{U}^d + \hat{\mathbf{Y}}^{oi}\hat{U}^i\right). \quad (24)$$

4. NUMERIČKA VERIFIKACIJA PRORAČUNA TOKOVA SNAGA

Na slici 4 data je miksovana distributivna test mreža za koju su rađeni proračuni tokova snaga. Dobijeni rezultati su analizirani u nastavku. Zaključci koji su ovde doneseni se odnose na jako male i slabo opterećene miksovane distributivne mreže, a ne na mreže uopšte.

Za mrežu sa slike su simulirani rezultati u cilju praćenja promene broja iteracija u zavisnosti od:

- 1) modela potrošnje;
- 2) opterećenja.



Slika 4. Miksovana distributivna mreža

Za početnu aproksimaciju napona uzimane su:

1. $\hat{U}^d = e^{j0}, \hat{U}^i = 0, \hat{U}^o = 0;$
2. $\hat{U}^d = 0.7e^{j0}, \hat{U}^i = 0.2e^{j35}, \hat{U}^o = 0.1e^{j195};$

Broj iteracija je u oba slučaja isti i iznosi 4, te se može zaključiti kako je ovaj postupak neosetljiv na početnu aproksimaciju napona. U sledećoj tabeli su dati rezultati za napone čvorova kada su potrošači predstavljeni kao konstantne struje.

Tabela 1. Naponi čvorova

Broj čvora	U_a [kV]	U_b [kV]	U_c [kV]
1	$11.537e^{j0.012}$	$11.532e^{j239.995}$	$11.537e^{j119.984}$
2	$11.538e^{j0.012}$	$11.530e^{j240.001}$	$11.534e^{j119.975}$
3	$11.529e^{j0.022}$	$11.522e^{j239.987}$	$11.531e^{j119.975}$
4	$0.230e^{j0.713}$	$0.225e^{j238.954}$	$0.234e^{j118.589}$
5	-	$11.529e^{j240.009}$	$11.533e^{j119.965}$
6	$0.229e^{j209.397}$	$0.230e^{j89.451}$	$0.230e^{j30.602}$
7	$11.521e^{j0.032}$	$11.513e^{j239.979}$	$11.526e^{j119.967}$
8	$0.229e^{j0.579}$	$0.229e^{j239.445}$	$0.230e^{j119.374}$
9	-	-	$11.531e^{j119.962}$
10	$0.230e^{j29.294}$	$0.224e^{j208.953}$	$0.234e^{j88.584}$
11	$11.515e^{j0.052}$	$11.504e^{j239.960}$	$11.527e^{j119.962}$
12	$11.522e^{j0.021}$	-	$11.521e^{j119.962}$
13	$0.229e^{j149.386}$	$0.229e^{j29.449}$	$0.230e^{j269.394}$
14	$11.509e^{j0.063}$	$11.502e^{j239.941}$	-
15	$11.515e^{j0.061}$	$11.499e^{j239.962}$	$11.525e^{j119.949}$
16	$11.523e^{j0.014}$	-	$11.518e^{j119.957}$
17	$11.505e^{j0.065}$	$11.502e^{j239.930}$	-
18	-	$11.499e^{j239.937}$	-
19	$0.229e^{j59.362}$	$0.230e^{j60.597}$	$0.230e^{j179.461}$
20	-	$11.496e^{j239.965}$	$11.526e^{j119.936}$
21	-	-	$11.515e^{j119.951}$
22	$11.502e^{j0.065}$	$11.502e^{j239.923}$	-
23	-	$11.498e^{j239.935}$	-
24	-	$11.493e^{j239.965}$	$11.526e^{j119.929}$
25	-	-	$11.514e^{j119.948}$
26	$11.501e^{j0.063}$	-	-
27	-	$11.492e^{j239.963}$	-

U nastavku je data tabela u kojoj je dat broj iteracija u zavisnosti od tipa potrošnje i opterećenja.

Tabela 2. Broj iteracija u zavisnosti od opterećenja i tipa potrošnje

Opterećenje	$k_{sp}=k_{sq}=1$ $k_{ip}=k_{iq}=0$ $k_{vp}=k_{vq}=0$	$k_{sp}=k_{sq}=0$ $k_{ip}=k_{iq}=1$ $k_{vp}=k_{vq}=0$	$k_{sp}=k_{sq}=0$ $k_{ip}=k_{iq}=0$ $k_{vp}=k_{vq}=1$	$k_{sp}=k_{sq}=0.1$ $k_{ip}=k_{iq}=0.2$ $k_{vp}=k_{vq}=0.7$
50%	4	4	4	4
100%	4	4	4	4
200%	5	5	5	5
300%	5	5	5	5

Iz tabele 2 se vidi da se broj iteracija povećava sa povećanjem opterećenja i da pri malim opterećenjima ne zavisi od tipa potrošnje.

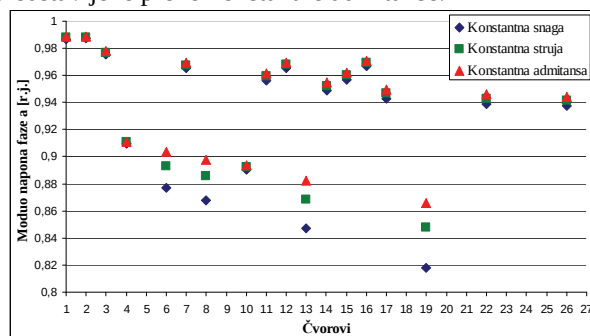
Tabela 3. Maksimalna apsolutna vrednost razlike normalizovanih napona kroz iteracije

Iter.	$k_{sp}=k_{sq}=1$ $k_{ip}=k_{iq}=0$ $k_{vp}=k_{vq}=0$	$k_{sp}=k_{sq}=0$ $k_{ip}=k_{iq}=1$ $k_{vp}=k_{vq}=0$	$k_{sp}=k_{sq}=0$ $k_{ip}=k_{iq}=0$ $k_{vp}=k_{vq}=1$
1	1.93	1.93	1.93
2	$1.25 \cdot 10^{-2}$	$1.25 \cdot 10^{-2}$	$1.25 \cdot 10^{-2}$
3	$1.47 \cdot 10^{-4}$	$1.39 \cdot 10^{-4}$	$1.40 \cdot 10^{-4}$
4	$3.69 \cdot 10^{-6}$	$3.93 \cdot 10^{-6}$	$4.45 \cdot 10^{-6}$

U okviru tabele 3 data je maksimalna apsolutna vrednost razlike normalizovanih napona čvorova tokom iterativnog postupka. Vidi se da je u prvoj iteraciji maksimalno odstupanje napona isto, te da se tokom iterativnog postupka smanjuje dok ne bude manje od kriterijuma konvergencije (10^{-5}), ali i da je maksimalno odstupanje u poslednjoj iteraciji najmanje u slučaju potrošača

predstavljenih preko konstantnih snaga, a najveće za potrošače predstavljene preko konstantnih admitansi.

Ako se posmatraju normalizovani naponi faze a (slika 5) za različite tipove potrošnje, jasno se vidi da su najveći padovi napona u slučaju potrošnje predstavljene preko konstantne snage, a najmanji u slučaju potrošnje predstavljene preko konstantne admitanse.



Slika 5. Modulo napona faze a za tri tipa potrošnje

5. ZAKLJUČAK

U okviru ovog rada opisan je proračun tokova snaga miksovanih distributivnih mreža u domenu simetričnih komponenti zasnovan na procedurama sumiranja struja i korekcija napona. Analizom rezultata zaključilo se da je proračun tokova snaga miksovanih distributivnih mreža neosetljiv na početnu aproksimaciju napona, da broj iteracija raste kako raste procenat opterećenja, te da broj iteracija ne zavisi od tipa potrošnje pri manjim opterećenjima i da su padovi napona najveći onda kada je potrošnja predstavljena kao konstantna snaga. Sve ove analize su rađene na maloj i slabo opterećenoj mreži, te zaključci koji su doneti ne važe za sve mreže.

4. LITERATURA

- [1] W.H.Kersting: Distribution System Modeling and Analysis; Boca Raton, FL:CRC, 2002.
- [2] D. Popović, D. Bekut, V. Treskanica, Specijalizovani DMS algoritmi. Srbija: Novi Sad, DMS Grupa DOO za elektroenergetski inženjering, 2004.
- [3] P.Vidović, Nesimetrični tokovi snaga distributivnih mreža, Magistarska teza, Fakultet tehničkih nauka Novi Sad, novembar 2008.
- [4] V. Strezoski, Osnovi elektroenergetike, FTN Izdavaštvo, Novi Sad, 2014.

Kratka biografija:



Andrija Mitrović rođen je u Loznici 1991. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranio je 2015. godine.



Predrag Vidović rođen u Vlasenici, Bosna i Hercegovina 1981. god. Diplomirao je, magistrirao i doktorirao na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi 2005, 2008. i 2015. godine, respektivno.

PRORAČUN TOKOVA SNAGA MIKSOVANE DISTRIBUTIVNE MREŽE U FAZNOM DOMENU

LOAD FLOW CALCULATION FOR A MIXED DISTRIBUTION SYSTEM IN PHASE DOMAIN

Mirjana Klimenta, Predrag Vidović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je obrađen postupak proračuna nesimetričnih tokova snaga miksovanih distributivnih mreža u faznom domenu zasnovan na procedurama sumiranja struja i korekcija napona. Dat je matematički model i postupak za rješavanje problema. Numerički proračun i analiza rezultata izvršena je za različite vrste potrošača i različite početne aproksimacije napona na primjeru jednostavne test mreže.

Abstract – In this paper a non-symmetric Load Flow calculation for a mixed distribution system in phase domain, based on backward/forward step, is analyzed. Mathematical model and algorithm for solving the problem is described. Results demonstrate the validity of the proposed method for different types of consumers and initial values of voltages in test distribution system.

Ključne reči: Tokovi snaga, fazni domen, miksovane distributivne mreže

1. UVOD

Proračun tokova snaga je najznačajniji i najšire korišten proračun u elektroenergetici. Sastoji se od proračuna promjenljivih stanja, odnosno potpunog režima mreže, na bazi poznatog napona izvora napajanja mreže i poznatih potrošnji u svim čvorovima mreže. Distributivne mreže, za razliku od prenosnih mreža, karakteriše relativno slaba upetljanost i radijalni pogon. Odnos R/X kod vodova i kablova je daleko viši nego u prenosnim mrežama. Navedene razlike su prouzrokovale da standardne metode za proračun tokova snaga u prenosnim mrežama imaju značajno manju efikasnost kada se primjenjuju u distributivnim mrežama [1]. Zbog toga se u distributivnim mrežama koriste algoritmi orjentisani ka granama, koji se svode na direktnu primjenu prvog i drugog Kirhofovog zakona. Problem proračuna nesimetričnih tokova snaga iskrsava odmah nakon utvrđivanja proračuna simetričnih tokova snaga. Nesimetrični režimi jesu realni režimi distributivnih mreža. Uzimanje u obzir nesimetrije povećava dimenziju problema na tri (mreža se ne može tretirati pofazno) [2]. Problem se svodi na uvažavanje prisustva distributivnih transformatora, naročito onih čiji su sekundarni namotaji povezani u trougao ili u zvijezdu sa izolovanim zvijezdištem. Kod miksovanih distributivnih mreža, u svakom koraku algoritma, je potrebno voditi računa o

faznosti elemenata mreže. Proračun se radi u domenu relativnih jedinica.

2. MATEMATIČKI MODEL ELEMENATA MIKSOVANE DISTRIBUTIVNE MREŽE

Modeli elemenata miksovane distributivne mreže (potrošača, vodova i transformatora) dati su u nastavku.

2.1. Model potrošača

Potrošači se modeluju funkcijom:

$$\hat{S}_x(U_x) = P_{ox} \left[k_{spx} + k_{ipx} \frac{U_x}{U_n} + k_{ypx} \left(\frac{U_x}{U_n} \right)^2 \right] - jQ_{ox} \left[k_{sqx} + k_{iqx} \frac{U_x}{U_n} + k_{yqx} \left(\frac{U_x}{U_n} \right)^2 \right], x \in \{a, b, c\}, \quad (1)$$

gdje je :

$$k_{spx} + k_{ipx} + k_{ypx} = 1; \quad (2a)$$

$$k_{sqx} + k_{iqx} + k_{yqx} = 1; \quad (2b)$$

P_{ox} – normalizovana specificirana aktivna snaga potrošača u fazi x;

Q_{ox} – normalizovana specificirana reaktivna snaga potrošača u fazi x.

Tri karakteristične grupe potrošača su:

1) konstantne snage (snaga ne zavisi od napona)

$$k_{spx} = k_{sqx} = 1; k_{ipx} = k_{ypx} = k_{iqx} = k_{yqx} = 0; \quad (3)$$

2) konstantne struje (snaga linearno srazmjerna sa naponom)

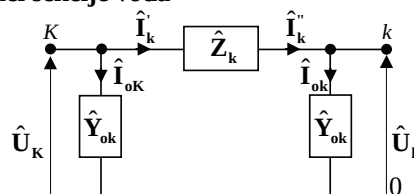
$$k_{ipx} = k_{iqx} = 1; k_{spx} = k_{ypx} = k_{sqx} = k_{yqx} = 0; \quad (4)$$

3) konstantne admitanse (snaga srazmjerna s kvadratom napona)

$$k_{ypx} = k_{yqx} = 1; k_{spx} = k_{ipx} = k_{sqx} = k_{iqx} = 0. \quad (5)$$

Pored ove tri karakteristične grupe potrošača, potrošači se mogu predstaviti i kao kombinacija opisanih grupa. Da bi se obezbjedilo ažuriranje nulte komponente napona u iterativnom proračunu, svi potrošači koji se napajaju sa transformatora čiji su sekundarni namotaji povezani u trougao ili izolovanu zvijezdu, su predstavljeni konstantnim admitansama [3].

2.2. Model sekcije voda



Slika 1. Načelna šema n-fazne sekcije (voda)

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Predrag Vidović, doc.

Matematički model sekcije voda sa slike 1, gdje su K i k čvorovi na početku i kraju sekcije, respektivno, glasi:

$$\hat{\mathbf{I}}_k' = \hat{\mathbf{I}}_k'', \quad (6)$$

$$\hat{\mathbf{I}}_{oK} = \hat{\mathbf{Y}}_{oK} \hat{\mathbf{U}}_K, \quad (7)$$

$$\hat{\mathbf{I}}_{ok} = \hat{\mathbf{Y}}_{ok} \hat{\mathbf{U}}_k, \quad (8)$$

$$\hat{\mathbf{U}}_k = \hat{\mathbf{U}}_K - \hat{\mathbf{Z}}_k \hat{\mathbf{I}}_k', \quad (9)$$

gdje su:

$\hat{\mathbf{U}}_K, \hat{\mathbf{U}}_k$ – vektori faznih napona na početku i kraju sekcije, dimenzija $n \times 1$;

$\hat{\mathbf{I}}_k', \hat{\mathbf{I}}_k''$ – međusobno jednaki vektori faznih struja na početku i kraju sekcije, dimenzija $n \times 1$;

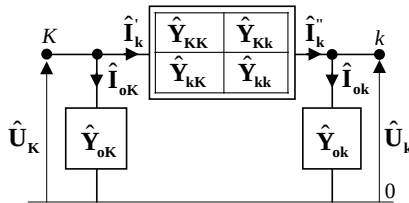
$\hat{\mathbf{I}}_{oK}, \hat{\mathbf{I}}_{ok}$ – vektori faznih struja otočnih parametara na početku i kraju sekcije, dimenzija $n \times 1$;

$\hat{\mathbf{Z}}_k, \hat{\mathbf{Y}}_{ok}$ – matricni reprezentanti rednih i otočnih parametara sekcije (jednaki na oba kraja sekcije), dimenzija $n \times n$;

n – broj faza voda.

2.3. Model transformatora

Na slici 2 je prikazana načelna šema trofaznog transformatora.



Slika 2. Načelna šema trofaznog transformatora

Matematički model transformatora glasi:

$$\hat{\mathbf{I}}_{oK} = \hat{\mathbf{Y}}_{oK} \hat{\mathbf{U}}_K, \quad (10)$$

$$\hat{\mathbf{I}}_{ok} = \hat{\mathbf{Y}}_{ok} \hat{\mathbf{U}}_k, \quad (11)$$

$$\hat{\mathbf{I}}_k' = \hat{\mathbf{Y}}_{KK} \hat{\mathbf{U}}_K + \hat{\mathbf{Y}}_{Kk} \hat{\mathbf{U}}_k, \quad (12)$$

$$-\hat{\mathbf{I}}_k'' = \hat{\mathbf{Y}}_{kK} \hat{\mathbf{U}}_K + \hat{\mathbf{Y}}_{kk} \hat{\mathbf{U}}_k, \quad (13)$$

gdje su:

$\hat{\mathbf{U}}_K, \hat{\mathbf{U}}_k$ – vektori faznih napona primara i sekundara, dimenzija 3×1 ;

$\hat{\mathbf{I}}_k', \hat{\mathbf{I}}_k''$ – međusobno različiti vektori faznih struja primara i sekundara, dimenzija 3×1 ;

$\hat{\mathbf{I}}_{oK}, \hat{\mathbf{I}}_{ok}$ – vektori faznih struja otočnih parametara primara i sekundara transformatora, dimenzija 3×1 ;

$\hat{\mathbf{Y}}_{oK}, \hat{\mathbf{Y}}_{ok}$ – matricni reprezentanti magnećenja transformatora (otočni parametri) predstavljeni alternativno na primaru, odnosno sekundaru transformatora, dimenzija 3×3 . Matricni reprezentanti rednih parametara transformatora, iz relacija (12 i 13), dimenzija 3×3 , dati su u tabeli 1. Matrice iz tabele 1 odnose se na sprege $\mathbf{Y}_y$ i $\mathbf{D}_d$ sa sprežnim brojem 0, a za $\mathbf{Y}_d$ i $\mathbf{D}_y$ sa sprežnim brojem 1 [3].

Tabela 1. Matricni reprezentanti rednih parametara transformatora

Sprega	$\hat{\mathbf{Y}}_{KK}$	$\hat{\mathbf{Y}}_{Kk}$	$\hat{\mathbf{Y}}_{kK}$	$\hat{\mathbf{Y}}_{kk}$
$\mathbf{Y}_{Ny_n}$	$\hat{\mathbf{Y}}_I$	$-\hat{\mathbf{Y}}_I$	$-\hat{\mathbf{Y}}_I$	$\hat{\mathbf{Y}}_I$

$\mathbf{Y}_{Ny_n}$	$\hat{\mathbf{Y}}_{II}$	$-\hat{\mathbf{Y}}_{II}$	$-\hat{\mathbf{Y}}_{II}$	$\hat{\mathbf{Y}}_{II}$
$\mathbf{Y}_{yy}$	$\hat{\mathbf{Y}}_{II}$	$-\hat{\mathbf{Y}}_{II}$	$-\hat{\mathbf{Y}}_{II}$	$\hat{\mathbf{Y}}_{II}$
$\mathbf{Y}_{Nd}$	$\hat{\mathbf{Y}}_I$	$\hat{\mathbf{Y}}_{III}$	$\hat{\mathbf{Y}}_{III}^T$	$\hat{\mathbf{Y}}_{II}$
$\mathbf{Y}_d$	$\hat{\mathbf{Y}}_{II}$	$\hat{\mathbf{Y}}_{III}^T$	$\hat{\mathbf{Y}}_{III}$	$\hat{\mathbf{Y}}_{II}$
$\mathbf{D}_{y_n}$	$\hat{\mathbf{Y}}_{II}$	$\hat{\mathbf{Y}}_{III}^T$	$\hat{\mathbf{Y}}_{III}$	$\hat{\mathbf{Y}}_I$
$\mathbf{D}_y$	$\hat{\mathbf{Y}}_{II}$	$\hat{\mathbf{Y}}_{III}^T$	$\hat{\mathbf{Y}}_{III}$	$\hat{\mathbf{Y}}_{II}$
$\mathbf{D}_d$	$\hat{\mathbf{Y}}_{II}$	$-\hat{\mathbf{Y}}_{II}$	$-\hat{\mathbf{Y}}_{II}$	$\hat{\mathbf{Y}}_{II}$

gdje su navedene matrice:

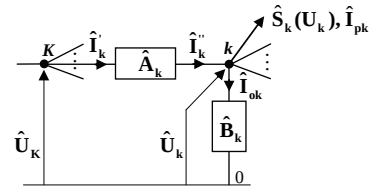
$$\hat{\mathbf{Y}}_I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{Y}}, \quad (14)$$

$$\hat{\mathbf{Y}}_{II} = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \hat{\mathbf{Y}}, \quad (15)$$

$$\hat{\mathbf{Y}}_{III} = \frac{1}{\sqrt{3}} \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix} \hat{\mathbf{Y}}. \quad (16)$$

2.3. Model mreže

Osnovu za proračun tokova snaga radialne mreže čini numeracija mreže po slojevima [4]. Mreža se predstavlja preko Γ segmenata složenih po slojevima.



Slika 3. Pogonska šema segmenta Γ_k distributivne mreže. Svaki Γ segment je asociran sekciji voda ili transformatoru. $\hat{\mathbf{A}}_k$ je matricni reprezentant rednih parametara grane asocirane Γ_k segmentu.

$\hat{\mathbf{B}}_k$ je admitantni matricni reprezentant otočnih parametara Γ_k segmenta i predstavlja sumu admitantnih matricnih reprezentata otočnih parametara kraja sekcije ili transformatora asociranih Γ_k segmentu, otočnih parametara na počecima sekcija ili transformatora koji se napajaju preko razmatrane sekcije ili transformatora i matricnih reprezentata elemenata direktno priključenih u čvoru k (npr. baterija kondenzatora, potrošač,...). Segmenti koji se napajaju direktno sa Γ_k segmenta, kao i elementi priključeni direktno u k -tom čvoru, generalno mogu biti različite faznosti od Γ_k segmenta. Zbog toga se u toku iterativnog postupka reprezentanti otočnih grana tih segmenata (elemenata) privremeno proširuju nulama na onim mjestima koja postoje u matricnom reprezentu otočne grane Γ_k segmenta. Tek nakon ovog koraka se sabiraju otočni reprezentanti svih grana koje se stiču u čvoru k .

3. PRORAČUN TOKOVA SNAGA MIKSOVANE DISTRIBUTIVNE MREŽE

Proračun tokova snaga se radi kroz iterativni postupak i to u tri koraka koja su opisana u nastavku.

3.1. Proračun struja otočnih elemenata

U svakom čvoru k , počevši od čvorova u prvom sloju se izračunava vektor struja potrošača na sledeći način:

$$\hat{\mathbf{I}}_{pk} = [\hat{S}_{kx} (U_{kx}) / \hat{U}_{kx}^*]^T, x \in \{a, b, c\}. \quad (17)$$

Prilikom obrade potrošača koji se nalaze iza transformatora čiji su sekundarni namotaji povezani u trougao ili izolovanu zvijezdu, struja potrošača se računa koristeći relaciju:

$$\hat{\mathbf{I}}_{pk} = [\hat{Y}_{ka} \hat{U}_{ka}, \hat{Y}_{kb} \hat{U}_{kb}, \hat{Y}_{kc} \hat{U}_{kc}]^T, \quad (18)$$

gdje su $\hat{Y}_{ka}, \hat{Y}_{kb}, \hat{Y}_{kc}$ konstantne admitanse kojima se modeluju potrošači.

Vektor otočnih struja u čvoru k se računa na sledeći način:

$$\hat{\mathbf{I}}_{ok} = \hat{\mathbf{B}}_k \hat{\mathbf{U}}_k. \quad (19)$$

Ako je potrošač u čvoru k različite faznosti od faznosti Γ_k segmenta sa kojeg se napaja, onda se vektor struja potrošača u čvoru k proširuje nulama, na mjestima onih faza Γ_k segmenta kojih nema u potrošaču. Nakon toga se izračunava struja otočnih elemenata u čvoru k :

$$\hat{\mathbf{I}}_k = \hat{\mathbf{I}}_{pk} + \hat{\mathbf{I}}_{ok}. \quad (20)$$

3.2. Procedura sumiranja struja

U proceduri sumiranja struja se polazi od segmenta k u poslednjem lejeru i ide do prvog segmenta u prvom lejeru. Kada se stigne do k -tog segmenta (Γ_k), vektori struja svih segmenata koje se napajaju sa Γ_k segmenta, a imaju manje faza od Γ_k segmenta, se proširuju, i to nulama, na onim mjestima koje odgovaraju fazama Γ_k segmenta, a kojih nema u vektoru struja datog segmenta. Ako je Γ_k segmenat asociran sekciji voda, struja segmenta je:

$$\hat{\mathbf{I}}_k = \hat{\mathbf{I}}_k'' + \sum_{j \in \alpha_k} \hat{\mathbf{I}}_j, \quad (21)$$

gdje je:

$\hat{\mathbf{I}}_j$ – vektor struja segmenta j koji se direktno napaja

sa segmenta k , proširen na gore naveden način;

α_k – skup indeksa svih segmenata koji se direktno napajaju sa segmenta k .

Ukoliko je Γ_k segment asociran transformatoru, struja sekundara se računa koristeći samo struje otočnih elemenata čvora k , pošto se sa transformatora ne napajaju grane, a iznošenje struja na primar u zavisnosti od sprege transformatora, može da se uradi na sledeći način:

Sprega $Y_{N\bar{N}}, D\bar{Y}_{\bar{N}}$

$$\hat{\mathbf{I}}_k = -\hat{\mathbf{Y}}_{KK} \hat{\mathbf{Y}}_{KK}^{-1} \hat{\mathbf{I}}_k'' + (\hat{\mathbf{Y}}_{KK} - \hat{\mathbf{Y}}_{KK} \hat{\mathbf{Y}}_{KK}^{-1} \hat{\mathbf{Y}}_{KK}) \hat{\mathbf{U}}_k, \quad (22)$$

Sprega $D\bar{Y}_{\bar{N}}, Y_{N\bar{N}}, D\bar{d}$

$$\hat{\mathbf{I}}_k = \hat{\mathbf{Y}}_{KK} \hat{\mathbf{U}}_K + \hat{\mathbf{Y}}_{Kk} \hat{\mathbf{U}}_k. \quad (23)$$

3.3. Procedura korekcije napona

U ovoj proceduri se polazi od čvorova u prvom lejeru do čvorova u poslednjem lejeru. Ukoliko je Γ_k segment asociran sekciji voda napon čvora na kraju segmenta se računa na sledeći način:

$$\hat{\mathbf{U}}_k = \hat{\mathbf{U}}_K - \hat{\mathbf{Z}}_k \hat{\mathbf{I}}_k. \quad (24)$$

Ako je čvor K različite faznosti od čvora k , u vektoru napona čvora K koriste se samo one faze koje odgovaraju fazama vektora čvora k . Na taj način se dobijaju odgovarajuće dimenzije vektora i matrica. Ukoliko je Γ segment asociran transformatoru, proračun napona

sekundara, u zavisnosti od sprege transformatora, može da se uradi na sledeći način:

Sprega $Y_{N\bar{N}}, D\bar{Y}_{\bar{N}}$

$$\hat{\mathbf{U}}_k = -\hat{\mathbf{Y}}_{KK}^{-1} (\hat{\mathbf{I}}_k'' + \hat{\mathbf{Y}}_{KK} \hat{\mathbf{U}}_K). \quad (25)$$

Sprega $Y_{N\bar{d}}, D\bar{d}$

Zbog singularnosti matrica, relacija (25) ne može da se primjeni za korekciju napona. Potrošači koji se napajaju sa sekundara ovih transformatora su predstavljeni konstantnom admitansom i nulta komponenta struje je jednaka nuli, pa može da se zapiše relacija:

$$0 = \hat{Y}_{ka} \hat{U}_{ka} + \hat{Y}_{kb} \hat{U}_{kb} + \hat{Y}_{kc} \hat{U}_{kc}. \quad (26)$$

Zamjenom treće relacije iz (13) relacijom (26), dobija se:

$$-\hat{\mathbf{I}}_k'' = (\hat{\mathbf{Y}}_{KK}) \hat{\mathbf{U}}_K + (\hat{\mathbf{Y}}_{KK}) \hat{\mathbf{U}}_k. \quad (27a)$$

Relacija (27a) matricno može da se zapiše na sledeći način:

$$\begin{bmatrix} -\hat{I}_{ka}'' \\ -\hat{I}_{kb}'' \\ 0 \end{bmatrix} = \begin{bmatrix} / & / & / \\ / & / & / \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{U}_{Ka} \\ \hat{U}_{Kb} \\ \hat{U}_{Kc} \end{bmatrix} + \begin{bmatrix} / & / & / \\ / & / & / \\ \hat{Y}_{ka} & \hat{Y}_{kb} & \hat{Y}_{kc} \end{bmatrix} \begin{bmatrix} \hat{U}_{ka} \\ \hat{U}_{kb} \\ \hat{U}_{kc} \end{bmatrix}. \quad (27b)$$

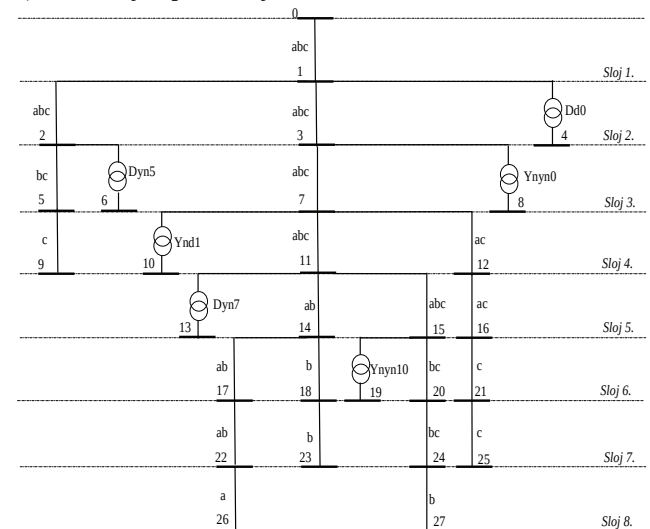
/ označava da su elementi isti kao u relaciji (13). Koristeći relacije (27) korekcija napona sekundara radi se relacijom:

$$\hat{\mathbf{U}}_k = -\hat{\mathbf{Y}}_{KK}^{-1} (\hat{\mathbf{I}}_k'' + \hat{\mathbf{Y}}_{KK} \hat{\mathbf{U}}_K). \quad (28)$$

4. NUMERIČKA VERIFIKACIJA PRORAČUNA TOKOVA SNAGA

Za miksovanu distributivnu test mrežu sa slike 4 su urađene simulacije rezultata da bi se utvrdila zavisnost broja iteracija od:

- 1) Modela potrošnje
- 2) Povećanja opterećenja.



Slika 4. Test miksovana distributivna mreža

Za različite početne aproksimacije napona:

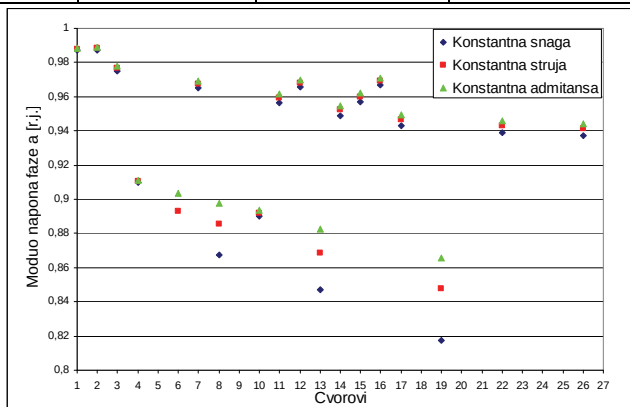
- 1) $\hat{U}_a^1 = e^{j0}$; $\hat{U}_b^1 = e^{-j120}$; $\hat{U}_c^1 = e^{j120}$,
- 2) $\hat{U}_a^1 = 0.8e^{j20}$; $\hat{U}_b^1 = 0.7e^{j60}$; $\hat{U}_c^1 = 0.9e^{j100}$,

u tabeli 2 su dati naponi čvorova, za potrošače predstavljene konstantnom admitansom. S obzirom da se dobijaju potpuno isti rezultati, kao i da je broj iteracija isti u oba slučaja, može se zaključiti da je postupak neosjetljiv na početnu aproksimaciju napona u slučaju malih i slabo opterećenih mreža. Najveći moduli napona u čvorovima su u slučaju potrošača čije su snage kvadratno zavisne od

napona, a najmanji u slučaju potrošača konstantne snage. Ovaj efekat je izraženiji na čvorovima koji su udaljeniji od izvora napajanja (slika 5).

Tabela 2. Naponi čvorova

Čvor	$\hat{U}_a$ (kV)	$\hat{U}_b$ (kV)	$\hat{U}_c$ (kV)
1	$11.537e^{j0.011}$	$11.533e^{j239.997}$	$11.537e^{j119.981}$
2	$11.538e^{j0.012}$	$11.531e^{j240.003}$	$11.535e^{j119.973}$
3	$11.529e^{j0.022}$	$11.523e^{j239.989}$	$11.531e^{j119.973}$
4	$0.230e^{j0.713}$	$0.225e^{j238.956}$	$0.234e^{j118.587}$
5	-	$11.529e^{j240.009}$	$11.533e^{j119.965}$
6	$0.229e^{j209.401}$	$0.230e^{j89.456}$	$0.230e^{j30.601}$
7	$11.521e^{j0.031}$	$11.514e^{j239.981}$	$11.527e^{j119.965}$
8	$0.229e^{j0.575}$	$0.229e^{j239.451}$	$0.230e^{j119.374}$
9	-	-	$11.532e^{j119.962}$
10	$0.230e^{j29.294}$	$0.224e^{j208.956}$	$0.234e^{j88.582}$
11	$11.515e^{j0.052}$	$11.505e^{j239.963}$	$11.528e^{j119.960}$
12	$11.522e^{j0.021}$	-	$11.522e^{j119.961}$
13	$0.229e^{j149.390}$	$0.229e^{j29.455}$	$0.230e^{j269.395}$
14	$11.509e^{j0.063}$	$11.502e^{j239.943}$	-
15	$11.515e^{j0.060}$	$11.500e^{j239.965}$	$11.526e^{j119.947}$
16	$11.523e^{j0.014}$	-	$11.518e^{j119.957}$
17	$11.505e^{j0.065}$	$11.502e^{j239.931}$	-
18	-	$11.500e^{j239.938}$	-
19	$0.229e^{j59.368}$	$0.230e^{j60.591}$	$0.230e^{j179.462}$
20	-	$11.496e^{j239.967}$	$11.526e^{j119.935}$
21	-	-	$11.516e^{j119.951}$
22	$11.502e^{j0.064}$	$11.502e^{j239.924}$	-
23	-	$11.498e^{j239.936}$	-
24	-	$11.494e^{j239.967}$	$11.526e^{j119.928}$
25	-	-	$11.514e^{j119.948}$
26	$11.501e^{j0.062}$	-	-
27	-	$11.492e^{j239.964}$	-



Slika 5. Normalizovani moduli napona faze a

Maksimalna apsolutna vrijednost razlike normalizovanih napona u dvije uzastopne iteracije je data u tabeli 3.

Tabela 3. Maksimalna apsolutna vrijednost razlike normalizovanih napona tokom iterativnog postupka

Iteracija	$k_{sp}=k_{sq}=1$ $k_{ip}=k_{iq}=0$ $k_{yp}=k_{yq}=0$	$k_{sp}=k_{sq}=0$ $k_{ip}=k_{iq}=1$ $k_{yp}=k_{yq}=0$	$k_{sp}=k_{sq}=0$ $k_{ip}=k_{iq}=0$ $k_{yp}=k_{yq}=1$
1	1.931852	1.931852	1.931852
2	$1.453145 \cdot 10^{-2}$	$1.453145 \cdot 10^{-2}$	$1.453145 \cdot 10^{-2}$
3	$1.976278 \cdot 10^{-4}$	$2.072350 \cdot 10^{-4}$	$2.333866 \cdot 10^{-4}$
4	$8.584932 \cdot 10^{-6}$	$9.262881 \cdot 10^{-6}$	$11.87392 \cdot 10^{-6}$
5	-	-	$5.397430 \cdot 10^{-7}$

Na osnovu rezultata se vidi da se kreće od iste maksimalne apsolutne vrijednosti razlike normalizovanih

napona u prvoj iteraciji, a da se tokom iterativnog postupka ta razlika najbrže smanjuje kada su potrošači predstavljeni konstantnom snagom, odnosno najsporije kada su potrošači predstavljeni konstantnom admitansom. Iz tabele 4 se vidi da povećanjem opterećenja raste broj iteracija. Pošto je test mreža slabo opterećena, broj iteracija se povećava jako malo. Za osnovno opterećenje je broj iteracija veći za jednu kada su potrošači predstavljeni konstantnom admitansom nego konstantnom snagom ili strujom.

Tabela 4. Broj iteracija u zavisnosti od opterećenja

OPT	$k_{sp}=k_{sq}=1$ $k_{ip}=k_{iq}=0$ $k_{yp}=k_{yq}=0$	$k_{sp}=k_{sq}=0$ $k_{ip}=k_{iq}=1$ $k_{yp}=k_{yq}=0$	$k_{sp}=k_{sq}=0$ $k_{ip}=k_{iq}=0$ $k_{yp}=k_{yq}=1$	$k_{sp}=k_{sq}=0.1$ $k_{ip}=k_{iq}=0.2$ $k_{yp}=k_{yq}=0.7$
50%	4	4	4	4
100%	4	4	5	5
150%	5	5	5	5
300%	5	5	5	5

OPT predstavlja procenat od osnovnog opterećenja.

5. ZAKLJUČAK

U ovom radu je opisan algoritam proračuna tokova snaga miksovane distributivne mreže, zasnovan na procedurama sumiranja struja i korekcija napona u faznom domenu. U svakom koraku algoritma je potrebno voditi računa o faznosti elemenata mreže. Proračun se radi za sve tri faze, a problem ažuriranja nulte komponente napona zbog singularnosti matrica se radi na gore opisan način. Pošto je mreža mala i slabo opterećena, početna aproksimacija napona ne utiče na broj iteracija. Moduli napona u čvorovima su najveći kada je potrošnja predstavljena konstantnom admitansom. Povećavanjem opterećenja raste broj iteracija. Modeli potrošača malo utiču na broj iteracija ako je mreža slabo opterećena.

6. LITERATURA

- [1] D.Popović, D.Bekut, V.Dabić, *Specijalizovani DMS algoritmi*, Prosveta, Srbija: Novi Sad, 2011.
- [2] A. Ulinuha, M.A.S. Masoum, S. M. Islam, *Unbalance Power Flow Calculation for a Radial Distribution System Using Forward – Backward Propagation Algorithm*, Power Engineering Conference, 2007. AUPEC 2007, Australasian Universities
- [3] P.Vidović, *Nesimetrični tokovi snaga distributivnih mreža*, Magistarska teza, Fakultet tehničkih nauka Novi Sad, novembar 2008.
- [4] V.C.Strezoski, *Analiza elektroenergetskih sistema*, skripta, Fakultet tehničkih nauka Novi Sad, 2011.

Kratka biografija:



Mirjana Klimenta je rođena u Trebinju, 1991. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranila je 2015. godine.



Predrag Vidović je rođen u Vlasenici, 1981. godine. Diplomirao je, magistrirao i doktorirao, na Fakultetu tehničkih nauka u Novom Sadu, iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi, 2005, 2008, odnosno 2015. godine, respektivno.

ДИСТРИБУИРАНИ СИСТЕМ ЗА ПОВЕЗИВАЊЕ C# ПРОЈЕКТА**DISTRIBUTED SYSTEM FOR BUILDING C# PROJECTS**

Радосав Белоица, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У оквиру овог рада укратко ће бити изложена имплементација архитектуре једног од могућих решења при пројектовању система за дистрибуирано повезивање програмског кода. Акцент ће бити стављен на архитектуру система, обрасце који су се користили при пројектовању решења као и предности које је у продукцији донело коришћење дистрибуираног повезивања програмског кода у односу на централизовано повезивање програмског кода. На основу резултата алгорита који су коришћени у дистрибуираном систему за повезивање програмског кода на различитом броју чворова извешће се закључак који алгоритам се најбоље показао у продукцији. Такође ћемо размотрити који је то оптималан број чворова који би требало да учествују у дистрибуираном повезивању у оквиру примене сваког од алгорита.

Abstract – In this paper we will expose the implementation for the architecture of one of the possible solutions in system design for distributed building of program code. Accent will be placed on the system architecture, the patterns that were used in the design of the solution and the benefits that distributed building of program code has brought into production in relation to centralized building of program code. Based on the results of algorithms that are used in a distributed system for building program code on a different number of nodes we will come to the conclusion which algorithm showed the best results in the production. We will also consider what is the optimal number of nodes which should be used to participate in a distributed networking within the application of each of the algorithms.

Кључне речи: повезивање пројекта, дистрибуирани системи, алгоритми

1. УВОД

Рад се бави пројектовањем архитектуре система, обрасца који су се користили при пројектовању решења као и предностима које су у продукцији донеле примене алгорита.

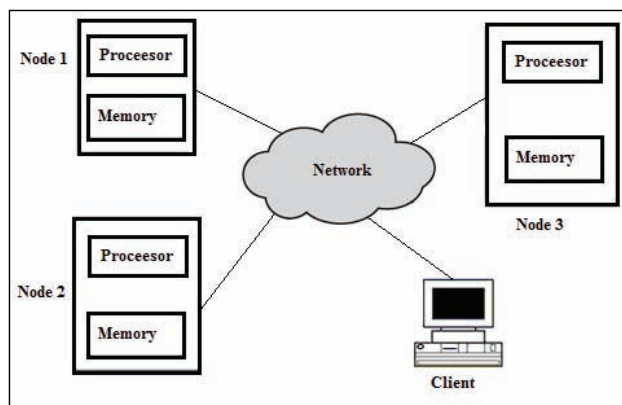
Пројектно решење засновано је на коришћењу три алгорита за дистрибуирано повезивање:

- алгоритам равномерне расподеле пројекта
- алгоритам груписања међусобно зависних пројекта
- алгоритам обраде последње измењених пројекта

На програмском нивоу апликација је заснована на *Controller/Worker* архитектонском шаблону [1] који раздваја архитектуру апликације на две компоненте *Controller* и *Worker* које имају одређене улоге у систему.

2. ОПИС ДИСТРИБУИРАНОГ СИСТЕМА

Дистрибуирани систем се састоји од скупа рачунара повезаних кроз рачунарску мрежу који преко заједничког дистрибуираног посредника усклађују своје активности и деле ресурсе система, тако да крајњи корисник дистрибуираног система доживљава систем као једну интегрисану целину [1].



Слика 1. Концептуални приказ дистрибуираног система

За потребе овог рада коришћен систем је организован по принципу кластера који спадају у групу дистрибуираних система за интензивно рачунање.

Кластер чини неколико идентичних рачунара повезаних у исту мрежу, где сваки рачунар поседује своју инстанцу оперативног система истог типа. Цео кластер се логички понаша као један рачунар [2].

3. ИМПЛЕМЕНТАЦИЈА ДИСТРИБУИРАНОГ СИСТЕМА

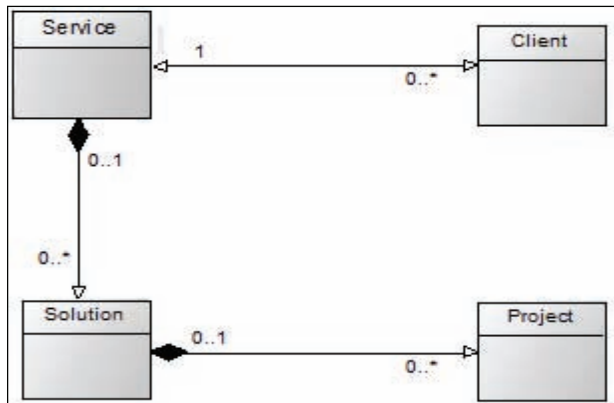
Пројектно решење моделовано је тако да се на њему може уочити неколико типова ентитета као и разне врсте веза између њих. У ентитету типа *Service* имплементирана је логика дистрибуираног система за повезивање кода и у себи садржи колекцију ентитета

НАПОМЕНА:

Овај рад је произтекао из мастер рада чији је ментор био др Дарко Чапко.

типа *Solution* која представља једну логичку целину, тачније у себи садржи ентитете типа *Project* над којим се врши дистрибуирано повезивање кода.

Ентитет типа *Client*, који у општем случају представља физичку машину, преко одговарајућег интерфејса, изложеног од стране сервиса, одабира тип дистрибуираног повезивања који ће се извршавати над жељеним ентитетом типа *Solution*.

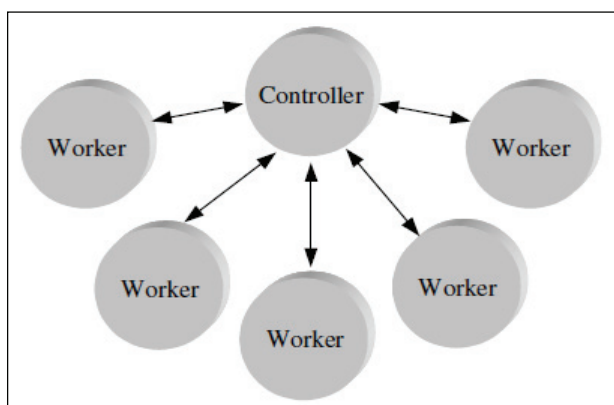


Слика 2. Ентитети пројектног решења

На програмском нивоу апликација је заснована на *Controller/Worker* архитектонском шаблону [1] који раздваја архитектуру апликације на следеће компоненте:

- Контролер (*Controller*)
- Радник (*Worker*)

Controller/Worker образац употребљавамо када задатак можемо ефикасније извршити тако што ћемо га поделити на мање делове, које ћемо путем рачунарске мреже дистрибуирати скупу радника. Контролер је повезан са скупом радника којима делегира подзатке и саставља одговор клијенту који је затражио извршење задатка. Радници ће посао извршења делова задатка урадити у паралели и на тај начин убрзати процес рачунања [1].



Слика 3. Програмска архитектура апликације

3.1. Контролер (*Controller*)

Улогу контролера у *Controller/Worker* обрасцу [1] има ентитет *Service* који прихвата захтеве од стране клијената који учествују у дистрибуираном повезивању, врши расподелу пројеката који се дистрибуирано повезују радницима и прослеђује резултате назад клијенту који је иницирао процес.

Радници се претходно морају пријавити сервису да би учествовали у процесу дистрибуираног повезивања. Сам клијент који иницира процес дистрибуираног повезивања такође учествује у њему тј. сервис ће и њему доделити одређени број пројеката, тако да се и клијент који је иницирао повезивање може посматрати као један од радника. Он се од осталих радника разликује по томе што ће на крају процеса сервис само њему проследити резултате рада осталих радника.

3.2. Радник (*Worker*)

Улогу радника у *Controller/Worker* обрасцу [1] има ентитет *Client* који дистрибуирано повезује пројекте које му је претходно доделио сервис. Након повезивања групе пројеката која му је додељена сваки клијент своје резултате шаље на заједнички репозиторијум (операција *push*).

Репозиторијум је локација на који се софтверски пакет може послати и са којег се софтверски пакет може додати [3]. При имплементацији пројектног решења коришћен је *Mercurial* дистрибуирани систем за контролу верзија и *TortoiseHG* који представља екстензију за рад са *Mercurial* дистрибуираним системом за контролу верзија на *Windows* оперативном систему [4].

3.3. Алгоритми дистрибуираног повезивања

При имплементацији дистрибуираног повезивања програмског кода коришћене су три врсте алгоритама који си независни једни од других и омогућавају клијенту да изабере начин на који ће се дистрибуирано повезивање извршити.

С обзиром да централна тема овог рада није начин имплементације алгоритама за дистрибуирано повезивање програмског кода, већ предности које је у продукцији донела њихова примена само укратко ћемо навести које су алгоритми коришћени при имплементацији пројектног задатка. Детаљан опис коришћених алгоритама као и начин имплементације истих описан је у раду "Алгоритми за дистрибуирано повезивање и превођење C# пројеката" [5], који заједно са овим радом ствара коментну слику функционисања и изгледа дистрибуираног система и указује на предности оваког система у односу на класичне централизоване системе.

3.3.1. Алгоритам равномерне расподеле пројеката

Овај алгоритам, као и остали, састоји се из два дела. У првом се врши груписање пројеката на основу одређеног критеријума, док се у другом врши расподела пројеката по клијентским машинама, а потом и њихово превођење и повезивање које се обавља по нивоима. Његова реализација одвија се и на клијентској и на серверској страни [5].

3.3.2. Алгоритам груписања међусобно зависних пројеката

Основна предност овог алгоритма је та што се клијенту на дистрибуирано повезивање одједном шаље читава подгрупа пројеката. На овај начин се смањује и комуникација између клијената, сервиса и удаљеног репозиторијума због тога што ће клијент

након повезивања целе подгрупе пројеката своје резултате одједном послати на заједнички репозиторијум [5].

3.3.3. Алгоритам обраде последње измењених пројеката

У оквиру овог алгоритма на дистрибуирано повезивање шаљу се само пројекти над којима је извршена нека измена. Не врши се провера зависности између пројеката који су измењени у односу на оне пројекте које нису измењени, због тога што се на репозиторијуму већ налази повезана верзија пројеката који нису измењени и она је свежа [5].

4. ПРИКАЗ ИМПЛЕМЕНТИРАНОГ РЕШЕЊА

Да би почели са радом над апликацијом претходно морамо подигнути и конфигурисати сервис који користе сви клијенти који учествују у дистрибуираном повезивању. Сервис који се подиже за размену података очекује податке на порту 4000 и користи *TCP (Transmission Control Protocol)* протокол који је конекционо оријентисан. *TCP* нуди поуздан пренос података са претходном успоставом везе уз контроле тока и загушења [6].

Након што смо конфигурисали сервис покрећемо клијенте које морамо пријавити сервису да би сервис знао који клијенти учествују у повезивању пројеката и доделио им пројекте.

Сервис региструје клијенте који су се пријавили за дистрибуирано повезивање пројеката и након иницирања почетка дистрибуираног повезивања од стране једног од клијената додељује им пројекте које клијенти дистрибуирано повезују. Појединачне резултате повезивања пројеката клијенти шаљу на заједнички репозиторијум, након чега сервис са репозиторијума клијенту који је иницирао повезивање пројеката шаље све измене које су доспеле на репозиторијум од стране клијената који учествују у повезивању пројеката.

5. РЕЗУЛТАТИ ПРИМЕНЕ АЛГОРИТАМА

У табели 1 приказани су резултати централизованог повезивања програмског кода за различит број пројеката.

Повезивање пројеката у VS 2013	1 машина
26 пројеката	8230ms
97 пројеката	279930ms
152 пројеката	332674ms
274 пројеката	960215ms
441 пројеката	1320397ms

Табела 1. Централизовано повезивање

5.1 Алгоритам равномерне расподеле пројеката

Примена овог алгоритма није побољшала време повезивања, стога ћемо понашање овог алгоритма приказати на само једном софтверском пакету који садржи 274 пројеката.

Алгоритам РРП	274 пројеката
2 машине	2113800 ms
4 машине	1629621 ms
6 машина	1872600 ms

Табела 2. Резултати алгоритма РРП

Из табеле 1 видимо да је време централизованог повезивања програмског кода, које за 274 пројеката износи 960215 ms што је око 16 минута, знатномање од времена дистрибуираног повезивања програмског кода из табеле 2 за било који број физичких машина. Најбоље време остварено је при коришћењу 4 машине и оно износи 1629621 ms што је око 27 минута. Из табеле 2 можемо видети и то да је остварено време дистрибуираног повезивања на 6 машина од 1872600 ms веће него време остварено на 4 машине што је директна последица превелике комуникације као и услова под којима се пројекти шаљу на повезивање. Стога констатујемо да примена овог алгоритма није препоручљива као алтернатива централизованом повезивању пројеката па стога нема сврхе даље тестирати алгоритам.

5.2 Алгоритам груписања међусобно зависних пројеката

Понашање овог алгоритма тестирано је на софтверским пакетима са различитим бројем пројеката.

Алгоритам ГМЗП	2 машине	4 машине	6 машина
26 пројеката	7356 ms	5982 ms	3956ms
97 пројеката	237537 ms	139742 ms	87762 ms
152 пројеката	236788 ms	138136 ms	111781 ms
274 пројеката	707127 ms	356790 ms	260619 ms
441 пројеката	900615 ms	620544 ms	489042 ms

Табела 3. Резултати алгоритма ГМЗП

Са табеле 3 видимо да разлика у оствареном времену повезивања при коришћењу алгоритма на 4 машине и 6 машина за било који број пројеката није значајно велика па је оптималан број машина на којем се препоручује примена алгоритма. Пошто се алгоритам ГМЗП показао најбоље од свих алгоритама за дистрибуирано повезивање у табели 4 јеи у процентима приказано убрзање које је алгоритам ГМЗП остварио за било који број машина и пројеката у односу на централизовано повезивање.

Убрзање у процентима	2 машине	4 машине	6 машина
26 пројеката	10.62 %	27.31%	51.93 %
97 пројеката	15.14 %	50.10 %	68.65 %
152 пројеката	28.82 %	58.48 %	66.40 %
274 пројеката	26.36 %	62.84 %	72.86%
441 пројеката	31.79 %	53.00 %	62.96 %

Табела 4. Убрзање повезивања алгоритма ГМЗП

Примера ради поредећи однос резултата између табеле 1 и табеле 3 за 274 пројеката, видимо да је време повезивања на оптималном броју машина (4 машине) смањено за 603425ms што је за око 10 минута мање. Изражено у процентима у табели 4 то време представља убрзање од 62.84 % у односу на време централизованог повезивања истог броја пројеката.

5.3 Алгоритам обраде последње измењених пројеката

У табели 5 приказани су резултати инкременталног централизованог повезивања програмског кода за различит број пројеката.

Инкрементално повезивање пројеката у VS 2013	1 машина
6 од 26 пројеката	2380ms
6 од 97 пројеката	10624ms
6 од 152 пројеката	16648ms
6 од 274 пројеката	30011ms
6 од 441 пројеката	48302 ms

Табела 5. Инкрементално повезивање

Понашање алгоритма тестирано је на софтверским пакетима са различитим бројем пројеката.

Алгоритам ПИП	2 машине	4 машине	6 машина
6 од 26 пројеката	1895ms	1206ms	973ms
6 од 97 пројеката	7257ms	5257ms	4939ms
6 од 152 пројеката	13143ms	8324ms	6788 ms
6 од 274 пројеката	23022ms	19510ms	14974ms
6 од 441 пројеката	37054ms	31401ms	24101ms

Табела 6. Резултати алгоритма ПИП

Поређењем добијених резултата између табеле 5 и табеле 6 видимо да је време централизованог повезивања програмског кода нешто више од времена добијеног применом алгоритма последње измењених пројеката за било који број физичких машина и пројеката. Обзиром да разлика у оствареном времену повезивања при коришћењу алгоритма на било којем броју машина није значајно велика оптималан број машина на којем се препоручује примена алгоритма је 2. Иако је алгоритам убрзао повезивање пројеката треба напоменути да се оно мери у секундама и није донело неко веће убрзање процеса повезивања пројеката.

6. ЗАКЉУЧАК

На основу резултата дошли смо до закључка да се дистрибуирано повезивање програмског кода може успешно користити као алтернатива централизованом. Видели смо и то да се нису сви алгоритми показали успешним. Тако се рецимо алгоритам равномерне расподеле пројеката показао веома лоше у пракси, док се са друге стране алгоритам груписања међусобно зависних пројеката показао као знатно боља алтернатива централизованом повезивању програмског кода.

Резултати добијени на основу коришћења алгоритама дистрибуираног повезивања, показали су:

- алгоритам равномерне расподеле пројеката
Резултати су показали да алгоритам није примењив као алтернатива јер је време остварено

његовом применом знатно веће од времена централизованог повезивања пројеката.

- алгоритам груписања међусобно зависних пројеката

Алгоритам се показао веома успешно у пракси значајно је умањио време повезивања пројеката. Оптималан број чворова над којим је препоручљиво извршење алгоритма је 4.

- алгоритам обраде последње измењених пројеката

Овај алгоритам се донекле показао успешно у пракси због тога што није значајно умањио време повезивања пројеката. Оптималан број чворова на којим је препоручљиво извршење алгоритма је 2.

На крају треба напоменути да се исплативост алгоритама мери и на основу тога колики број пројеката учествује у повезивању. За софтверске пакете који садрже велики број пројеката, око четири стотине и више остварује се знатно веће убрзање повезивања у односу на софтверске пакете где број пројеката не прелази стотину и педесет, где временска разлика између централизованог и дистрибуираног повезивања није много већа у корист дистрибуираног.

7. ЛИТЕРАТУРА

- [1] George Coulouris , Jean Dollimore, Tim Kindberg, Gordon Blair, *DISTRIBUTED SYSTEMS Concepts and Design*, Addison-Wesley, ISBN: 978-0-13-214301-1
- [2] Срђан Вукмировић, *Cloud засновани SmartGrid системи*,
<http://esi.ftn.uns.ac.rs/index.php/predmeti/80-cloud>
- [3] Mercurial-selenic, *Repository*,
<https://mercurial.selenic.com/wiki/Repository>
- [4] Tortoisehg, *TortoiseHG*,
<http://tortoisehg.bitbucket.org/>
- [5] Никола Васић, Алгоритми за дистрибуирано повезивање и превођење C# пројеката, Библиотека ФТН Нови Сад, 2015
- [6] Мита Чокић, Напредни рачунарски системи са критичном мисијом у електроенергетици,
<http://esi.ftn.uns.ac.rs/index.php/predmeti/78-napredni-sistemi>

Кратка биографија:



Радосав Белоња рођен је у Краљеву 1990 године. Дипломирао је на Факултету техничких наука 2014 године. Мастер рад брани на Факултету техничких наука из области Електротехника и рачунарство у октобру 2015 године.

PRIMENA INTELIGENTNIH PID REGULATORA U SISTEMIMA ZA KLIMU, GREJANJE I HLAĐENJE**APPLICATION OF INTELIGENT PID CONTROLLERS IN HEATING, VENTILATING AND AIR CONDITIONING SYSTEMS**

Danilo Banjanin, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu prikazana je primena inteligentnih PID regulatora u sistemima za KGH. Urađena je simulacija dva tipa regulatora. Fuzzy-PID regulator koji koristi fuzzy logiku za podešavanje parametara PID regulatora i BPNN-PID regulator koji koristi veštačku neuronsku mrežu sa back-propagation algoritmom za podešavanje parametara PID regulatora.

Abstract – In this paper the application of intelligent PID controllers in HVAC systems has been presented. The simulation of two types of controllers has been undertaken - Fuzzy-PID controller based on fuzzy logic adjustment of PID controller parameters and BPNN-PID controller, which uses artificial neural network with back-propagation algorithm to adjust PID controller parameters.

Ključne reči: Sistemi za KGH, PID regulatori, fuzzy, veštačka neuronska mreža, fuzzy-PID, BPNN-PID

1. UVOD

Zahtevi za što boljom regulacijom temperature i kvaliteta vazduha u prostorijama doveli su do velike upotrebe sistema za KGH. Jedan od načina da se zadovolji zahtev za dobrom regulacijom temperature i kvaliteta vazduha je korišćenje naprednih regulacionih tehnika. U ovom radu će biti razmatrana upotreba naprednih regulacionih tehnika u sistemima za KGH.

Glavne osobine klasičnih PID regulatora su velika rasprostranjenost i jednostavnost. Cilj rada je kombinacija dobrih osobina klasičnih regulatora sa inteligentnim regulatorima zasnovanim na fuzzy logici i veštačkim neuronskim mrežama kako bi se postiglo što bolje upravljanje sistemima za KGH.

2. SISTEMI ZA KGH

Izraz „sistemi za KGH” se odnosi na opremu koja obezbeđuje grejanje, hlađenje, filtriranje i kontrolu vlažnosti vazduha u prostorijama [1].

2.1. Vrste sistema za KGH

Jednozonski KGH sistemi se koriste za regulaciju jedne ili više zona unutar zgrade ukoliko te zone imaju slične zahteve za klimatizaciju ili se rezlike u zahtevima mogu kompenzovati.

Višezonski sistemi imaju mogućnost da klimatizuju svaku zonu kao odvojenu celinu.

Postoje još dva tipa sistema za KGH, oni se dele u odnosu na način ubacivanja vazduha u prostoriju. Sistemi koji ubacuju konstantnu odnosno promenljivu količinu vazduha.

3. MODEL SISTEMA

Veoma je teško napraviti regulator sa više petlji koji u isto vreme kontrlšiše temperaturu i koncentraciju CO₂. Dva regulatora, od kojih svaki kontrolise po jedan proces, se koriste umesto jednog regulatora. Takođe, teško je napraviti regulator nezavisno od modela procesa i primeniti ga na realan proces, zato nam je neophodan model procesa. U ovom poglavlju su predstavljeni modeli procesa za temperaturu i koncentraciju CO₂.

3.1 Matematički model temperature u zoni

Na temperaturu u nekom prostoru utiče temperatura vazduha, grejanje, zapremina prostorije, gubici toplote kroz zidove i vlažnost vazduha, čiji uticaj ćemo zane-mariti. Poštujući princip očuvanja energije temperatura unutar prostorije može biti predstavljena sledećom jednačinom [1]:

$$\rho_a \cdot C_p \cdot V_i \cdot \frac{dT_i}{d\tau} = Q_{h1} - U_w \cdot A_w \cdot (T_i - T_o) \quad (1)$$

gde je $\rho_a (kg/m^3)$ gustina vazduha, $C_p (J/kg \cdot ^\circ C)$ je toplotni kapacitet vazduha, $V_i (m^3)$ je zapremina prostorije, $T_i (^\circ C)$ je temperatura prostorije, $\tau (s)$ je vreme, $Q_{h1} (W)$ je rad grejača, $T_o (^\circ C)$ je spoljašnja temperatura, $A_w (m^2)$ je površina zidova, $U_w (W/m^2 \cdot ^\circ C)$ je koeficijent ukupnog prenosa toplote kroz zid.

Nakon Laplasove transformacije i pojednostavljenja matematički model prostorije može biti predstavljen kao:

$$\frac{T_i(s)}{Q_{h1}(s)} = \frac{K_{it}}{T_{it} \cdot s + 1} \quad (2)$$

gde je $T_{it} = \frac{\rho_a \cdot V_i \cdot C_p}{U_w \cdot A_w}$ i predstavlja vremensku konstantu, $K_{it} = \frac{1}{U_w \cdot A_w}$ je pojačanje funkcije prenosa.

NAPOMENA:

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

3.2 Matematički model koncentracije CO2 u zoni

Kao što je dato u [1] koncentracija CO2 može biti predstavljena:

$$C_{ico_2}(t) = \frac{E_{co_2} + f \cdot C_{oco_2}}{\lambda \cdot V_i + f} \cdot \left(1 - e^{-\frac{t}{\tau_{ico_2}}}\right) + C_{ion} \cdot e^{-\frac{t}{\tau_{ico_2}}} \quad (3)$$

Gde je $V_i(m^3)$ zapremina prostorije, $C_{ico_2}(ppm)$ je koncentracija CO2 u prostoriji, $C_{oco_2}(ppm)$ je spoljašnja koncentracija CO2, $\tau(s)$ je vreme, $E_{co_2}(ppm/s)$ je prosečna količina CO2 koju emituje svaka osoba u prostoriji, $\lambda(s^{-1})$ je konstanta vremenskog kašnjenja CO2, $f(m^3/s)$ je zapreminski protok svežeg vazduha.

4. FUZZY-PID REGULATOR

Fuzzy-PID regulator se sastoji od dva dela: PID regulatora i regulatora zasnovanog na fuzzy logici. Fuzzy-PID regulator je samopodešavajući i napravljen je pomoću inkrementalnog fuzzy logičkog regulatora.

PID regulator je zadužen za kontrolu temperature dok se fuzzy logički regulator koristi za podešavanje parametara PID regulatora K_p , K_i i K_d da bi se postigle što bolje performanse u različitim situacijama.

Da bismo napravili fuzzy regulator neophodna nam je baza fuzzy pravila i funkcije pripadnosti. Da bismo napravili bazu fuzzy pravila potrebno je razumeti uticaj svakog od parametara PID regulatora [2].

Onda je moguće odrediti veze između izlaza iz fuzzy regulatora K_p , K_i i K_d i ulaza e i ec kako bi se napravila baza fuzzy pravila. U

Tabela 1-Tabela 3. su prikazana fuzzy pravila za K_p , K_i i K_d .

Tabela 1. Fuzzy pravila za K_p

ec e	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZO	ZO
NM	PB	PB	PM	PS	PS	ZO	NS
NS	PM	PM	PM	PS	ZO	NS	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	PS	ZO	NS	NM	NM	NM	NB
PB	ZO	ZO	NM	NM	NM	NB	NB

Tabela 2. Fuzzy pravila za K_i

ec e	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZO	ZO
NM	NB	NB	NM	NS	NS	ZO	ZO
NS	NB	NM	NS	NS	ZO	PS	PS
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PS	PM	PB
PM	ZO	ZO	PS	PS	PM	PB	PB

PB	ZO	ZO	PS	PM	PM	PB	PB
----	----	----	----	----	----	----	----

Tabela 3. Fuzzy pravila za K_d

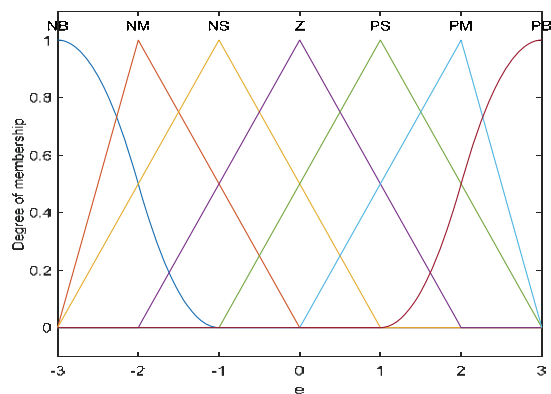
ec e	NB	NM	NS	ZO	PS	PM	PB
NB	PS	NS	NB	NB	NB	NM	PS
NM	PS	NS	NB	NM	NM	NS	ZO
NS	ZO	NS	NM	NM	NS	NS	ZO
ZO	ZO	NS	NS	NS	NS	NS	ZO
PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
PM	PB	NS	PS	PS	PS	PS	PB
PB	PB	PM	PM	PM	PS	PS	PB

Baza fuzzy pravila se sastoji od tri matrice koje prikazuju kako će se parametri K_p , K_i i K_d menjati (ΔK_p , ΔK_i i ΔK_d) u odnosu na promenu greške i njenog izvoda ec . Baza fuzzy pravila se dobija korišćenjem više ako-onda pravila kao što je prikazano u

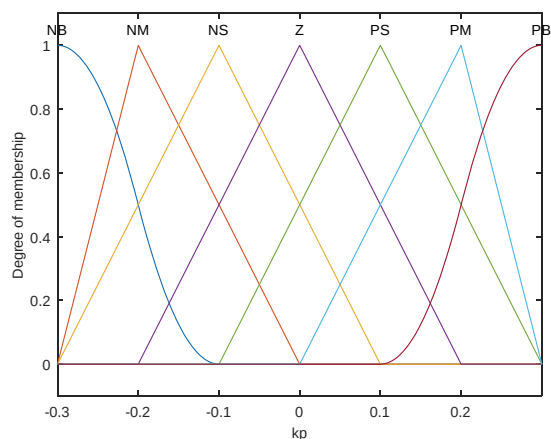
Tabela 1-Tabela 3.

Fuzzy promenljive za bazu pravila su definisane na sledeći način: NB (negativno veliko), NM (negativno srednje), NS (negativno malo), ZO (nula), PS (pozitivno malo), PM (pozitivno srednje), PB (pozitivno veliko).

Nakon što smo definisali bazu fuzzy pravila neophodno je odrediti funkcije pripadnosti za e , ec , ΔK_p , ΔK_i i ΔK_d . Funkcija pripadnosti određuje kako se svaka ulazna vrednost mapira na vrednost pripadnosti (stepen pripadnosti) i može imati vrednost u opsegu od 0 do 1. Na Sliku 1 i Sliku 2 su prikazane funkcije pripadnosti za e i kp .



Slika 1. Funkcija pripadnosti greške



Slika 2. Funkcija pripadnosti K_p

5. BPNN-PID REGULATOR

Struktura BPNN-PID regulatora sastoji se iz klasičnog PID regulatora i veštačke neuronske mreže zasnovane na back-propagation algoritmu koja podešava parametre PID regulatora. Algoritam klasičnog PID regulatora može biti predstavljen jednačinom:

$$u(k) = u(k-1) + K_p\{e(k) - e(k-1)\} + K_i e(k) + K_d\{e(k) - 2e(k-1) + e(k-2)\} \quad (4)$$

Ukoliko neuronska mreža sadrži dovoljan broj neurona, ona može da aproksimira bilo koju neprekidnu funkciju uz pomoć samo jednog skrivenog sloja [3]. Iz tog razloga se koristi neuronska mreža sa samo jednim skrivenim slojem. Mreža ima četiri ulaza, tri izlaza i tri sloja: ulazni, jedan skriveni i izlazni sloj.

Algoritam BPNN regulatora za podešavanje parametara PID regulatora može biti opisan sledećim koracima:

- 1) Odrediti početne vrednosti težina ulaznog $w_{ij}^{(2)}(k)$ i izlaznog $w_{li}^{(3)}(k)$ sloja kao i brzinu učenja η i uticaj momenta α za $k = 1$
- 2) Na osnovu zadate vrednosti $r(k)$ i izlaza sistema $y(k)$ odrediti grešku sistema $e_y = r(k) - y(k)$
- 3) Izračunati ulaz i izlaz svih neurona i parametre PID regulatora K_p , K_i i K_d
- 4) Izračunati vrednost upravljanja PID regulatora iz jednačine (4)
- 5) Podesiti težinu za svaki neuron uz pomoć back-propagation algoritma
- 6) Povećati $k = k + 1$ i vratiti se na korak 2)

6. REZULTATI I SIMULACIJA

6.1 Fuzzy-PID regulator

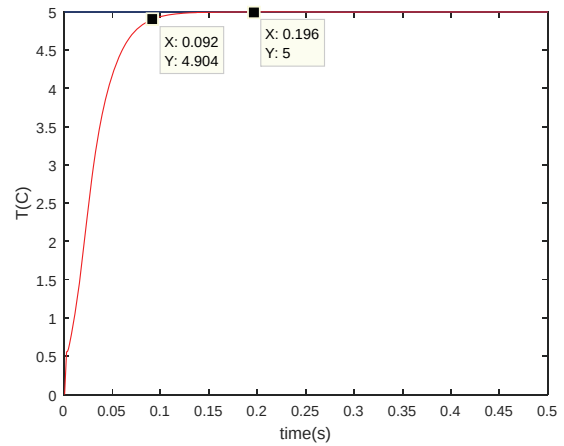
Fuzzy-PID regulator će biti korišćen za regulaciju temperature u prostoriji. Model sistema može biti predstavljen jednačinom:

$$\frac{T_i(s)}{Q_{h1}(s)} = \frac{0.028}{506.52 \cdot s + 1} \quad (5)$$

Pretpostavimo da je temperatura u prostoriji niska i da je potrebno zagrejati prostoriju da bi se komforno boravilo u njoj. Postavljamo željenu temperaturu i regulator počinje da radi da bi dostigao tu željenu temperaturu. Step signal može biti iskorišćen da simulira ovaj proces.

Pretpostavimo da je razlika između trenutne i željene temperature u prostoriji 5°C. Prema tome kao referentnu vrednost sistema dovodimo step signal ($r(k) = 5$) u vremenskom trenutku $t = 0$, izlaz sistema je dat na Slika 3. Vreme smirenja je $t_s = 0.092s$ za toleranciju od 1%, sistem brzo dostiže željenu vrednost.

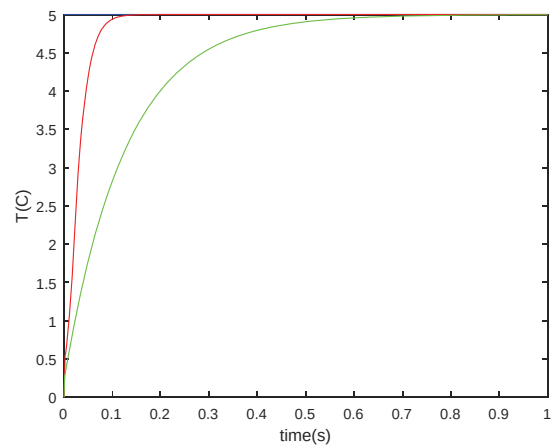
Pored brzog odziva ne dolazi do preskoka a greška u ustaljenom stanju je 0. Što znači da regulator ima dobre karakteristike brz odziv, ne dolazi do prekoka i ima dobru stabilnost.



Slika 3. Odziv sistema regulisanog fuzzy-PID regulatorom

Na Slika 4 je uporedno prikazan odziv sistema regulisanog klasičnim PID regulatorom i fuzzy-PID regulatorom.

Na početku oba regulatora imaju iste parametre ali se parametri fuzzy-PID regulatora menjaju. Usled promene parametara fuzzy-PID je uspeo da postigne bolje rezultate od klasičnog PID regulatora. Sa slike se vidi da mnogo brže dostiže željenu vrednost i pri tome ne dolazi do prebačaja.



Slika 4. Odziv sistema regulisanog fuzzy-PID (crveno) i klasičnim PID (zeleno) regulatorom

6.2 BPNN-PID regulator

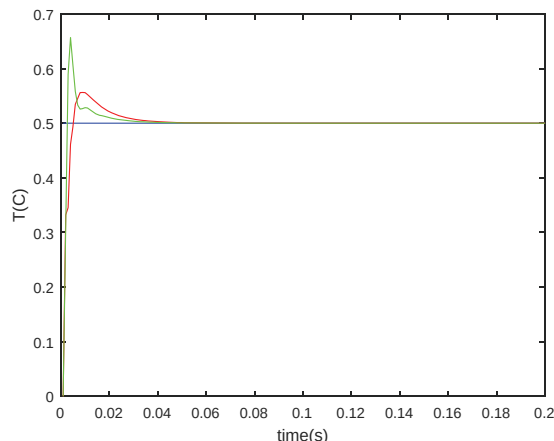
BPNN-PID regulator koristimo za regulaciju koncentracije ugljen dioksida (CO₂). Model sistema može biti predstavljen jednačinom:

$$y(k) = \frac{a(k) \cdot y(k-1)}{1 + y^2(k-1)} + u(k-1) + 0.2u(k-2) \quad (6)$$

gde $a(k)$ predstavlja parametar koji se menja tokom vremena da bi se uvela nepredvidiva promena koncentracije CO₂, kao što je otvaranje vrata, ulazak ili izlazak ljudi iz prostorije.

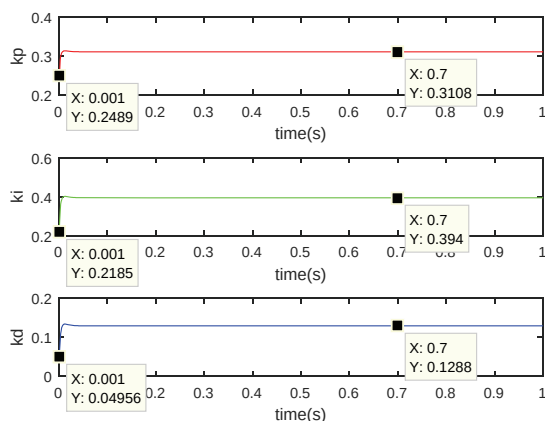
Step signal $r(k) = 1$ je doveden na ulaz sistema u trenutku $t = 0$. Odziv sistema je prikazan na Slika 5. Odziv sistema regulisanog klasičnim PID (zeleno) i BPNN-PID (crveno) regulatorom. Kao što se vidi na slici sistem jako brzo dostiže željenu vrednost i ima relativno mali preskok. Vreme smirenja je $t_s = 0.007s$ a

maksimalni preskok $\sigma = 4\%$. Na slici se takodje vidi da slučajno izabrane težine nisu uzrokovale nestabilnost sistema na samom početku regulacije. Nakon relativno malog preskoka od 4% sistem je ušao u ustaljeno stanje. U poređenju sa PID regulatorom se vidi da su oba regulatora uspjela da postignu željenu vrednost ali je BPNN-PID regulator imao manji preskok.



Slika 5. Odziv sistema regulisanog klasičnim PID(zeleno) i BPNN-PID(crveno) regulatorom

Podešavanje parametara PID regulatora K_p , K_i i K_d je prikazano na Slika 6. *Podešavanje parametara PID regulatora.* Na početku procesa regulacije parametri imaju vrednost $K_p = 0.25$, $K_i = 0.22$ i $K_d = 0.04$. Kada počne simulacija veštačka neuronska mreža podešava parametre PID regulatora kako bi sistem dostigao željenu vrednost, na Slika 5. Odziv sistema regulisanog klasičnim PID(zeleno) i BPNN-PID(crveno) regulatorom se vidi da je sistem uspeo da dostigne željeno stanje. Kada sistem dođe u ustaljeno stanje vrednosti parametara su $K_p = 0.31$, $K_i = 0.39$ i $K_d = 0.12$.



Slika 6. Podešavanje parametara PID regulatora

7. ZAKLJUČAK

U radu su predstavljeni inteligentni PID regulatori zasnovani na fuzzy logici i veštačkim neuronskim mrežama. Glavni cilj je bio da se isprave nedostaci klasičnog PID regulatora, jedan od nedostataka je da regulator mora biti napravljen u odnosu na model zgrade i veoma je podložan greškama ukoliko model nije dobar.

Podešavanje parametara za novi model oduzima dosta vremena. Prilikom simulacije je prikazano da predloženi regulatori veoma dobro kompenzuju greške prilikom modeliranja sistema i imaju dobru otpornost na poremećaje, takode se veoma dobro prilagođavaju novim uslovima što je prikazano u sistemu koji je vremenski promenljiv.

8. LITERATURA

- [1] Y. Song, Intelligent PID controller based on fuzzy logic control and neural network technology for indoor environment quality improvement, University of Nottingham, 2014.
- [2] J. Zhong, PID Controller Tuning: A Short Tutorial, 2006.
- [3] J. Villiers / E. Barnard, Backpropagation neural nets with one and two hidden layers, IEEE Trans. Neural Networks, vol. 4, 1993.
- [4] M. Rolloos, HVAC Systems and Indoor Air Quality, Indoor and Built Environment, 1993.
- [5] D. L. Loveday / G. S. Virk, Artificial intelligence for buildings, Applied Energy, 1992.

Kratka biografija:



Danilo Banjanin rođen je u Zrenjaninu 1986. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Automatika i upravljanje sistemima odbranio je 2015.godine.

INTEGRISANA UPOTREBA WII I KINECT KONTROLERA U FIZIKALNOJ TERAPIJI

INTEGRATED USE OF THE WII AND KINECT CONTROLLERS IN PHYSICAL REHABILITATION

Vladimir Benčić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je opisana implementacija aplikacije za rehabilitaciju u fizikalnoj terapiji, koja koristi Kinect verzije 2 i Wii kontroler. Radi postizanja što boljih rezultata pri merenju u realizaciji su uporedo korištena oba kontrolera.

Abstract – *The paper gives detailed information about building an application for physical therapy using Kinect v2 and Wii Remote. In order to achieve the best possible results during the measurement process, both controllers were used simultaneously.*

Ključne reči: *Kinect, Wii, praćenje pokreta korisnika, Kinect skeletni sistem, rehabilitacija, fizikalna terapija*

1. UVOD

Tehnologije osetljive na pokret (eng. *motion sensitivity technologies*) su tehnologije koje imaju sposobnost da prate pokrete. Prva primena ove tehnologije bila je prilikom izrade radara tokom Drugog svetskog rata. Ovo otkriće je imalo značajan uticaj na konačan ishod rata. Nakon toga, tehnologije osetljive na pokret su korišćene u izradi sigurnosnih sistema za praćenje pokreta u zatvorenim prostorima, a danas se nalaze svuda: na automatskim ulazno/izlaznim vratima marketa, automatskim česnama i sušaćima za ruke u toaletima, pametnim mobilnim telefonima, automobilima, igračkim konzolama i mnogim drugim mestima. Upotreba tehnologija osetljivih na pokret postala je svakodnevnica modernog čoveka, i iz dana u dan primena ove tehnologije je sve zastupljenija u svim granama privrede.

Tehnologije osetljive na pokret mogu se sresti i u medicini. Naime, postoji dosta rešenja koja koriste tehnologije osetljive na pokret kako bi se unapredio kvalitet i efikasnost lečenja. Postoje senzori koji su namenski razvijani za potrebe medicine: senzori koji upozoravaju osoblje ukoliko se pacijent pomera kako bi se sprečio mogući pad pacijenta sa kreveta, senzori koje pacijenti nose ugrađene u specijalne pojaseve kako bi se prikupljali podaci o njihovim aktivnostima, itd. Takođe, postoje i primeri gde se već postojeći komercijalni sistemi namenjeni drugim oblastima, pre svega igranju, primenjuju u medicini. Na tržištu postoji mnoštvo softverskih rešenja koja koriste kontrolere igračkih konzola koji poseduju *motion sensitive* tehnologiju. Ta softverska rešenja isključivo služe kao motivišuća asistencija pri fizikalnoj rehabilitaciji, a ne kao zamena za klasičnu rehabilitaciju.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivetić, red.prof.

Neka od tih rešenja su: *Ortho[m]easure, Kinetisense, Jintronix, SeeMe* i *WiiHabilitation*. Svaka od ovih implementacija na svoj način fizijatrima pruža dosta mogućnosti po pitanju praćenja stanja pacijenata i njihovog napretka, a pacijentima jedno novo, zanimljivije iskustvo tokom procesa rehabilitacije.

Većina softverskih rešenja koja se koriste u fizikalnoj terapiji, koristi samo jedan tip igračkih kontrolera osetljivih na pokret. Cilj istraživanja ovog rada je da se u implementaciji rešenja istovremeno koriste barem dva tipa kontrolera kako bi se došlo do boljih rezultata. Takođe, jedan od ciljeva istraživanja je i da se napravi ekonomski prihvatljivo rešenje. Ovo se pokušava postići tako što su za realizaciju ove aplikacije korišćeni *Kinect* i *Wii* kontroleri, kontroleri koji su bili korišćeni uz prošlu generaciju igračkih konzola, i koji već zastarevaju, a samim tim i njihova vrednost opada [1]. Dobar deo starijih konzola zajedno sa njihovim kontrolerima završi napušteno na otpadu kada na tržište izađu nove. Implementacijom rešenja iz istraživanja bi se dodatno produžio životni vek ovih kontrolera.

2. FIZIKALNA REHABILITACIJA ZASNOVANA NA IGRI

Na tržištu postoji mnoštvo aplikacija koje se koriste u fizikalnoj terapiji, a koje u svojoj implementaciji koriste kontrolere igračkih konzola. Ova rešenja su implementirana na različite načine, i svako od rešenja ima neka svojstva koja su mu karakteristična i čime sebe ističe u odnosu na druge. Međutim, većina rešenja u svojoj implementaciji koristi samo jedan tip kontrolera. U implementaciji rešenja koje je opisano u ovom radu korišćena su dva kontrolera različitih proizvođača, kako bi se postigli što bolji rezultati merenja i poboljšalo korisničko iskustvo prilikom korišćenja aplikacije.

3. IMPLEMENTACIJA INTEGRALNOG REŠENJA

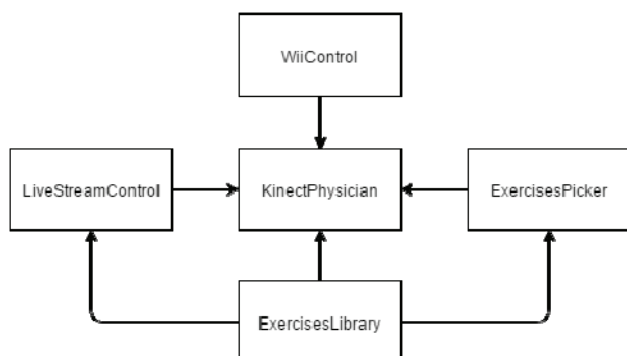
Aplikacija koja je razvijana za potrebe ovog istraživanja je zamišljena da bude primer koncepta za sinhronu upotrebu više *motion sensitive* kontrolera pri oporavku pacijenata. U ovoj aplikaciji implementirane su samo vežbe za rehabilitaciju ramena. U implementaciji rešenja koristi se *Kinect* kontroler kako bi se pratili pokreti prilikom izvođenja vežbe, dok se *Wii* kontroler koristi kako bi se pratio tempo izvođenja iste.

U okviru aplikacije postoji podrška za vođenje evidencije o pacijentima. Ova funkcionalnost bi omogućila medicinskim ustanovama da imaju zabeležene podatke o pacijentima koji su koristili ovaj vid oporavka i obezbedila im da brzo i lako dođu do podataka o traženom pacijentu. Takođe, korisnički interfejs aplikacije

je zamišljen tako da osoblju maksimalno olakša korišćenje i smanji potrebno vreme za obuku. Kako bi se to postiglo, razvijeno je nekoliko korisničkih kontrola: kontrola za odabir vežbi, kontrola za prikaz videa sa *Kinect*-ove kamere sa označenim delovima tela koji se koriste tokom izvođenja vežbe i kontrola za prikaz stanja *Wii* kontrolera na kojoj se prikazuje tempo izvođenja vežbe.

Glavni fokus u aplikaciji je na samom procesu vežbanja (sesiji), koji je implementiran korišćenjem *Kinect* v2 i *Wii* kontrolera. Nakon završene sesije pojavljuje se grafik sa postignutim rezultatima tokom sesije.

Radi lakšeg održavanja i razvoja, rešenje je podeljeno u 5 projekata (slika 1).



Slika 1: Organizacija rešenja

Svaki projekat ima jedinstveni programski identifikator maksimalne deskriptivnosti funkcije:

- *ExercisesLibrary* – projekat u okviru koga se nalaze klase koje definišu vežbe,
- *LiveStream* – projekat u okviru koga je implementirana kontrola za prikazivanje video snimka sa *Kinect*-ove kamere, i koja iscrta delove *Kinect* skeleta koji predstavljaju delove tela koje treba koristiti za izvođenje izabrane vežbe,
- *ExercisePicker* – projekat u okviru koga je implementirana kontrola za odabir mesta traume pacijenta prilikom odabira vežbe,
- *WiiControl* – projekat u kome je implementirana kontrola koja prikazuje stanje *Wii* kontrolera,
- *KinectPhysician* – glavni projekat aplikacije.

4. IMPLEMENTACIJA PRAĆENJA PRILIKOM IZVOĐENJA VEŽBI

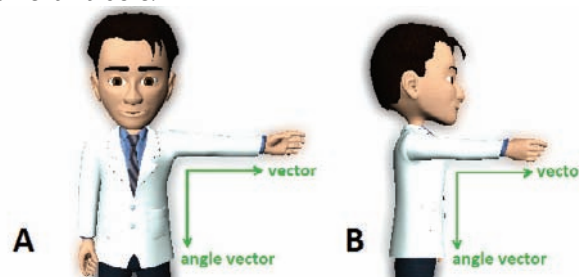
U aplikaciji realizovanoj u ovom radu praćenje pokreta tela korisnika implementirano je korišćenjem *Kinect*-ovog skeleta. *Kinect* skelet pruža podatke o poziciji zglobova korisnika u prostoru.

U aplikaciji su implementirane sledeće vežbe:

- odručenje (eng. *abduction*)
- predručenje (eng. *flexion*)

Na slici 2A može se videti avatar koji izvodi vežbu odručenja. Postignuti ugao prilikom izvođenja ove vežbe se dobija izračunavanjem ugla između vektora *vector* i *angleVector*, gde *vector* predstavlja vektor od ramena ka

laktu korisnika, a *angleVector* predstavlja vektor od ramena na dole.



Slika 2: Prikaz vektora koji učestvuju u izračunavanju postignutog ugla prilikom izvođenja vežbe

Na slici 2B može se videti avatar koji izvodi vežbu predručenja. Prilikom izvođenja ove vežbe korisnik mora da bude bočno okrenut prema *Kinect* kontroleru kako pri izvođenju vežbe ne bi šakom zaklanjao rame i lakat i samim tim umanjivao preciznost merenja. Ovom promenom orijentacije, pored toga što je *Kinect*-u omogućeno da bolje „vidi“ delove tela potrebne za izvođenje vežbe, postignuto je i to da se za izračunavanje postignutog ugla može koristiti algoritam korišćen u implementaciji praćenja za vežbu odručenja.

4. OPIS APLIKACIJE

Nakon startovanja aplikacije pojavljuje se glavni ekran na kome se može videti lista pacijenata koji koriste aplikaciju (slika 3). Listu pacijenata moguće je sortirati klikom na zaglavlje kolone i pretraživati unosom teksta u polje iznad kolone. Sa ovog ekrana moguće je otići na ekran za dodavanje novih pacijenata, izmeniti podatke o selektovanom pacijentu, obrisati pacijeta iz liste, ili otići na ekran za odabir vežbi.



Slika 3: Glavni ekran aplikacije

Ukoliko korisnik klikne na dugme *Add Patient*, otvara se ekran za dodavanje pacijenata (slika 4). Nakon što se popune polja na ovom ekranu i klikne na dugme *Add*, pacijent se dodaje u listu pacijenata i ponovo se prikazuje glavni ekran.

Ukoliko korisnik u bilo kom trenutku tokom korišćenja aplikacije želi da se vrati na prethodni ekran, to može da uradi klikom na dugme *Back* koje je predstavljeno u vidu strelice koja se nalazi u gornjem levom uglu ekrana.

Kinect Physician

Email:

First Name:

Last Name:

Address:

Date of birth:

Add Cancel

Slika 4: Ekran za dodavanje pacijenata

Nakon što se odabere pacijent iz liste pacijenata, funkcije za izmenu podataka o pacijentu, brisanje pacijenta iz liste i pokretanje sesije postaju omogućene. Ukoliko korisnik klikne na dugme *Edit Patient*, otvara se ekran za izmenu podataka o pacijentu (slika 5).

Kinect Physician

Email: john.doe@yahoo.com

First Name: Joh

Last Name: Doe

Address: Cirpanova 25/3, 21000 Novi Sad

Date of birth: 11/3/1991

Save Cancel

Slika 5: Ekran za izmenu podataka o pacijentu

Posle unosa izmena, potrebno je kliknuti na dugme *Save* kako bi se unete izmene sačuvale. Nakon klika na dugme *Save* ponovo se pojavljuje glavni ekran aplikacije. Da bi se pacijent uklonio iz evidencije, potrebno je odabrati željenog pacijenta iz liste i zatim kliknuti na dugme *Delete Patient*, nakon toga se pojavljuje poruka u kojoj se traži potvrda za brisanje izabranog pacijenta.

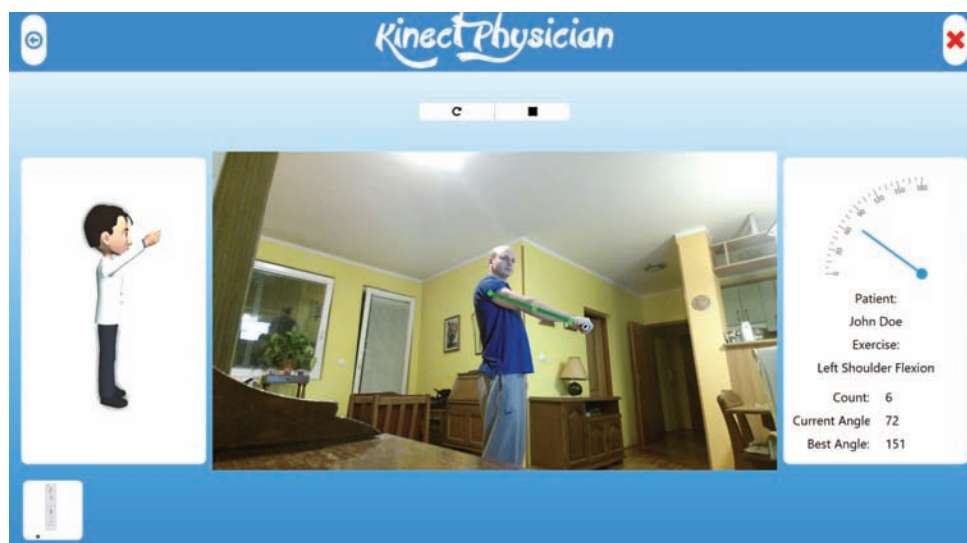
Ukoliko korisnik potvrdi brisanje, pacijent se briše iz liste pacijenata.

Ukoliko je potrebno pokrenuti sesiju za određenog pacijenta, potrebno je odabrati pacijenta iz liste pacijenata sa glavnog ekrana i kliknuti na dugme *Go to Session*. Nakon toga, pojavljuje se ekran za odabir vežbi (slika 6). Na ovom ekranu korisnik treba da odabere vežbu koju je potrebno izvoditi tokom sesije. Vežba se bira tako što korisnik kursomom miša odabere regiju traume klikom na deo tela avatara, i zatim odabere jednu od ponuđenih vežbi iz liste koja se pojavi na desnoj strani ekrana. Odabir vežbe se potvrđuje klikom na dugme *Start Session*.



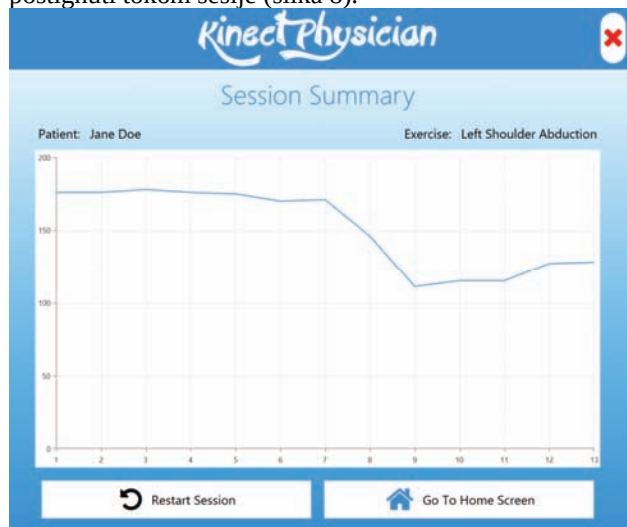
Slika 6: Ekran za izmenu podataka o pacijentu

Nakon potvrde odabira vežbe prikazuje se ekran sesije (slika 7). Ovaj ekran se koristi tokom izvođenja vežbe. Na levoj strani ekrana prikazan je avatar koji demonstrira izvođenje vežbe kako bi korisniku bilo jasno kako da se pozicionira u odnosu na Kinect kontroler i kako pravilno da izvodi pokret. U sredini ekrana korisnik može da vidi snimak sa kamere Kinect-a na kojem su zelenom linijom označeni delovi tela korisnika koji se u izvođenju vežbe. Ukoliko korisnik nije dobro pozicioniran, ili ne izvodi vežbu pravilno, prikazaće se odgovarajuća poruka koja će ga upozoriti na grešku.



Slika 7: Ekran sesije

Na desnoj strani ekrana nalazi se tabela u kojoj su prikazani detalji sesije. Na njoj korisnik može da vidi podatke o maksimalnom postignutom uglu tokom sesije, trenutnom postignutom uglu, broju ponavljanja, itd. U donjem levom uglu ekrana nalazi se kontrola koja prikazuje stanje *Wii* kontrolera. Tempo izvođenja vežbe se prati uz pomoć *Wii*-a. Indikacija o tempu izvođenja je implementirana preko led dioda na samom kontroleru. Plavi kvadrati ispod slike *Wii*-a prikazuju trenutno stanje led dioda sa kontrolera. Ukoliko se prekorači dozvoljena brzina izvođenja vežbe, korisnik se upozorava vibracijom *Wii*-a, i kontrola koja prikazuje stanje kontrolera dobija crveni okvir. Ukoliko se *Wii* kontroler ne koristi pri izvođenju vežbe, kontrola koja prikazuje njegovo stanje neće biti prikazana. Nakon što korisnik završi sesiju, prikazuje se ekran koji na kome se mogu videti rezultati postignuti tokom sesije (slika 8).



Slika 8: Ekran za prikaz pregleda rezultata sesije

Sa ovog ekrana moguće je ponoviti sesiju klikom na dugme *Restart Session* ili otići na glavni ekran aplikacije klikom na dugme *Go To Home Screen*.

5. ZAKLJUČAK

U ovom radu je implementirana aplikacija koja predstavlja primer koncepta razvoja aplikacija za rehabilitaciju u kojima se istovremeno koristi više kontrolera sa senzorima osetljivim na pokret. U razvoju aplikacije korišćeni su *Kinect* kontroler verzije 2 i *Wii* kontroler. Korišćena je *Kinect*-ova mogućnost praćenja skeleta za praćenje pokreta pacijenta, dok je *Wii*-ov akcelerometar korišćen kako bi se pratio tempo izvođenja vežbi. Takođe, korišćena je i *Wii*-ova vibracija kako bi upozorila pacijenta ukoliko prebrzo izvodi vežbu. Integrisana upotreba ova dva kontrolera se pokazala dobro, jer su iskorišćene jače strane svakog od kontrolera kako bi se došlo do što tačnijih rezultata, a ujedno se i poboljšalo korisničko iskustvo, jer pored uobičajenih vizuelnih povratnih informacija sa ekrana, korisnik dobija i fizičku povratnu vrednost putem vibracije *Wii*-ja.

Prilikom izrade rada bilo je mnogo izazova. Najveći izazov bio je definisati vežbe za rehabilitaciju ramena i napraviti intuitivni dizajn koji je prijatan za oko korisnika i lak za korišćenje. Da bi problem odabira odgovarajućih vežbi bio rešen, kontaktiran je stručnjak iz oblasti medicine. Problem dizajna interfejsa je rešen uz

konsultaciju sa stručnjakom iz oblasti dizajna. Implementacija je bila dosta olakšana upotrebom template mehanizma *Windows Presentation Foundation*-a. Ovaj mehanizam se pokazao kao veoma fleksibilan i lak za korišćenje. Mnoštvo izazova pri izradi ovog rada koji su uspešno prevaziđeni pokazali su da je za izradu ovakvog tipa aplikacije obavezna saradnja sa stručnjacima iz drugih oblasti.

U aplikaciji ima prostora za dalji razvoj. Neke od mogućnosti su sledeće:

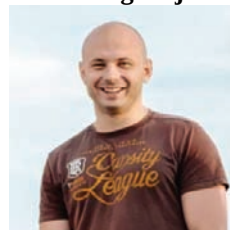
- Implementirati mogućnost potpunog upravljanja interfejsom uz pomoć gestikulacija, bez potrebe korišćenja miša i tastature, čime bi se dodatno olakšalo i ubrzalo korišćenje aplikacije.
- Kako bi aplikacija zaista mogla da se koristi u praksi, implementirati još vežbi.
- Napraviti aplikaciju koja bi bila u formi igre, gde bi pacijent kroz igru i vežbao, što bi bilo posebno prikladno za pacijente mlađeg uzrasta.
- Čuvati istorije svih sesija pacijenata u bazi podataka i implementirati ekran na kojem bi bilo moguće pregledati sumirane rezultate.
- Razviti višeslojnu aplikaciju, gde bi klijent imao deo aplikacije za vežbanje, baza podataka bi bila na nekom serveru, a lekar bi u svojoj kancelariji imao aplikaciju u kojoj bi mogao da proveri trenutno stanje pacijenta, prati njegov napredak i po potrebi menja program vežbi.

Iz ovog rada može se zaključiti da upotreba *Kinect* kontrolera u kombinaciji sa *Wii* kontrolerom ima potencijala da zaživi u izradi aplikacija za medicinsku rehabilitaciju.

4. LITERATURA

- [1] Ivetic D. & Petrovic V. Active extending the life expectancy of motion-sensing game consoles. *AWERProcedia Information Technology & Computer Science*. [Online]. 2013, 04, pp 130-135. Dostupno na: www.awercenter.org/pitcs

Kratka biografija:



Vladimir Benčić rođen je u Sremskoj Mitrovici 1989. godine. Diplomski – bečelor rad je odbranio na Fakultetu tehničkih nauka u Novom Sadu 2013. godine. Diplomski – master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Virtualna realnost odbranio je 2015. godine.

ПРИМЕНА СОЛАРНИХ СИСТЕМА ЗА ЗАГРЕВАЊЕ ТОПЛЕ ВОДЕ У ДОМАЋИНСТВУ**DOMESTIC HOT WATER HEATING WITH SOLAR SYSTEMS**

Владо Оравец, Факултет техничких наука, Нови Сад

Област - ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај: Овај рад представља један начин пројектовања соларног система за загревање топле воде у домаћинству. Извршена је анализа система на територији града Београда. Такође је одрађена и економска процена и процена исплативости овакво пројектованог система.

Кључне речи: соларни систем, соларни колектори, заштита животне средине

Abstract: This thesis presents one possible way of solar hot water heat system for domestic residence. System analysis is conducted on territory of Belgrade city. It was also determined economic assessment and evaluation of the projected profitability of such systems.

Key words: solar system, solar collector, environmental protection

1. УВОД

Соларна енергија је обновљиви извор енергије. Сунце је нама најближа звезда те, непосредно или посредно, извор готово све расположиве енергије на Земљи.



Слика 1. Сунце као извор енергије

Сунчева енергија потиче од нуклеарних реакција у његовом средишту, где температура достиже и 15 мил °С. Постоје три врсте претварања соларне енергије:

- Соларни панели - Сунчева енергија се директно претвара у топлотну енергију и сам о њима ће бити реч у овом раду.
 - Концентрисање соларне енергије - Користи огледала или поља огледала за усмеравање соларног зрачења у једну тачку, где се загрева флуид, који се затим користи за погон турбина-генератора за производњу електричне енергије.
 - Соларне фото-напонске ћелије - претварање соларне енергије директно у електричну.
- Соларна енергија је енергија која се добија зрачењем Сунца Без буке и отпадних гасова. Сврстава се у групу еколошки неутралних извора енергије.

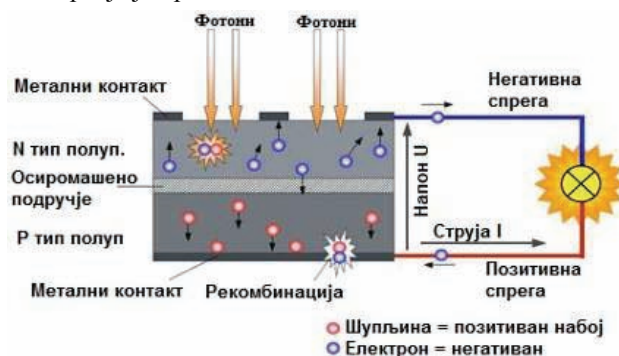
НАПОМЕНА:

Овај рад проистекао је из мастер рада, чији ментор је био др Велимир Чонградац, доцент.

2. СОЛАРНЕ ЋЕЛИЈЕ

Соларне ћелије су елементи који директно претварају енергију сунчевог зрачења у електричну енергију. Ефикасност им је од 10% за јефтиније израде са аморфним силицијумом, до 25% за скупље израде. За сада су још увек економски нерентабилни.

Фотонапонске ћелије могу да се користе као самостални извори енергије или као додатни извор енергије. Као самостални извор енергије користи се нпр. на сателитима, саобраћајним знаковима, калкулаторима и удаљеним објектима који захтевају дуготрајни извор енергије. Пример фотоелектричне конверзије је приказан на слици 2. [1].



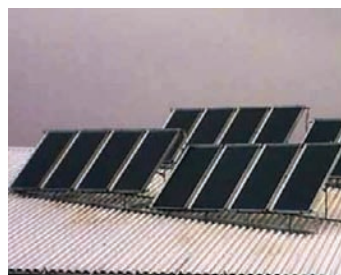
Слика 2. Фотоелектрична конверзија ПН спој

3. ПРИПРЕМА ТОПЛЕ ВОДЕ ПОМОЋУ СОЛАРНИХ СИСТЕМА

Соларни системи се генерално могу поделити на активне и пасивне. У оквиру овог рада пажњу смо посветили активним соларним системима.

Основни уређај активног соларног система је пријемник сунчеве енергије (ПСЕ) или, како се још често назива *соларни колектор* [5]. По својој конструкцији пријем-ник може бити:

- раван ПСЕ,
- цевни или
- параболични.



Слика 3. Равни пријемник сунчеве енергије

Максimalно искоришћење сунчеве енергије примајуће површине остварује се када је она нагнута под одеђеним углом β , постављена према југу.

Интензитет зрачења, које доспева до равни неке површине нагнуте за угао β , већи је у односу на интензитет зрачења које доспева до хоризонталне површине исте величине [1].

За Европу те вредности су између 30 °C и 45 °C, у односу на хоризонталу, док је препоручљив минимални угао 15 °C.

3.1. Индиректни систем са принудном циркулацијом флуида, једним резервоаром и допунским извором топлоте

У одређеном тренутку јављају се веће потребе за топлотом од оних које може да обезбеди соларни

систем. У тим случајевима се поставља додатни извор за грејање СТВ, који може бити електрични грејач или топоводни (што је случај код система који раде преко целе године). [3] Пример таквог система је дат на слици 4.

Соларни бојлер служи за загревање воде користећи размењивач топлоте. Такође служи за складиштење топле воде, која се употребљава у домаћинству.

Соларни модул се састоји од циркулационе пумпе и експанзионе посуде. Циркулациона пумпа омогућава кружења флуида кроз цеовод уради бојлера. Експанзиона посуда прихвата вишак воде која настаје усред загревања.



Слика 4. Систем са ПСЕ и једним складишником топлоте и допунским извором за грејање

Загрејана вода из колектора, флуид, под дејством пумпе, се потискује у бојлер. Ту се хлади и поново радом пумпе се враћа у колектор.

4. ПРОЈЕКТНИ УСЛОВИ И ДИНАМИКА ПОТРОШЊЕ СТВ

Табела 1. Просечна дневна потрошња по члану домаћинства

Укупна дневна потрошња за домаћинства	
Мањи захтеви	10 - 20 л/дневно особа
Средњи захтеви	20 - 40 л/дневно особа
Велики захтеви	40 - 80 л/дневно особа

Годишња потребна енергија за припрему СТВ се израчунава преко једначине, а према стандарду СРПС ЕН 15316:

$$Q_w = \rho_w * c_w * V_w * (\theta_w - \theta_o) \text{ [kWh / a]} \quad (4.1)$$

, где је:

V_w - годишња потрошња воде [m^3 / a], θ_w - температура воде у резервоару [$^{\circ}\text{C}$], θ_o - температура воде из водовода [$^{\circ}\text{C}$], $\rho_w * c_w = 1.16 \text{ [kWh / m}^3\text{K]}$,

За потрошњу енергије такође је важно осигурати мерење потрошње санитарне топле воде. Праћењем потрошње могу се утврдити одступања од уобичајених вредности или нерационално трошење, а уз додатно мерење температуре и удео топлоте за припрему санитарне топле воде у енергетском билансу зграде.

5. АНАЛИЗА СИСТЕМА ЗА ЗАГРЕВАЊЕ СТВ

Просечно годишње глобално зрачење Сунца на хоризонталну раван за нека места у Србији су приказани у табели 2 испод.

Табела 2. *Просечно годишње глобално зрачење Сунца [kWh/m²]*

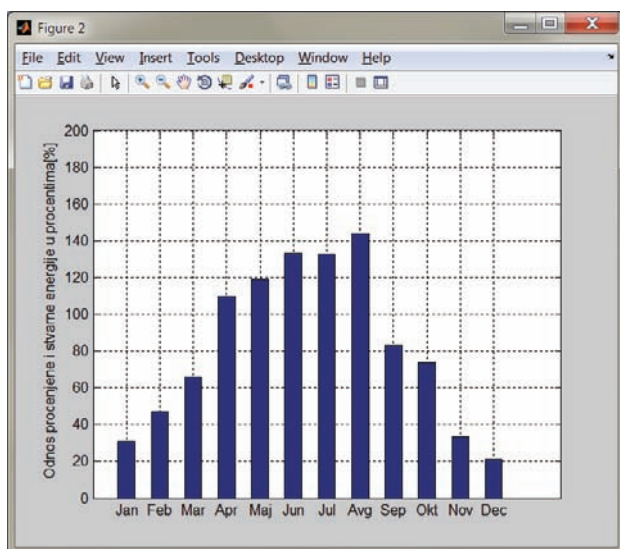
Град	Укупно годишње
Београд	1446.8
Нови Сад	1392.64
Вршац	1424.75
Зрењанин	1419.45

Приликом процене потребне енергије у обзир је узет само број чланова домаћинства и препоручена просечна потрошња воде по члану(10-50l). У прорачуну за процењену потребну енергију на месечном нивоу у обзир нису узети временски услови, захтеви за удобнији живот станара и слично, иако они потенцијално утичу на коришћење топле воде. [2]

Укупна процењена потребна енергија је 3186.1 kWh, што је за 45.4% више од стварне прикупљене енергије која износи 1446.8 kWh.

На слици се може видети да у току зимског периода стварна прикупљена енергија задовољава између 21.1% и 65.9%, док у летњем периоду прикупљена соларна енергија може да задовољи и до 143% енергетских захтева.

Овако пројектовани систем чини да уколико систем користи за допунско загревање воде електрични бојлер снаге грејача 2kW. Време рада овог бојлера се смањује у просеку са 70 сати месечно на 37 сати, а самим тим се постиже и уштеда у новцу.



Слика 5. *Процентуални приказ односа стварне и процењене енергије потребне за загревање СТВ*

Период отплате соларних система за загревање топле воде је значајан фактор за случај пословног одлучивања. Укупни трошкови за конструкцију су 1705 €, а процењени период отплате је 12.4 година.

Поред економске исплативости, овако пројектован систем има и еколошке предности. Уколико би се грејање воде у бојлеру вршило само енергијом

добијеном од лож уља, на годишњем нивоу требало би око 153.8кг, што је еквивалентно 481.58кг емисије CO₂, док се применом само мрког угља троши око 337.19кг, што је еквивалентно 603.58кг емисије CO₂.

6. ЗАКЉУЧАК

Предмет овог рада ја било пројектовање соларног система за загревање санитарне топле воде. Поред тога што је вођено рачуна о исплативости система, такође је акценат стављен и на заштиту и очување животне средине.

Комплетан систем се састоји од соларних панела, бојлера са регулатором, експанзионе посуде и термостата. Поглавље један описује Сунчеву енергију и даје увод у енергетску моћ соларне енергије. Кроз поглавље два је описано како се енергија прикупља. У трећем поглављу су наведени типови соларних колектора и приказан је један од начина како се систем може испројектовати.

Графички приказ процентуалног односа стварне и процењене вредности прикупљене енергије је приказан у поглављу пет.

Генерални закључак је да поред тога што се постиже новчана уштеда из разлога што користимо потпуно обновљиви извор енергије помоћу система који има минималне захтеве за одржавањем. Овакав систем постиже да се смањи емисија CO₂, и, његова концентрација у атмосфери, а самим тим и да се смањи акумулирање топлоте и пораст температуре планете.

У овом раду можемо да видимо да се емисија CO₂ смањила за скоро 55% у односу на системе који за загревање користе друге видове топлотне енергије.

7. ЛИТЕРАТУРА

- [1] Miroslav Lambić, Priručnik za solarno grejanje, Naučna knjiga, Beograd, 1992
- [2] Tomislav M. Pavlović, Branislav d. Čabrić, Fizika i tehnika solarne energetike, Građevinska knjiga, Beograd, 2007
- [3] Jasmina M. Radosavljević, Tomislav M. Pavlović, Miroslav R. Lambić, Solarna energetika i održivi razvoj, Građevinska knjiga, Beograd, 2004
- [4] Miroslav Lambić, Energetika, Univerzitet u Novom Sadu, Tehnički fakultet "Mihajlo Pupin". Zrenjanin, 2007.
- [5] Dr Miroslav Lambić, Mr Ivan Tasić, Dr Novica Pavlović, Dr Dragan Stojićević
- [6] <http://aliquantum.rs/energetska-efikasnost-sistema-grejanja-i-klimatizacije/>
- [7] http://www.izvorienergije.com/obnovljivi_izvori_energije.html
- [8] <http://www.energetskiportal.rs/obnovljivi-izvori-energije/energija-sunca/>
- [9] <http://www.centrometal.hr/wp-content/uploads/2011/12/CPK-7210N-Alu-jednolisni-03-2013.pdf>

[10] <http://www.azsolarcenter.org/tech-science/solar-for-consumers/solar-hot-water/solar-hot-water-a-primer.html>

Кратка биографија:



Владо Оравец рођен је у Вршцу 15.07.1987. год. Дипломски рад на Факултету Техничких Наука из области Електротехника и Рачунарство - Рачунарство и аутоматика одбранио је 2015. године.

PARTICIONISANJE MODELA PODATAKA DISTRIBUTIVNE MREŽE U CLOUD OKRUŽENJU**DATA PARTITIONING OF DISTRIBUTION NETWORK IN CLOUD ENVIRONMENT**

Mladen Korica, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je predstavljen problem podele modela podataka distributivne mreže u Cloud okruženju. Opisane su osnove Cloud računarstva i Microsoft Windows Azure platforme, kao i strategije particionisanja podataka. Nakon toga, predložena je arhitektura rešenja, izmerene performanse i analizirani su dobijeni rezultati.

Abstract – This paper presents problem of data partitioning of distribution network in Cloud environment. Basics of Cloud computing and Microsoft Azure Windows platform are described, along with data partition strategies. After that, the proposed architecture is shown along with the performance measurement.

Ključne reči: Cloud, Microsoft Windows Azure, particionisanje modela podataka, distributivna mreža, CIM

1. UVOD

Jedan od ozbiljnih problema velikih sistema, kao što je sistem za snabdevanje električnom energijom, jeste skladištenje ogromne količine podataka i njihova kasnija upotreba. Zbog upotrebe podataka distributivne mreže u različitim sistemima, došlo je do potrebe da se oni predstave na zajednički način. Najrasprostranjeniji standard jeste CIM (Common Information Model), koji predstavlja apstraktni model kojim se opisuju svi osnovni elementi elektroenergetskog sistema. Dizajn skladišta podataka može imati značajan uticaj na performanse, propusnost, i skalabilnost sistema. Ovaj rad se fokusira na podelu modela podataka distributivne mreže koji je predstavljen CIM standardima, međutim particionisanje podataka nekom od strategija se može primeniti i na druge sisteme.

Oslanjajući se na CIM standarde može izvršiti particionisanje modela podataka distributivne mreže funkcionalnom strategijom tako što će se grafički podaci elemenata distributivne mreže smestiti u jedno skladište, podaci o električnim karakteristikama elemenata u drugo skladište i ostali podaci u treće skladište. Element breaker je iskorišten za demonstraciju particionisanja.

U cilju istraživačkog segmenta ovog rada napravljen je servis u vidu radne uloge, koja je opsluživala ostale učesnike sistema kao što su aplikacije za grafički prikaz distributivne mreže, servisi koji se bave električnim proračunima, aplikacije za unos podataka distributivne mreže i sl. potrebnim atributima elemenata.

NAPOMENA:

Ovaj rad proistekao je iz -master rada čiji mentor je bio dr Darko Čapko, docent.

Skladišta podataka su realizovana korištenjem Azure Table Storage-a.

Takođe, u cilju merenja i upoređivanja performansi, realizovana je kontrolna aplikacija koja je celokupan model podataka čuvala u jednom skladištu.

2. OSNOVE CLOUD RAČUNARSTVA I MICROSOFT WINDOWS AZURE PLATFORME

Cloud računarstvo (engl. *cloud computing*) je skup nekoliko računarskih koncepata i njegov cilj je isporučivanje računarskih resursa (procesorska moć, memorija, prostor za skladištenje podataka itd.) putem Interneta. Osnova cloud računarstva su *cluster* i *grid* računarstvo, kao i virtualizacija. Arhitektura Cloud računarstva se može opisati sa pet slojeva[1]:

- Sloj aplikacije (engl. *Cloud Application Layer*) je sloj koji je najvidljiviji krajnjim korisnicima. Ono što ovaj model donosi jeste da korisnici više nisu dužni da brinu o održavanju softvera i hardvera, kao i mogućnost upotrebe učinkovitijih računarskih resursa bez potrebe za investiranjem u lokalne računarske resurse. Drugi naziv ovog modela je „softver kao servis“ (engl. *Software as a Service, SaaS*).
- Sloj softverskog okruženja (engl. *Cloud Software Environment Layer*) se naziva i platformskim slojem. Korisnici ovog sloja se bave razvojem cloud aplikacija, dok provajderi pružaju okruženje sa definisanim programskim jezikom i aplikativnim interfejsom u cilju efikasnijeg korišćenja okruženja kao i unapređenja skalabilnosti razvijane cloud aplikacije. Drugi naziv ovog modela je „platforma kao servis“ (engl. *Platform as a Service, PaaS*).
- Sloj softverske infrastrukture (engl. *Cloud Software Resources*) je zadužen za snabdevanje esencijalnih resursa višim slojevima. Servisi ovog sloja se mogu podeliti u tri kategorije: računarski resursi, prostor za skladištenje podataka i komunikacija. Ovaj model se još naziva i „infrastruktura kao servis“ (engl. *Infrastructure as a Service, IaaS*).
- Jezgro softvera (engl. *Software Kernel*) je sloj koji se bavi osnovnim softverskim upravljanjem fizičkim serverima koji su deo cloud sistema. Softversko jezgro ovog sloja može biti implementirano kao jezgro operativnog sistema, hipervizor, kontroler virtuelnih mašina i/ili srednji sloj klastera.
- Hardver i upravljački softver (engl. *Hardware and Firmware*) predstavlja najniži sloj koji čini osnovu cloud-a. Korisnici ovog sloja najčešće su velike kompanije koje imaju potrebu za iznajmljivanjem hardvera. Ovaj koncept se naziva „hardver kao servis“ (engl. *Hardware as a Service, Haas*).

Windows Azure predstavlja najbitniju komponentu Microsoft Windows Azure platforme pod čiju nadležnost spada izvršavanje aplikacija i skladištenje podataka u Cloud-u[2].

- **Compute** je servis koji omogućava izvršavanje aplikacija. Aplikacija mora biti implementirana kao uloga (engl. *Role*), koja može biti web uloga (engl. *Web Role*), radna uloga (engl. *Worker Role*) ili uloga virtualne mašine (engl. *VM Role*). Radne uloge se koriste za procese u pozadini i mogu da izvrše bilo koji kod namenjen za Windows operativni sistem. Za razliku od radnih uloga, web uloge se koriste za prednji sloj aplikacija. Uloga virtualne mašine je instanca koja ima Windows Server operativni sistem koju korisnik može sam konfigurisati u zavisnosti od potreba.
- **Storage** služi za skladištenje različitih podataka u Windows Azure skladištima. Pored SQL Azure-a koji će biti naknadno opisan, postoje još tri tipa skladišta podataka: *Blob* (engl. *Binary Large Object*), *tabele* (engl. *Tables*) i *redove* (engl. *Queues*). *Blob* je tip skladišta koji se koristi za čuvanje binarnih podataka. *Blob*-ovi su organizovani u kontejnere (engl. *Containers*) i mogu sadržati metapodatke. Tabele podsećaju na SQL bazu podataka i u njima su podaci organizovani u tabele, a zatim na particije i redove. Redovi, za razliku od predhodna dva tipa skladišta, za zadatak imaju da omoguće asinhronu komunikaciju između web uloga.
- **Fabric Controller** je komponenta koja kreira virtualne mašine i upravlja njima ukoliko korisnik napravi neku od uloga. Takođe, ona je zadužena da nadgleda sve podignute instance, ukoliko neka od instanci uloga iz bilo kog razloga prestane sa radom, ova komponenta će napraviti novu instancu uloge kako bi održala zadati broj instanci.
- **CDN – Content Delivery Network** nudi mehanizam kojim se omogućuje keširanje podataka iz skladište kojima aplikacije često pristupaju u toku svog rada.
- **Connect** predstavlja skup protokola preko kojih aplikacije u Cloud-u mogu da komuniciraju sa spoljnim svetom.

3. PODELA MODELA PODATAKA U CLOUD OKRUŽENJU

Većina Cloud aplikacija i servisa rad zasniva na obradi podataka. Dizajn skladišta podataka može imati značajan uticaj na performanse, propusnost, i skalabilnost sistema. Jedan od pristupa predlaže da se izvrši particionisanje podataka u cilju lakše manipulacije podacima. Podela podataka u particije kao rezultat može imati:

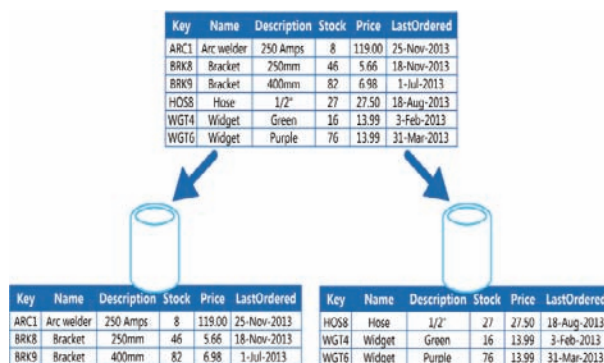
- **Poboljšanje skalabilnosti.** Podelom podataka na više particija, gde se particije nalaze na zasebnim serverima, postoji mogućnost neograničenog skaliranja sistema.
- **Poboljšanje performansi.** Ako se uzme u obzir da performanse aplikacije zavise od operacija upisa i čitanja, efikasnost se može poboljšati ukoliko su podaci pogodno particionisani. Takođe, operacije koje se izvršavaju nad više particija je moguće paralelizovati.
- **Poboljšanje dostupnosti.** Ukoliko jedan od servera iz bilo kog razloga prestane sa radom, odnosno

postane nedostupan, nedostupni su samo podaci na njemu. Raspoređivanjem podataka na više servera omogućava se da se operacije nad ostalim podacima nesmetano odvijaju. Replikacijom podataka se dodatno smanjuje nedostupnost podataka.

- **Poboljšanje bezbednosti.** U zavisnosti od prirode podataka i načina na koji su podeljeni, moguće je razdvojiti osetljive i neosetljive podatke u različite particije koje se čuvaju na različitim serverima.
- **Pružanje operativne fleksibilnosti.** Particionisanje nudi razne opcije podešavanja operacija, maksimalnu administrativnu efikasnost i minimalnu cenu.

U zavisnosti od zahteva aplikacija i servisa koji koriste podatke bira se strategija particionisanja. Postoje tri strategije particionisanja podataka: horizontalna, vertikalna i funkcionalna [3]. U zavisnosti od potreba dizajna možemo kombinovati ove tri strategije.

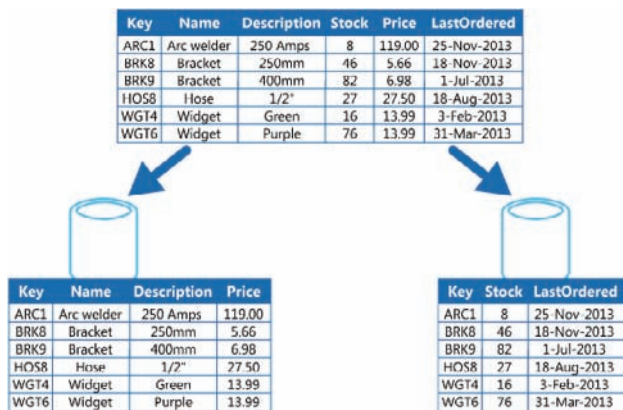
Kod horizontalne strategije (engl. *Sharding*) svaka particija sadrži podskup podataka, pri čemu sve particije imaju istu šemu. Horizontalno particionisanje omogućava raspoređivanje opterećenja na više računara čime se smanjuje uporedno pristupanje istim resursima što dovodi do poboljšanja performansi. Najvažnija stvar pri implementaciji jeste odabir ključa horizontalne strategije koji treba da obezbedi da podaci budu particionisani tako da bude što ravnomernije opterećenje na svakoj od particija. Npr. korišćenjem rezultata heš funkcije nad identifikatorom korisnika umesto prvog slova imena korisnika za ključ horizontalne strategije se izbegava nebalansirana distribucija podataka po particijama. Na narednoj slici je prikazan primer horizontalnog particionisanja:



Slika 1. Horizontalno particionisanje

Kod vertikalne strategije svaka particija sadrži podskup atributa entiteta koji se nalaze u skladištu podataka. Atributi su podeljeni u skladu sa načinom korišćenja, npr. atributi entiteta kojima se često pristupa se smeštaju u jednu particiju, dok se u drugu smeštaju atributi entiteta kojima se ređe pristupa.

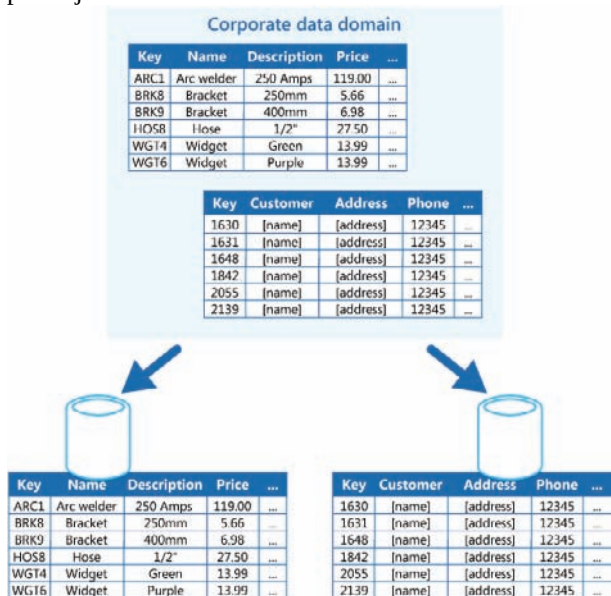
Na slici 2. prikazan je primer vertikalnog particionisanja, gde su ime, opis i cena proizvoda smešteni u jednu particiju zato što aplikacija šalje upite kojim dobavlja pomenute attribute proizvoda zajedno kad god korisnik zahteva detalje o nekom proizvodu.



Slika 2. Vertikalno particionisanje

U drugoj particiji čuvaju se informacije o stanju zaliha, kao i datum poslednje porudžbine proizvoda jer se ova dva atributa često koriste zajedno. Takođe, ovakvim particionisanjem atributi entiteta koji se ne menjaju ili se jako sporo menjaju su odvojeni od dinamičnih atributa entiteta.

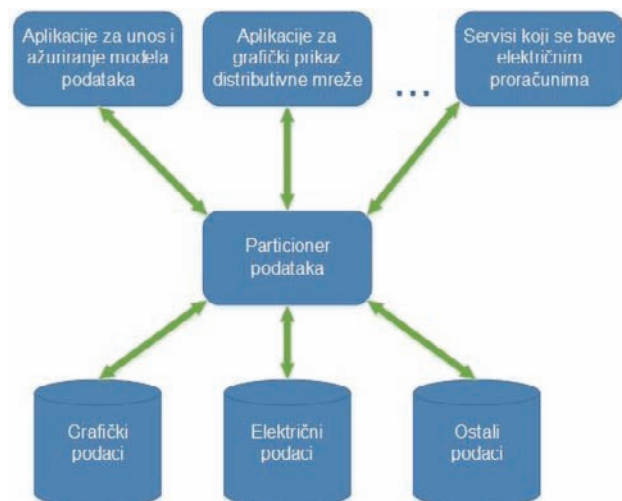
Funkcionalna strategija se koristi u slučaju kada je moguće identifikovati funkcionalno nezavisne celine podataka u sistemu, odnosno ova strategija pruža mogućnost izolacije podataka, kao i bolje performanse pri pristupu podacima. Na slici 3. prikazan je komercijalni sistem u kom se podaci o korisnicima čuvaju u jednoj particiji, dok se podaci o stanju inventara čuvaju u drugoj particiji.



Slika 3. Funkcionalno particionisanje

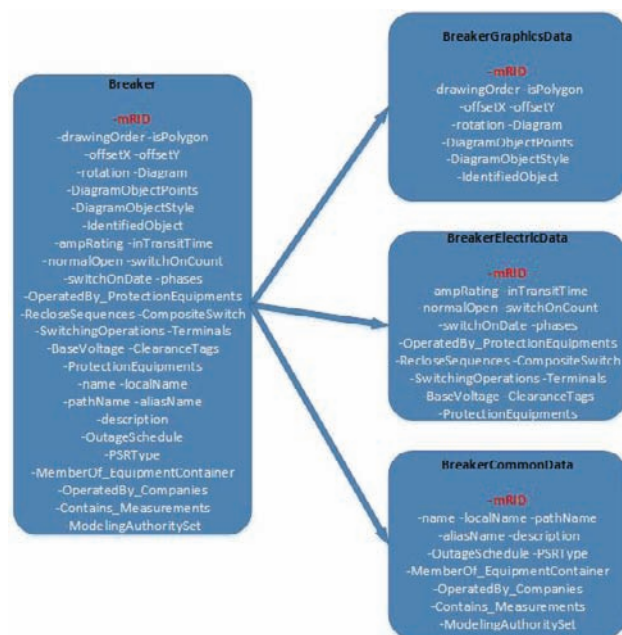
4. PODELA MODELA PODATAKA DISTRIBUTIVNE MREŽE

Na slici 4 je prikazana arhitektura sistema u kom bi postojala potreba za particionisanje modela podataka distributivne mreže. Sistem se sastoji iz komponenti koje su realizovane kao radne uloge (particioner, servisi koji se bave električnim proračunima, ...) i komponenti koje su realizovane kao web uloge (aplikacije za unos i ažuriranje modela podataka, aplikacije za grafički prikaz distributivne mreže, ...) ili aplikacije koje se nalaze van Cloud okruženja i povezane su sa ostalim komponentama putem Interneta.



Slika 4. Arhitektura sistema

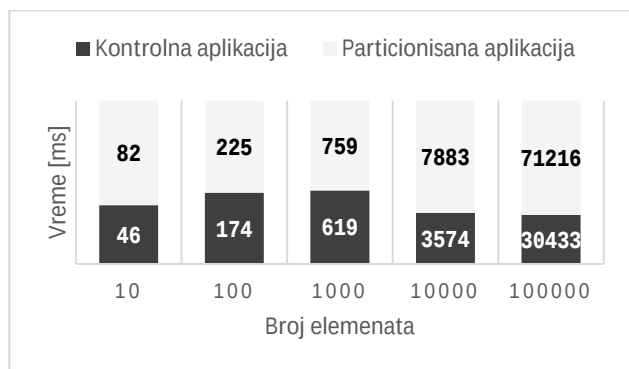
Zadatak particionera podataka je da pri unosu novih elemenata u sistem izvrši razdvajanje atributa u skladu sa njihovom namenom. Takođe, particioner mora da zna u kojem skladištu su smešteni određeni atributi elemenata ukoliko neka od komponenta sistema potražuje iste zadržavajući svojih funkcija. Na narednoj slici su prikazani atributi breakera po CIM standardima[4][5] i prikazan način na koji je on particionisan, odnosno koji atributi breakera su okarakterisani kao grafički, električni i ostali.



Slika 5. Particionisanje breakera

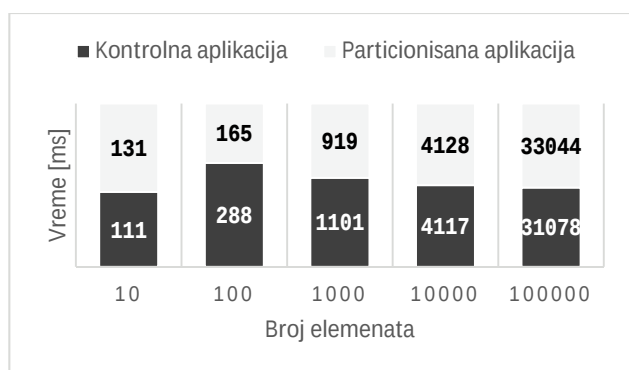
5. EKSPERIMENTALNI REZULTATI

Zarad merenja i upoređivanja performansi realizovana je kontrolna aplikacija koja sve attribute elemenata distributivne mreže smešta jedno skladište, odnosno nije izvršena podela modela podataka. Na grafiku 1 je prikazano trajanje upisa određenog broja elemenata u kontrolnoj i particionisanoj aplikaciji. Sa grafika se može zaključiti da performanse particionisane aplikacije prilično zaostaju, ali to ne bi trebalo da predstavlja problem s obzirom da najveći deo podataka unosi pri inicijalnoj postavci sistema.



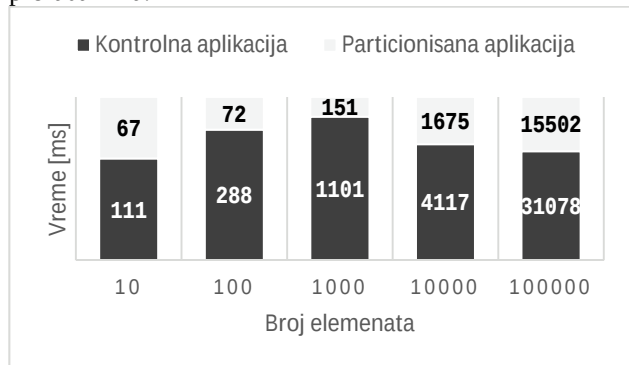
Grafik 1. Upis elemenata

Na narednom grafiku prikazano je trajanje čitanja svih atributa određenog broja elemenata u kontrolnoj i particionisanoj aplikaciji, i sa grafika se može videti da su performanse slične u oba slučaja.



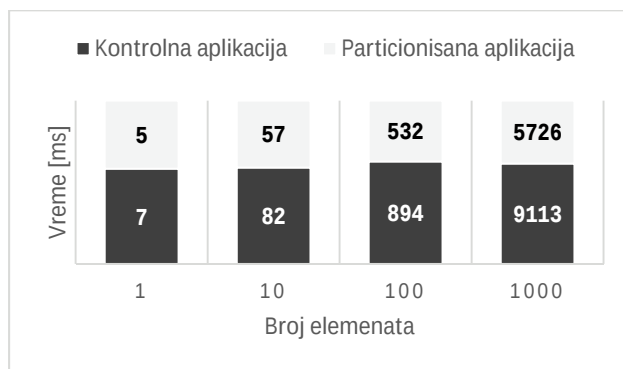
Grafik 2. Čitanje svih atributa elemenata

Grafik 3 prikazuje trajanje čitanja samo električnih atributa svih elemenata koji se nalaze u skladištu kontrolne i particionisane aplikacije. Sa grafika se može zaključiti particionisana aplikacija mnogo brže čita električne attribute elemenata što može pozitivno uticati na performanse servisa koji se bave električnim proračunima.



Grafik 3. Čitanje električnih atributa svih elemenata

Za razliku od prethodnog grafika na grafiku 4 je prikazano trajanje čitanja samo električnih atributa nasumično odabranog dela elemenata, pri čemu se u skladištu nalazi 100000 elemenata. Kao i u prethodnom slučaju, particionisana aplikacija daje značajno bolje rezultate u odnosu na kontrolnu aplikaciju.



Grafik 4. Nasumično čitanje električnih atributa dela elemenata

6. ZAKLJUČAK

U ovom radu predstavljeno je rešenje problema particionisanja modela podataka distributivne mreže koji je opisan CIM standardima. Skladišta podataka su realizovana korišćenjem Azure Table Storage-a, dok je aplikacija particionera napravljena u vidu radne uloge. Particionisana aplikacija je u slučajevima čitanja podataka iz skladišta davala iste ili bolje rezultate od kontrolne aplikacije, dok su u slučajevima manipulacije podacima rezultati na strani neparticionisane aplikacije. Jedan od pravaca daljeg istraživanja i razvoja bi bila dodatno particionisanje koje bi podatke o elementima distributivne mreže koji nemaju električne karakteristike čuvalo u posebnom skladištu, kao i horizontalno particionisanje u slučajevima velikog broja određenih elemenata. Druga stvar koja bi se mogla poboljšati jeste paralelizam unutar aplikacije particionera.

7. LITERATURA

- [1] Youseff, L., Butrico, M., Da Silva, D.: *Toward a unified ontology of cloud computing*, 2008.
- [2] Chapell, D.: *Introducing the Windows Azure platform*, 2010.
- [3] Data Partitioning Guidance - <https://msdn.microsoft.com/en-us/library/dn589795.aspx>
- [4] IEC 61970 Energy management system application program interface (EMS-API) – Part 301: Common Information Model (CIM) Base, 2007.
- [5] IEC 61970 Energy management system application program interface (EMS-API) – Part 453: Diagram layout profile, 2014.

Kratka biografija:



Mladen Korica rođen je 1988. godine u Subotici. Završio je Tehničku školu u Subotici, 2007. godine. Osnovne studije je završio na Fakultetu Tehničkih Nauka u Novom Sadu 2011. godine. Ispunio je sve obaveze i položio je sve ispite predviđene studentskim programom

IMPLEMENTACIJA SISTEMA ZA KONTROLU VERZIJA U OKVIRU FENEK ALATA IMPLEMENTATION OF VESION CONTROL SYSTEM WITHIN FENNEC TOOL

Darko Čolak, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je predstavljena implementacija sistema za kontrolu verzija u okviru Fenek alata. Rad se fokusira na kontrolu verzija šeme baze podataka. Fenec alat je implementiran kao web aplikacija.

Abstract – This paper presents implementation of version control system within Fennec tool. Emphasis of this paper is the version control of database schema. Fennec tool is implemented as a web application.

Ključne reči: Fennec, verzioniranje šeme baze podataka, Git, Python, django.

1. UVOD

Kontrola verzija podrazumeva rukovanje promenama u dokumentima, izvornom kodu ili statičkim datotekama. Najbitnije osobine bilo kog sistema za kontrolu verzija su: praćenje promena nad datotekama, njihovo opisivanje, omogućavanje saradnje više ljudi kao i omogućavanje pouzdanog načina prikaza bilo kog prošlog stanja same datoteke [1].

Predmet ovog rada je kontrola verzija šeme baze podataka.

Verzioniranje šeme baze podataka je od iste važnosti kao i verzioniranje izvornog koda. Čak i pored toga ono se često zanemaruje usled kompleksnosti samog problema verzioniranja. Cilj ovog rada je predstavljanje alata za modelovanje i verzioniranje šema baza podataka.

Trenutno najčešći vid verzioniranja šema baza podataka je ručno praćenje SQL datoteka za izmenu šema. Iako ovakav pristup pruža određene rezultate, izuzetno je nezahvalan i podložan ljudskim greškama. Par ovakvih rešenja su Database Version Control – dvc [2] i Liquibase [3]. Oba se baziraju na praćenju promena u tekstualnim datotekama.

Najkompletnije pronadjeno rešenje predstavlja ERWin Data Modeler [4]. ERWin uključuje rešenja za modelovanje šema baza podataka, praćenje njihovih verzija, itd. Iako rešava problem verzioniranja ERWin nije idealno rešenje jer je komercijalni alat.

Fenek je dizajniran sa namenom da pruži funkcije modelovanja i verzioniranja šema baza podataka kao besplatan alat dostupan pod GNU general public licencom.

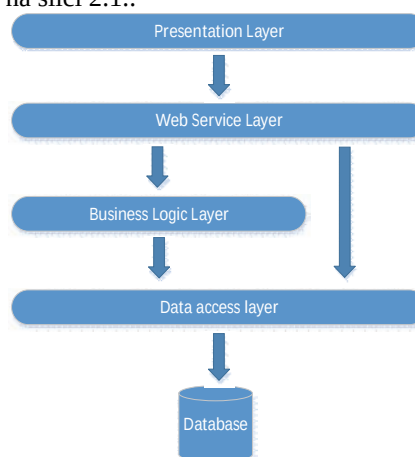
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Gordana Milosavljević.

2. SPECIFIKACIJA SISTEMA

Ovaj rad će se fokusirati na implementaciju serverske strane te se neće baviti detaljima implementacije i tehnologijama korišćenim za izradu klijentskog dela aplikacije (tzv. front end).

Fenek je dizajniran kao slojevita aplikacija u smislu logičke separacije komponenti gde komponente komuniciraju medjusobno. Pregled slojevite arhitekture je prikazan na slici 2.1.:



Slika 2.1. Slojevita arhitektura Fennec alata.

Kao što je prikazano na slici 2.1. aplikacija je podeljena u 4 zasebna sloja: prezentacijski sloj, sloj web servisa, sloj poslovne logike i sloj pristupa bazi podataka.

Prezentacijski sloj je jedini dostupan klijentima. Sastoji se iz tri dela: generalna administracija, administracija projekata i sekcija za modelovanje. Odgovoran je za omogućavanje korisničkih akcija za rukovanje projektima i modelovanje samih šema baza podataka. Prezentacijski sloj je dostupan klijentima preko internet pregledača (browser-a). Prezentacijski sloj komunicira isključivo sa slojem web servisa.

Sloj web servisa omogućava prezentacijskom sloju da pristupi poslovnoj logici potrebnoj za opsluživanje korisničkih akcija. Sloj web servisa pruža sledeće servise preko REST API-ja:

- administracija korisnika i grupa korisnika,
- administracija projekata i članova projekata,
- administracija razvojnih grana i
- administracija verzija razvojnih grana.

Administracija verzija razvojnih grana je najbitnija jer sadrži metode za rukovanje promenama kao i pristupu stanja korisničkih modela šema baza podataka i njihovih

dijagrama. Sloj web servisa je korišćen isključivo od strane prezentacijskog sloja. Sam sloj web servisa komunicira sa slojem poslovne logike i slojem pristupa bazi podataka.

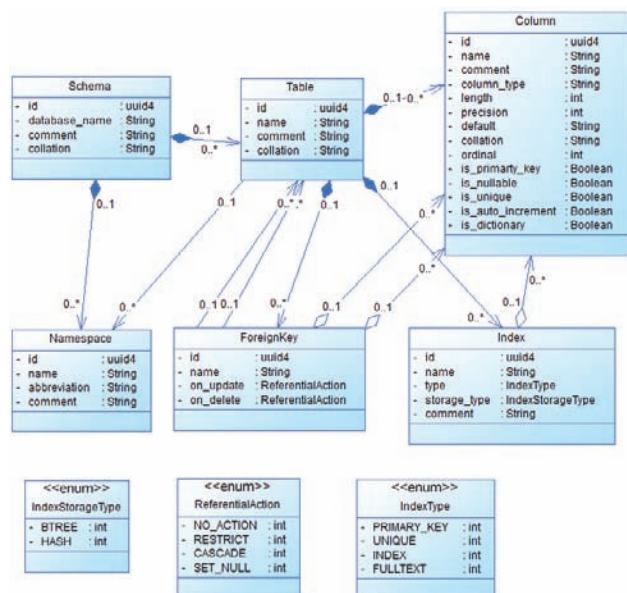
Sloj poslovne logike se sastoji od glavne logike odgovorne za glavne procese u aplikaciji. Sloj sadrži sledeće operacije:

- pristup sandbox okruženju (okruženje projekta u kome korisnik menja model),
- izgradnja objekata koji predstavljaju određeno stanje korisničkog modela,
- izgradnja objekata koji predstavljaju stanje korisničkog sandbox okruženja i
- operacija pravljenja trajnih promena od izmena koje je korisnik napravio na modelu.

Sloj se bavi i prevodjenjem objektnog modela u i iz JSON formatiranih stringova. Ovo je bitno jer se sve korisničke promene snimaju u internoj bazi u vidu stringova.

Sloj pristupa bazi podataka je odgovoran za čuvanje i manipulaciju svih podataka korišćenih u aplikaciji. Sloj sadrži i servise za keširanje podataka. Keširanje podataka je od izuzetne važnosti za performanse same aplikacije. Kako bi aplikacija izgradila stanje korisničkog modela, potrebno je agregirati sve promene nastale od početka projekta. Ukoliko keširanje ne bi bilo uključeno, ova operacija bi se morala izvoditi svaki put kada korisnik otvori model, ili zatraži pristup nekoj ranijoj verziji modela.

Fennec aplikacija koristi podskup CWM-a (Common Warehouse Metamodel [5]). CWM omogućava modelovanje relacionih šema, skladištenja podataka, itd. Podskup modela je prikazan na slici 2.2.



Slika 2.2. Podskup CWM-a korišćen u aplikaciji

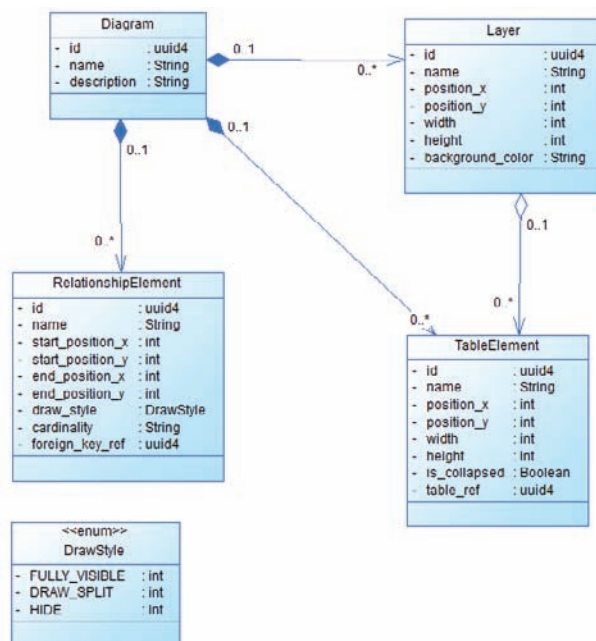
Metamodel sa slike 2.2. sadrži sledeće metaklase:

- šema baze podataka (služi za grupisanje tabela i imenskih prostora koji čine jednu šemu baze podataka),

- imenski prostor (služi za grupisanje tabela koje pripadaju istom imenskom prostoru, sadrži i prefix koji se upotrebljava prilikom generisanja imena tabela),

- tabela,
- kolona,
- index i
- strani ključ.

Metamodel grafičkih elementa korisničkih modela je prikazan na slici 2.3.



Slika 2.3. Metamodel grafičkih elemenata

Metamodel sa slike 2.3. sadrži sledeće klase:

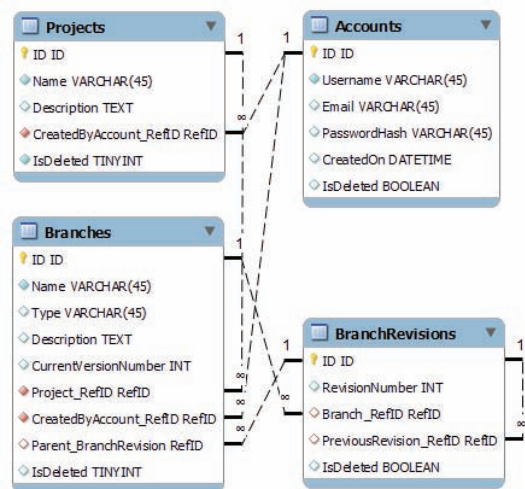
- dijagram (predstavlja jedan korisnički dijagram u okviru projekta),
- sloj (služi za logičko grupisanje tabela u okviru dijagrama),
- element table (predstavlja vizuelnu prezentaciju CWM table) i
- element veze (predstavlja vizuelnu prezentaciju veze između dve CWM table).

Unutrašnja struktura skladištenja podataka unutar Fennec aplikacije se deli na dve celine: administracija korisnika i projekata i rukovanje promenama.

U smislu organizacije i uloge, projekat je ekvivalentan repozitorijumu iz sistema za kontrolu verzija (git npr.). Projekat je organizaciona struktura koja obuhvata set šema baza podataka uz njihove odgovarajuće dijagrame. Projekat se sastoji iz jedne ili više nezavisnih grana. Grane, kao i sam projekat kreira korisnik.

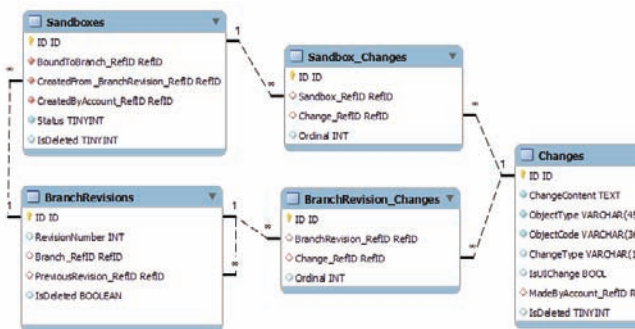
Grana predstavlja nezavisan tok razvoja korisničkog modela. Postojanje grana omogućava kako paralelan razvoj između više učesnika, tako i paralelan razvoj

odvojenih celina u okviru projekta. Svaku granu čini aciklični graf čvorova koji se nazivaju revizije grane. Svaki čvor predstavlja reviziju, odnosno skup promena nastalih u odnosu na prethodni čvor. Opisana organizacija projekta je prikazana na slici 2.4.



Slika 2.4. Organizacija projekta

Sandbox predstavlja privremenu verziju revizije grane u kojoj korisnik pravi promene. Promene u okviru sandbox okruženja su vidljive samo korisniku koji ih je napravio. Ove promene postaju trajne posle commit (skladištenje) operacije. Nakon operacije skladištenja sve promene koje su bile sačuvane kao privremene, postaju trajne i vezuju se za novu reviziju grane. Dijagram koji prikazuje odnos privremenih (Sandbox_Changes) i trajnih promena (BranchRevision_Changes) je prikazan na slici 2.5.:



Slika 2.5. Privremene i trajne promene

Sama promena sačinjava jediničnu izmenu koja je napravljena na modelu. Promena može biti izmena elementa šeme baze podataka ili izmena grafičkog prikaza nekog od elemenata. Promena sadrži kolonu SadržajPromene koja sadrži tekstualnu prezentaciju objekta koji je izmenjen u JSON formatu. Pored toga sadrži i dodatne informacije o promeni: da li je vizuelnog ili semantičkog karaktera, kog je tipa objekat promene (tabela, kolona, element veze, itd.), kog je tipa promena (dodavanje, izmena ili brisanje) i reference na korisnika koji je promenu napravio. Primer sadržaja promene je prikazan u listing 2.1.

```
{
  "id": 1,
  "content": "{ 'collation': 'utf-8', 'comment': 'minor',
    'schemaRef': 'e0cc3067-c31c-4273-8252-a4f750150962',
    'id': '2cd611e4-e9a3-472c-a221-5dba18877050',
    'name': 'TableOne' }",
  "object_type": "Table",
  "object_code": "",
  "change_type": 2,
  "is_ui_change": false,
  "object_code": 2,
  "made_by": 2
}
```

Listing 2.1 Primer objekta promene u JSON formatu

Izgradnja stanja revizije grane se izvršava tako što se agregiraju sve promene načinjene od početka projekta sve do trenutne revizije grane, uključujući i promene koje sačinjavaju izabranu reviziju.

Agregiranje promena znači sabiranje, oduzimanje ili izmenu elemenata svih elemenata (koji su podložni promenama) tokom životnog toka grane. Na primer dodavanje tabela, uklanjanje pojedinih kolona iz tabele, pomeranje grafičkih elemenata.

3. IMPLEMENTACIJA SISTEMA

Serverska strana aplikacije je implementirana u Python programskom jeziku. Intenzivno je korišćen popularan web framework za Python: Django Rest Framework.

Za potrebe razvoja korišćena je SQL Lite baza koja omogućava brz razvoj i zadovoljavajuće performanse tokom faze razvoja. Django models paket je korišćen kod samog pristupa bazi. Paket se koristi tako što se nasledjivanjem models.Model klase dobijaju entitet klase koje enkapsuliraju pristup tabelama na koje se mapiraju. Na listingu 3.1. prikazana je Project entitet klasa koji se mapira na "Projekat" tabelu:

```
class Project(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=64)
    description = models.CharField(max_length=512, null=True)
    created_by = models.ForeignKey(settings.AUTH_USER_MODEL,
        help_text="project author")
    is_deleted = models.BooleanField(default=False)
```

Listing 3.1. Project klasa

Web metode su dostupne preko REST Web servisa koje pruža Django Rest Framework. Listing 3.2. prikazuje putanje na kojima su dostupni CRUD servisi (omogućavaju CRUD operacije snimanja, čitanja, brisanja i ažuriranja):

```
/auth/
/api/users/
/api/groups/
/api/project-members/
/api/projects/
/api/projects/{id}/branches/
/api/projects/{id}/branches/{b-id}/branch-revisions
```

Listing 3.2. Spisak putanja CRUD servisa

Branch-Revision web servis sadrži i dodatne operacije koje su prikazane na listingu 3.3.:


```

/api/projects/{id}/branches/{b-id}/branch-revisions/{br-id}
- /branch POST
- /change POST
- /commit POST
- /project_state GET
- /metadata GET
- /diagram?id={id} GET

```

Listing 3.3. Dodatne operacije nad revizijom grane

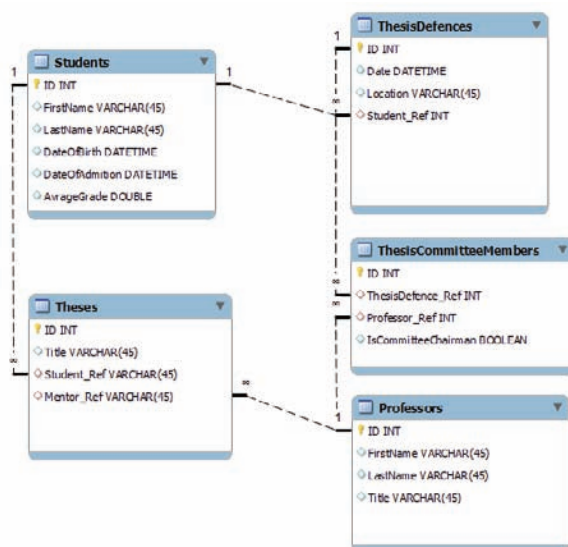
Dodatne operacije omogućavaju korisniku grananje od trenutne grane, pravljenje izmena nad svojim sandbox okruženjem, snimanje napravljenih izmena kao i pribavljanje celokupnog stanja projekta, pribavljanje samo metapodataka ili pribavljanje informacija o pojedinim dijagramima.

4. STUDIJA IZVODLJIVOSTI

U okviru rada prikazana je studija izvodljivosti prikazanog rešenja sistema za kontrolu verzija. Cilj studije je da prikaže mogućnosti i performanse serverske strane dela aplikacije u radu sa modelima baza podataka raznih veličina.

U svrhu testiranja rada aplikacije, razvijena su dva dodatna modula, MySQL Workbench parser i Fennec importer (moduli za parsiranje i uvlačenje modela iz xml datoteke).

Na slici 4.1. prikazan je model korišćen u studiji.



Slika 4.1. Primer modela – Odbrana diplomskog rada

Nakon parsiranja i uvlačenja xml datoteke koja predstavlja MySQL model sa slike 4.1. unutar prethodno kreiranog projekta, može se videti rezultat Fennec aplikacije, kao stanje celokupnog projekta u obliku JSON teksta.

Listing 4.1. prikazuje deo ispisa koji čini stanje uvučenog projekta.

```

{
  "id": "a956c2e6-5276-48ca-a2ab-85f67301a717",
  "databaseName": "mydb",
  "comment": "Sample schema used for case study.",
  "collation": "utf8_general_ci",
  "tables": [
    {
      "id": "02655097-dd01-43e1-9560-57c179231e9e",
      "name": "ThesisCommitteeMembers",
      "schemaRef": "a956c2e6-5276-48ca-a2ab-85f67301a717",
      "columns": [
        {
          "id": "a4571c9c-dc3c-4411-b983-39eca9a306b8",
          "name": "Professor_Ref",
          "column_type": "INT",
          "ordinal": 2,
          "primary": false,
          "nullable": true,
          "unique": false,
          "autoIncrement": false,
          "dictionary": false,
          "tableRef": "02655097-dd01-43e1-9560-57c179231e9e"
        },
        {
          "id": "ae668b40-9154-4278-bcc8-720f98c6ad1e",
          "name": "ID",
          "column_type": "INT",
          "ordinal": 0,
          "primary": false,
          "nullable": false,
          "unique": false,
          "autoIncrement": false,
          "dictionary": false,
          "tableRef": "02655097-dd01-43e1-9560-57c179231e9e"
        }
      ],
      "indexes": [
        {
          "id": "30049061-dc10-4190-a137-94a29cee0466",
          "name": "PRIMARY",
          "storageType": "PRIMARY",
          "columns": [
            "ae668b40-9154-4278-bcc8-720f98c6ad1e"
          ],
          "tableRef": "02655097-dd01-43e1-9560-57c179231e9e"
        }
      ],
      "foreignKeys": [
        {
          "id": "1f3ddd01-ec52-4ac6-8ca9-3ae5a4995edd",
          "name": "fk_ThesisCommitteeMembers_Professors1",
          "sourceColumn": "a4571c9c-dc3c-4411-b983-39eca9a306b8",
          "referencedColumn": "cce6241b-965b-4ffa-bee4-f3be03e45435",
          "tableRef": "02655097-dd01-43e1-9560-57c179231e9e"
        }
      ]
    }
  ]
}

```

Listing 4.1. JSON rezultata za primer modela.

5. ZAKLJUČAK

U ovom radu prikazana je implementacija serverske strane web aplikacije Fennec. Cilj aplikacije je da pruži korisnicima mogućnosti modelovanja i kontrole verzija šema baza podataka.

6. LITERATURA

- [1] <http://www.catb.org/esr/writings/version-control/version-control.html>, Eric Raymond's "Understanding Version-Control Systems"
- [2] <http://dbv.vizuina.com/>, Database version control library
- [3] <http://www.liquibase.org/index.html>, Liquibase software
- [4] <http://erwin.com/products/data-modeler/>, ERwin data modeler software
- [5] <http://www.omg.org/spec/CWM/>, Common Warehouse Metamodel

Kratka biografija

Darko Čolak je rođen u Kikindi, 1990. god. Fakultet tehničkih nauka upisao je 2008. godine, odsek Računarstvo i automatika. Osnovne akademske studije završio je 2012.

SIMULACIONI MODEL FOTONAPONSKOG SOLARNOG SISTEMA A SIMULATION MODEL OF THE PHOTOVOLTAIC SOLAR SYSTEM

Miloš Savić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su dati osnovni podaci o sunčevoj energiji, objašnjen je princip rada solarne ćelije, dat je model fotonaponskog panela.

Abstract – The paper presents basic facts about solar energy, explains the principle of the solar cell, a model of photovoltaic panel.

Ključne reči: Sučeva energija, solarne ćelije, fotonaponski paneli.

1. UVOD

Višedecenijsko neracionalno trošenje različitih izvora energije dovelo je do pojave krize energenata. Svet se našao u opasnosti od gubitka izvora energije kako danas tako i u budućnosti. Rasipanje energije i neizvesna budućnost podstakli su stručnjake i članove društvene zajednice da se okrenu primeni alternativnih izvora energije. U alternativne izvore energije se ubrajaju: sunčeva energija, energija vetra, energija morskih talasa, geotermalna energija.

2. ENERGIJA SUNČEVOG ZRAČENJA

Snaga Sunčevog zračenja iznosi oko $3,8 \cdot 10^{26}$ W, od čega na Zemlju padne oko $1,7 \cdot 10^{17}$ W, odnosno oko $4 \cdot 10^{24}$ J energije. To je 20 000 puta više nego što iznosi ukupna godišnja potrošnja energije iz svih primarnih izvora. Dotok energije Sunčevim zračenjem van Zemljine atmosfere predstavljen je solarnom konstantom, koja iznosi 1367 W/m^2 , pri srednjoj udaljenosti Zemlje od Sunca, uz upadni ugao od 90 stepeni. Pri prolasku kroz atmosferu deo energije se troši u složenim procesima, a deo se reflektuje i reemituje u svemir. Taj deo iznosi oko 1/3 energije koja je dospela na rub atmosfere, pa dotok energije do površine Zemlje iznosi prosečno 920 W/m^2 . Ako je projekcija površine Zemlje $127.731.670 \text{ km}^2$, snaga zračenja iznosi 117.484 TW

Zbog rotacije Zemlje, ta energija se raspoređuje po celoj površini Zemlje ($510.065.600 \text{ km}^2$), pa je prosečna snaga oko 230 W/m^2 , odnosno $5,52 \text{ kWh/m}^2$ dnevno.

3. FOTONAPONSKI SISTEMI

Osnovni elementi fotonaponskog sistema su: fotonaponski paneli, akumulator (baterija), inverter, regulator (kontroler), potrošač. Potrošači mogu biti naizmeničnog i jednosmernog karaktera. Ilustracija fotonaponskog sistema data je na slici 1.

NAPOMENA:

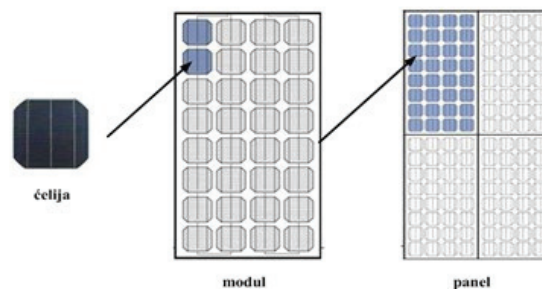
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Velimir Čongradac.



Slika 1. Ilustracija fotonaponskog sistema

3. 1 Fotonaponski paneli

Kako je snaga pojedinačne FN ćelije veoma mala (do 2 W), pa se kao takva ne može upotrebljavati naširoko, FN ćelije se mehanički i električno vezuju u veće celine koje se nazivaju moduli. U cilju dobijanja još većih snaga, moduli se vezuju u panele, čije snage idu i do reda MW. Na slici 2 dat je prikaz ćelije, modula i panela.



Slika 2. Ćelija, modul, panel

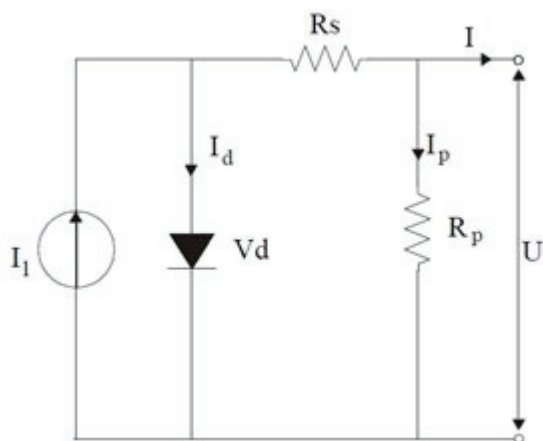
Nije teško zaključiti da ako želimo od odabranog broja FN ćelija, da dobijemo najveću moguću struju, onda te ćelije treba (električno) vezati paralelno. Ako je struja jedne ćelije I_1 , a broj ćelija n , onda je, u idealnom slučaju, ukupna struja paralelno vezanih ćelija $I_{uk} = n \cdot I_1$. Slično važi i za napon. Ako od odabranog broja ćelija, želimo dobiti što je moguće veći napon, onda date ćelije treba vezati redno. Ako je napon jedne ćelije U_1 , a broj ćelija n , onda je ukupni napon, redno vezanih ćelija $U_{uk} = n \cdot U_1$. Sve ovo važi i u slučaju vezivanja većeg broja modula.

3.2 Fotoelektrični efekat

Fotonaponska (FN) ćelija temeljni je blok fotonaponskog sistema. Ona predstavlja p-n spoj koji pomoću fotonaponskog efekta sunčevo zračenje pretvara u električnu energiju. Sačinjena je od poluprovodničkog

materijala (najčešće silicijuma). Usled apsorpcije sunčevog zračenja, u p-n spoju se javljaju parovi elektron-šupljina. Prilikom sunčevog zračenja unutar ili u blizini p-n spoja, unutrašnje električno polje razdvaja elektrone i šupljine. Pri tome se elektroni kreću prema n strani, a šupljine prema p strani p-n spoja, i kao posledica toga javlja se potencijalna razlika, odnosno napon. Kada se FN ćelija poveže u strujno kolo, kroz potrošač će poteći struja.

Osvetljena FN ćelija ponaša se dakle kao izvor stalne struje. Njena ekvivalentna šema prikazana je na slici 3. Izvor stalne struje paralelno je spojen sa diodom. Serijski otpor R_s predstavlja otpornost p-n spoja, zavisi od materijala od koga je izgrađena FN ćelija i poželjno je da bude što manji. Paralelni otpor R_p potiče od mikrodefekata i nečistoća unutar FN ćelije i zavisi od osobina ćelije i većinom je dovoljno veliki da ga možemo smatrati beskonačnim. Tipične vrednosti za R_s i R_p , za silicijumske FN ćelije iznose $R_s < 0,1 \Omega$ i $R_p > 500 \Omega$.



Slika 3. Šema fotonaponske ćelije

Na osnovu date ekvivalentne šeme FN ćelije, i uz određena zanemarenja ($R_s=0$ i $R_p \rightarrow \infty$), može se napisati jednačina izlazne struje I:

$$I = I_f - I_d = I_f - I_0 \cdot \left(e^{\frac{qU}{kT}} - 1 \right) \quad (3.1)$$

gde je:

I_0 – inverzna struja zasićenja diode,

I_f – generisana fotostruja,

U, I – izlazni napon i struja, respektivno.

3.3 Parametri FN ćelija

Jedan od najvažnijih parametara FN ćelije jeste njena elektromotorna sila, tj. napon na njenim krajevima u slučaju praznog hoda. Kada se stavi $I=0$ u jednačinu (3.1), dobija se napon praznog hoda:

$$U_{ph} = \frac{kT}{q} \cdot \ln \left(\frac{I_f}{I_0} - 1 \right) \quad (3.2)$$

Na osnovu datog izraza (3.2) vidi se da napon praznog hoda U_{ph} raste sa porastom I_f , a opada sa povećanjem I_0 . Pošto inverzna struja zasićenja diode I_0 raste sa povećanjem temperature, sledi da sa porastom temperature opada napon praznog hoda. U uslovima kada serijski otpor nije zanemariv, dobija se:

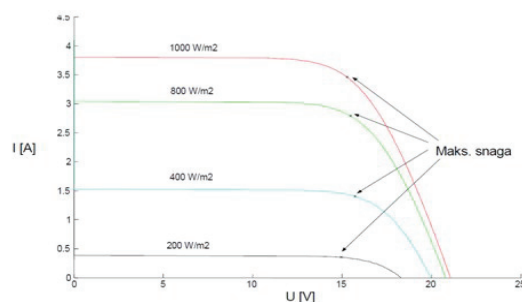
$$U_{ph} = \frac{kT}{q} \cdot \ln \left(\frac{I_f}{I_0} - 1 \right) - R_s I \quad (3.3)$$

Elektromotorna sila FN ćelije zavisi od fotostruje (koja raste sa porastom nivoa zračenja) i od struje zasićenja diode za koju je poželjno da je što manja. Važan podatak za ćeliju je i struja kratkog spoja I_{ks} , tj. jačina struje u slučaju kada su krajevi ćelije kratko spojeni. Kada se u jednačini (3.1) stavi da je $U=0$, dobijamo da je struja kratkog spoja jednaka fotostruji:

$$I_{ks} = I_f \quad (3.4)$$

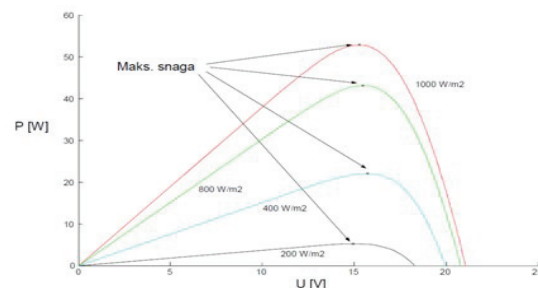
Promena intenziteta zračenja (iradijacije) bitno utiče na UI karakteristiku ćelije. Struja kratkog spoja praktično je proporcionalna intenzitetu zračenja, slika 4.

$$I_{ks} = I_{ks, nom} \cdot \frac{G}{G_{nom}} \quad (3.5)$$



Slika 4. Zavisnost struje kratkog spoja od sunčevog zračenja

Za razliku od struje kratkog spoja, napon praznog hoda se menja po logaritamskoj funkciji sa intenzitetom zračenja. Ako se zračenje smanji dva puta (npr. sa 1000 W/m^2 na 500 W/m^2) struja kratkog spoja smanji se dva puta, a napon praznog hoda oko 5%, slika 5.



Slika 5. Zavisnost napona praznog hoda od sunčevog zračenja

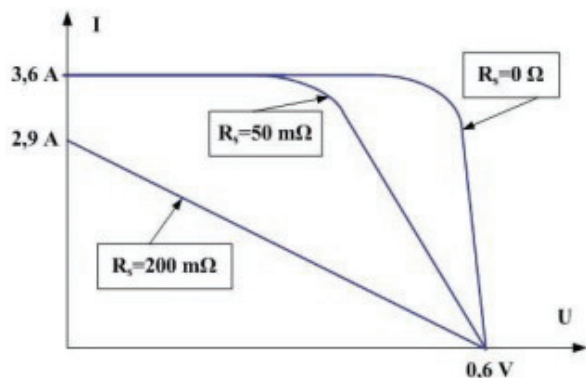
Svaka FN ćelija ima sopstvenu otpornost, koja se može ekvivalentirati preko serijskog i paralelnog otpora. UI karakteristika FN ćelije itekako zavisi od vrednosti ove dve otpornosti. Ovaj uticaj se može zanemariti u slučaju da je serijski otpor mnogo puta manji od karakterističnog

otpora ćelije $R_s \ll R_k$ ($R_k = \frac{U_{ph}}{I_{ks}}$), odnosno ako je paralelni otpor mnogo puta veći od karakterističnog otpora $R_p \gg R_k$.

U realnom modelu FN ćelije potrebno je uzeti u obzir serijski R_s i paralelni otpor R_p . Jednačina izlazne struje realne ćelije je:

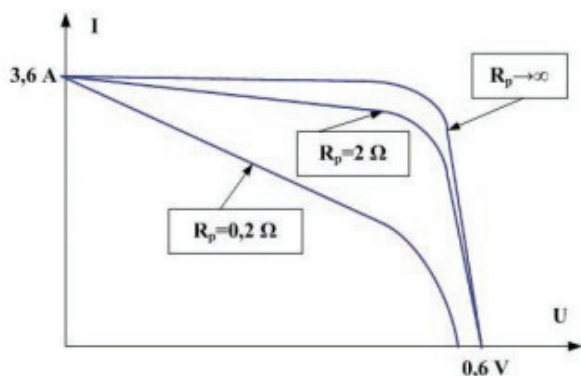
$$I = I_f - I_d = I_f - I_0 \cdot \left(e^{\frac{q(U+IR_s)}{nkT}} - 1 \right) - \frac{U + IR_s}{R_p} \quad (3.6)$$

Na osnovu izraza (3.6) na sledeće dve slike biće prikazano kako promene pomenutih otpornosti utiču na UI karakteristiku FN ćelije.



Slika 6. Uticaj otpornosti R_s na oblik UI karakteristike FN ćelije

Sa slike se može uočiti da postojanje male vrednosti serijske otpornosti R_s , itekako degradira UI karakteristiku FN ćelije. Zbog toga je veoma bitno prilikom projektovanja same FN ćelije voditi računa da ta otpornost bude što je moguće manja.



Slika 7. Uticaj otpornosti R_p na oblik UI karakteristike FN ćelije

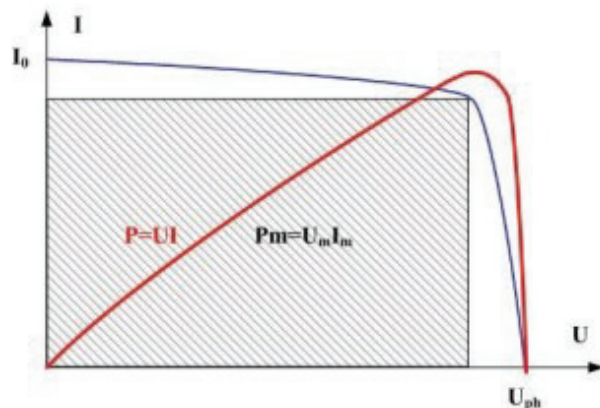
Paralelna otpornost R_p je reda nekoliko stotina oma, pa se sa slike može videti da prisustvo paralelne otpornosti ne utiče mnogo na oblik UI karakteristike FN ćelije.

3.4 Snaga fotonaponske ćelije

Snaga koju daje fotonaponska ćelija dobija se kao proizvod napona i struje:

$$P = UI = U \cdot \left(I_f - I_0 e^{\frac{qU}{kT}} + I_0 \right) \quad (3.7)$$

Maksimalna snaga koju idealna ćelija može dati $P_m = U_m I_m$ označena je na slici 8. šrafirom površinom. Računski se može odrediti traženjem pravougaonika sa najvećom površinom. Maksimalna snaga FN ćelije može se izraziti i preko napona praznog hoda i struje kratkog spoja:



Slika 8. Snaga fotonaponske ćelije

$$P_m = U_m I_m = U_{ph} \cdot I_{ks} \cdot F \quad (3.8)$$

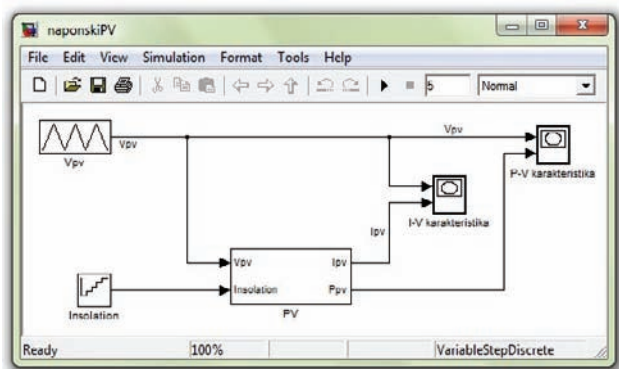
gde je:

$$F = \frac{U_m I_m}{U_{ph} I_{ks}} \quad (3.9)$$

Ovaj odnos je poznat kao faktor ispunenosti ili filing faktor FN ćelije. To je odnos površine pravougaonika sa stranicama U_m i I_m , i pravougaonika sa stranicama U_{ph} i I_{ks} (obično važi da je $0,7 < F < 0,9$). On pokazuje koliko se realna ćelija približava idealnoj, odnosno koliki je uticaj serijskog otpora ćelije. Filing faktor F opada linearno sa odnosom R_s/R_k i R_k/R_p .

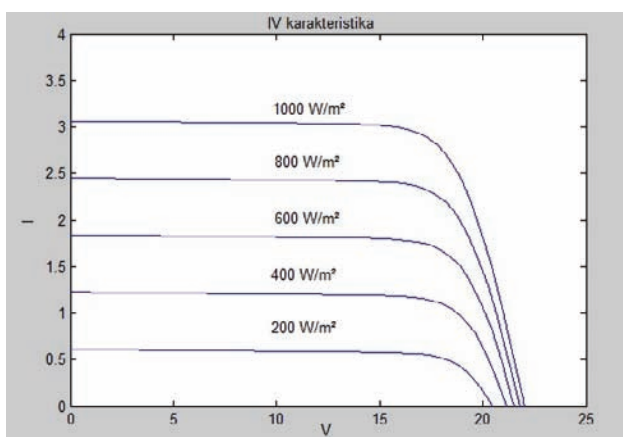
4. REALIZACIJA MODELA FN PANELA

Matlab/Simulink je postao nezamenljiv i standardan alat u fazi projektovanja i analizi rada, modelovanih fotonaponskih sistema. Zahtevi za maksimalnim povećanjem efikasnosti, zatim činjenice da su ovi sistemi nelinearni, primorali su inženjere da u radu koriste kompjuterske simulacije za proučavanje i analizu rada, a da pri tome ne moraju da ih izgrađuju. U praksi postoje dva postupka ili dve metode modelovanja FN modula, koji se koriste u zavisnosti da li nam na izlazu FN modula treba veći napon ili struja. Dakle postoji: serijski model, gde su solarne ćelije vezane serijski i paralelni model, kod koga su solarne ćelije vezane paralelno. Na slici 9. je prikazan naponski tip FN modela. Model pokazuje uticaj promene intenziteta zračenja na izlazne karakteristike pri konstantnoj temperaturi.

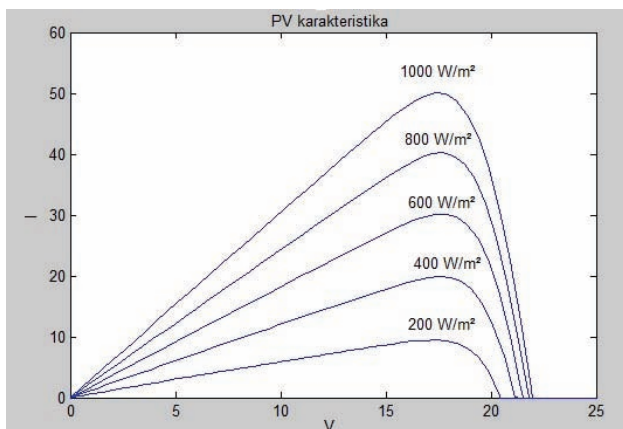


Slika 9. MATLAB model FN panela

Na sledećim slikama prikazani su rezultati simulacije, sa kojih se jasno može videti kako i uolikoj meri promena intenziteta zračenja utiče na promenu izlaznih karakteristika FN modula.



Slika 10. Promena UI karakteristike FN modula sa promenom intenziteta zračenja



Slika 11 Promena PV karakteristike FN modula sa promenom intenziteta zračenja

Na slikama se vidi da struja, a takođe i snaga fotonaponskih modula, pokazuje linearnu zavisnost sa promenom inteziteta sunčevog zračenja.

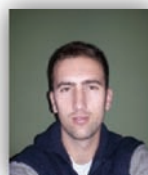
5. ZAKLJUČAK

Sunce je ogroman i nepresušan izvor energije koja je svima dostupna. Ako nauka i tehnologija u potpunosti ovladaju korišćenjem solarne energije i njenim efikasnim i ekonomski isplativim pretvaranjem u električnu energiju, čovečanstvo i planeta Zemlja neće osetiti posledice globalnog zagađenja i nestašice fosilnih energenata u budućnosti. Fotonaponski pretvarači solarne u električnu energiju su, verovatno, najbrži i najčistiji put rešavanja ovih problema. Ipak, da bi se uspešno ovladalo bilo kojom novom tehnologijom, potrebna su prvenstveno ulaganja u obrazovanje. Specifičnost i multidisciplinarnost solarne energetike zahtevaju široku akciju obrazovnih institucija na svim nivoima, od osnovnih škola do Univerziteta, ali i podizanje opšte informisanosti u ovoj oblasti putem medija, jer neznanje i neinformisanost, po pravilu, koštaju više i od najskuplje tehnologije.

7. LITERATURA

- [1] A. Perteši, „Solarna energija“, 2009
- [2] U. Jeremić, „Fotonaponsko pretvaranje energije (solarne ćelije)“, 2007
- [3] J. Dunlop, „Photovoltaic Systems“, 2009
- [4] R. Messenger, J. Ventre, „Photovoltaic Systems Engineering“, 2010
- [5] J. Nelson, „The Physics of Solar Cells“, 2003

Kratka biografija:



Miloš Savić rođen je u Kruševcu 1989. godine. Osnovne studije na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo – Računarstvo i automatika završio je 2012. god. Trenutno student master studija na smeru Automatika i upravljanje sistemima.

**ANALIZA STATIČKE SIGURNOSTI ELEKTROENERGETSKIH SISTEMA SA
PRIMENOM KOREKTIVNIH MERA****POWER SYSTEM SECURITY ANALYSIS WITH REMEDIAL ACTIONS**

Dušan Stanivuković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su opisane korektivne akcije i njihova primena u analizi statičke sigurnosti elektroenergetskih sistema. Korektivne akcije su modelovane u skladu sa standardom CGMES v2.5.2. Njihova primena je verifikovana na primeru test mreže.

Abstract – This paper contains description of remedial actions in power system security analysis. Remedial actions are defined in accordance with CGMES v2.5.2. model exchange standard. Remedial actions are verified on test network.

Ključne reči: sigurnost, kriterijumi sigurnosti, ENTSOE, korektivne akcije

1. UVOD

Osnovna funkcija elektroenergetskih sistema (EES) je obezbeđenje napajanja potrošača kvalitetnom električnom energijom, vodeći se pritom osnovnim principima rada, a to su pouzdanost, sigurnost i ekonomičnost. Od suštinskog interesa za ovaj rad je sigurnost EES koja predstavlja sposobnost sistema da bez većih posledica podnese neke nepredviđene, ali karakteristične i vrlo verovatne poremećaje. Kao poremećaj obično se definiše ispad nekog elementa: generatora, transformatora, voda ili nagla i iznenadna promena opterećenja, napona i učestanosti [1].

U radu je opisan način obezbeđivanja sigurnosti Evropskih operatora primenom dva pristupa, determinističkog N-1 kriterijuma sigurnosti i probabilističkog pristupa zasnovanog na pojmu rizika. U trećoj glavi su opisane preventivne i korektivne akcije i prikazan je način njihove integracije u postojeći N-1 proračun. Prikazane su metode za identifikaciju preventivnih i korektivnih akcija, kao i najčešće vrste akcija koje se primenjuju u upravljanju prenosnim sistemima. Njihova verifikacija na test mreži predstavljena je u petoj glavi.

2. STATIČKA SIGURNOST EVROPSKIH TSO

Da bi se obezbedio rad sistema u normalnom radnom režimu, vrše se analize statičke sigurnosti i upravlja se mrežom na takav način, da se dovede u stanje koje zadovoljava osnovne kriterijume sigurnosti [1]. Za analizu statičke sigurnosti, od interesa su stacionarna stanja u koja sistem dolazi nakon poremećaja i završenih prelaznih pojava. Dakle, vrše se sekvencijalne simulacije

neraspoloživosti nekog od elemenata sistema. Ukoliko za neki ispad postoje narušena pogonska ograničenja, kaže se da sistem ne zadovoljava kriterijum sigurnosti. Potom se vrši identifikacija korektivnih akcija i njihova primena u slučaju pojave kritičnog režima, kako bi se sistem vratio u normalni radni režim.

U slučaju ispada dalekovoda ili transformatora, dolazi do preraspodele tokova snaga po ostalim granama, te može doći do narušavanja njihovog maksimalnog opterećenja ili proboja granica napona. U slučaju međusistemskih vodova i transformatora, njihov ispad se može ekvivalentirati ispadom proizvodnje/potrošnje, koji za posledicu ima odstupanje ugovorene snage razmene na interkonektivnim vodovima (u slučaju povezanih sistema), ili povećanje/pad frekvencije usled narušavanja bilansa snaga (u slučaju izolovanih sistema). Zato je potrebno unapred izvršiti analizu statičke sigurnosti i identifikovati preventivne i korektivne akcije, čijom će se koordinisanom primenom sprečiti narušavanje pogonskih ograničenja usled neraspoloživosti bilo kog elementa sistema iz definisane liste kritičnih elemenata.

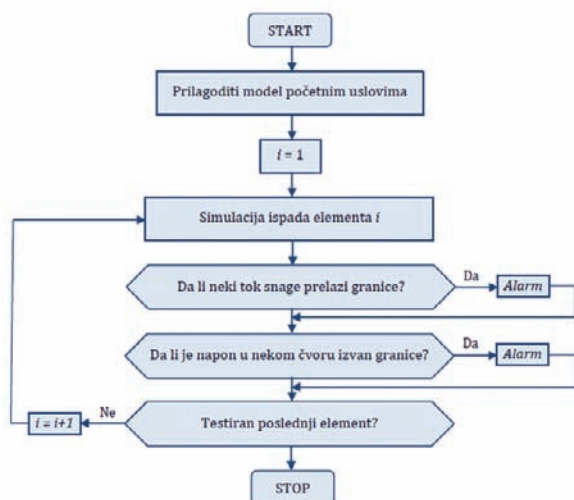
Procena sigurnosti Evropskih operatora se i dalje oslanja na deterministički N-1 kriterijum. Sistem zadovoljava N-1 kriterijum sigurnosti, ukoliko usled neraspoloživosti bilo kojeg elementa iz unapred definisane liste kritičnih elemenata, ne dolazi do narušavanja ograničenja maksimalnog opterećenja elemenata, niti do proboja granica napona. Iako razumljiv i jednostavan za implementaciju, ovakav pristup ima nekoliko mana [2]. Po N-1 kriterijumu, postoje samo dva rezultata analize sigurnosti, siguran i nesiguran sistem. Ukoliko su dva režima sigurna, nemoguće je porediti njihove nivoe sigurnosti, tj. odrediti kvantitativnu meru sigurnosti, a samim tim ni integrisati sigurnost u ekonomske analize. Pored toga, ovaj pristup se pokazao kao neefikasan pri analizi simultanih i kaskadnih ispada. Raspad prenosnih mreža su se dešavali iako je analiza sigurnosti pokazala da je mreža sigurna, jer ispadi koji su dovodili do njih nisu bili razmatrani usled izrazito male verovatnoće. Blok dijagram N-1 algoritma je prikazan na slici 2.1.

Probabilistički pristup rešava problem određivanja globalnog pokazatelja nivoa sigurnosti mreže. Oslanja se na pojam rizika, koji zavisi od verovatnoće pojave ispada i posledice (štete) ispada. Nivo rizika se prikazuje na „iso – risk“ krivama (krive zavisnosti verovatnoće ispada i štete) [4], gde je rizik konstanta. Mana pristupa je u tome što je ozbiljnost pojave ispada nepoznata sve dok se ispad ne desi, pa je potrebno sprovesti dodatne simulacije određivanja ove nepoznate. S obzirom da analiza sigurnosti predstavlja oblik „brute force“ proračuna koji

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Goran Švenda, red.prof.

je zahtevan u pogledu računarskog vremena, današnja softverska rešenja se uglavnom baziraju na $N-1$ kriterijumu sigurnosti [2].



Slika 2.1 - Opšti dijagram $N-1$ proračuna [3]

Lista kritičnih elemenata, čija se nerazpoloživost simulira, formira se na osnovu verovatnoće ispada. Ranije je analiza sigurnosti podrazumevala proveru robusnosti sistema samo na jednostruke ispadе, vodeći se $N-1$ pravilom. Sa druge strane, današnji prenosni sistemi rade pod znatno višim opterećenjem, pa je i verovatnoća pojave višestrukih ispada uvećana.

Održavanje sigurnosti sistema po $N-2$ ili $N-X$ pravilu nije realna, jer bi u tom slučaju bila potrebna znatno veća finansijska ulaganja u infrastrukturu.

Kompromis je pronađen na nivou provere sigurnosti na sve jednostruke ispadе i neke karakteristične, vrlo moguće višestruke ispadе, kao npr. ispad sabirnice, dva paralelna voda itd. Takođe, lista se može formirati korišćenjem rezultata OPF (Optimal Power Flow) funkcije ili dodatnim analizama osetljivosti, ali za tim obično nema potrebe, jer su liste slabo promenljive (iskustveno se pravi početna lista, a kasnije se ažurira ukoliko dođe do značajnije izmene topologije mreže ili dodavanja novih elemenata i sl.) [5].

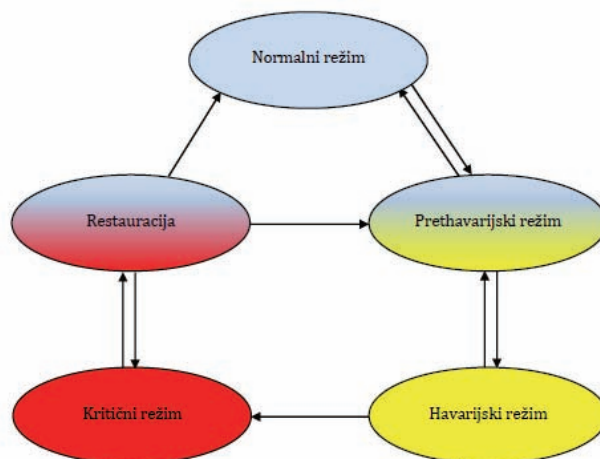
2.1 Radni režimi EES

U toku rada, EES se može nalaziti u nekoliko radnih režima, koji zavise od zadovoljenja uslova:

- bilans proizvodnje i potrošnje,
- pogonska ograničenja,
- sigurnosna ograničenja.

Bilans aktivnih i reaktivnih snaga predstavlja osnovni preduslov rada sistema u stacionarnom stanju. U slučaju ispada proizvodnje/potrošnje dolazi do narušavanja bilansa snaga i upravljačke akcije su usmerene ka ponovnom uspostavljanju bilansa.

U pogonska ograničenja spadaju maksimalni tokovi snaga po granama, dozvoljeni opseg napona čvorova, opseg snaga proizvodnje generatora itd. Sigurnosna ograničenja su vezana za ispunjenost nekog od kriterijuma sigurnosti. Odnos režima, kao i mogući prelasci iz jednog u drugi, prikazani su na slici 2.2.



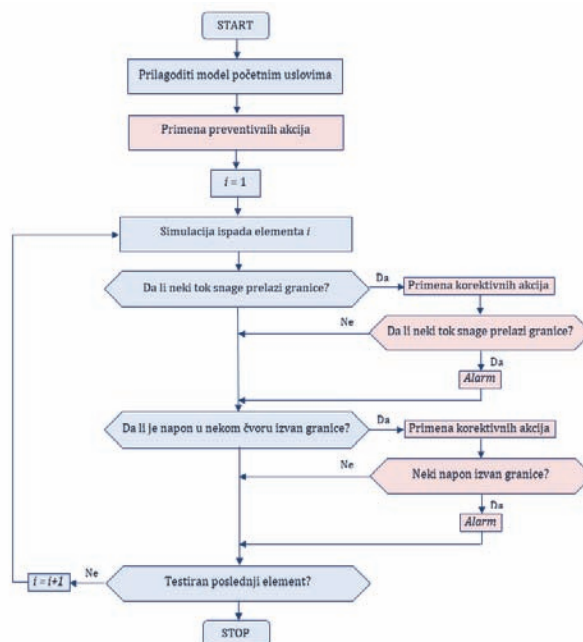
Slika 2.2 – Radni režimi EES [4]

3. PREVENTIVNE I KOREKTIVNE MERE

Drugi korak u obezbeđenju sigurnosti prenosnih sistema, nakon detekcije mogućih problema, jeste identifikacija upravljačkih akcija, koje će eliminisati preopterećenja [4]. Akcije mogu biti preventivne ili korektivne, u zavisnosti od toga da li se primenjuju pre ili posle pojave stvarnog ispada. Preventivne akcije se primenjuju kako bi sistem nakon ispada imao minimalna preopterećenja. Korektivne akcije definitivno eliminišu preostala preopterećenja i vraćaju sistem u normalni radni režim. Blok dijagram $N-1$ proračuna sa uvažnim akcijama prikazan je na slici 3.1.

3.1. Identifikacija korektivnih mera

Korektivne akcije se mogu identifikovati na osnovu proračuna stacionarnih tokova snaga po granama i topologije sistema, dakle iskustveno ili „nagađanjem“. Iako se na ovaj način ne mogu odrediti „najbolje“ korektivne akcije, one su u najvećem broju slučajeva dovoljno efikasne za eliminisanje preopterećenja.



Slika 3.1 - $N-1$ sa korektivnim merama

Softveri koji se bave identifikacijom popravnih akcija uglavnom su bazirani na problemu optimalnih tokova snaga, koji pored jednačina snaga injektiranja, sadrže i

nejednačine kojima su obuhvaćena pogonska ograničenja, kao i kriterijumsku funkciju koja je formulisana tako da rešenje zadovoljava minimalne troškove proizvodnje, minimalne gubitke aktivne snage, minimalan broj potrebnih akcija i sl.

Osnovni OPF problem se može modifikovati tako, da se u postojeći skup (ne)jednačina inkorporiraju nejednačine koje će opisati sigurnosna ograničenja, dobijajući time SCOPF (*security constrained OPF*) problem. SCOPF predstavlja kompleksan, nelinearan problem, koji se mora rešavati iterativnim postupkom. Iako on daje najpreciznije rezultate, na mrežama velikih dimenzija, vreme potrebno za proračun je često neprihvatljivo, pogotovo u realnom vremenu, te je potrebno napraviti kompromis. Radi smanjenja vremena proračuna, može se koristiti DC proračun, koji predstavlja linearan problem i daje dobru aproksimaciju tokova aktivnih snaga i opterećenja vodova. Ukoliko je potreban uvid i u napone i tokove reaktivnih snaga, može se dodati nezavisni skup jednačina, dobijajući time brzi, raspregnuti OPF [5].

Dodatnu pomoć u pronalasku korektivnih akcija mogu dati faktori osetljivosti, koji predstavljaju odnos priraštaja izlazne promenljive, usled male promene ulazne promenljive, uz pretpostavku da su ostale veličine konstante, pa se njihovo izračunavanje svodi na proračun parcijalnih izvoda [1].

Faktori osetljivosti aktivnih snaga (*PTDFs – Power transfer distribution factors*) mogu biti:

- Faktori preraspodele tokova aktivne snage P_l grane $l = 1, 2 \dots L$, pri promeni snage injektiranja P_i čvora $i = 1, 2 \dots N$. Do promene snage injektiranja može doći usled promene angažovanja ili potpunog ispada proizvodnih jedinica ili potrošača. Definišu se kao [1]:

$$a_l^i = \frac{\Delta P_l}{\Delta P_i} \dots \dots \dots (4.1)$$

gde su ΔP_l i ΔP_i mala promena toka aktivne snage grane l i mala promena aktivne snage injektiranja čvora i , respektivno. Pozitivne vrednosti a_l^i znače da porast snage injektiranja u čvoru i izaziva porast toka snage kroz granu l . Negativne vrednosti a_l^i znače da porast snage injektiranja u čvoru i izaziva pad toka snage kroz granu l . Ovi faktori se koriste kada je preopterećenje elementa potrebno rešiti izmenom proizvodnje jednog ili više generatora.

- Distribucionni faktori promene toka aktivne snage P_l grane $l = 1, 2 \dots L$, pri promeni toka snage P_k grane $k = 1, 2 \dots L$. Do promene toka kroz granu k može doći usled izmene topologije, delovanja na sklopke regulacionih transformatora ili ispada prenosnog voda / transformatora iz pogona. Definišu se kao [1]:

$$d_l^k = \frac{\Delta P_l}{\Delta P_k} \dots \dots \dots (4.2)$$

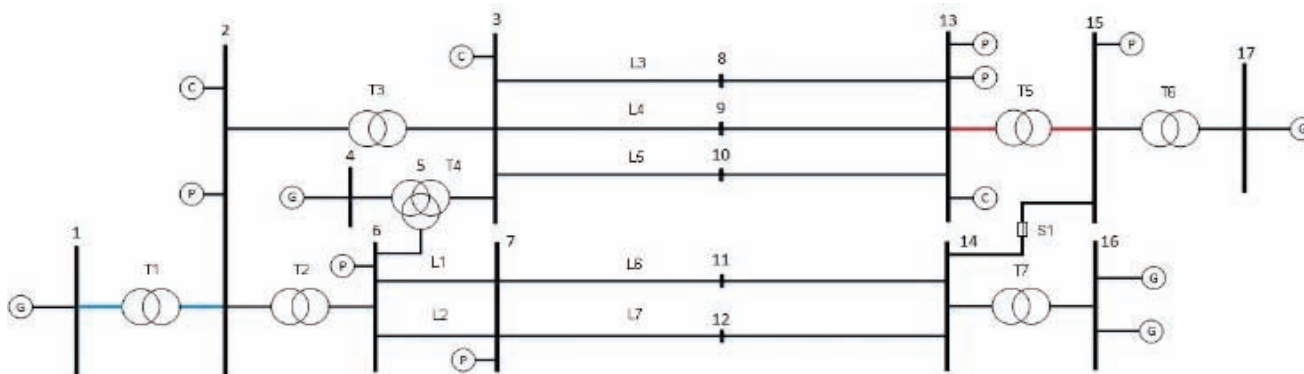
Ovi faktori se koriste pri identifikaciji popravnih akcija tipa izmene pozicija regulacionih sklopki ili promene statusa uključenosti rasklopne opreme.

3.2 Jeftine i skupe korektivne akcije

Imperativ u eksploataciji EES je pružanje što kvalitetnijeg napajanja, ali ne po svaku cenu. Sa poboljšanjem kvaliteta, troškovi održavanja eksponencijalno rastu. Sa niskim kvalitetom plaćaju se veliki penali korisnicima. Troškovi održavanja, između ostalog, obuhvataju i cenu primene korektivnih akcija. Zato je pogodno za svaki identifikovani paket akcija uraditi ekonomsku analizu, čime se određuje isplativost njihove primene. Osim toga, prilikom odabira akcija potrebno je uvažiti njihovu brzinu delovanja. Najčešće korektivne mere su date u tabeli 3.1 [4].

Tabela 3.1 Korektivne akcije

Akcije	Brzina delovanja	Cena
Predefinisane automatske akcije (smanjenje/isključenje generacije ili redistribucija/isključenje potrošnje usled ispada)	Milisekunde	Bez troškova
Izmena topologije delovanjem osigurača	Minuti	Bez troškova
Izmena topologije delovanjem prekidača	Minuti ili sati	Bez troškova
Promena pozicije sklopke transformatora	Minuti	Bez troškova
Promena pozicije PST	Minuti	Bez troškova
Uključenje/isključenje kondenzatora/zavojnice	Minuti	Bez troškova
Redistribucija potrošnje	Minuti	Bez troškova
Izmena proizvodnje reaktivne snage generatora	Minuti	Jeftina
Pokretanje/zaustavljanje generatora	Minuti ili sati	Jeftina
Redispečing u domaćoj mreži	Minuti	Jeftina
Redispečing u domaćoj mreži i susedima	Minuti	Jeftina
Izmena plana isključenja elementa	/	Skupa
Povratak u pogon isključenog/ispalog elementa	Više sati	Skupa
Ograničenje snage razmene sa susedima	/	Skupa
Izmena ugovorene snage razmene sa susedima	/	Skupa
Isključenje/sečenje potrošača	Minuti	Skupa



Slika 4.1 – Konfiguracija test mreže

4. VERIFIKACIJA KOREKTIVNIH AKCIJA

Izvršena je analiza sigurnosti na primeru test mreže prenosnog dela EES Belgije i Holandije (slika 4.1). Ova mreža ima 17. čvorova, pri čemu 5 čvorova predstavljaju fiktivne čvorove interkonektivnih vodova. Belgijski deo mreže je napajan u čvoru 1, gde je prikačen generator, dok se holandski deo mreže napaja u čvorovima 16 i 17. Čvor 17 je izabran kao balansni čvor. Simuliran je ispad blok transformatora T_1 , što predstavlja ujedno i izolovanje generatora u čvoru 1. Manjak generacije u belgijskoj prenosnoj mreži, uzrokuje dodatno angažovanje generatora u balansnom čvoru 17. Na slici 4.1 plavom bojom je predstavljen element u ispadu, dok su preopterećeni elementi predstavljeni crvenom bojom. Preopterećenje transformatora T_5 iznosi 110%.

Tabela 4.1 – Preopterećenja transformatora

Transformator	Opterećenje (%)
T_2	4.8
T_3	60.7
T_4	21.7
T_5	62.4
T_6	71.6
T_7	26.5

Jedini transformator u mreži sa mogućnošću zakretanja faznog stava jeste transformator T_1 . Regulacijom pozicije sklopke može se upravljati tokovima snaga u „gornjem“ delu mreže. Međutim, regulacione mogućnosti ovog transformatora su nedovoljne, s obzirom da u krajnjem položaju sklopke opterećenje kritičnog transformatora iznosi 102.5%. Izmena snaga proizvodnje u ovom slučaju takođe nije moguće, jer je element u ispadu izolovao generator G_1 . Preostaje otvaranje prekidača S_1 , pri čemu su tokovi snaga po transformatorima u tom slučaju prikazani u tabeli 4.1.

5. ZAKLJUČAK

Provera statičke sigurnosti predstavlja jedan od bazičnih proračuna u analizi elektroenergetskih sistema. Njeni rezultati su liste ispada, pri kojima dolazi do problema u mreži. Za takve ispade, definišu se upravljačke akcije (preventivne ili korektivne) čijim delovanjem se eliminišu problemi. Da bi se potvrdila efikasnost identifikovanih akcija, potrebno je izvršiti njihovu verifikaciju u

simulacionom modu. U slučaju stvarnog ispada, vrši se odabir paketa akcija na osnovu potrebne brzine delovanja i troškova njihove primene. Prema CGMES v2.5.2. standardu, postoje tri tipa akcija: izmena proizvodnje generatora, promena pozicije regulacione sklopke transformatora i promena statusa uključenosti rasklopne opreme.

6. LITERATURA

- [1] M.S.Čalović, A. T. Sarić: *Eksploatacija elektroenergetskih sistema u uslovima slobodnog tržišta*, 1999.
- [2] Qin Wang, Aimin Yang, Fushuan Wen: *Risk-based security-constrained economic dispatch in power systems*, 2013.
- [3] V.Strezoski: *Analiza elektroenergetskih sistema II*, skripta, Fakultet Tehničkih Nauka, Novi Sad, 2010.
- [4] Entso - e: *Supporting Document NC on Operational Security*, Brisel, 2013.
- [5] Umbrella Project: *Risk-based assessment concepts for system security*, 2012.

Kratka biografija



Kandidat Dušan Stanivuković je rođen 12.3.1989. u Sremskoj Mitrovici. Master rad je odbranio 2015. godine na Fakultetu tehničkih nauka u Novom Sadu iz oblasti Elektrotehnike i računarstva.

IDENTIFIKACIJA I AUTORIZACIJA BAZIRANA NA CLAIM-OVIMA U SMART GRID SISTEMIMA**IDENTIFICATION AND AUTHORIZATION IN SMART GRID BASED ON CLAIMS**Radomir Marković, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu su analizirane mogućnosti sistema za identifikaciju i autorizaciju na bazi claim-ova u Smart Grid okruženju. Podešeno je eksperimentalno okruženje i razvijena softverska rješenja za analizu rešenja baziranih na claim-ovima. Analizirana su različita scenarija upotrebe i uočene su pozitivne i negativne strane ovog mehanizma identifikacije i autorizacije. Utvrđeno je da identifikacija i autorizacija na bazi claim-ova ima jasne prednosti na polju rada sa korisnicima koji pristupaju sa Interneta i kod sprege dva bezbednosna domena, npr. između dve kompanije, ili između SCADA i poslovne mreže u okviru jedne kompanije.

Abstract – The goal of this paper is to analyze the applicability of claims-based identification and authorization in Smart Grids. A test environment was set up and a software solution was developed for the analysis of claims-based security mechanisms. Various usage scenarios were analyzed and the advantages and disadvantages of claims-based identification and authorization were discussed. It was shown that a claims-based solution has clear benefits for users who access a system from the Internet, as well as in systems with two security domains, e.g. between two companies or between the SCADA and business network.

Ključne reči: identifikacija, autorizacija, claims, Smart Grid

1. UVOD

Najčešće upotrebljen način autentifikacije korisnika, servisa i računara je preko upotrebe lozinke. Održavanje i skladištenje tih lozinki predstavlja izazov kako tokom administracije, tako i prilikom svakodnevnog korišćenja. Koji god sistem autentifikacije bio korišćen podaci o korisnicima se moraju održavati svežim i dopunjavati. U javno dostupnim sistemima to uglavnom obavljaju sami korisnici, dok u kompanijskim sistemima zatvorenog tipa, taj teret najčešće pada na sistem administratore. Kako broj aplikacija u jednom sistemu raste, tako raste i potreba za dodatnim informacijama o korisnicima. Česta je potreba za prijavljivanjem na sisteme više puta, često sa različitim kredencijalima, tj. korisnik mora da pamti korisnička imena i lozinke za razne aplikacije i sisteme. Mnogi korisnici ne mogu da zapamte sve lozinke, što dovodi do bezbednosnih rizika, npr. zapisivanje lozinke na papir ili postavljanja iste lozinke na različitim sistemima.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Imre Lendak, docent.

Kada se na sve ovo doda da različite organizacije moraju međusobno da sarađuju i da pruže pristup svojim aplikacijama korisnicima iz drugih organizacija, ovaj problem postaje ozbiljan, jer organizacije pored održavanja podataka i naloga svojih korisnika, moraju da vode računa i o korisnicima drugih organizacija. Korisnici u drugim organizacijama često mijenjaju poziciju ili čak napuste organizaciju, pa se sinhronizacija bezbednosnih informacija dodatno komplikuje.

Pristup u ovom radu u borbi sa gore navedenim problemima biće upotreba sistema za identifikaciju i autorizaciju korisnika bazirana na tzv. *claim*-ovima. Identifikacija i autentifikacija se prebacuje na servis za izdavanje i rad sa *claim*-ovima, tako da mi u aplikaciji ne moramo implementirati, niti održavati taj dio, već se samo oslanjamo na već postojeće servise. Korisnik dobija određene *claim*-ove kojima naša aplikacija vjeruje i na osnovu njih vršimo autorizaciju. Sve se zasniva na međusobnom povjerenju između servisa za izdavanje *claim*-ova i aplikacija ili sistema koji koriste te *claim*-ove. Dodatna prednost ovakvog rešenja je da neće biti nikakvih problema ako se promjeni način pristupa aplikaciji, npr. ako sistem pređe na prepoznavanje pametnom karticom umjesto korisničkog imena i lozinke, sve će i dalje funkcionisati bez potrebe za bilo kakvom promjenom na aplikaciji.

2. OPIS PROBLEMA

Smart grid je električna mreža koja koristi informacione i komunikacione tehnologije da prikupi i automatski djeluje na osnovu informacija o proizvodnji i potrošnji, u cilju unapređenja efikasnosti, pouzdanosti, ekonomičnosti i održivosti proizvodnje i distribucije električne energije [8].

Problem prepoznavanja korisnika i dodijele prava istima je jedan od najstarijih problema u računarskim naukama, to proizilazi iz svakodnevnih problema prepoznavanja ljudi i njihovih osobina u svakodnevnom životu. Ovi problemi se javljaju u svim savremenim sistemima, a jedan od predstavnika tih sistema jeste *Smart Grid*.

Postoji pregršt načina i mogućnosti da se rješavaju ovi problemi. Potrebno je vagati između prednosti jednog načina autentifikacije u odnosu na druge, jer da postoji jedan univerzalni i najbolji način drugi se ne bi ni razmatrali. Ova različitost i razne implementacije identifikacije, autentifikacije i autorizacije korisnika komplikuju kako razvoj, tako i održavanje sistema. Autor je tokom istraživanja analizirao sledeće probleme u sistemima za identifikaciju, autentifikaciju i autorizaciju korisnika:

- Nedostatak informacija o korisnicima,
- Dupla administracija i
- Razdvajanje biznis logike i autorizacije.

Nekada nam o korisniku treba više informacija nego što možemo da dobijemo iz korisničkog imena, lozinke i eventualno grupe kojoj korisnik pripada. Šta ako nam treba broj telefona korisnika u aplikaciji ili ako nam treba kontakt od nadređenog za zaposlenog iz neke kompanije? Npr. slučaj kada želimo da kontaktiramo nadređenog ako je osoba koja nam treba na odmoru ili odsutna iz kompanije, da bi našli zamjenu.

Takođe veliki izazov predstavlja kada našoj aplikaciji treba da dopustimo pristup korisnicima iz druge organizacije. Kao rešenje možemo da svakom od korisnika iz druge organizacije koji imaju potrebu da pristupaju našoj aplikaciji dodijelimo korisnički nalog samo za tu potrebu. Tu nastaje problem duple administracije korisnika, moramo da vodimo računa o korisnicima iz druge organizacije. Problem nastaje ako zaposleni u drugoj organizaciji dobije otkaz, ako promjeni poziciju u organizaciji ili neke druge atribute. Sve su to problemi koji se javljaju u ovom svakodnevnom scenariju kada nekim servisima ili aplikacijama jedne organizacije treba da pristupaju korisnici partnerske organizacije.

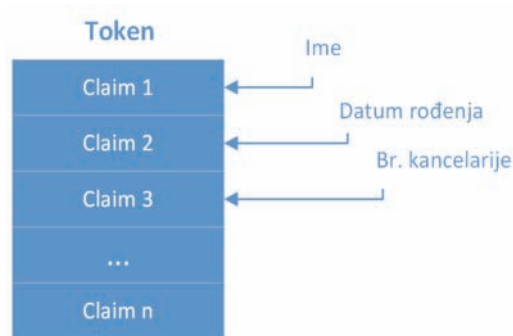
3. CLAIMS-BASED ARHITEKTURA

Claims-based bezbednosna arhitektura nam dozvoljava da delegiramo autentifikacionu logiku nekom drugom „entitetu“. Ovaj drugi „entitet“ garantuje aplikaciji da je korisnik autentifikovan. Pored dokaza da je korisnik autentifikovan, dobijamo i informacije o korisniku u obliku *claim*-ova koji će aplikaciji omogućiti da izvrši autorizacione odluke.

Osnovni pojmovi *claims-based* arhitekture su:

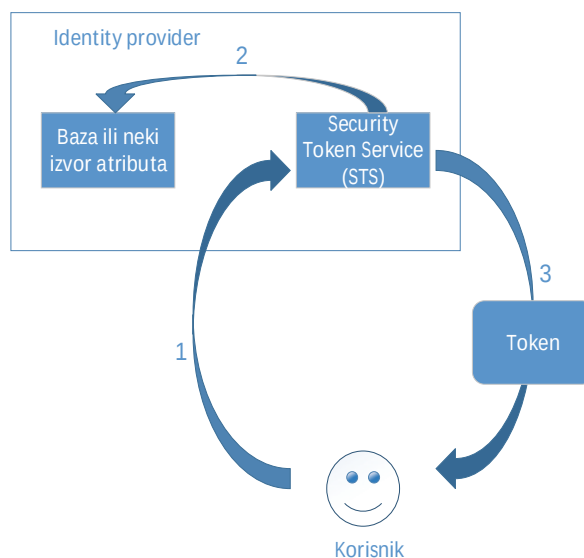
- Subjekat ili korisnik predstavlja entitet koji treba da bude autentifikovan. On može biti korisnik koji pristupa aplikaciji ili neka druga aplikacija koja želi da pristupi našoj aplikaciji.
- *Relaying party* je aplikacija ili web servis koji želi da delegira sa sebe autentifikacionu logiku.
- *Identity provider* izdaje, upravlja i verifikuje identitete entiteta koji pristupaju aplikaciji ili servisu.
- *Claims-based* identitet predstavlja način da aplikacije prikupe informacije o identitetu korisnika iz njihove unutrašnje organizacije, iz drugih organizacija i na internetu [2].
- *Claim* predstavlja izjavu koju neki subjekat izjavi o sebi ili o nekom drugom subjektu. Ova izjava može biti ime, identitet, ključ, grupa, privilegija, ili mogućnost. *Claim*-ovi se izdaju od strane *Identity Provider*-a. Na osnovu sadržaja *claim*-ova aplikacija vrši autorizacione odluke [1].
- *Token* predstavlja kriptografski potpisanu jedinicu koja sadrži jedan ili više *claim*-ova.

Na slici 1 je prikazan izgled *token*-a.



Slika 1: Izgled tokena

Security token service (STS) predstavlja servis koji kreira, potpisuje i izdaje bezbednosni *token*-e u skladu sa protokolima i standardima koji opisuju taj proces [3]. Na slici 2 se može vidjeti jedan od scenarija dobijanja identiteta tj. izdavanja bezbednosne *token*-a korisniku od strane STS. Korisnik se obraća STS-u i tu se autentifikuje, STS čita podatke o korisniku iz nekog izvora atributa, na osnovu toga se kreira *token* i vraća korisniku.



Slika 2: Izdavanje tokena korisniku

Kao STS koji se koristi u ovom radu izabran je *Active Directory Federation Services (ADFS)*. ADFS je aplikacija koja dolazi kao standardno podržana u *Windows Server* operativnom sistemu. Pruža pojednostavljen pristup i *single sign-on* pristup u aplikacijama koje se nalaze u jednoj organizaciji, cloud-u ili na internetu. Služi kao *identity provider*, tj. izdavalac *token*-a koji omogućava korisnicima *single sign-on* u okviru sistema partnerskih organizacija.

ADFS može da se konfigurise na dva načina, kao *Identity Service Provider* i kao *Relaying Party*. U prvom slučaju ADFS autentifikuje korisnika i izdaje im *security token*. Kada je konfigurisan kao *relaying party* ADFS prima dolazeće bezbednosne *token*e od drugih ADFS, validira ih i dopušta pristup aplikacijama koje imaju povjerenje u njega. *Token* se potpisuje digitalnim potpisom [5] i ukoliko je *token* modifikovan ili nije sa pravog izvora to se može lako utvrditi.

Velika prednost *claims-based* pristupa je to da se teret autentifikacije korisnika prebacuje sa aplikacije na STS. Pošto je to njegova primarna aktivnost, gledano sa strane bezbjednosti, dobijamo to da ne moramo uopšte

implementirati, niti održavati autentifikacionu logiku, već za to koristimo servise koji su komercijalni i za koje imamo odgovarajuću podršku. Što se tiče aplikacije dodatno olakšanje je to što aplikacija ne mora brinuti gdje su i kako podaci o korisnicima skladišteni i da li ima dovoljno informacija o njima. Krajnji korisnik također ima pogodnosti, jer na osnovu *claim*-ova možemo da im prikazujemo samo one funkcionalnosti koje im, kao pojedincima, pripadaju i trebaju za obavljanje njihovih aktivnosti.

4. IDENTIFIKACIJA I AUTORIZACIJA KORISNIKA

Cilj ovog rada je da se napravi eksperimentalno softversko okruženje i da se implementiraju pokazni primjeri za ispitivanje mogućnosti primjene sistema sigurnosti i bezbjednosti baziranih na *claim*-ovima u kontekstu kompleksnih softverskih sistema među koje spada i *Smart Grid* sistem. Pokazni primjeri su izvedeni na osnovu aktivnosti identifikovanih u produkcionim *Smart Grid* sistemima.

Pokazane su prednosti *claims-based* identiteta na dva primjera. U prvom primeru imamo korisnika, koji pristupa sistemu sa mobilnog uređaja i preko otvorenog interneta korišćenjem Web aplikacije (eng. *field client*).

Drugi primjer prikazuje prednosti i mane *claims-based* rešenja na slučaju dva bezbednosna domena, između dve *Smart Grid* kompanije, ili između SCADA i poslovnih mreže u okviru jedne kompanije. Analizirani su primeri dodavanja i brisanja korisnika, kao i promjena atributa korisnika.

4.1 Test okruženje

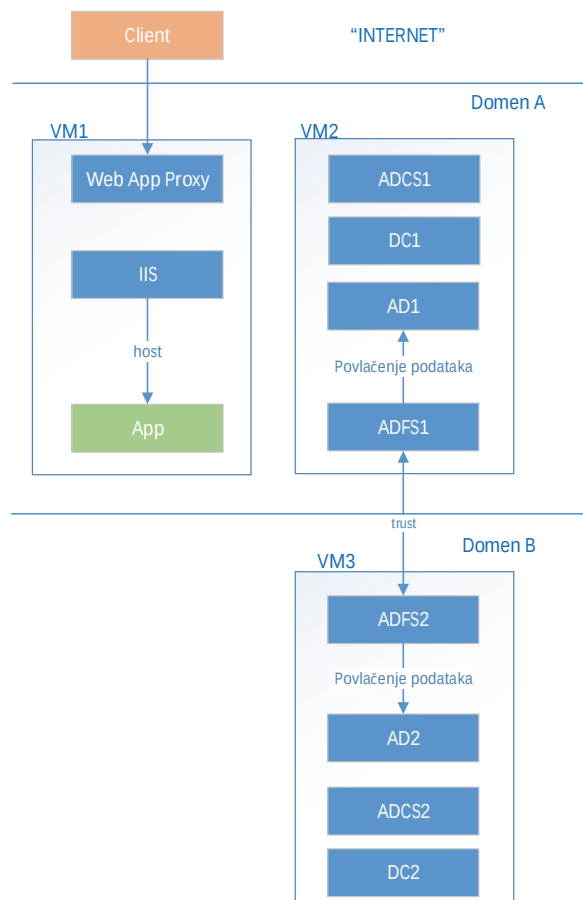
Testno okruženje je podešeno na jednoj fizičkoj mašini na kojoj se nalaze tri virtualne mašine koje simuliraju dva bezbednosna domena i otvoreni internet. Na slici 3 se vidi arhitektura sistema koja je u principu osnova za sve pokazne primjere, sa tim što neki primjeri koriste samo jedan dio arhitekture. Plavi pravougaonici predstavljaju postojeće servise koji su instalirani i konfigurisani, zeleni pravougaonik aplikaciju implementiranu u C#, a narandžasti pravougaonik klijenta.

Na prvoj virtualnoj mašini je postavljena aplikacija koja je implementirana u C# programskom jeziku. Aplikacija se publikuje preko IIS i *Web Application Proxy* koji pruža mogućnost *proxy* pristupa web aplikacijama unutar sistema, kako bi dozvolio pristup sistemu sa bilo kog uređaja izvan kompanijske mreže [7].

Druga virtualna mašina je podešena kao domen kontroler. Na njoj je instaliran *Active Directory* (AD) [6] i podešen *Active Directory Certificate Services* (ADCS) uz pomoć koga je kreiran javni sertifikat za potpisivanje *token*-a. Ovde je takođe instaliran i podešen ADFS, koji sarađuje sa AD i ADCS, tako što podatke o korisnicima povlači iz AD, a *token*-e potpisuje sa sertifikatom koji je izdat od strane ADCS.

Treća virtualna mašina je opcionalna i njeno prisustvo zavisi od eksperimentalnog scenarija. U slučaju dva domena, tj. dvije organizacije koristi se i ova virtualna mašina, a ona je u principu ista kao druga, samo što se radi o drugom domenu.

Prilikom primjera sa dva domena uspostavljeno je povjerenje između dva ADFS, tako što je izvršena konfiguracija oba ADFS da vjeruju *token*-ima izdatim od druge strane.



Slika 3: *Claims-based* pristup sa dvije organizacije

4.2 Pristup korisnika preko Interneta

U prvom primeru se razmatra pristup *Smart Grid* sistemu sa Interneta. Za realizaciju ovog eksperimenta dovoljan je prvi dio sa slike 3 koji obuhvata „internet“ i „Domen A“. U *Smart Grid* sistemu korisnik ili entitet pristupa sistemu preko otvorenog interneta koristeći mobilni uređaj na terenu (eng. *field client*). Entitet predstavlja bilo šta što se može jedinstveno identifikovati [4], u ovom slučaju to je aplikacija označena kao „Client“ na slici 3.

Korisnik pristupa aplikaciji kroz *Web Application Proxy*. Ukoliko se utvrdi da korisnik nije autentifikovan, dalje biva preusmjeren na ADFS. Korisnik upisuje svoje kredencijale i prijavljuje se na sistem. Ako je uspješno izvršena autentifikacija korisnika, ADFS čita potrebne podatke o korisniku iz AD. Korisnik se preusmjerava nazad na aplikaciju. Dalje se pristupa aplikaciji i vrši autorizacija korisnika na osnovu *claim*-ova koje sadrži *token* koji je izdat od strane STS. Korisnik može da vidi i koristi sve što mu dozvoljavaju prava koja se utvrde iz *claim*-ova unutar *token*-a.

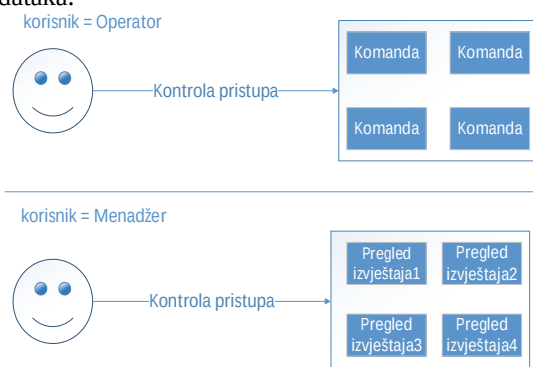
4.3 Dupla administracija

Zaposlenje novog radnika često podrazumeva unos njegovih podataka u bezbednosnim domenima partnerskih organizacija. Za to je potrebno stupiti u kontakt sa administratorima drugog sistema i napraviti korisnički nalog. Sa ovim dolazi do nagomilavanja redundantnih

podataka i do otežane administracije bezbednosnih informacija. Takođe dolazi do čestih promjena u pozicijama zaposlenih. Ako neki korisnik u partnerskoj organizaciji dobije otkaz, njegove podatke treba vrlo brzo obrisati iz svih domena. To često može da potraje, jer administratori iz jednog domena trebaju da kontaktiraju administratore drugog domena. Može doći do greške, može se desiti da administratori u jednom sistemu budu neodgovorni pa ne izvrše nalog koji je došao iz druge organizacije. Kao posljedica svega ovoga može doći do raznih problema.

Prilikom dodavanja korisnika svaka organizacija dodaje novog korisnika u svoj domen i svako je odgovoran za dodavanje svojih korisnika. Korisnik je dodat u domen B sa slike 3, i automatski može sa svojim kredencijalima da se prijavi na domen A. Unutar aplikacije domena A čitaju se *claim*-ovi iz *tokena* i vrši se kontrola pristupa aplikaciji. Ovo znatno olakšava saradnju između dvije organizacije, a olakšava život i samom korisniku. Korisnik bi trebao da ima različite lozinke na oba sistema, što bi za njega bio dodatni pritisak i umanjivalo bi njegovu produktivnost. Isto važi i za brisanje korisnika, korisnik se briše samo na jednom mjestu.

U domenu B sa slike 3 postoji korisnik koji pristupa servisima iz domena A. Taj korisnik je u ulozi Operatora i njegov posao je da upravlja nekim procesom izdavanjem komandi koristeći *web* aplikaciju iz domena A. Kada se korisnik prijavi na sistem, unutar aplikacije vidi komande i interfejs vezan za upravljanje i komandovanje. Upotrebom *Active Directory Users and Computers* alata atribut korisnika je u domenu B promjenjen sa Operatora na Menadžera. Sledećom prijavom korisnika iz domena B na aplikaciju iz domena A, korisnik više ne vidi komande za upravljanje i interfejs vezan za upravljanje i komandovanje. Prijavljeni korisnik sada vidi interfejs gdje može da pregleda izvještaje i da prati promjene vezane za njegovu novu ulogu, tj. ulogu Menadžera. Slika 4 ilustruje scenario prije i posle ažuriranja korisničkih podataka.



Slika 4: Ažuriranje korisničkih podataka

4.4 Diskusija

Korišćenjem bezbednosne arhitekture bazirane na *claim*-ovima rješavamo problem duple administracije i pružamo jednostavan način pristupa našim aplikacijama sa terena. Uvođenje ovog pristupa u već postojeći funkcionalan sistem može biti veoma komplikovan i naporan proces. Zavisno od veličine samog sistema, količine atributa vezanih za korisnike i broja aplikacija u sistemu može biti potrebno mnogo truda i rada prilikom prebacivanja na ovu bezbednosnu arhitekturu i ovaj pristup.

5. ZAKLJUČAK

Upotrebom *claims-based* bezbednosne arhitekture može se olakšati održavanje i upravljanje korisničkim nalogima. Identifikacija i autorizacija korisnika u sistemu takođe može znatno da se olakša i pojednostavi. Podaci o korisnicima iz partnerskih organizacija koji treba da pristupaju funkcionalnostima sistema druge organizacije biće u potpunosti održavani i upravljani od strane njihove organizacije.

U radu su predstavljena su dva primera primene ove paradigme. Prvi je rješenje za problem pristupa korisnika sa interneta, tj. korisnika sa terena u *Smart Grid* sistemu. Drugi primjer rješava problem duple administracije do koje može doći kada korisnici treba da „prelaze“ granicu između dva bezbednosna domena, npr. organizacija A i B.

Kao mogućnost daljeg razvoja javlja se mogućnost prebacivanja rešenja na *cloud*, tačnije integracija nekog *cloud* baziranog servisa za izdavanje *token*-a koji bi se našao u ulozi *ADFS*.

6. LITERATURA

- [1] Baier D. B., Bertocci V. B., Brown K. B., Densmore S. D., Pace E. P., Woloski M. W., 2011. A Guide to Claims-Based Identity and Access Control. 2nd ed. Microsoft Corporation.
- [2] Chappell D. C., 2011. "[Claims Based Identity for Windows](#)". Microsoft Corporation. Septembar 2015
- [3] Brown K., Sesha M., 2008-09 "Microsoft Windows Identity Foundation (WIF) Whitepaper for Developers" Microsoft Corporation.
- [4] International Telecommunication Union, "[NGN Identity management framework](#)". Septembar 2015.
- [5] Microsoft, "[Understanding AD FS Terminology](#)". Septembar 2015.
- [6] Microsoft, "[So What Is Active Directory](#)". Septembar 2015.
- [7] Microsoft, "[Web Application Proxy](#)". Septembar 2015.
- [8] Atlagić B., 2014. Softver sa kritičnim odzivom u elektroenergetskim sistemima.

Kratka biografija:



Radomir Marković je rođen 1991. godine u Sarajevu. Diplomski rad iz oblasti Elektrotehnike i računarstva odbranio je 2014. godine na Fakultetu tehničkih nauka u Novom Sadu. Na istom fakultetu odbranio je master rad iz oblasti Elektrotehnike i računarstva 2015. godine.

Performance analysis of resistive switching devices based on BaTiO₃ thin films

Natasa Samardzic¹, Tijana Kojic¹, Jelena Vukmirovic², Djordjije Tripkovic²,
Branimir Bajac², Vladimir Srdic² and Goran Stojanovic¹

¹Department of Microelectronics, Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia

²Department of Materials Engineering, Faculty of Technology, University of Novi Sad, Novi Sad, Serbia

Abstract. Resistive switching devices, memristors, have recently attracted much attention due to promising performances and potential applications in the field of logic and memory devices. Here, we present thin film BaTiO₃ based memristor fabricated using ink-jet printing technique. Active material is a single layer barium titanate film with thickness of ~100 nm, sandwiched between metal electrodes. Printing parameters were optimized aiming to achieve stable drop flow and uniform printed layer. Current-voltage characteristics show typical memristive behavior with pinched hysteresis loop crossed at the origin, with marked differences between High Resistive State (HRS) and Low Resistive State (LRS). Obtained resistive states are stable during numerous switching processes. The device also shows unipolar switching effect for negative voltage impulses. Variable voltage impulse amplitudes leads to the shifting of the energy levels of electrode contacts resulting in changing of the overall current through the device. Structural characterization have been performed using XRD analysis and SEM micrography. High-temperature current-voltage measurements combined with transport parameter analysis using Hall effect measurement system (HMS 3000) and Impedance Analyzer AC measurements allows deeper insight into conduction mechanism of ferroelectric memristors.

1. Introduction

Expanding interest for the devices with memristive behaviour is caused by favourable future applications such as in: new generation of memory device, artificial synapses, unconventional computing systems and overcoming of the Moore's law. Various materials and mechanism have gained memristive effect, however one of the most promising types are ferroelectric junctions, due robustness during read-out process, long retention time, possibilities for scaling down to few unit cells, chemical stability and high R_{OFF}/R_{ON} ratio [1].

Widely used ferroelectric memristive structures are based on BaTiO₃ thin films [2], while underlying switching mechanisms is explained as field induced charge redistribution at the interface metal/ferroelectric, which lowers the ferroelectric barrier height and enables conduction. Accumulation of the charge carriers at the interface contributes to the stability of the memorized

NAPOMENA:

- a) Ovaj rad proistekao je iz master rada Tijane Kojić. Mentor je bio prof. dr Goran Stojanović.
- b) Rad je prethodno publikovan na konferenciji IC-MAST, Mikonos, Grčka, septembar 2015.

states, wherein interface property, i.e. selection of the electrode material is critical issue for the current control [1].

In this paper, we present detailed performance analysis of Au/BaTiO₃/Ag memristive device. Ink-jet printing fabrication technique was chosen for deposition of active BaTiO₃ layer with advantages of additive, controlled and low-cost deposition. Memristive effect was identified through current-voltage measurement, while further electrical and structural analysis bring contribution in still open issues of underlying resistive switching effects mechanism.

2. Materials and Methods

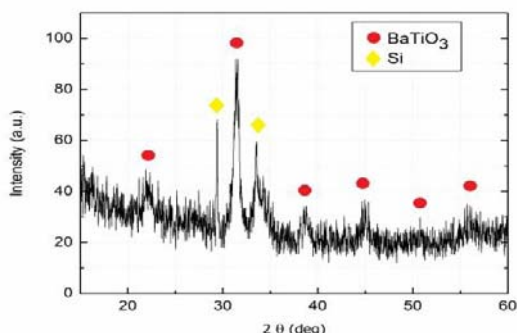
Active material is BaTiO₃ thin film deposited using Dimatix DMP 3000 ink-jet printer on Si wafer (SG 108, CrysTec) substrate on which was previously evaporated Au layer. Details about BaTiO₃ ink synthesis and properties have been reported elsewhere [3].

Stable drop flow during from 10 pl nozzles was obtained with trapezoidal actuation voltage waveform with amplitude of 26V and 1 kHz printing frequency. Deposition was performed in the clean room environment; class ISO 7 at room temperature. Drop spacing was set to 35 μ m, matching the half diameter of the printed drop on Si wafer substrate. Square-shaped patterns of BaTiO₃ one-layer film were well defined and reproducible. Post-processing step involves sintering at temperature of 700°C for 1 hour. Top electrode of the device is Ag two-component conductive paste (CircuitWorks® Conductive Epoxy), cured for 24 hours at room temperature.

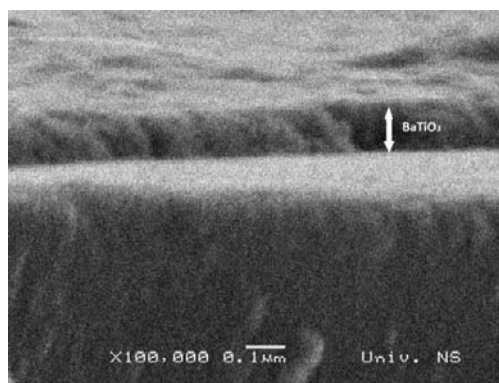
3. Results and discussion

Structural and morphology analysis

Structural XRD analysis (Rigaku XRD Miniflex 600), resolved characteristics peaks for BaTiO₃ crystalline phase (Figure 1a), with the presence of Si peaks from the substrate. SEM micrograph (JEOL JSM 6460 LV), Figure 1b, showed uniform layer of BaTiO₃ thin film with thickness $\sim$ 100 nm.



(a)



(b)

Figure 1. Structural analysis of Au/BaTiO₃/Ag resistive switching device a) XRD patterns b) SEM micrograph

3.2. Electrical analysis

The current-voltage characteristics were measured using Keithley 2410 High Voltage Source Meter in voltage source mode with 0.1V per second step voltage sweep. Bottom electrode Au was grounded during measurement, while voltages pulses are applied to the top (Ag) electrode with amplitude of $\pm$ 5V. Pinched hysteresis loop crossed at origin, the fingerprint of memristive effect, with bipolar switching behavior shown in Figure 2. Dissymmetry of the characteristics originates from Joule heating on the contact regions, as well as differences between barrier heights and metal work functions

at Au/BaTiO₃ and Ag/BaTiO₃ interfaces. Exponential behavior of current-voltage relation is most likely caused by dominant Fowler-Nordheim tunneling mechanism above critical electric field [4], while low voltage switching is controlled by barrier height modulation due to charge accumulation and dissipation at the interface. Namely, in the ON state barrier height is reduced, which lowers the resistance values, in contrary in the OFF state resistance increases as a consequence of charge dissipation.

Unipolar negative voltage pulses were applied from -1V to -5V, however only above 3V amplitude hysteretic I-V curve is visible, Figure 2 (inset). The relation between R_{OFF} and R_{ON} state, decreases as the voltage amplitude decrease, which is the resulted of less attracted carriers included in the current flow. Considering that BaTiO₃ acts as n-type of semiconductor, at room temperature [5] it is expected that main accumulated charge carriers included in the transport are electrons.

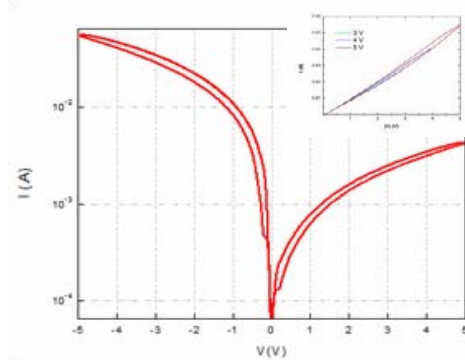


Figure 2. The current-voltage characteristics of Au/BaTiO₃/Ag printed memristor

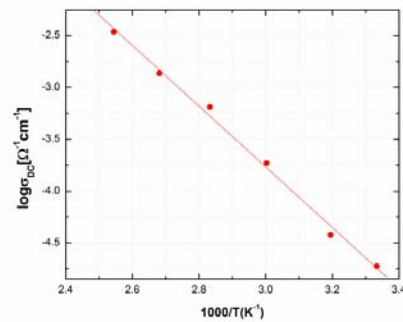
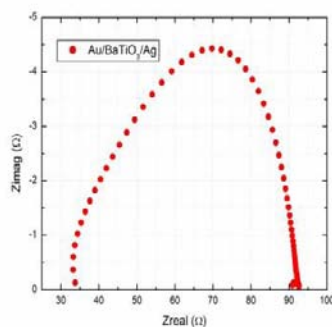


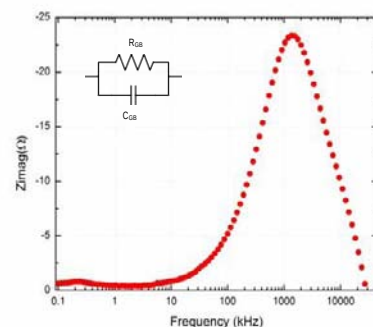
Figure 3. Temperature dependence of DC conductivity for BaTiO₃ memristive device

Specific conductivity of the Au/BaTiO₃/Ag sample was measured in temperature range from 300K-400K, results are shown in Figure 3. Results are in accordance with typical semiconductor behavior described with Arrhenius relation $\sigma_{DC}(T) = \sigma_0 \exp(-\Delta E/kT)$, where ΔE is activation energy, k -Boltzmann's constant, T -absolute temperature and pre-exponential factor [$\Omega \text{ cm}$]⁻¹. According to measured results and fitted curve, activation energy is estimated = 0.25 eV, corresponding to previously obtained value for BaTiO₃ thin films [6]. Activation energy indicates that film has shallow traps near conduction band edge, which could origin from the vacancies introduced during high temperature treatment or impurities [6].

Further electrical analysis was performed by means of impedance spectroscopy (Impedance Analyzer HP4194A), resulting in Cole-Cole diagram shown in Figure 4a, for frequency range from 100 Hz- 40 MHz.



(a)



(b)

Figure 4. Cole-Cole diagram (a) and $Z_{\text{imag}}=f(\text{frequency})$ (b)

As it is known from the literature [7], the semicircular in the Cole-Cole diagram formed from resistance and capacitance of the grain boundaries are much larger to then the resistance and capacitance of the bulk grains. Accordingly, dominant middle-frequency semicircular, Figure 4a, corresponds to grain boundaries contribution to the total conductivity of the BaTiO₃ sample.

Grain resistivity can be estimated according to graphical impedance spectroscopy analysis approach proposed in [8], as $R_g \approx 35 \Omega$, which is equal to the dislocation of the semicircle from the origin, while parallel resistance and capacitance of the grain boundary arc is estimated to $R_{GB} \approx 110 \Omega$ and $C_{GB} \approx 1 \text{ nF}$, respectively. Maximum frequency of the complete semicircular is evident in Figure 4b, is $\omega \approx 1.53 \text{ MHz}$. Accordingly grain boundary relaxation time amount $\tau \approx \frac{1}{\omega} = R_{GB} C_{GB} = 1.1 \cdot 10^{-7} \text{ s}$.

Additionally, we performed Hall effect measurement (HMS 3000) of one layer BaTiO₃ film deposited on Si wafer substrate connected in four points with the sample holder. Measurement results showed negative Hall coefficient $R_H = -5.91 \times 10^3 [\text{cm}^3/\text{C}]$, which confirms n-type conductivity of the sample, bulk concentration $N_b = 1.05 \times 10^{15} [\text{1/cm}^3]$ and mobility $\mu_n = 0.1 \times 10^{-2} [\text{cm}^2/\text{Vs}]$ in good agreement with results from literature [6].

4. Conclusion

In this paper we originally presented Au/BaTiO₃/Ag thin film memristor fabricated using ink-jet printing deposition technique. Current-voltage characteristics is in the form of pinched hysteresis, for both bipolar and unipolar impulse. Ratio of $R_{\text{OFF}}/R_{\text{ON}}$ state could be improved by increasing the contribution of ferroelectric phase in the BaTiO₃. High temperature DC conductivity measurement, revealed typical semiconductor behavior, with 0.25 eV shallow trap near the conductive band gap, probably due to introduced vacancies upon high-temperature treatment. Transport parameter measurement, proved that the sample is n-type semiconductor, thus electrons present the main charge carriers. Impedance spectroscopy graphical analysis revealed separate influence of the grain and grain boundaries in the overall conduction.

Acknowledgements

This work was supported by the Ministry of Education and Science, Republic of Serbia under the Project III-45021.

References

- [1] D. J. Kim et al., "Ferroelectric Tunnel Memristor," *Nano Lett.*, vol. 12, pp. 5697-5702, 2012.
- [2] X. Chen, C. H. Jia, Y. H. Chen, G. Yang and W. F. Zhang, "Ferroelectric memristive effect in BaTiO₃ epitaxial thin films," *Journal of Physics D: Applied Physics*, vol. 47, p. 365102, 2014.
- [3] Vukmirovic J, Tripkovic Dj, Bajac B, Kojic S, Stojanovic G and Srdic V, "Comparison of characteristics of barium titanate thin films prepared by inkjet printing and spin coating," *Processing and Application of Ceramics*, 2015.
- [4] V. Garcia and M. Bibes, "Ferroelectric tunnel junctions for information," *Nature Communications*, vol. 5, p. 4289, 2014.
- [5] J. Nowotny and M. Sloma, "Surface electrical properties of BaTiO₃ at elevated temperatures," *Solid State Ionics*, vol. 49, pp. 129-133, 1991.
- [6] Z. Xiong et al., "Room-temperature resonant quantum tunneling," *Nanoscale*, vol. 6, p. 13876, 2014.
- [7] Y.K. Vayunandana Reddy and D.Mergel, "Frequency and temperature-dependent dielectric properties of BaTiO₃," *Physica B*, vol. 391, pp. 212-221, 2007.
- [8] E. Barsoukov and J. Ross Macdonald, *Impedance Spectroscopy*. Hoboken, New Jersey.: John Wiley & Sons, 2005.

IDEJNO REŠENJE MIKROMREŽE U LABORATORIJU FTN CONCEPTUAL DESIGN OF A MICROGRID IN THE FTS LABORATORY

Dušan Grče, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj - U ovom radu opisan je koncept idejnog rešenja mikromreže koja bi se formirala u Laboratoriji za obnovljive izvore električne energije, a koja je smeštena u zgradi Mašinskih instituta Fakulteta tehničkih nauka (FTN) u Novom Sadu. Dat je prikaz tehničkih karakteristika kako postojeće opreme, tako i nove opreme koju treba nabaviti, u svrhu kompletiranja ovakve mikromreže i njene bolje funkcionalnosti u samostalnom radu. Napravljeni su proračuni proizvodnje električne energije iz korišćenih distribuiranih izvora unutar mikromreže. Na kraju su dati zaključci o opravdanosti izgradnje ovakve mikromreže.

Abstract - In the paper a conceptual solution of a microgrid to be formed in the Laboratory of Renewable Electrical Energy Sources, which is placed in the Mechanical institute building of Faculty of Technical Sciences (FTS) in Novi Sad is described. Technical characteristic of existing and new equipment which need to be purchased in order to complete the microgrid and its functionality in off-grid mode are presented. Cost calculations of the new equipment are made. Estimate of microgrid's distributed generators electrical energy production is made. At the end, some conclusions on the feasibility of the microgrid construction are given.

1. UVOD

Sve većom ekspanzijom korišćenja obnovljivih izvora, omogućuje se postepena zamena klasičnih izvora električne energije ovim ekološki prihvatljivijim. Posebno su interesantna rešenja sa fotonaponskim sistemima, vetrogeneratorima, malim hidroelektranama, zatim generatorima električne energije na bazi biomase itd. Ove sisteme moguće je povezati na različitim mestima u elektro-energetskom sistemu, tako da on dobija novu strukturu, potpuno različitu od do sada korišćene vertikalne organizacije. Sistemi skladištenja, odnosno akumulisanja električne energije koriste baterije, superkondenzatore, zatim pretvaranje u vodoničnu kombinaciju sa gorivnim ćelijama i druge mogućnosti. Ovi sistemi se takođe uključuju u moderne mreže, kao kratkotrajni izvori energije, ali i sa ulogom u balansiranju energetskih tokova između izvora energije i potrošača. Međusobno povezivanje ovih izvora korišćenjem najnovijih rešenja energetskih elektronskih pretvarača, obezbeđuje osnove za formiranje manjih, nezavisnih

jedinica u distributivnoj mreži, koje su sposobne da u nekom vremenu rade izdvojeno od glavne mreže i koje se nazivaju mikromreže. Budući da svaki od izvora, sistema za skladištenja ili potrošača karakterišu različita dinamička ponašanja, upravljanje mikromrežama predstavlja velik izazov i zahteva posebna rešenja. Dosadašnja istraživanja na području mikromreža AC i DC tipa usmerena su bila na obradu pojedinačnih tehnologija dobijanja električne energije iz obnovljivih izvora (sunce, vetar, biomasa), skladištenja energije, njihove međusobne interakcije i distribucije energetskih tokova unutar mikromreže u skladu sa potrebama potrošača (*demand side management*) [1,2,3]. Ona su ukazala na potrebu formiranja manjih, laboratorijskih sistema za istraživanje modela ponašanja mikromreža u realnim situacijama i u različitim scenarijima rada. Neke od takvih pilot mikromreža su već predstavljene u literaturi [4, 5,6].

U ovom radu biće prikazan primer koncepta idejnog rešenja mikromreže, koja bi se formirala u Laboratoriji za obnovljive izvore električne energije trenutno smeštene u zgradi Mašinskih instituta Fakulteta tehničkih nauka (FTN) u Novom Sadu. Razmatrana mikromreža ovog idejnog rešenja obuhvataće sledeće distribuirane izvore električne energije, sisteme za skladištenje i sisteme za upravljanje i kontrolu:

- Fotonaponska elektrana
- Mini vetrogenerator
- Kogeneraciono postrojenje
- Sistem za akumulisanje električne energije
- Kontroler mikromreže
- Grid emulator

2. MIKROMREŽE

Mikromreža predstavlja deo elektroenergetskog sistema koju čini skup nekoliko osnovnih tehnologija. Ova tehnologija, koja je poslednjih godina značajno uznapredovala, mikromreži omogućava kako autonoman rad tako i rad paralelan s NN mrežom. Osnovni elementi, odnosno tehnologija koje čini sve ovo mogućim je sledeća:

- Distribuirana proizvodnja električne energije
- Sistemi za skladištenje električne energije
- Sistemi za povezivanje na NN mrežu
- Sistemi za upravljanje radom mikromreže

Na slici 1 prikazana je tipična struktura mikromreže.

Distribuirana proizvodnja predstavlja proizvodnju električne energije smeštene na samoj lokaciji potrošača ili u njegovoj neposrednoj blizini. Ovo omogućava bržu i sigurniju isporuku električne energije potrošaču, obezbeđujući na ovaj način i značajne uštede u vidu smanjenja gubitaka usled prenosa električne energije na daljinu, odnosno izgradnje novih i proširenja postojećih sistema za prenos električne energije.

NAPOMENA:

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



Slika 1. Struktura mikromreže

3. IDEJNO REŠENJE KONCEPTA MIKROMREŽE U LABORATORIJU FTN-a

3.1 Projektni zadatak

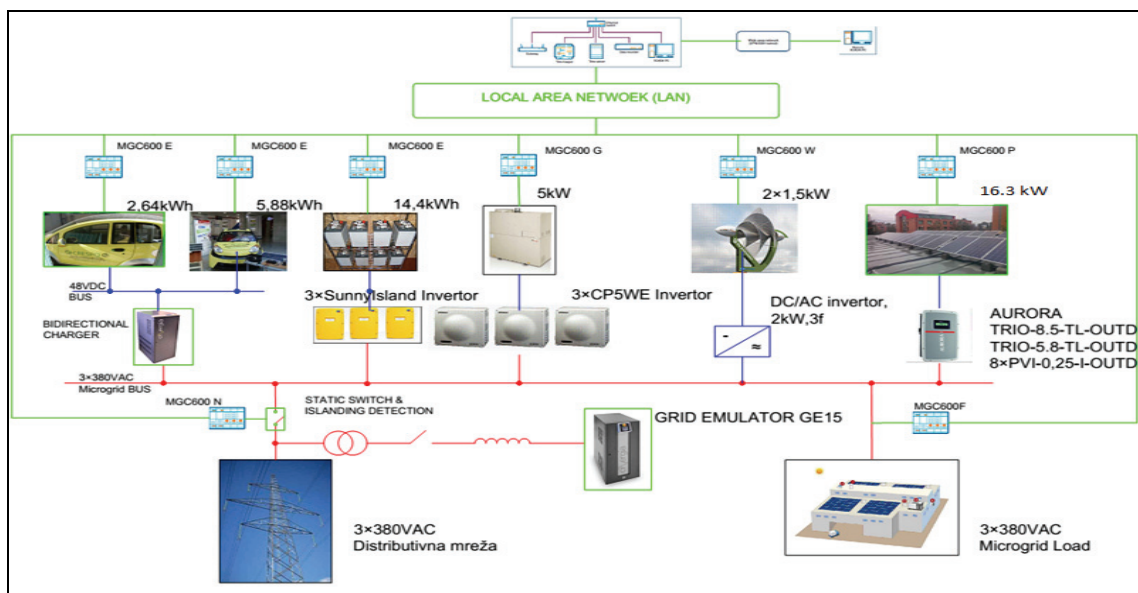
Potrebno je napraviti idejno rešenje koncepta mikromreže na lokaciji Laboratorije za obnovljive izvore električne energije smeštene u prostorijama Mašinskih instituta u sklopu Fakulteta tehničkih nauka u Novom Sadu. Kao osnovne sisteme i komponente za projektovanje mikromreže koristiti sisteme i komponente kojima raspolaže laboratorija. Koncept mikromreže treba da obuhvati jedan deo kompleksa, koji se posmatra kao kritično opterećenje sa prosečnom dnevnom potrošnjom od 5 kWh, odnosno prosečnom mesečnom potrošnjom od 3.600 kWh. Predviđeno je da mikromreža bude priključena na distributivnu električnu mrežu NN i da ima sposobnost da uz pomoć predviđenih distribuiranih generatora (DG), u vidu obnovljivih izvora energije, obezbedi dovoljno električne energije za ostrvski režim rada i eventualno toplotne energije u slučaju kogeneracije. To znači da se u slučaju poremećaja u distributivnoj mreži ili u slučaju pojave neke druge nestabilnosti ona može

odvojiti i nastaviti samostalan rad. U suprotnom treba predvideti da sistem DG svu proizvedenu električnu energiju preda u mrežu.

3.2 Tehnički opis

Projektovana mikromreža kombinuje različite vrste obnovljivih izvora energije za proizvodnju električne, ali i toplotne energije. Pored solarne i vetroelektrane u mikromrežu je uključeno i mikro kogeneraciono postrojenje (KGP), koje predstavlja distribuirani generator za proizvodnju i električne i toplotne energije, s tim da distribucija toplotne energije, koju proizvodi KGP (grejanje objekta, sanitarne vode, itd.), nije predmet ovog projekta. Potrošači u ovoj mikromreži su PC računari, deo osvetljenja u laboratoriji i mali ventilatori, a višak energije se može lagerovati u akumulatorskim baterijama za skladištenje ili za punjenje baterija električnih automobila. Struktura ovog idejnog rešenja mikromreže prikazana je na slici 2 i sastoji se iz sledećih DG-a, potrošača i distribuiranih sistema skladištenja (DS):

- Fotonaponska (FN) elektrana locirana na krovu mašinskog instituta (instalisan snage 16,3 kWp)
- Mali vetrogeneratori – 2 kom. (max 2 x 1,5 kW)
- Mikro-kogeneraciono postrojenje (5 kW_e+9,9 kW_t)
- Električna vozila (snage 2,64 kW+5,88 kW, sa dvostranim sporim punjačem)
- Akumulatorske baterije za skladištenje električne energije (ukupnog kapaciteta 14,4 kWh)
- Grid Emulator
- Električni potrošači (grupa od 5 PC računara – ukupna snaga 1 kW, osvetljenje 0,5 kW, mali ventilatori 0,5 kW, električna vozila 2, 6- 5,9 kW).



Slika 2. Koncept idejnog rešenja mikromreže

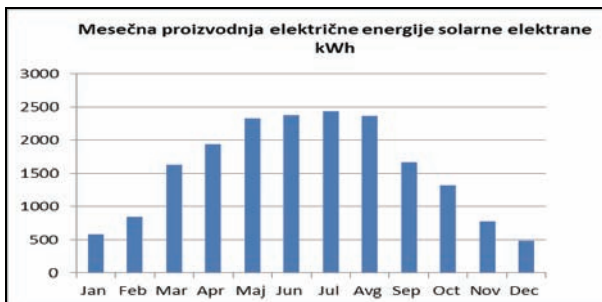
3.3 Energetski pokazatelji za obnovljive izvore

Na slici 3 dat je izgled postojeće FN elektrane od 16,3 kW_p, koja se nalazi na krovu mašinskog instituta u susedstvu laboratorije, a na slici 4 grafički prikaz prosečne proizvodnje električne energije [7]. Rezultati su dobijeni pomoću on-line softvera PVGIS, koji omogućava

predviđanje nivoa zračenja i proračun proizvedene električne energije na željenoj lokaciji i za datu snagu FN panela. Vidi se da se godišnje može očekivati oko 18.800 kWh „zelene“ energije.



Slika 3. FN elektrana 16,3 kWp [7].

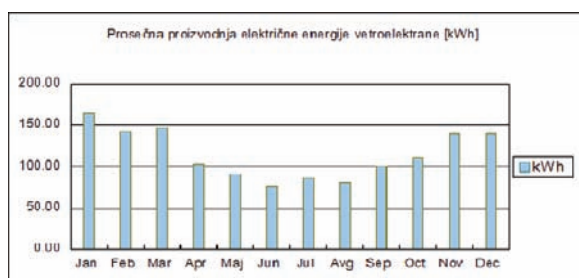


Slika 4. Prikaz mesečne proizvodnje solarne elektrane

Proizvodnja novo-projektovanih malih vetrogeneratora Liam F1, od kojih je jedan virtuelno prikazan na slici 5 [8], predstavljena je na slici 6. Proračuni brzine vetra dobijeni na osnovu srednje brzine vetra od 2,78 m/s [5] i inteziteta tokom godine za region Novog Sada. Može se očekivati oko 1.380 kWh godišnje.



Slika 5. Mali vetrogenerator Liam F1 [8].



Slika 6. Mesečna proizvodnja iz dva Liam F1.

Na slici 7 prikazana je ukupna procenjena mesečna proizvodnja električne energije iz oba izvora, tj. iz FN i vetroelektrane.

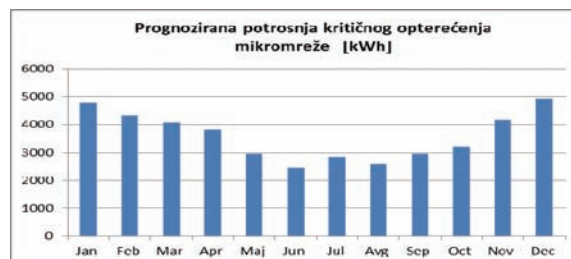
Vidi se da se može očekivati oko 21.560 kWh godišnje, s tim da je udeo vetrogeneratora od 5%-20% mesečno, odnosno 6,4% u proseku godišnje.



Slika 7. Ukupna proizvodnja solarne elektrane i vetroelektrane

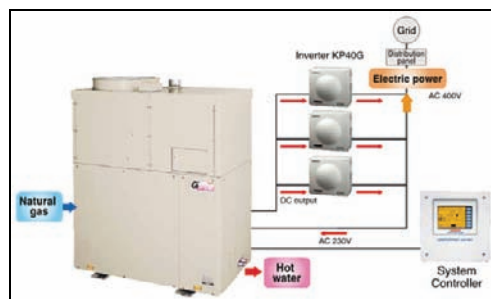
3.4 Energetski bilans

Raspodela očekivane potrošnje tokom godine prikazane na slici 8 dobijena je na osnovu procenutalne raspodele potrošnje tokom godine za poslovne korisnike prema podacima EPS-a pomnožene sa prognoziranim prosečnom potrošnjom posmatranog opterećenja od 3.600 kWh na mesečnom nivou. Na osnovu slika 7 i 8 može se zaključiti da proizvodnja solarne i vetroelektrane nije dovoljna za snabdevanje predviđenih potrošača mikromreže i da će morati da se koriste proizvodni kapaciteti mikro KGP.



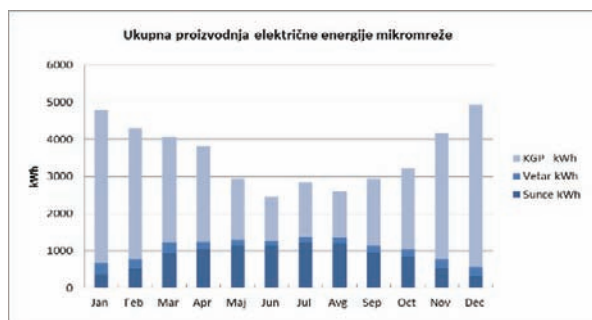
Slika 8. Prognozirana potrošnja opterećenja mikromreže

Za primenu u ovoj mikromreži odabrano je mikro kogeneracijsko postrojenje YANMAR CP5WG1 5 kWe+9,9 kWt, 3f, 400V koje za pogon koristi prirodni gas (slika 9). Gasni motor sa unutrašnjim sagorevanjem pokreće električni generator, dok se toplota, koja se oslobađa radom, koristi za podršku sistemu grejanja ili za pripremu potrošne tople vode u kombinaciji sa međurezervoarom tehničke vode. Varijabilni broj obrtaja motora omogućava optimalnu prilagođenost toplotne i električne snage trenutnim potrebama.



Slika 9. Blok šema kogeneracione jedinice [9].

Sada se može upotpuniti energetski bilans mikromreže, što je prikazano na slici 10. Vidi se nivo prosečne proizvodnje, koju mikromreža treba da obezbedi iz svojih distribuiranih izvora za pokrivanje sopstvenih potreba u toku godine. Sa slike 10 se vidi da najveći udeo imaju solarna elektrana i mikro KGP.



Slika 10. Ukupna proizvodnja električne energije mikromreže.

Može se zaključiti da mikro KGP ima najviše potrebe za proizvodnjom električne energije upravo tokom zimskog perioda kada je proizvodnja solarne elektrane mala, odnosno kada je potrebno obezbediti i najviše toplotne energije za grejanje. Tokom letnjeg perioda toplota koju proizvodi KGP može da se koristi za grejanje sanitarne vode. Konstrukcija motora i KGP omogućava varijabilnu proizvodnju, odnosno proizvodnju onoliko energije koliko je potrebno.

4. ZAKLJUČAK

Cilj ovog zadatka bio je da se obezbedi pouzdano i nesmetano napajanje, u ovom slučaju jednog dela potrošača, usled bilo kakvog poremećaja u elektroenergetskoj mreži, odnosno da se obezbedi mogućnost rada sistema u tzv. ostrvskom režimu. Može se videti da je moguće kombinovati proizvodnju iz različitih izvora energije, kako bi se obezbedilo dovoljno energije u određenom trenutku. U našem slučaju za zadatu potrošnju posmatranog opterećenja mikromreže proizvodnja električne energije iz solarne i vetroelektrane nije dovoljna, pa smo morali obezbediti dodatni izvor električne energije kako bi nadomestili potrebe za potrošnjom. Konstrukcija mikro KGP, odnosno njegovog gasnog motora je takva da može skoro jednako efikasno da radi i kada nije pod punim opterećenjem. Odnosno ima sposobnost varijabilne proizvodnje prema potrebi za električnom i toplotnom energijom. Ukoliko je potreba za toplotnom energijom veća, njegova proizvodnja se može povećati, a višak električne energije ukoliko ga ima predati u mrežu ili skladištiti u sisteme za akumulaciju.

Cena koja prati svaku vrstu nove i savremene tehnologije, pa i ove, predstavlja osnovni problem u razvoju i implementaciji mikromreža. Razvoj mikromreža je predmet proučavanja velikih država poput SAD, Japana, Kanade i razvijenih članica EU na nivou intezivnih laboratorijskih ispitivanja i sigurno je da će u skorije vreme naići na odobrenje za masovnu realizaciju pa samim tim i pristupačniju cenu.

5. LITERATURA

- [1] V. Mijailović, "Distribuirani izvori energije – principi rada i eksploatacioni aspekti", Akademska misao, Beograd, 2011.

- [2] J.J. Justo, F. Mwasilu, J. Lee, J.-W. Jung, "AC - microgrids versus DC-microgrids with distributed energy resources: A review", *Renewable and Sustainable Energy Reviews*, Vol.24, 2013, pp.387-405.
- [3] E. Prieto - Araujo, P. Olivella - Rosell, M. Cheah-Mane, R. Villafafila-Robles, O. Gomis-Bellmunt, "Renewable energy emulation concepts for microgrids", *Renewable and Sustainable Energy Reviews*, Vol.50, 2015, pp.325-345.
- [4] F. Huerta, J.K. Gruber, M. Prodanovic, P. Matatagui, T. Gafurov, "A laboratory environment for real-time testing of energy management scenarios", 3rd Int. Conf. on Renewable Energy Research and Applications, Milwaukee (USA) 19-22 Oct 2014, pp.514-519.
- [5] Z. Čorba, V. Katić, M. Mišić, G. Vamović, "Renewable Energy Sources Hybrid System", 15th Int. Symp. on Power Electronics Ee-2009, Novi Sad, 28-30 Oct. 2009, Paper No.T7-1.5, pp.1-5.
- [6] M. Roman-Barri, I. Cairo-Molins, A. Sumper, A. Sudria-Andreu, "Experience on the Implementation of a Microgrid Project in Barcelona", Catalonia Institute for Energy Research, Barcelona, 2010.
- [7] D. Rekanović, "Fotonaponska elektrana snage 16,3 kW na krovu zgrade Mašinskog instituta", Završni rad, Fakultet tehničkih nauka, Novi Sad, 2014.
- [8] Archimedes-wind turbines Liam F1, <http://dearchimedes.com/liam/>
- [9] Yanmar Co.- Yanmar Micro Cogeneration system, www.yanmar.co.jp/en/energy/

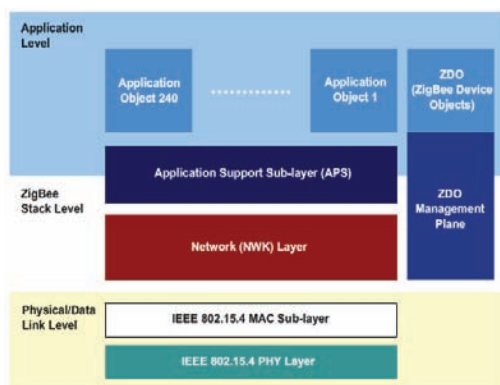
KRATKA BIOGRAFIJA:



Dušan Grče rođen je u 1984. god. Srednju elektrotehničku školu, završio je u Zrenjaninu, 2003 god, a diplomirao je na Fakultetu tehničkih nauka 2013. god. Master studije na smeru Elektroenergetika - Distribuirani elektroenergetski resursi završio je oktobra. 2015. god.



Vladimir Katić je rođen 1954. god. u Novom Sadu. Od 2002. god. redovni je profesor Univerziteta u Novom Sadu. Trenutno je prodekan Fakulteta tehničkih nauka i šef Katedre za energetsku elektroniku i pretvarače. Oblasti interesovanja su energetska elektronika, obnovljivi izvori električne energije, električna vozila i kvalitet električne energije.

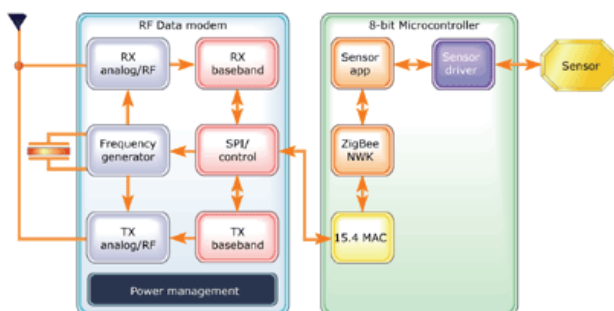


Слика 2. Архитектура ZigBee протокола

ZigBee намењен је бежичним сензорским мрежама са великим бројем сензорских чворова који имају релативно малу размену података. Формирање једне овакве мреже – сензорског поља се врши постављањем великог броја уређаја распоређених на малим растојањима које најчешће износи до десетак метара у области у која се посматра. Пошто се овакви чворови напајају из сопствених извора енергије као што су батерије, долази се до закључка да је животни век једног сензорског модула условљен енергетском ефикасношћу самог модула. Уколико дође до губитка једног од чворова из било ког разлога сензорска мрежа не само да остаје без података које је он прикупљао него и без могућности да се врши даље прослеђивање података преко тог чвора ка другим чворовима. Физички распоред уређаја у сензорском пољу није праволинијски, него више произвољан. У датој мрежи сви чворови имају могућност прикупљања података и њиховог слања ка циљу.

3.2. ZigBee уређај

ZigBee уређај са којим се жели остварити комуникација се састоји из дела за напајање, батерија или соларне плоче, микроконтролера чија је улога контрола и координација рада сензора и примопредајника, обрада података и имплементација мрежних протокола и протокола рутирања затим сензорске јединице, ствараоца података као и радио примопредајна јединице која има задатак да прослеђује како своје тако и туђе податке кроз сензорску мрежу

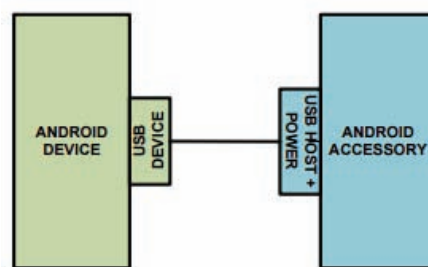


Слика 3. Блок шема ZigBee уређаја

4. USB ACCESSORIES ПРОТОКОЛ

Андроид режим отвореног приступа дозвољава повезивање периферних уређаја на Андроид платформу где је Андроид USB уређај а периферија USB домаћин. Овај протокол повезивања поседује следеће предности :

- Непостојање потребе за развојем посебних драјвера за периферни хардвер
- Елиминисање потребе за рутовањем уређаја да би се дала дозвола за учитавање драјвера
- Електрична енергија за напајање порта се обезбеђује од стране периферије омогућавајући да се батерија мобилног уређаја не троши када екстерни уређај буде прикључен



Слика 4. Андроид режим отвореног приступа

4. РЕАЛИЗАЦИЈА УРЕЂАЈА

Реализовани уређај користи микроконтролер Texas Instruments CC2530 за рад са ZigBee протоколом као и интегрисано коло Future Technology Devices International FT311D као мост између Андроид платформе и ZigBee уређаја.

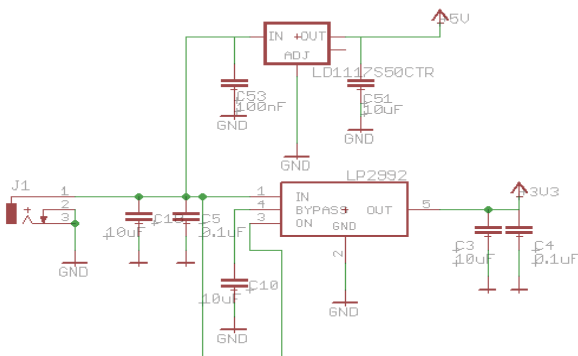


Слика 5. Изглед готовог уређаја

4.1. Физичка реализација уређаја

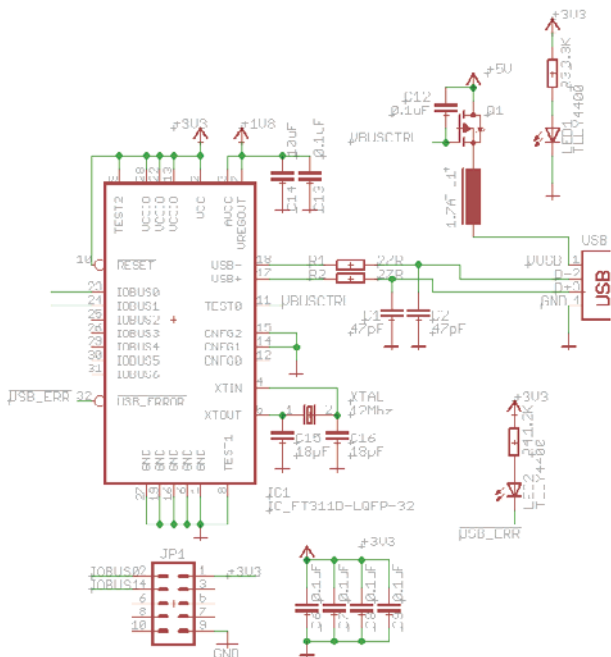
Напајање овог уређаја врши се помоћу два напонска регулатора. Помоћу напонског регулатора фирме STMicroelectronics, LD1117S50CTR врши се напајање USB-а односно напон добијен са излаза овог интегрисаног кола, који износи +5V се користи за реализацију USBAccessories протокола, по коме напајање USB порта врши уређај који је прикључен на Андроид платформу, на електричној шематској ознаци на којој се види да овај напон је VUSB. Струја коју на свом излазу може да реализује овај напонски регулатор износи до 800mA, па је самим тим обезбеђено и пуњење електричном енергијом батерије Андроид платформе. Електрична шема повезивања овог кола добијена је са интернет странице произвођача кола.

Напајање FTDI311D чипа добија се помоћу Texas Instruments LP2992 напонског регулатора. Он се најчешће употребљава код реализације преносивих уређаја као што су лаптопови, мобилни телефони, персонални дигитални асистенти, камере и тако даља. Напонског регулатора LP2992 на свом излазу даје између 1.5 V и 5 V, за напоне на улазу од 2.2 V до 16 V и струју до 250mA, што је више него довољно за чип који се користи у овом раду, у овој конфигурацији повезивања даје +3V3 на излазу што је и напон напајања самог FTDI311D чипа.



Слика 6. Електрична шема напајања

Електрична шема уређаја за комуникацију између Андроид платформе и ZigBee уређаја представљена је на доњој слици.



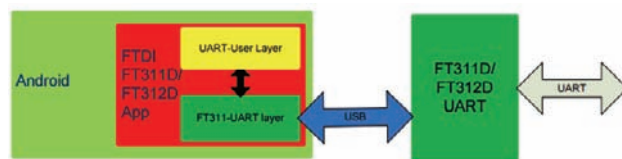
Слика 7. Електрична шема за повезивање са Андроид платформом

Након њиховог физичког повезивања врши се истовремено довођење напона са излаза напонског регулатора LD1117S50CTR од +5V на VUSB преко транзистора Q1, чија је улога да уколико је напајање мање од +5V онемогући успоставу везе блокирањем електронског кола FTDI311D. Затим +3V3 се води са излаза LP2992 на пине 13, 22, 28, 10 електронског кола FTDI311D. Од тог момента креће софтверско

повезивање које је објашњено у петом поглављу овог рада. На слици се може видети да је конфигурација чипа извршена довођењем пинова 14 и 15 на нулти потенцијал а остављање пина 12 практично нигде не прикљученог. Пин 32 служи за светлосну идентификацију грешке уколико до ње дође приликом повезивања уређаја са Андроидом. Повезивање овог чипа за ZigBee модулом је обављено преко улазно / излазних пинова 23 и 24 чија је улога у овој конфигурацији серијска комуникација. Напајање од +3V3 се преко JP1 користи за напајање ZigBee модула.

5. ПРОГРАМСКА ПОДРШКА

Програмска подршка за рад са уређајем реализована је у виду Андроид апликације која ће касније бити употребљена као тест за проверу рада комплектног уређаја. Апликација поседује две класе, као што се може видети на слици.



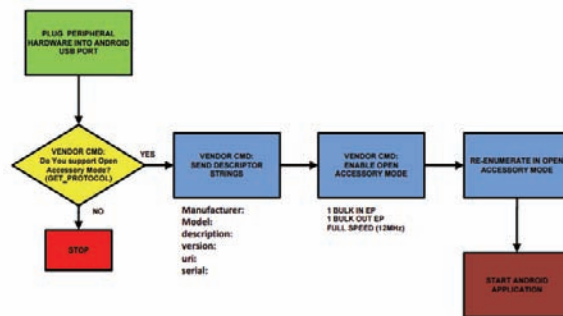
Слика 8. Структура Андроид тест апликације

Кориснички слој (User Layer) омогућава конфигурисање параметара преноса података помоћу серијске комуникације и унос података. Док се у UART Layer-у врши реализација USB Accessorie протокола као и одабир формата слања поруке, код ове реализације употребљава се хексадецимални формат.

Када је периферни уређај прикључена на Андроид платформу мора послати захтев Андроиду да утврди да ли Андроид уређај подржава режим отвореног приступа. Ово је Get_protocol команда.

Ако је одговор не онда се комуникација зауставља овде.

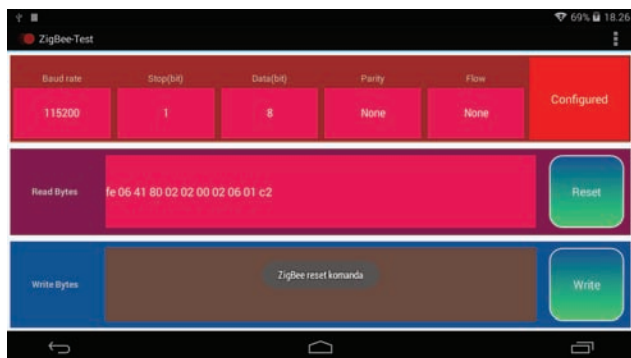
Међутим уколико је одговор потврдан периферија затим шаље јединствен стринг дескриптор да би се идентификовала. Када се изврши енумерација створиће се USB BULK IN и USB BULK OUT крајње тачке као и крајња контролна тачка.



Слика 9. Успостављање везе између Андроид платформе и периферног уређаја

6. ИСПИТИВАЊЕ И РЕЗУЛТАТИ

Управљачки део комплетног уређаја, Андроид апликација врши конфигурисање параметара серијске комуникације као и слања ресет поруке ка Zigbee уређају. Одговор Zigbee уређаја враћа се на екран односно врши се његов испис на екрану. Тестирање уређаја је извршено на неколико Андроид уређаја као што су Nexus player, Google ADT – 1, Alcatel PIXI7, Alcatel one touch, Asus, Samsung SM-T230 као и RK2030 set top box, код оних уређаја који поседују могућност рада у USB Accessorie режиму односно који подржавају OTG USB, а то су најчешће таблети и мобилни телефони, могућ је рад са овим уређајем док генерацијских старијих уређаја као што је set top box није јер или не подржавају софтверски или хардверски USB Accessorie протокол.



Слика 11. Слање ресет поруке ка ZigBee уређају и његов одговор



Слика 12. Уређај повезан са Андроид платформом

7. ЗАКЉУЧАК

Задатак овог рада био реализација уређаја који успоставио комуникацију између Андроид и Zigbee уређаја коришћењем USB Accessorie протокола затим писање Андроид тест апликације за проверу рада уређаја.

Да би се реализовао овај уређај било је неопходно стећи знање из области бежичних сензорских мрежа као и програмирање Андроид апликација. Сви задаци постављени спецификацијом овог рада су успешно решени. Даља разрада пројектованог уређаја је могућа усавањем Андроид апликације, додавањем различитих могућности за управљање радом Zigbee уређаја као и омогућавање, комуникације са другим Zigbee уређајима у циљу формирања сензорске мреже.

8. ЛИТЕРАТУРА

- [1] Vladimir Kovačević: *Logičko projektovanje računarskih sistema I –projektovanje digitalnih sistema*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2001
- [2] Asymmetric Encryption in Wireless Sensor Networks, Gustavo S. Quirino, Admilson R. L. Ribeiro, Edward David Moreno, DOI: 10.5772/48464
- [3] Home networking with Zigbee, Mikhail Galeev, April 20 2004
- [4] Drew Gislason „ZigBee Wireless Networking“ ZigBee alliance „ZIGBEE CLUSTER LIBRARY SPECIFICATION“ www.zigbee.org, 2015-10-19
- [5] Урош Јеремић: Инфраструктура за мобилно учење, Факултет техничких наука, Чачак, 2009
- [6] Future Technology Devices International Ltd, <http://www.ftdichip.com/Products/ICs/FT311D.html>,
- [7] Texas Instruments, CC2530, A true System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications, <http://www.ti.com/lit/ds/symlink/cc2530.pdf>, 2015-09-08
- [8] Урош Јеремић: Инфраструктура за мобилно учење, Факултет техничких наука, Чачак, 2009

Кратка биографија

Иван Иванчић, рођен у Новим Саду 1986. године. Дипломски рад – Бечелор из Електротехнике и рачунарства – Примењена електроника одбранио је 2013. Године. ivancic21000@gmail.com

Милош Живанов, доктор техничких наука, редовни професор на Факултету техничких наука у Новом Саду. Области рада: физика и моделовање полупроводничких компоненти, управљачка електроника, оптоелектроника, примењена електроника, електрична возила и геофизичка каротажна мерења. zivanov@uns.ac.rs

PROGRAMSKO OKRUŽENJE ZA KONVERZIJU RAZLIČITIH IZVORA PODATAKA U CIM/XML**SOFTWARE ENVIRONMENT FOR CONVERTING VARIOUS DATA SOURCES INTO CIM/XML**

Žolt Kanjo, Fakultet tehničkih nauka, Novi Sad

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je predstavljena analiza konverzije iz različitih vrsta izvora podataka u ciljnu CIM/XML format. Pod konverzijom se podrazumeva ceo ETL proces. Izvori podataka su: CSV datoteka, Microsoft Excel dokument, tabela u bazi podataka koja je dostupna ODBC interfejsom, kao i XML dokument. Implementirana je Microsoft Windows aplikacija za sprovođenje ETL procesa nad izvornim podacima u cilju kreiranja ciljnih CIM/XML dokumenata sa odgovarajućim CIM objektima, koji su definisani RDFS i XSD datotekama. Rezultat rada aplikacije predstavlja konzistentan model izgrađen na osnovu izvornih podataka.

Abstract – In this paper an analysis of the conversion of various data sources into CIM/XML is presented. In this case conversion means the entire ETL process. The data sources are: CSV files, Microsoft Excel documents, database tables with an ODBC interface and XML documents. A Microsoft Windows application is implemented for the execution of ETL process for these data sources in order to create the resulting CIM/XML documents with the corresponding CIM objects defined by RDFS and XSD schemas. The result of the application is a consistent model based on the data from data sources.

Cljučne reči: CIM (Common Information Model), XSD (XML Schema Definition), CSV (Comma Separated Values), ODBC (Open Database Connectivity), XML, ETL (Extract-Transform-Load), RDFS (Resource Description Framework Schema).

1. UVOD

Porastom primene informacionih tehnologija u elektroenergetskim sistemima dolazi do potrebe za korišćenjem standarda, koji na precizan način propisuju pravila predstavljanja elemenata elektroenergetske mreže i razmene podataka. Obezbeđivanje modela podataka nezavisnog od platforme, predstavlja osnovni cilj Common Information Model-a (CIM-a). U slučaju kada je u sistem sa tačno definisanim modelom potrebno integrisati postojeće podatke iz sistema sa različitim modelom, sprovodi se ETL proces. On se sastoji iz tri dela: Ekstrakcija, Transformacija i Učitavanje.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Milan Gavrić.

Ekstrakcija je proces izdvajanja podataka iz različitih izvora i platformi u oblik pogodan za dalju obradu. U većini slučajeva ona predstavlja najvažniji aspekt ETL-a, kako njeno ispravno izvršenje postavlja početno stanje, važno za uspeh narednih koraka.

U koraku **transformacije** se nad izdvojenim podacima izvršava skup pravila ili funkcija kao priprema za njihovo učitavanje u ciljno skladište podataka. Važna funkcija transformacije je čišćenje, koja kroz ovaj korak propušta samo podatke koji su validni po definisanim kriterijumima.

Učitavanje predstavlja smeštanje transformisanih podataka u ciljno skladište podataka, koje se definiše u skladu sa potrebama sistema [1].

Kako su sve opisane faze vremenski zahtevne njih je potrebno u što većoj meri izvršavati paralelno, da bi se postigle što bolje performanse. Razvijeno programsko okruženje (ETL alat) omogućava paralelno izvršavanje zadataka, kao i podršku za čitanje CIM profila iz RDFS i XSD datoteka, što mu je prednost u odnosu na analizirane alate. Implementirana Windows aplikacija sprovodi ETL proces, tako što sadrži komponente za izvršenje njegovih faza. Za fazu ekstrakcije koristi ulazne komponente odgovarajuće za podržane izvore, naime: CSV, XLS, tabela iz baze podataka sa ODBC interfejsom i XML. Za fazu transformacije sadrži komponente koje omogućavaju filtriranje, spajanje i samo transformisanje ekstrahovanih podataka. Za fazu učitavanja ima jednu komponentu koja je zadužena za smeštanje transformisanih podataka u jedini podržani ciljni format, naime u CIM/XML datoteku.

Aplikacija je izrađena u C# programskom jeziku, korišćenjem WPF (Windows Presentation Foundation) tehnologije. Korišćen je Windows 8.1 operativni sistem.

2. POSTOJEĆI ETL ALATI

Pre implementacije novog softverskog rešenja je potrebno istražiti i analizirati postojeća. U cilju istraživanja trenutnog stanja na tržištu (State of the Art) ETL aplikacija su testirani alati: Talend Open Studio, CloverETL Designer i FME Desktop.

Talend Open Studio je moćno i jednostavno programsko rešenje otvorenog pristupa (Open Source) za integraciju, migraciju i ETL podataka. On poseduje jednostavno grafičko okruženje za razvoj zadataka (Jobs) koje je bazirano na Eclipse alatu, lako se proširuje i lako se podešava u zavisnosti od potreba korisnika [2]. Zadaci su osnovni elementi za pokretanje. Njegove glavne komponente su metapodaci sa kojima se lako upravlja i

koji se međusobno jednostavno mapiraju. Metapodaci se koriste za ulazne, a takođe i za izlazne podatke.

CloverETL Designer je vizuelni alat za dizajniranje integracije podataka. Omogućuje kreiranje vizuelnih grafova ETL podataka. On se takođe bazira na *Eclipse* alatu [3]. Kao što su u prethodnom alatu bili zadaci, ovde su osnovni elementi grafovi. Grafovi su osnovni elementi za pokretanje. Na njih se mogu postavljati komponente koji se međusobno povezuju. Od skupa komponenata je moguće napraviti podgraf koji sakriva unutrašnju implementaciju skupa i ističe samo njegove ulaze i izlaze koji su potrebni korisniku.

FME Desktop je moćan ETL alat, koji podržava preko 335 formata podataka i sadrži stotinu funkcija za transformaciju strukture i vrednosti podataka [4]. Njegov osnovni element je radna površina (*Workspace*) na koji se dodaju elementi. Elementi imaju predefinisane ulazne i izlazne čvorove i koji se povezuju sa čvorovima drugih elemenata.

Svi testirani alati su se dobro pokazali u test scenarijima i uspeali su da izvrše zahtevane zadatke, sa određenim razlikama u implementaciji rešenja. Oni mogu značajno ubrzati i olakšati proces integracije podataka. Međutim, postoje razlike u mogućnostima i lakoći korišćenja tih alata. Jedini zadatak koji se ne može uraditi sa *CloverETL Designer* alatom je generisanje strukture izlazne datoteke direktno iz uzorka XML fajla. To je moguće uraditi jedino pomoću XSD šeme ili ručnim kreiranjem cele izlazne strukture.

FME Desktop se pokazao kao najmoćniji i najfleksibilniji alat, koji ima najmanje ograničenja i najviše funkcija, ali sa druge strane je daleko najteži za korišćenje i zahteva znatno veći napor za proučavanje dokumentacije od ostalih alata. *Talend Open Studio* i *CloverETL Designer* su veoma slični po funkcionalnostima, organizaciji komponenata i lakoći korišćenja. Ipak se *Talend Open Studio* pokazao za nijansu bolje u tim aspektima.

Po performansama se ovi alati znatno razlikuju. *CloverETL* se pokazao kao daleko najbrži alat. Sa povećanjem broja uzoraka njegova brzina raste, što pokazuje da je paralelizam dobro implementiran.

Drugi alat po brzini jeste *FME Desktop*, čija brzina ne zavisi od broja uzoraka. Daleko najsporiji alat od testiranih je *Talend Open Studio*. Njegova brzina drastično opada sa povećanjem broja uzoraka.

3. PREGLED PODRŽANIH ULAZNIH FORMATA

3.1. CSV

CSV datoteke se često koriste za razmenu podataka. One imaju strukturu tabele i podatke čuvaju u tekstualnom obliku. Svaki red u datoteci predstavlja jedan objekat, a svaki od njih sadrži polja, razdvojena zarezom ili nekim drugim separatorom.

Najčešći separatori su: zarez, tabulator, razmak, dvotačka i tačka sa zarezom. U prvom redu datoteke se mogu navesti imena polja, takođe razdvojena izabranim separatorom.

Ovaj oblik skladišta podataka se često koristi zbog njegove jednostavnosti i zbog njegove velike podrške od strane različitih alata i programskih jezika.

3.2. MICROSOFT EXCEL DOKUMENT

Microsoft Excel je alat za kreiranje tabela i za njihovo sačuvanje koristi *.xls* i novije *.xlsx* formate datoteka. XLS datoteka se bazira na binarnom pristupu i podatke skladišti direktno u tom obliku, dok XLSX implementira Open Office XML format. Jedan dokument može sadržati više tabela (*sheet-a*).

Excel tabele se vrlo često koriste i u elektroenergetskim sistemima za skladištenje kataloških informacija o uređajima mreže.

3.3. TABELA U BAZI PODATAKA SA ODBC INTERFEJSOM

Open Database Connectivity obezbeđuje standardni softverski interfejs za pristup bazi podataka sistema. On predstavlja univerzalni međusloj između baze podataka i aplikacije i dizajniran je tako da bude nezavisan od programskog jezika, baze podataka i operativnog sistema. To znači da je aplikacija koja implementira ODBC interfejs sposobna za komunikaciju sa svim bazama podataka koji podržavaju isti.

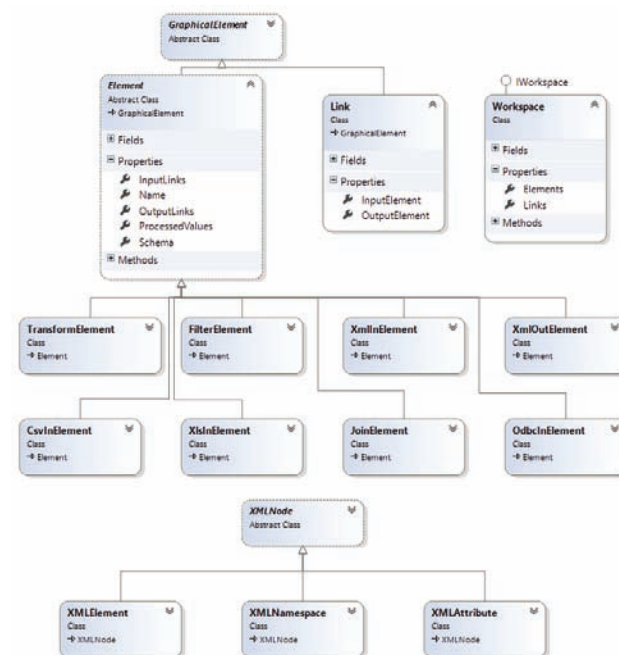
Različite baze podataka su sastavni deo elektroenergetskih sistema, iz čega proističe potreba za implementaciju ODBC konekcije od strane aplikacije koja je namenjena za ETL njihovih podataka.

3.4. CIM/XML

Obezbeđivanje modela podataka elektroenergetskih sistema nezavisnog od platforme, predstavlja osnovni cilj *Common Information Model-a* (CIM-a). Za ovako predstavljene podatke je neophodno omogućiti njihovu razmenu, u vidu poruka, između različitih sistema upotrebom standardizovanih jezika. Jedan od ovih jezika je XML [5].

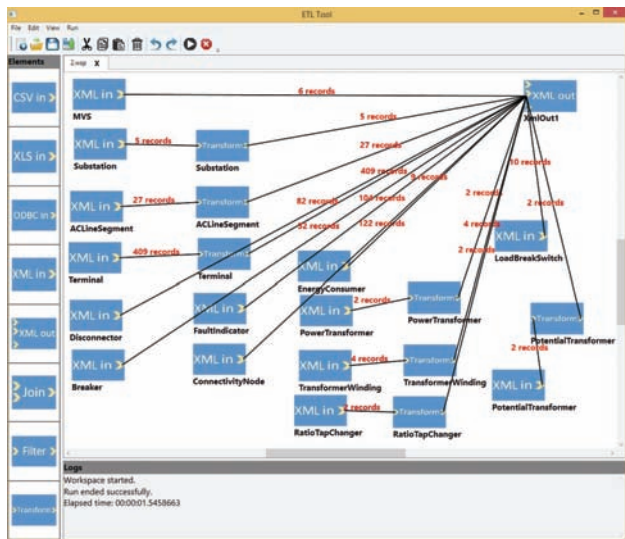
CIM/XML poruke sadrže skup objekata čija je struktura definisana CIM modelom. Kako ovi dokumenti predstavljaju osnovni deo u razmeni informacija, postoji realna potreba za njihovu podršku od strane ETL alata.

4. IMPLEMENTACIJA ETL ALATA



Slika 1. Klasni dijagram implementiranog ETL alata

Korisnički interfejs se oslanja na ovaj model i proširuje ga View klasama (*WorkspaceView*, *GraphicalElementView*, *ElementView* i *LinkView*). On takođe poseduje po jedan prozor za svaki tip elementa i jedan glavni prozor koji je prikazan na slici 2.

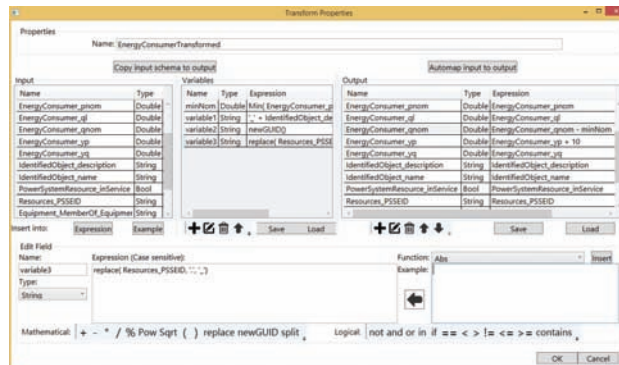


Slika 2. Korisnički interfejs ETL alata sa izvršenom radnom površinom

Za svaki element se definiše šema izlaznih objekata, na osnovu koje se oni kreiraju pri izvršavanju ETL zadatka radne površine. Na osnovu ove šeme se definišu transformacije i mapiranja polja izlaznih objekata elementa u narednim fazama ETL procesa. Rezultat izvršenja ekstrakcije ili transformacije je niz objekata i na njegovim izlaznim vezama je vidljiva informacija o broju elemenata tog niza, čije vrednosti korisnik može videti pomoću *Preview* opcije. Elementi se izvršavaju paralelno, što može znatno ubrzati ceo proces. Alat kontinualno pruža povratnu informaciju o obrađenim objektima i izvršenje radne površine može da se zaustavi u bilo kom trenutku.

Transformacije ekstrahovanih podataka se definišu u prozoru *Transform Properties* (slika 3) dodatog *TransformElement*-a. Ovaj element za ulaz mora imati tačno jednu vezu, na čijim objektima on izvršava definisane transformacije. Tabela na levoj strani predstavlja polja ulaznog entiteta, u desnoj se mogu

definisati izlazna polja i izrazi za računanje njihovih vrednosti, a u srednju se mogu dodati promenljive, u kojim se mogu definisati kompleksniji ili zajednički izrazi za više izlaznih polja, i one se ne propagiraju na izlaz elementa.



Slika 3. Transform Properties prozor TransformElement-a

Izlaznu šemu ovog elementa je moguće generisati iz nekog RDFS ili XSD dokumenta, tako što se nakon njihovog parsiranja bira jedan od dobijenih entiteta.

Drugi važan element je *XmlOutElement* koji je zadužen za kreiranje strukture izlaznog dokumenta i mapiranje transformisanih entiteta na tu strukturu. Izlazna struktura se može kreirati ručno, dodavanjem novih XML tagova i atributa ili generisanjem iz postojećeg RDFS, XSD ili XML dokumenta. Pri generisanju iz RDFS i XSD dokumenata se vrši njihovo parsiranje i kreiranje njima odgovarajućih .cs klasa C# programskog jezika. U šemu se nakon toga dodaju tagovi koji odgovaraju tim klasama. Tako dobijena šema sadrži informacije o tipovima polja entiteta i sposobna je na validaciju tipa polja pri mapiranju. Sa druge strane XML dokument ne sadrži informacije o tipovima polja svojih entiteta i generisanje strukture iz njega se vrši na drugačiji način. Naime, alat tretira XML dokument kao uzorak ciljnog dokumenta, što znači da generiše istu strukturu, s tim da ignoriše ponavljajuće tagove. Svi tagovi tako dobijene strukture se tretiraju kao stringovi i ne postoji opcija za validaciju tipa pri mapiranju.

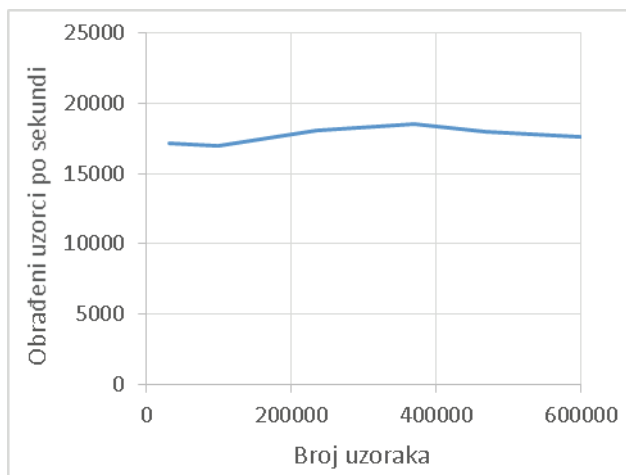
5. PERFORMANSE IMPLEMENTIRANOG ALATA

Za analizu performansi alata je preuzet CIM model mreže iz ENTSOE zvanične dokumentacije za CIM Interoperability testove [6]. Test scenario predstavlja ekstrakciju 8 tipova entiteta iz preuzetog CIM/XML dokumenta, njihovu transformaciju i zapis u izlazni CML/XML dokument. Kako karakteristike računara na kojem su izvršeni testovi performansi utiču na dobijene rezultate, treba napomenuti da on ima procesor Intel Core i3-2120 od 3,3GHz sa 2 jezgra i 4 thread-ova, 16 GB radne memorije i HDD na 7200rpm. Ukupan broj ekstrahovanih objekata je približno 100.000, koji su prošli kroz ETL proces za 6 sekundi, pri čemu su svi *thread*-ovi procesora bili ravnomerno opterećeni i zauzeto je 70 MB-a radne memorije. Ovi entiteti su imali prosečno 6 polja. Brzina obrade objekata direktno zavisi od broja njihovih polja. Obrada elemenata radne površine se vrši u paralelnim *Task*-ovima, dok se obrada objekata i njihovih polja na nivou elementa vrši sekvencijalno.

Isti test scenario je izvršen i bez paralelnog izvršavanja elemenata. U tom slučaju se ETL proces izvršio za 9

sekundi, što predstavlja 50% usporenja u odnosu na paralelno izvršavanje. Što više elemenata sadrži radna površina to je veće ubrzanje dobijeno paralelnim pristupom.

Izvršeni su i dalji testovi sa istom ulaznom CIM/XML datotekom, ali sa povećanim brojem ekstrahovanih entiteta, gde se analizira broj obrađenih objekata u sekundi u odnosu na njihov ukupan broj. Rezultati ovih testova su prikazani na slici 4. Dobijeni rezultati su u većini slučajeva u opsegu od 17000 do 19000 obrađenih ulaznih entiteta po sekundi, što predstavlja zadovoljavajuće performanse. Na slici 4 se vidi blago usporenje obrade preko određene cifre ulaznih entiteta, što je rezultat sekvencijalne obrade podataka na nivou jednog elementa, u ovom slučaju jedinog izlaznog elementa, naime *XML out-a*, koji predstavlja potencijalno usko grlo radne površine.



Slika 4. Broj obrađenih uzoraka po sekundi u odnosu na njihov ukupan broj

Kod velike količine ulaznih podataka radna memorija računara predstavlja kritičan resurs. Nakon što aplikacija zauzme svu slobodnu radnu memoriju, prinuđena je da koristi virtuelnu memoriju i tada se i preko deset puta usporava njeno izvršavanje. Drugi važan test scenario je analiza zauzeća radne memorije, gde je kao ulaz uzeta CSV datoteka sa preko 20.000.000 redova od po 29 polja. Za njenu ekstrakciju i zapis u izlaznu datoteku je bilo potrebno 20 minuta i 13 GB radne memorije.

6. ZAKLJUČAK

Implementiran alat se dobro pokazao sa realnim primerima iz prakse i pokazano je da može podržati integracije različitih izvora podataka. Njegove performanse i funkcionalnosti su konkurentne sa testiranim postojećim alatima na tržištu.

Mogućnost definisanja matematičkih izraza i manipulacija stringova pri transformaciji i filtriranju objekata u tekstualnom obliku daje veliku slobodu korisnicima alata i omogućuje im da definišu vrlo specifične transformacije.

Alat je dizajniran tako da je moguće njegovo proširivanje novim elementima i da je izvršavanje ETL zadatka radne površine omogućeno iz komandne linije, kako je model aplikacije potpuno razdvojen i nezavisan od korisničkog interfejsa. Za pokretanje iz komandne linije je prvo potrebno kreirati radnu površinu pomoću korisničkog

interfejsa, a nakon snimanja radne površine, moguće je izvršiti prosleđivanjem njene putanje kao parametra.

Pri prvobitnoj implementaciji alata, za čuvanje obrađenih entiteta na izlazima elemenata radne površine su korišćene matrice, tačnije nizovi nizova referenci, gde su redovi predstavljali entitete, a kolone polja tih entiteta. U kasnijim fazama razvoja su ove matrice zamenjene sa nizovima C# objekata, čija polja predstavljaju polja entiteta. Ova zamena je za rezultat imala do 40% manje zauzeće memorije i omogućila je validaciju tipova podataka, ali i do 20% manju brzinu izvršavanja. Ipak, ako se uzme u obzir da je pri velikim količinama podataka aplikacija prinuđena da koristi virtuelnu memoriju, tada je obrada podataka za red veličine sporija, te se ova zamena u takvim slučajevima može smatrati kao poboljšanje brzine alata.

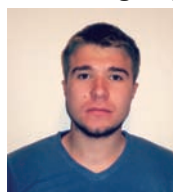
Neki od mogućih smerova razvoja su:

- dodavanje novih elemenata u cilju podržavanja drugih ulaznih ili izlaznih tipova podataka, kao i predefinisanih transformacija
- dodavanje podrške za generisanje izlaznih struktura elemenata iz drugih tipova šema
- dodavanje podrške za polja entiteta tipa liste
- modifikacija ulaznih elemenata tako da postanu sposobni za paralelnu ekstrakciju više od jednog tipa entiteta
- implementacija dodatnih uobičajenih funkcionalnosti grafičkih editora, kao što su zumiranje radne površine, laso selekcija elemenata, kreiranje projekata sa radnim površinama, Drag & Drop funkcionalnost, itd.

7. LITERATURA

- [1] Stanisław Kozielski, Robert Wrembel: New Trends in Data Warehousing and Data Analysis, Chapter 3: Near Real Time ETL, 2009, pp 19-50.
- [2] Open Studio Integration Software Platform | Talend, <https://www.talend.com/products/talend-open-studio/>, preuzeto 6.8.2015
- [3] CloverETL Designer | CloverETL Data Integration, <http://www.cloveretl.com/products/designer/>, preuzeto 26.8.2015
- [4] FME Desktop | Safe Software, <https://www.safe.com/fme/fme-desktop/>, preuzeto 19.8.2015
- [5] Dr Alan W. McMorran: An Introduction to IEC 61970-301 & 61968-11, 2007
- [6] Common Grid Model Exchange Standard | ENSTOE, https://www.entsoe.eu/Documents/CIM_documents/Grid_Model_CIM/CGMES_v2.4.15_TestConfigurations_v4.0.3.zip, preuzeto 7.10.2015

Kratka biografija:



Žolt Kanjo je rođen 1991. godine u Somboru. Školske 2010/2011 se upisao na FTN u Novom Sadu odsek Elektrotehnika i računarstvo, smer Računarstvo i automatika. Školske 2014/2015 je upisao master diplomatske akademske studije na FTN u Novom Sadu, smer Elektroenergetski softverski inženjering.

SOFTVERSKO RASPOREĐIVANJE POSLOVA PO IZVRŠIOCIMA**AUTOMATED CREW SCHEDULING**Nikola Pap, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je prikazan razvoj sistema, koji omogućava upravljanje mobilnim ljudskim resursima u smislu povere radnih zadataka pojedincima i timovima. Sistem pruža usluge prikaza i modifikacije podataka, kao i ručno i softversko kreiranje rasporeda.

Abstract – This paper describes development of a part of field service management system which is used for crew scheduling. System visualizes data and allows its modification, along with features for manual and automatic scheduling.

Ključne reči: softversko raspoređivanje, raspored, prikaz rasporeda

1. UVOD

U ovom radu prikazana je implementacija sistema za upravljanje mobilnim ljudskim resursima u smislu povere radnih zadataka pojedincima i timovima. Ovaj sistem pruža usluge prikaza i modifikacije podataka, kao i ručno i softversko kreiranje rasporeda. Ideja kreiranja ovakvog sistema je nastala iz potrebe za automatizacijom procesa raspoređivanja. Trendovi u softveru za upravljanje mobilnim ljudskim resursima pridaju sve veću važnost softverskom raspoređivanju poslova po izvršiocima, jer to omogućava smanjenje vremena u kome radnik nije na zadatku u toku radnog vremena.

2. OPIS REŠAVANOG PROBLEMA

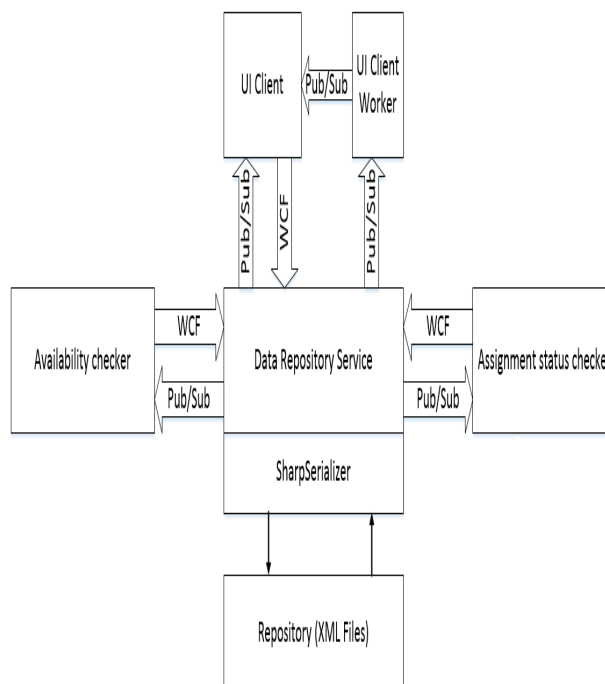
Sa porastom broja radnika kojima je potrebno dodeliti zadatke, raste broj dispečera koji se bave dodelom radnih zadataka. To predstavlja veliki trošak za kompanije, jer obuka dispečera može da traje i do dve godine da bi dispečer mogao samostalno i pouzdano da obavlja svoj posao. Pored jednostavnog kreiranja rasporeda na papiru ili uz pomoć alata za tabelarni prikaz podataka, kompanije zahtevaju softverske alate, koji će omogućiti automatizaciju procesa raspoređivanja. Kompanije ovaj problem najčešće rešavaju angažovanjem odgovarajućeg kadra, radi razvoja potrebnog softvera. Pošto je ovakav softver je najčešće nestabilan, nepouzdan i nefleksibilan, alternativa je nabavka profesionalnog softvera koji radi u realnom vremenu, sa velikom fleksibilnošću i pouzdanošću, koji vrlo brzo može da modifikuje raspored i prilagodi ga trenutnom stanju saobraćaja, udaljenosti radnika i slično [1].

Prilikom rešavanja problema automatskog raspoređivanja poslova po izvršiocima, potrebno je raspoređivanje poslova izvršiti spram radnog vremena izvršioca i sposobnosti izvršioca za izvršenje željenog posla. Implementacija automatskog raspoređivanja se može bazirati na genetskom algoritmu. Rešenje treba da poseduje grafički korisnički interfejs koji, pored prikaza rasporeda, omogućiti prikaz i modifikaciju podataka o poslovima i izvršiocima.

3. OPIS REŠENJA PROBLEMA

U cilju rešavanja problema automatskog raspoređivanja poslova po izvršiocima, razvijen je sistem koji omogućuje prikaz, unos, izmenu i brisanje relevantnih podataka. Sistem je takav da razlikuje dve vrste korisnika koji imaju različita prava za upotrebu podataka. Kako bi konzistentnost podataka bila očuvana, sistem poseduje jedinstveni izvor podataka. Manipulacija podacima se odvija iz klijentskih aplikacija realizovanih upotrebom **WPF** tehnologije kao i konzolne aplikacije, pozivom operacija servisne aplikacije, koja je realizovana upotrebom **WCF** tehnologije.

Arhitektura rešenja je prikazana na slici 1.



Slika 1. Arhitektura rešenja

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Hajduković, red. prof.

Centralno mesto u arhitekturi rešenja ima **Data Repository Service** servisna aplikacija realizovana upotrebom **WCF** tehnologije, čija je svrha upravljanje

podacima. Ona se oslanja na modul za serijalizaciju **SharpSerializer** [2]. **Availability checker** i **Assignment Status Checker** su konzolne aplikacije, koje kontorlišu podatke o izvršiocima, odnosno poslovima. Očekivano je da postoji po jedna instanca ovih aplikacija u sistemu. **UI Client** i **UI Client Worker** su aplikacije realizovane upotrebom **WPF** tehnologije, a njihova namena je prikaz podataka sa mogućnošću njihovog dodavanja, izmene i brisanja, u zavisnosti koja se aplikacija koristi. Očekivano je da u sistemu postoji više instanci ovih aplikacija.

Svaka aplikacija, osim **Data Repository Service Host** aplikacije, poseduje svoju lokalnu kopiju podataka, koja se nalazi u operativnoj memoriji. Izmenjeni podaci se iz aplikacije šalju do **Data Repository Service** aplikacije, koja zapisuje promene u **XML** datoteke. Nakon završenog pisanja u datoteke, podaci se šalju prema ostalim aplikacijama, odnosno njihovim lokalnim kopijama podataka, uz pomoć **Pub/Sub** mehanizma. U lokalnoj kopiji podataka aplikacija **UI Client** i **UI Client Worker** se memoriše podatak o trenutnom korisniku aplikacije.

U sistemu postoji podsistem za vođenje žurnala upotrebe sistema (**CrFSMLogger**) koji se koristi u svim aplikacijama.

3.1. Model strukture podataka

Sve klase u modelu strukture podataka nasleđuju korensku klasu (**root class**) **IdentifiedObject**. Klasa **Worker** predstavlja generalizaciju dva tipa izvršioca posla, radnika (**CrewMember**) i grupe radnika (**Crew**). Klasa **Worker** se nalazi u asocijaciji sa klasama **Shift** i **Assignment**. Izvršilac radi u jednoj smeni i jednom izvršiocu može biti dodeljeno više poslova. Jedan radnik (**CrewMember**) može, a ne mora, biti član jedne ekipe (**Crew**). Svakom radniku odgovara tačno jedan set kredencijala (**username** i **password**) i jedna uloga u sistemu, predstavljene enumeracijom **UserType**. Svaki radnik može iskoristiti svoj godišnji odmor (**Vacation**) u više termina. Svaki objekat nosi podatak o svom tipu, koji je modelovan enumeracijom **EntityType**, ukoliko je klasa objekta na poslednjem nivou hijerarhije nasleđivanja.

3.2. Način perzistencije podataka

Podaci se čuvaju u **XML** datotekama kojima manipuliše **SharpSerializer** modulu za serijalizaciju. Ovom modulu se pristupa preko **WCF** servisa, **Data Repository Service**, koji serijalizuje objekat tipa **Hastable**. Datoteke u koje se serijalizuju podaci dobijaju nazive po tipu entiteta koji se serijalizuje (**EntityType**).

3.3. Data Repository Service

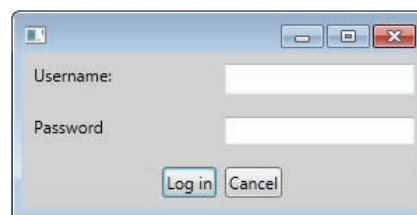
Data Repository Service je modul koji ima ulogu isporuke podataka aplikacijama i pisanja podataka na disk, ako aplikacija to zatraži. Ovaj modul je realizovan kao **WCF** servis. **Host** ovog servisa je aplikacija **DataRepositoryServiceHost**. **DataRepositoryServiceHost** aplikacija postavlja adresu servisa i tip komunikacionog kanala. Korišćeni tip komunikacionog kanala je **NetTcpBinding**.

3.4. Lokalna kopija podataka

Lokalna kopija podataka (**Local Cache**) je pomoćna struktura čiji je cilj da ubrza procese čitanja podataka. Za razliku od podataka koje koristi **Data Repository Service**, koji se nalaze na disku, podaci u lokalnoj kopiji podataka se nalaze u operativnoj memoriji, što obezbeđuje kraće vreme pristupa podacima zbog prirode operativne memorije.

3.5. Korisničke aplikacije

U okviru sistema postoje dve korisničke aplikacije. **UI Client** je aplikacija koja služi za prikaz i manipulaciju podacima. Korisnik ove aplikacije ima ulogu administratora ili menadžera sistema. **UI Client Worker** aplikacija je namenjena samo za prikaz podataka o prijavljenom korisniku. Korisnik **UI Client Worker** aplikacija može da podnese zahtev za izmenu ličnih podataka korisniku **UI Client** aplikacije koji može da pogleda, prihvati ili odbije taj zahtev. Obe aplikacije omogućuju prikaz i manipulaciju rasporedom u okviru svojih ograničenja. Prijava na sistem se vrši preko **LogIn** forme prikazane na slici 2. Kada korisnik unese svoje kredencijale, sistem inicijalizuje odgovarajuću aplikaciju.



Slika 2. LogIn forma

UI Client poseduje forme za prikaz i manipulaciju podacima o radnicima, grupama radnika, poslovima, smenama rada i kalendaru. Svaka forma, osim forme za prikaz kalendara, se sastoji iz dva dela. U levom delu tabelarno su prikazani podaci o svim entitetima željenog tipa sa prikazanim vrednostima bitnijih atributa. U desnom delu forme se nalaze polja za unos i izmenu podataka o entitetu.

Na kartici **Calendar** se nalazi prikaz rasporeda posla po izvršiocima. Kalendar pruža uvid u dnevni, nedeljni i mesečni raspored, odabir datuma za prikaz, mogućnost ručne manipulacije i aktivaciju procesa softverskog raspoređivanja poslova po izvršiocima. Nedeljni raspored se prikazuje kao pet instanci dnevnog rasporeda za datume radnih dana izabrane nedelje za prikaz. Raspored je prikazan tabelarno, gde redovi predstavljaju vreme u intervalu od ponoći do 23 časa, a kolone predstavljaju izvršioce u koje spadaju grupe radnika i pojedinačni radnici koji nisu ni u jednoj grupi. U prvom redu zaglavlja kolona se nalazi dan i datum dnevnog rasporeda, a u drugom redu su prikazani izvršioci. Mesečni raspored poseduje deo za svaki dan, obeležen datumom, u kome se nalaze informacije o nazivu posla, njegovom izvršiocu, početku izvršavanja i kraju izvršavanja se nalaze u žutim pravougaonicima.

Korisnik **UI Client Worker** aplikacije ima ulogu radnika (**Worker**) i on preko ove aplikacije ima uvid samo u svoje podatke. Aplikacija se sastoji iz dve forme: forma za prikaz i izmenu ličnih podataka i forma za prikaz rasporeda. Na formi za prikaz rasporeda radnik ima mogućnost prikaza rasporeda na dnevnom, nedeljnom i mesečnom nivou koji ima istu formu kao u aplikaciji **UI Client**, ali je personalizovan odnosno prikazuje podatke vezane samo za ulogovanog korisnika sistema.

3.6. Availability checker i Assignment status checker aplikacije

Availability checker aplikacija je namenjena da prati radno vreme radnika i menja podatke o dostupnosti radnika u skladu sa radnim vremenom. Realizovana je tako da periodično proverava da li je nekom od radnika počelo ili se završilo radno vreme. Dostupnost radnika je modelovana enumeracijom **AvailabilityStatus**.

Assignment status checker aplikacija je namenjena da vrši izmene statusa nekog posla u zavisnosti od vremena. Realizovana je tako da periodično proverava da li je izvršavanje posla započeto po rasporedu i da li je izvršavanje posla završeno po rasporedu. Status posla je modelovan enumeracijom **AssignmentStatus**.

Parametar	Tip	Opis
Population Size	int	Veličina populacije
Replace by population	int	Broj jedinki koji se menja prelaskom iz generacije u generaciju
Mutation size	int	Veličina mutacije
Mutation probability	double	Verovatnoća mutacije
Crossover probability	double	Verovatnoća ulaska jedinke u ukrštanje
Crossover points	int	Broj tačaka na kojima se vrši ukrštanje
Schedule from	DateTime	Početak intervala za koji se pravi raspored
Schedule to	DateTime	Kraj intervala za koji se pravi raspored
Elitism	bool	Elitizam
Random selection	bool	Način odabiranja jedinki za ukrštanje
Assignments	List <Assignment>	Poslovi koje je potrebno rasporediti

Tabela 1. Parametri algoritma

3.7. Softversko raspoređivanje

Softversko raspoređivanje poslova po izvršiocima se u sistemu obavlja pozivom iz menadžerske aplikacije. Korisnik ima mogućnost podešavanja parametara preko odgovarajuće forme. Za potrebe rešenja problema

rapoređivanja implementiran je genetski algoritam. Parametri algoritma su prikazani u tabeli 1.

3.7.1. Model jedinke

Model jedinke predstavlja jednu verziju rasporeda poslova po izvršiocima. Raspored je predstavljen kao statički niz sa pomoćnim strukturama. Dužina niza se računa kao suma dužina radnog vremena svakog izvršioca, izdvojena na intervale dužine pet minuta. Računanje dužine niza je predstavljeno sledećom formulom:

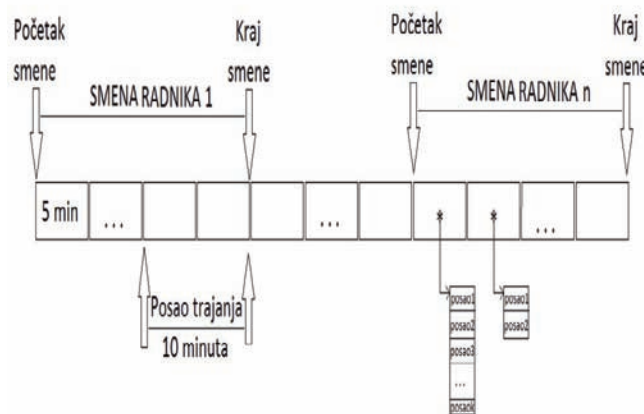
$$d = \sum_{i=1}^n [s_i/5]$$

d – dužina niza

s_i – dužina smene radnika u minutima

n – broj izvršilaca

Elementi statičkog niza su liste stringova odnosno globalnih identifikatora poslova. Kada se posao dodeli radniku, globalni identifikator tog posla se dodaje u broj sekvencijalnih lokacija niza koji odgovara gornjoj granici količnika trajanja posla u minutima i intervala dužine pet minuta. Na slici 3 je prikazana skica statičkog niza.



Slika 3. Skica statičkog niza

Pošto statički niz nosi informacije samo o dužinama smena i poslovima dodeljenim izvršiocima, uvodi se struktura koja će nositi informacije o radnicima. Kao pomoćna struktura korišćen je statički niz čiji su elementi tipa strukture koja, nosi podatke o izvršiocu posla i o poslednjoj poziciji u nizu.

U cilju optimizacije i uprošćavanja implementacije genetskih operatora, uvedena je pomoćna struktura tipa sortirane kolekcije ključ-vrednost parova, gde je ključ globalni identifikator posla, a vrednost je njegova pozicija u nizu.

3.7.2. Operator selekcije

Namena operatora selekcije je da na neki način odabere jedinke iz populacije koje će ući u proces rekombinacije. U okviru ovog rešenja implementirana su dva načina selekcije: selekcija tipa ruleta [3] i nasumična selekcija. Upotreba načina selekcije se odabira vrednošću parametra **Random selection**. Selekcija tipa ruleta predstavlja način selekcije kod kog se za svaku jedinku računa verovatnoća selekcije na osnovu prilagođenosti jedinke, što obezbe-

duje odabir prilagođenijih jedinki za rekombinaciju. Ovakav način selekcije vrlo lako može da dovede do prevremene konvergencije algoritma odnosno pronalaženje suboptimalnog rešenja [4].

3.7.3. Operator rekombinacije

Proces rekombinacije se vrši tako što se nasumično generiše **Crossover points** tačaka. Geni roditeljskih jedinki se rekombinuju po generisanim tačkama tako da se dobiju dva deteta jedinke. Da bi se sprečilo nazadovanje algoritma, u novu populaciju se uključuje najprilagođenija od dve roditeljske jedinke i dva deteta jedinke.

3.7.4. Operator mutacije

Mutacija jedinke se vrši tako što se jedan ili više nasumično odabranih poslova prebaci u drugi termin odnosno promeni svoju poziciju u statičkom nizu. Broj poslova koje je potrebno premestiti određen je vrednošću parametra **Mutation size**, a parametar **Mutation probability** određuje verovatnoću dešavanja mutacije.

3.7.5. Funkcija za izračunavanje prilagođenosti

Implementirano rešenje vrši evaluaciju jedinke na osnovu dva pravila:

- Jedan izvršilac može da obavlja samo jedan posao u datom trenutku.
- Izvršilac mora da poseduje veštine koje su potrebne za uspešno obavljanje zadatka.

Svaki posao može imati najviše dva boda. Bodovi se dobijaju na osnovu broja zadovoljenih pravila. Jedna jedinka može da ostvari maksimalno dva puta više bodova od broja poslova koje je potrebno rasporediti. Prilagođenost jedinke se računa kao količnik ostvarenog broja bodova jedinke i maksimalnog broja bodova jedinke.

3.7.6. Ažuriranje modela strukture podataka sistema

Kada se softversko raspoređivanje poslova po izvršiocima završi, dobijeni raspored se rekonstruiše iz statičkog niza i pomoćnih struktura i primenjuje se na model strukture podataka sistema, tako što se ažuriraju podaci o dodeljenom vremenu i izvršiocu svakog posla na datum koji je prosleđen parametrom **Schedule from**. Ukoliko se neki od poslova ponavlja u nekom periodu, ponavljanje se unosi u periodu, zaključno sa datumom koji je prosleđen parametrom **Schedule to**.

3.8. CrFSMLogger modul

CrFSMLogger modul predstavlja podsistem za vođenje žurnala korišćenja sistema. Uveden je radi lakše kontrole toka programa i jednostavnijeg rešavanja problema i otklanjanja grešaka.

Modul je realizovan kao **Singleton** klasa sa dve metode istog naziva koje primaju različite parametre. Za pisanje u tekstualnu datoteku koristi se **.NET** klasa **System.IO.TextWriter**.

4. ZAKLJUČAK

U ovom radu prikazan je razvoj sistema za upravljanje mobilnim ljudskim resursima sa mogućnošću softverskog raspoređivanja poslova po izvršiocima. Sistem je realizovan tako da se više aplikacija istovremeno oslanja na jedan izvor. Bilo je potrebno implementirati genetski algoritam za potrebe automatizacije kreiranja rasporeda poslova po izvršiocima. Rešenje je realizovano tako da se perzistencija podataka vrši preko biblioteke **SharpSerializer** upisom podataka u **XML** datoteke. Tačka pristupa biblioteci, samim tim i podacima, odvija se preko servisne aplikacije **Data Repository Service** kojoj se obraćaju aplikacije **Availability checker**, **Assignmet status checker**, **UI Client** i **UI Client worker**.

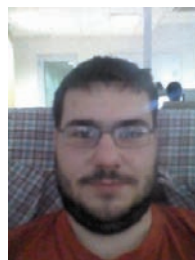
Komunikacija aplikacija prema servisu se vrši putem **WCF**-a, a komunikacija servisa sa aplikacijama putem **Pub/Sub**-a. Implementirani genetski algoritam može na više načina da izvršava svoje operacije nad podacima, što je omogućeno podešavanjem parametara algoritma.

Mogući pravci razvoja su implementacija bezbednosnih mehanizama, optimizacija algoritma raspoređivanja, unapređenje algoritma da očuva postojeće podatke, odnosno uvođenje ograničenja pri kreiranju rasporeda, razvoj sistema koji će raditi u realnom vremenu po ugledu na postojeća komercijalna rešenja, uvođenje relacije baze podataka sa sistemom za upravljanje bazom podataka, zbog jednostavnijeg modelovanja ograničenja i proširenja samog modela podataka.

5. LITERATURA

- [1] Selecting a dynamic scheduling engine for field service management, <http://www.ifsworld.com/en/sitecore/media-library/assets/2014/07/16/16/08/white-papers-selecting-a-dynamic-scheduling-engine-for-field-service-management/>
- [2] SharpSerializer, <https://sharpserializer.codeplex.com>
- [3] Genetic Algorithms, <http://www.graham-kendall.com/papers/sgk2005.pdf>
- [4] Genetic Algorithms Demystified, <http://www.codeproject.com/Articles/707505/Genetic-Algorithms-Demystified>

6. PODACI O KANDIDATU



Nikola Pap je rođen u Novom Sadu. Završio je gimnaziju „Jovan Jovanović Zmaj“ u Novom Sadu 2010. godine. Osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu završio je 2014. godine. Master akademske studije na Fakultetu tehničkih nauka u Novom Sadu je upisao 2014. godine.

IMPLEMENTACIJA IZVEDENIH POLJA KORIŠĆENJEM OCL IZRAZA

IMPLEMENTATION OF DERIVED FIELDS USING OCL EXPRESSIONS

Marijana Rackov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu opisana je implementacija alata za generisanje java koda na osnovu zadatih OCL ograničenja.

Abstract – This paper describes implementation of tool for generating java application using OCL expressions.

Ključne reči: OCL, ograničenje, izvedena polja

1. UVOD

U vremenu kada su računari postali naša svakodnevica, poseže se za pojednostavljivanjem već dosta uprošćenih procesa u svim sferama našeg života. Zahtevi za tehničkim unapređenjem su sve učestaliji i zahtevniji u smislu da se poboljšanje očekuje u veoma kratkom periodu, gde se često dešava da dolazi do nerazumevanja oko vremenskog okvira i resursa koji su potrebni za novonastale zahteve. Samim tim kao rezultat nezadovoljstva korisnika, konkurencije na tržištu i olakšanja procesa rada zaposlenima u softverskim kompanijama, uvode se pravila i standardi za kreiranje odnosno programiranje aplikacija po zahtevu korisnika. Naime, kako bi se izbegao takozvani „problem kreiranja softverskog rešenja od nule“ odnosno od početne tačke, osmišljene su različite metodologije, pravila, standardi i tehnike koje doprinose bržem i jednostavnijem razvoju softverskih proizvoda. Samo neki od njih su: pristupi i alati za brzi razvoj softvera, metode i tehnike brzog razvoja softvera, generatori koda, metodoloških pristupi razvoju softvera (agilne i tradicionalne metoda), prototipski razvoj softvera, razvoj softvera na bazi modela (MDA - Model Driven Architecture) itd.

Jedna od oblasti koja je veoma zastupljena i korisna za rešavanje navedenih problema jeste MDE – Model Driven Engineering odnosno istaknuti predstavnik ove oblasti MDA. MDA proces je podeljen u tri koraka, gde se u prvom kreira model nezavisan od implementacijske platforme (PIM – Platform Independent Model), u drugom se kreirani model transformiše u jedan ili više modela zavinih od platforme (PSM – Platform Specific Model), dok se u trećem završnom koraku generiraju programski kod na osnovu PSM-a. Konkretno tema ovog rada je vezana isključivo za treći deo procesa pa će tome biti i posvećena najveća pažnja. Pri generisanju koda uzimaju se u obzir polja na entitetima u kojima su specificirana poslovna pravila određenog sistema. Takve specifikacije su deo OCL-a (OCL – Object Constraint Language), čije tumačenje i generisanje

ograničenja na osnovu protumačenih iskaza predstavljaju temu ovog rada.

2. OCL – OSNOVNE INFORMACIJE

Prva verzija OCL-a je razvijena od strane IBM-a 1995. godine. OCL je 1997. godine postao deo UML-a i objavljen je kao deo OMG (Object Management Group) specifikacije u verziji 1.1. Razvoj verzija UML-a 2.0 i OCL-a 2.0 je počeo 1999. godine, a verzije su objavljene 2004. Danas je OCL postao opšteprihvaćeni standard koji podržava sve veći broj alata a napravljen je tako da bude lak za čitanje i pisanje za većinu korisnika, za razliku od drugih jezika ovog tipa koji su bazirani na matematičkim formalizmima. OCL je formalni jezik pomoću kojeg se definišu ograničenja koja nije moguće iskazati samo UML-om. OCL je deo UML standarda i uveden je kao proširenje samog UML-a, kako bi upotpunio potrebu za pisanjem dodatnih izraza.

2.1 Oblik OCL izraza

context NazivEntiteta stereotip: OCL izraz

Context je obavezna službena reč na početku. Stereotip može biti:

- Inv <<invariant>> ograničenje - mora biti tačno u celom toku životnog ciklusa objekta
- Pre <<precondition>> - uslovi koji moraju biti zadovoljeni pre poziva neke metode
- Post <<postcondition>> - uslovi koji moraju biti zadovoljeni posle poziva neke metode

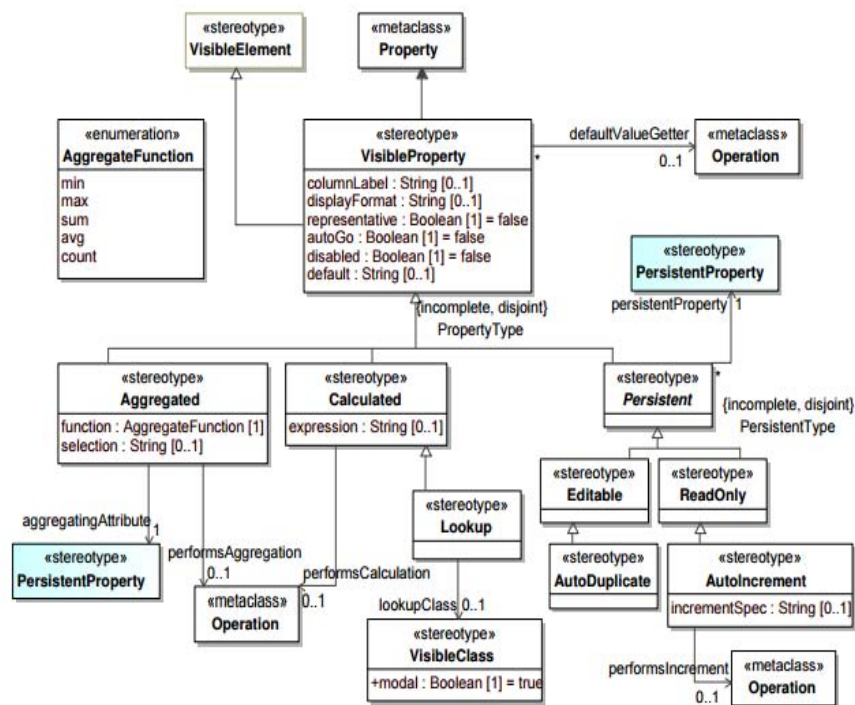
Službena reč self označava instancu tekućeg objekta, a result povratnu vrednost. Umesto self može biti korišćena i druga reč.

2. OCL POLJA DEFINISANA U KROKI ALATU

Kroki (fr. croquis – skica) je alat namenjan za interaktivni razvoj poslovnih aplikacija baziranih na skicama. Za razliku od uobičajene prakse, ovde se skica koristi u toku celog procesa razvoja softverskog proizvoda kao podloga za automatsko izvršavanje ili generisanje koda poslovnih aplikacija. Kroki implementira konkretnu sintaksu datog EUIS (Enterprise User Interface Specification) DSL-a (Domain Specific Language), koja je projektovana tako da se omogući modelovanje korisničkog interfejsa na „prirodan“ način. Pokretanje skica obezbeđuju dve generičke aspect-orientisane (AOP) web odnosno desktop aplikacije kreirane na bazi EUIS DSL-a. Kroki podržava skiciranje više vrsta formi, poštujući standard koji definiše njihov izgled i funkcionalnost [5].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Gordana Milosavljević.



Slika 1 Dijagram metaklasa polja na formama [4]

3. IMPLEMENTACIJA REŠENJA

Za potrebe ovog rada Kroki model proširen je klasama uz čiju pomoć se definišu ograničenja zadata kroz interfejs Kroki alata. U paketu kroki.app.generators.utils KrokiMockupTool-a dodate su klase Constraint.java, Element.java, Operation.java i Parametar.java. Klasa Constraint.java predstavlja, kako joj i sam naziv kaže, strukturu jednog ograničenja odnosno njegov entitet. Proces započinje u klasi ProjectExporter.java gde se vrši parsiranje OCL izraza odnosno njihovo pripremanje za dalju analizu pomoću klase ConstraintAnalyzer, ConstraintBody–Analyzer odnosno pozivima klase koje obrađuju pogodeni tip OCL izraza. Posle toga, tako pripremljen skup ograničenja prosledjuje se klasi ConstraintGenerator.java gde se pomoću constraint.ftl šablona generešu EntitetConstraint.java klase. Proces parsiranja ograničenja se vrši u metodi getStandardPanelData klase ProjectExporter.java, gde se pripremaju svi elementi panela za kreiranje generisanih klase.

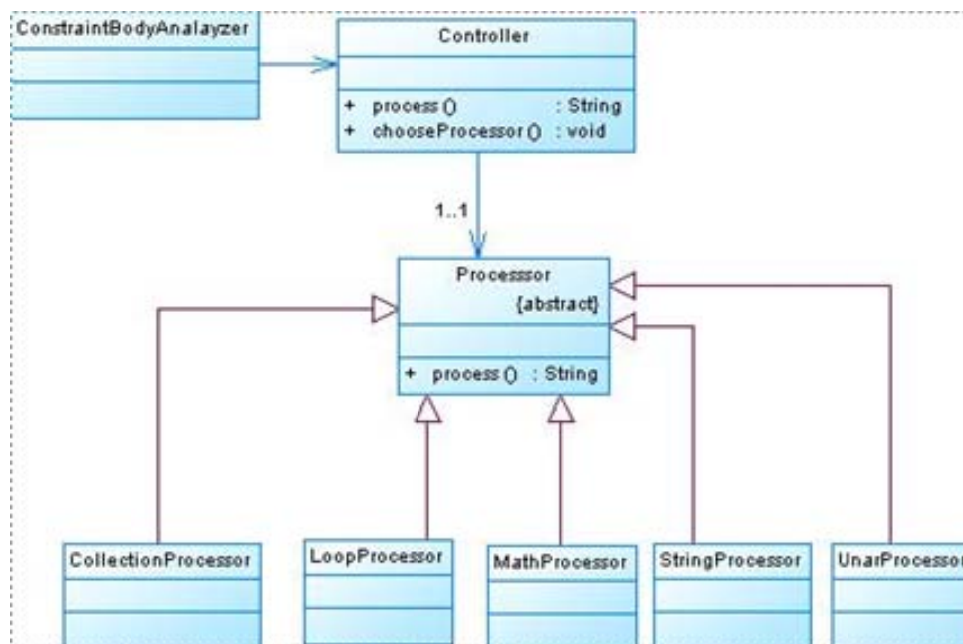
Upotrebom standardne OCL biblioteke (The Standard OCL Eclipse Library) dolazi do velikih problema prilikom parsiranja. Naime, parsiranje se vrši tako što je potrebno kreirati instancu OCL-a preko ecore modela, što traje neupotrebljivo dugo (više od 10 sekundi), nakon čega se kreira OCLHelper kome se kao parametar navodi biblioteka iz koje će koristiti pravila za parsiranje ograničenja. Svaka od tih biblioteka je specijalizovana za određenu problematiku, tj. Ne postoji neka univerzalna biblioteka koja bi bila pogodna za ovo rešenje. Rezultat najjednostavnijih OCL izraza je bio neuspešan sa porukom o nepoznatoj sintaksi.

Nakon neuspešnog pokušaja upotrebe prethodno opisane biblioteke, upotreba DresdenOCL biblioteke dala je mnogo bolji rezultat. Iako je to biblioteka koja podržava

kompletnu sintaksu definisanu OCL 2.3 specifikacijom, u okviru ovog rada korišćena je samo za parsiranje formula. Kao ni ostale, ni ona nije napravljena bez nedostataka, npr. nije podržana operacija OclAny.oclLocale(), nisu podržani izrazi kojima se može definisati višestrukost (multiplicity) u okviru ograničenja, nije podržan ni tip podataka OclMessage i drugi nedostaci čijim bi rešenjem bila znatno jednostavnija upotreba samog DresdenOCL-a.

Kao i kod Eclipse-ove biblioteke, potrebna je upotreba modela nad kojim se pravila izvršavaju. Na samom početku procesa učitava se ecore bazirani model potreban za parsiranje OCL izraza. Nakon njegovog učitavanja prolazi se kroz listu svih VisibleProperty-a gde se svaki element koji je instanca kalkulisanog, agregiranog ili lookup polja prosledjuje Ocl22Parser-u. Tako parsirano ograničenja se dodaje u listu DresdenOCL ograničenja koja se prosledjuje ConstraintAnalyzer-u gde se vrši obrada i priprema elemenata za dalje generisanje koda. Tako pripremljena lista ograničenja se presipa u listu inicijalizovanu na početku procesa kao listu ograničenja entiteta za koji se vrši analiza elemenata panela.

Kao što je već naglašeno, analiza ograničenja započinje u klasi ConstraintAnalyzer.java. Unutar klase ConstraintBodyAnalyzer prolazi se kroz listu izraza pri čemu se u slučaju pojave operacije, informacija prosledjuje Controller-u radi analize o kojem se tipu operacije radi i daljem prosledjivanju odgovarajućem Processor-u. Implementirano je pet klase za obradu različitih tipova tela ograničenja: CollectionProcessor, LoopProcessor, MathProcessor, StringProcessor i UnarProcessor (slika 2).



Slika 2 Dijagram obrade operacija

Nakon kompletne analize ograničenja dobijena je lista Constraint-a definisanih u modelu. Popunjavanjem liste ograničenja klase EJBClass vrednostima ograničenja dobijenih analizom u ConstraintAnalyzer-u, klasa EJBClass kompletno je pripremljena za dalji tok procesa odnosno generisanje java koda. Nastavak procesa se odvija u klasi ConstraintGenerator.java gde se dobavlja šablon za generisanje java klase definisanih za implementaciju ograničenja.

Nakon definisanja naziva paketa klase, importovanja svih ostalih potrebnih klasa i kreiranja konstruktora, dolazi se do dela koji se odnosi na generisanje ograničenja tipa invariant, postcondition i precondition. Na početku se prolazi kroz listu ograničenja, pri čemu se za svako ograničenje privremeno čuva u promenljivoj con, a zatim se vrši provera tipa ograničenja. \${con.operation.header} će biti zamenjen vrednošću iz modela koja predstavlja zaglavlje metode. Zatim se prolazi kroz listu deklaracija potrebnih promenljivih i svaka deklaracija se ispisuje na mestu u šablonu gde se nalazi promenljiva \${dec}. Nakon toga se proverava da li se u ograničenju javlja if-else struktura i ispisuju se uslovi koji moraju biti zadovoljeni. Na kraju se formira poruka o grešci i aktivira se izuzetak u slučaju da ograničenje nije zadovoljeno.

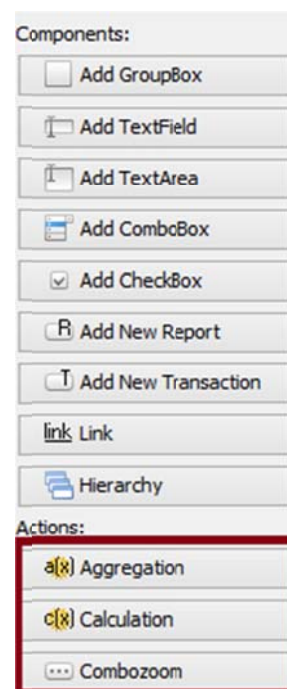
4. NAČIN UPOTREBE OGRANIČENJA U KROKI ALATU

Kao što je već pomenuto u okviru Kroki alata postoje polja preko kojih se mogu zadati OCL ograničenja za buduće generisanu aplikaciju. Na meni bar sa desne strane su ona izdvojena u delu „Akcije“. Kao što se vidi na slici 3 to su: agregirana, kalkulisana i lookup polja.

Upotreba agregiranog polja ogleda se u odabiru jedne od ponuđenih finkcija u combobox-u oivičenom na slici 3. Moguće vrednosti funkcije su: min, max, sum, avg i count.

Kalkulisana polja su nešto komplikovanija za upotrebu od agregiranih samim tim što je za dobijanje ograničenja

potrebno znati osnove sintakse OCL-a a takođe poznavati i model aplikacije koja se generiše. OCL izraz je potrebno upisati kao expression konkretnog elementa. Takođe, kao i kod kalkulisanih polja i kod lookup polja izraz se piše u expression delu.



Slika 3 Akcije u Kroki alatu

5. ZAKLJUČAK

Implementacija OCL ograničenja u Kroki alat donela bi još jednu veom bitnu i korisnu funkcionalnost u njegovoj daljoj upotrebi i budućem procesu razvoja. Prezentaciono to je jedan veoma isplativ novitet čija se primena može demonstrirati na licu mesta što je uvek ocena više u očima budućih korisnika.

U okviru ovog rešenja omogućeno je generisanje ograničenja na nivou entiteta i njegovih atributa tako što obuhvata sve tipove ograničenja i većinu definisanih operacija. Podržane su logičke i matematičke funkcije, operacije za rad sa stringovima, kolekcijama, if-else strukture, upotreba enumeracija, nadovezivanje operacija do proizvodnje dubine, rad sa iteratorima i drugi tipovi.

Ono što u okviru ovog rada nije podržano su tipovi podataka Tuple i OclMessage, operacije za rad sa ovim tipovima kao ni operacije za koje je potrebno koristiti AspectJ odnosno operacije kao što su sortedBy, isSignalSent, oclAsNew, allInstances i druge. Ovo bi svakako mogao biti predmet daljeg rada u ovoj oblasti.

6. LITERATURA

- [1] OMG OCL <http://www.omg.org/spec/OCL/>
- [2] Kroki, alat za interaktivni razvoj poslovnih aplikacija baziranih na skicama, www.kroki-mde.net
- [3] Predavanja sa predmeta Metodologije brzog razvoja softvera - Gordana Milosavljević
- [4] UML Profile for Specifying User Interfaces of Business Applications - <http://www.doiserbia.nb.rs/img/doi/1820-0214/2011/1820-02141100010P.pdf>
- [5] M. Filipović, V. Marsenić, G.Milosavljević, I. Dejanović. Kroki: alat za interaktivni razvoj poslovnih aplikacija baziranih na skicama

Kratka biografija:

Marijana Rackov rođena je 2.5.1990. godine u Novom Sadu. Osnovnu školu „Branko Radičević“ i Gimnaziju „Isidora Sekulić“ završila je u rodnom gradu, gde je 2009. godine upisala Fakultet tehničkih nauka, odsek Računarstvo i automatika, smer Primenjene računarske nauke i informatika. Nakon završenih osnovnih studija 2013. godine upisala je master studije na istom fakultetu smer Softversko inženjerstvo i informacione tehnologije. Od februara 2014. godine zaposlena u softverskoj kompaniji Prozone iz Novog Sada na poziciji mlađeg programera.

ANALIZA UTICAJA VISOKONAPONSKIH ELEKTROENERGETSKIH SISTEMA NA GASOVODE

ANALYSIS OF THE INFLUENCE OF HIGH VOLTAGE POWER SYSTEMS ON GAS PIPELINES

Nemanja Jovičić, Fakultet tehničkih nauka, Novi Sad

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U okviru ovog rada razmotren je uticaj visokonaponskih elektroenergetskih sistema na gasovode. Predstavljeni su osnovni tipovi i karakteristike ovih uticaja kao i princip njihovog proračun.

Abstract – This guideline examined the analysis of the influence of high voltage power systems on gas pipelines. It contains basic types and characteristics of these interferences as well as their calculations.

Ključne reči: Uticaj elektroenergetskih sistema na gasovode, galvanski uticaj, induktivni uticaj, kapacitivni uticaj, zaštita metalnih gasovoda od uticaja elektroenergetskih sistema.

1. UVOD

Opštim povećanjem potrošnje električne energije i razvojem elektroenergetske mreže adekvatnu trasu za gasovod je sve teže pronaći, naročito u gusto naseljenim predelima, pa se sve više koriste trase dalekovoda za polaganje gasovoda. Iz tih razloga je opasnost od posledica uticaja visokonaponskih elektroenergetskih vodova na gasovode vremenom sve veća. Gasovodi se izgrađuju kao dugi metalni neizolovani ili gasovodi izolovani bitumenom ili polietilenom.

Gasovodi se mogu posmatrati kao provodnici izolovani od zemlje. Svojom dužinom mogu biti izloženi velikom broju tipova uticaja, a najznačajniji je uticaj elektroenergetskih vodova. Taj uticaj može biti posledica tri vrste sprezanja: kapacitivna sprega, induktivna sprega i galvanska sprega.

Usled ovih uticaja javljaju se negativne posledice u vidu ugroženosti bezbednosti ljudi i oštećenja na gasovodu. Najznačajniji problem je svakako opasnost po ljude koji su u dodiru sa metalnim delovima sistema gasovoda. Kada se govori o oštećenjima gasovoda, može nastati oštećenje izolacionog sloja, zida gasovoda, izolacionog spoja i opreme povezane sa gasovodom.

2. KARAKTERISTIKE GASOVODA

2.1. Osnovni parametri gasovoda

Preciznost proračuna zavisi od poznavanja parametara gasovoda. U tabeli 2.1.1. je prikazan pregled parametara gasovoda koji su od značaja za proračun uticaja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragutin Salamon, vanred.prof.

Oznaka	VELIČINA	Jedinica
z	Redna impedansa gasovoda	Ω/m
z_{int}	Sopstvena impedansa gasovoda	Ω/m
y	Admitansa gasovoda	$1/\Omega/\text{m}$
μ_0	Magnetska permeabilnost vazduha	H/m
μ_r	Relativna permeabilnost gasovoda	
ρ	Otpornost tla	Ω/m
ρ_p	Otpornost gasovoda	Ω/m
ρ_c	Otpornost izolacionog sloja gasovoda	Ω/m
ϵ_0	Električna permitivnost vazduha	F/m
ϵ_r	Relativna permitivnost izolacionog sloja gasovoda	
ϵ	Električna permitivnost tla	
r_c	Specifična otpornost izolacionog sloja gasovoda ($r_c = \rho_c \delta_c$)	Ω/m^2
D	Prečnik gasovoda	m
a	Poluprečnik gasovoda	m
a'	Ekvivalentni poluprečnik podzemnog gasovoda $a' = \sqrt{a^2 + 4h_p^2}$	m
h_p	Visina centra gasovoda	m
h'_p	Dubina podzemnog gasovoda	m
δ_c	Debljina izolacionog sloja gasovoda	m
α	Konstanta slabljenja gasovoda	m^{-1}
γ	Konstanta prostiranja gasovoda ($\gamma = \alpha + j\beta$)	m^{-1}
Z_c	Karakteristična impedansa kola gasovod-zemlja	Ω
λ_p	Karakteristična dužina gasovoda	m

Tabela 2.1.1. Električni parametri gasovoda

2.2. Proračun električnih parametara gasovoda

Preciznost proračuna zavisi od poznavanja električnih veličina gasovoda, i to:

- 1) redna impedansa gasovoda za nadzemne gasovode
- 2) redna impedansa gasovoda za podzemne gasovode
- 3) sopstvena impedansa gasovoda
- 4) admitansa nadzemnog gasovoda
- 5) admitansa podzemnog gasovoda
- 6) otpornost i specifična otpornost izolacionog sloja
- 7) prečnik i poluprečnik gasovoda
- 8) dubina podzemnog gasovoda
- 9) visina gasovoda u odnosu na površinu
- 10) debljina izolacionog sloja
- 11) konstanta slabljenja gasovoda
- 12) karakteristična impedansa gasovoda
- 13) konstanta prostiranja gasovoda
- 14) karakteristična dužina gasovoda

3. KAPACITIVNI UTICAJ

Kapacitivno sprezanje se dešava kod nadzemnih gasovoda u blizini nadzemnih elektroenergetskih vodova kao posledica kapacitivnosti između voda i gasovoda. Podzemni gasovodi nisu pod uticajem

kapacitivne sprege zbog zaštitnog efekta zemlje u odnosu na električno polje.

U zavisnosti od toga da li je gasovod uzemljen ili ne kapacitivna sprega izaziva pojavu napona između gasovoda i zemlje ili tok struje u gasovodu i njegovom uzemljivačkom sistemu.

Kapacitivni uticaj nadzemnog voda na obližnji gasovod zavisi od sledećih parametara:

- 1) napona na vodu - uticaj raste srazmerno sa naponom na vodu
- 2) rastojanja između voda i gasovoda - indukovani napon se smanjuje sa porastom rastojanja
- 3) režima rada - pri redovnom pogonu svaki provodnik ee voda je u sprezi sa gasovodom. Zbog različitih vrednosti kapacitivnosti, uticaji se poništavaju samo delimično. Rezultujuća struja i indukovani napon se dobijaju superpozicijom uticaja svih provodnika. Za sisteme sa više paralelno vođenih vodova raspored faza može biti od velikog značaja.
- 4) dužine pojasa uticaja - indukovani napon ne zavisi od dužine pojasa uticaja nadzemnog voda na gasovod. Struja koja prolazi kroz osobu u kontaktu sa gasovodom ili kroz uzemljenje gasovoda raste sa porastom dužine pojasa uticaja. Za paralelno vođeni vod i gasovod struja raste proporcionalno porastu dužine pojasa uticaja.

3.1. Proračun vrednosti kapacitivnog uticaja

Metode koje se koriste za proračun vrednosti kapacitivnog uticaja su klasičan metod koji se zasniva na korišćenju koeficijenata potencijala i pojednostavljena metoda.

3.1.1. Jednostavan metod za određivanje nivoa uticaja

Radi pojednostavljivanja, pojas uticaja se može izdeliti na sekcije koje se predstavljaju paralelizmima. Procedura proračuna je tada:

- 1) proračun napona u svakoj sekciji
- 2) proračun struja u uzemljenju svake sekcije
- 3) proračun struje jedinstvenog uzemljenja gasovoda (zbir struja u sekcijama)
- 4) proračun efektivne vrednosti indukovnog napona

3.1.2. Metod koeficijenata potencijala

Moraju se usvojiti pretpostavke:

- elektroenergetski vod i gasovod su dugački u odnosu na međusobno rastojanje. Međusobno su paralelni i paralelni su u odnosu na zemlju.
- efekat zemlje je uvažan korišćenjem teoreme likova.

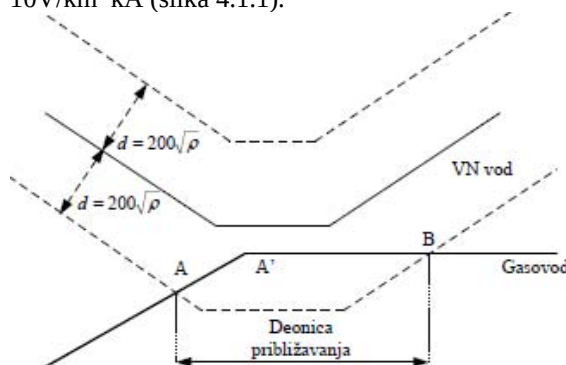
4. INDUKTIVNI UTICAJ

Nadzemni i podzemni gasovodi, čija trasa ide paralelno sa vodovima i kablovima i nalaze se u njihovoj blizini, podložni su indukovanim naponima prouzrokovanim magnetnim poljem koje stvara struja u vodu. Indukovana elektromotorna sila (EMS) uzrokuje pojavu struje u gasovodu i napone između gasovoda i okolnog tla. Induktivni uticaj visokonaponskog voda ili kabla na obližnji gasovod u osnovi zavisi od četiri parametra:

- 1) Intenziteta struje i pogonskog stanja mreže - prilikom redovnog pogona, indukovana EMS zavisi od struja u fazama. Tokom zemljospoja, amplituda

EMS zavisi od struje kvara. Indukovani naponi mogu biti mnogo veći nego u normalnom stanju, ali je njihovo trajanje kratko

- 2) Razdaljine između energetskeg voda i gasovoda - kao i za kapacitivan uticaj, rastojanje između gasovoda i energetskeg voda ima veliki značaj u određivanju indukovnog napona. Sa povećanim rastojanjem, uticaj razdaljine se smanjuje.
- 3) Dužine deonice približavanja - deonica približavanja predstavlja deo gasovoda koji se nalazi u pojasu uticaja energetskeg voda. Dužina deonice približavanja je vrednost ortogonalne projekcije gasovoda na elektroenergetski vod u pojasu uticaja. Smatra se da je pojas uticaja definisan predelom gde je indukovana EMS usled struje kvara sa povratnim putem kroz zemlju veća od $10\text{V/km}\cdot\text{kA}$ (slika 4.1.1).



Slika 4.1.1. Pojas uticaja i deonica približavanja

- 4) Prirode elektroenergetskog voda - uticaji od nadzemnog voda i podzemnog kabla u sličnim uslovima se mogu značajno razlikovati. Razlike su najizraženije usled redukcionog faktora kablova prilikom kvarova, kao i zbog manjeg rastojanja između faznih provodnika kod kablova, usled čega su manji poremećaji u redovnom pogonu.

4.1. Posledice induktivnog uticaja

Indukovani naponi mogu ugroziti bezbednost ljudi koji dođu u kontakt sa gasovodom koji se nalazi u blizini VN vodova. Domaći standard SPRS.N.CO 105-1987 dozvoljava napon dodira od 65V na neograđenim pristupačnim delovima gasovoda a 125V na ograđenim pristupačnim delovima gasovoda. Tokom trajanja zemljospoja dozvoljeni su veći naponi. Razlog za to je činjenica da struja kvara traje veoma kratko. Rizik posledica kvara je ograničen zbog male verovatnoće da će neko biti u kontaktu sa gasovodom u momentu kvara.

Tokom zemljospoja naponi mogu prekoračiti podnosive napone izolacionih spojeva. Isto važi i za opremu povezanu sa gasovodom, naročito za sistem katodne zaštite od korozije. Po SRPS.N.CO 105-1987 ispravljački uređaji, kao i ostali električni pogoni čiji su sastavni delovi i strujna kola provodljivo spojeni sa podzemnim delovima gasovoda, moraju se napajati preko zaštitnog transformatora za galvansko odvajanje.

4.2. Proračun indukovane elektromotorne sile

Proračuni napona koji se javljaju u gasovodima se uglavnom rade u dva koraka:

- 1) određivanje elektromotorne sile duž gasovoda
- 2) proračun napona prema zemlji i struja koje teku, a koji su posledica indukovane EMS.

Treba razlikovati EMS i napone koji se javljaju na gasovodima. EMS su virtuelni naponski generatori unutar kola gasovod-zemlja koji se pojavljuju usled uticaja induktivnog sprezanja. EMS uzrokuju napone u cevima, i samo ovi naponi predstavljaju stvarna naprezanja gasovoda i prateće opreme.

4.3. Određivanje indukovane elektromotorne sile

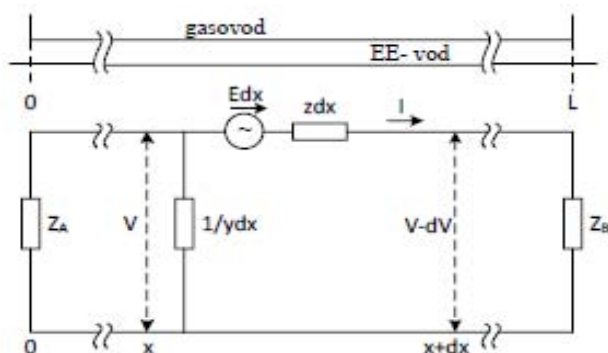
Indukovana elektromotorna sila se određuje za više različitih uslova. U zavisnosti od stanja mreže, postoje dva slučaja i to: određivanje stalne EMS pri redovnom pogonu i određivanje maksimalne kratkotrajne EMS tokom trajanja kvara.

4.4. Proračun napona na gasovodu

Za proračun treba usvojiti sledeće pretpostavke:

- da su gasovod i elektroenergetski vod paralelni
- da je admitansa gasovoda konstantna. Podužna otpor. izolacionog sloja je uniformna na celom gasovodu i ne zavisi od napona
- da je otpornost tla duž cele trase konstantna.

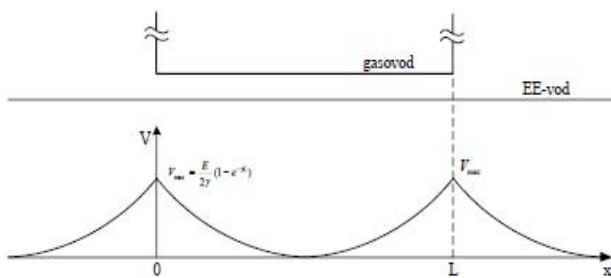
Na slici 4.4.1. prikazano je kolo gasovod-zemlja.



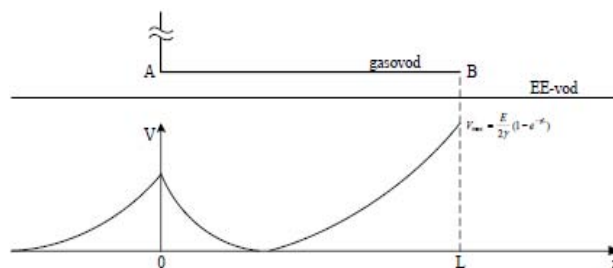
Slika 4.4.1. Kolo gasovod-zemlja

Proračun napona vrši se za više slučajeva:

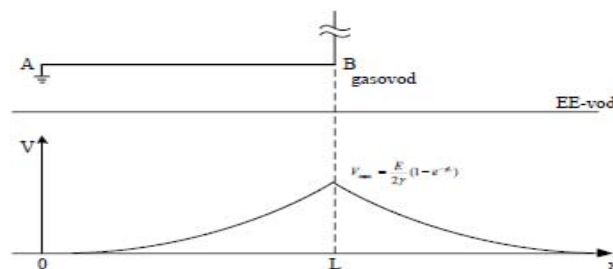
- 1) kada se gasovod nastavlja van pojasa uticaja bez uzemljenja (slika 4.4.2)
- 2) kada se gasovod nastavlja van pojasa na jednom kraju a na drugom se zaustavlja (slika 4.4.3.)
- 3) kada se gasovod na jednom kraju nastavlja van pojasa uticaja a na drugom kraju je uzemljen (slika 4.4.4).



Slika 4.4.2. Napon na gasovodu koji se prostire izvan pojasa uticaja



Slika 4.4.3. Napon na gasovodu koji se na jednom kraju prostire izvan pojasa uticaja



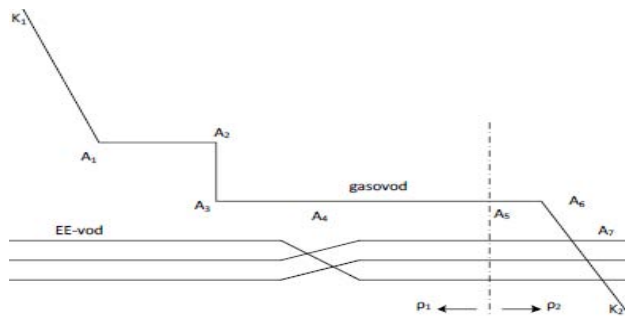
Slika 4.4.4. Napon na gasovodu koji je na jednom kraju uzemljen a na drugom se prostire izvan pojasa uticaja

4.5. Opšti slučaj

Proračun se zasniva na podeli pojasa uticaja na relativno veliki broj sekcija. Tada se određuje vrednost napona na velikom broju pozicija duž cevovoda.

4.6. Najkritičnije mesto kvara sa aspekta naprezanja na gasovodu

Najveće vrednosti potencijala na gasovodu, usled zemljospoja, se javljaju na tačkama diskontinuiteta električnih ili geometrijskih osobina sistema gasovod-elektroenergetski sistem. Te tačke su: spoj dve sekcije cevovoda gde se naglo menja rastojanje paralelizma, naspram mesta preplitanja faza na ee vodu, na granici dve vrste zemljišta sa drastično različitim specifičnim otpornostima, na mestima ukrštanja. Ove tačke su prikazane na slici 4.6.1.



Slika 4.4.4. Tačke diskontinuiteta u sistemu GV-EE vod

4.5. Mere i sredstva zaštite

U toku projektovanja mora se nastojati da elektroenergetski objekti i gasovodi ne budu locirani u pojasu uticaja tj. da njihova rastojanja budu što veća. Mere zaštite vezane za elektroenergetske vodove su: ugradnja zaštitnog užeta, redovno preplitanje faznih provodnika, adekvatan izbor rasporeda faza. Mere vezane za gasovode su vođenje provodnika paralelno sa gasovodom, ugradnja izolacionih spojeva, uzemljavanje gasovoda, izrada ekvipotencijalne nagazne rešetke.

5. GALVANSKI UTICAJ

Kada se desi zemljospoj na elektroenergetskom objektu struja koja protiče kroz uzemljivač uzrokuje porast potencijala uzemljivača i okolnog zemljišta u odnosu na referentnu zemlju. Galvanska sprega se ostvaruje između elektroenergetske opreme i obližnjeg gasovoda na jedan od sledećih načina:

- ako je gasovod direktno povezan sa uzemljivačem visokonaponskog sistema (unutar postrojenja),
- ako je gasovod unutar naponskog levka uzemljivača elektroenergetskog objekta. Velika razlika potencijala se pojavljuje duž izolacionog sloja gasovoda usled porasta potencijala okolnog tla. Galvanska sprega se u praksi najčešće ostvaruje na taj način.

5.1. Posledice galvanskog uticaja

Kao posledica galvanskog uticaja elektroenergetskog voda na gasovod može doći do povećanja napona dodira i koraka, oštećenja izolacionog sloja gasovoda, oštećenja metalne cevi gasovoda, oštećenja izolacionog spoja cevi, oštećenja sistema katodne zaštite.

5.2. Proračun uticaja galvanske sprege

Da bi se predvideo ovaj uticaj potrebno je odrediti:

- porast potencijala zemljišta u okolini gasovoda
- napon kojem je izložen izolacioni sloj gasovoda
- potencijal prenesen na gasovod
- napon kojem je izložen sistem katodne zaštite

5.3. Mere i sredstva zaštite od galvanskog uticaja

To mogu biti smanjivanje porasta potencijala tla na lokaciji cevovoda, povećanje dielektrične izdržljivosti izolacionog sloja cevovoda, ugradnja izolacionih spojeva visoke izdržljivosti itd. U svakom slučaju, minimalno rastojanje se mora ispoštovati (od 1m do nekoliko metara u zavisnosti od otpornosti tla). Kod galvanskog sprežavanja, povećavanje rastojanja ima dva efekta: sprečava pojavu luka između gasovoda i uzemljivača stuba elektroenergetskog voda pri zemljospoju i usporava degradaciju izolacionog sloja na gasovodu.

6. MERENJA

Merenja se izvode radi:

- određivanja kratkotrajnog uticaja tokom kvarova u EE sistemu
- određivanja trajnog uticaja pri redovnom pogonu
- predviđanja budućeg nivoa uticaja
- merenja sopstvene impedanse cevi gasovoda

7. PRIMER ANALIZE UTICAJA NADZEMNOG VODA DV 400kV NA BUDUĆU GASOVODNU MREŽU

Primerom je prikazan način određivanja dugoročne prognoze uticaja ee sistema na gasovod srednjeg pritiska za najnepovoljnije uslove (najveću struju kvara).

8. ZAKLJUČAK

Nakon detaljne analize uticaja elektroenergetskih vodova na gasovode dobijeni rezultati ukazuju da oni mogu biti veoma značajni.

Primenom mera zaštite, poput ugradnje izolacionih umetaka na prelazima nadzemni-podzemni gasovod, ograđivanja nadzemnih delova gasovoda i instalacija, primene najstrožijih uslova uzemljenja nadzemnih delova gasovoda, održavanje električne neprekidnosti gasovoda, privremeno uzemljavanje na mestu rada prilikom remonta ka, korišćenje zaštitne opreme i drugih, može se smanjiti ili potpuno izbeći negativne posledice ovih uticaja.

9. LITERATURA

- [1] Guide on the influence of high voltage AC power systems on metallic pipelines, Working Groupe 36.02, Electromagnetic Compatibility with telecommunication circuits, low voltage networks and metallic structures, *Cigre*, 1995
- [2] SRPS N.CO 105-1987, „Tehnički uslovi zaštite podzemnih metalnih cevovoda od uticaja elektroenergetskih postrojenja, Jugoslovenski standard“, *Službeni list SFRJ*, br. 68/86, 1985
- [3] David A. Weston, *Electromagnetic compatibility*, Marcel Dekker, 2001.
- [4] Henry W. Ott, *Electromagnetic Compatibility Engineer*, John Wiley and sons, 2009.
- [5] Ian B. Darney, *Circuit Modeling for Electromagnetic Compatibility*, SciTech Publishing, 2013 [6] J. L. Norman Viollette, Donald R. J. White, Micheale F. Violette, *Electromagnetic Compatibility Handbook*, Springer Science+Business Media New York, 1987
- [7] V. Prasad Kodali, *Engineering Electromagnetic Compatibility*, Institute of Electrical and Electronics Engineers, 1996
- [8] Interni standard IS – EMS 123:2014 Uzemljenje elektroenergetskih postrojenja, 2014
- [9] F. P. Dawalibi, R. Southey, J. Ma, Y. Li, „On the mechanisms of electromagnetic interference between electrical power systems and neighboring pipelines“, *NACE 2000 T10B Symposium on DC & AC Interference*, Orlando, March 26-31, 2000.
- [10] Earl L. Kirkpatrick, „Basic Concepts of Induced AC Voltages on Pipelines“, *Cathodic and anodic protection*, July 1995
- [11] Dan Doru Micu, Iosif Lingvay, Carmen Lingvay, Laura Cret, Emil Simion, „Numerical Evaluation of induced Voltages in the Metallic underground Pipelines“, *Revue Romaine des Sciences Techniques*, 53, 175-184, 2008
- [12] Francesco Lattarulo, *Electromagnetic Compatibility in Power Systems*, Elsevier New York, 2006

Kratka biografija:



Nemanja Jovičić rođen je u Novom Sadu 1986. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranio je 2015.godine.

KOMPARATIVNA ANALIZA STANDARDA H-264 I H-265

COMPARATIVE ANALYSIS OF H-264 AND H-265 STANDARDS

Milan Meljanac, Željko Trpovski, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu opisani su standardi za kodovanje videa H-262, H-263, H-264 kao i najnoviji standard H-265. Standard H-265 detaljno je analiziran, objašnjena je njegova sintaksa, arhitektura i tehnike kodovanja. Izvršeno je poređenje standarda H-265 sa H-264 eksperimentalnim putem. Eksperimenti su realizovani uz pomoć softverskog alata Handbrake.

Abstract – In this paper, video coding standards H-262, H-263, H-264 and also the latest video coding standard H-265 are described. Standard H-265 was analyzed in detail, its syntax, architecture and coding techniques are also explained. Experimental comparison of H-264 and H-265 is done. Experiments are done with Handbrake software tool.

Ključne reči: H-265, H-264, Standardi za kodovanje videa.

1. UVOD

Svedoci smo sve veće popularnosti gledanja video sadržaja na televizorima, računarima pa i mobilnim telefonima. Ono što je zajedničko za video sadržaje na svim platformama jeste da korisnici žele što veći kvalitet istih. Polako se od video sadržaja HD rezolucije (1920x1080) ide ka rezolucijama poput 4K (3840x2160), pa čak i ogromnih rezolucija poput 8K (7680x4320). Rešavanje problema gledanja takvih videa krije se u optimizaciji kodne efikasnosti odnosno smanjenja bitske brzine koja se koristi za njihov prenos. Tu na scenu stupaju standardi za kodovanje videa bez kojih bi bilo nemoguće gledanje takvih videa. Zahvaljujući standardu H-265 i njegovoj odličnoj kodnoj efikasnosti gledanje video sadržaja u 4K ili 8K rezoluciji je moguće bez ikakvih poteškoća. Iako je standard relativno nov, jer je nastao 2013.god, on je već sastavni deo televizora novije generacije koji su u stanju da prikazuju sadržaj koji je veće rezolucije od sada već uobičajene HD rezolucije.

2. ITU-T PREPORUKA H.264 (MPEG-4 AVC)

H.264/MPEG-4 AVC je drugi video standard razvijen od strane ITU-T VCEG i ISO/IEC MPEG. On i dalje koristi koncept makroblokova veličine 16x16, ali sadrži mnoge dodatne karakteristike. Makroblokovima se mogu podeliti na kvadratne ili pravougaone blokove veličina od 4x4 do 16x16 lumenentnih odbiraka. H.264/MPEG-4 AVC takođe podržava višestruke referentne slike. Vektori pomeraja prenose se koristeći preciznost 1/4 odbiraka u

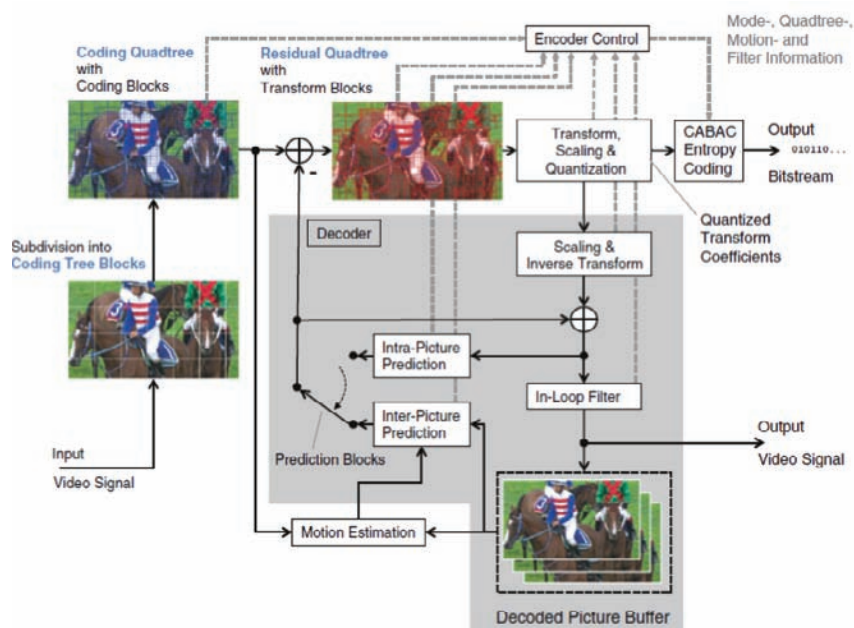
odnosu na lumenentnu rešetku odbiraka. Generalno, vektori pomeraja dobijaju se predikcijom medijana vektora pomeraja tri susedna prethodno dekodovana bloka. U odnosu na prethodne standarde za kodovanje, generalizuje se koncept slika. Umesto I, P i B slika standard navodi I, P i B slajs (engl. slice, traka blokova duž cele slike). Slika može da sadrži slajsove različitih tipova, a može se koristiti kao referentna slika za interpredikciju narednih slika. Za transformaciono kodovanje u H.264/MPEG-4 AVC koristi se 4x4 i 8x8 transformacija. Dok se za hrominentne blokove uvek koristi 4x4 transformacija, za lumenentne blokove veličina transformacije bira se na osnovu makrobloka. Za intra makroblokove veličina transformacije vezana je sa korišćenom veličinom bloka intra predikcije. U odnosu na prethodne standarde, inverzna transformacija opisuje se celobrojnim operacijama, tako da su u okruženjima bez grešaka rekonstruisane slike u koderu i deoderu uvek iste. Transformacioni koeficijenti predstavljaju se uz pomoć kvantizera uniformne rekonstrukcije. Slično H.262/MPEG-2 i H.264/MPEG-4 AVC takođe podržava korišćenje kvantizacionih težinskih matrica.

3. ITU-T PREPORUKA H.265 (HEVC - HIGH EFFICIENCY VIDEO CODING)

HEVC standard je najnoviji zajednički video projekat ITU-T (VCEG) i ISO/IEC (MPEG) organizacija za standardizaciju. Oni su zajedno radili u partnerstvu poznatom kao Joint Collaborative Team on Video Coding (JCT-VC). Prva verzija HEVC standarda završena je u januaru 2013. Glavni standard za kodovanje videa, koji je prethodio HEVC projektu bio je H.264/MPEG-4 AVC. H.264/MPEG-4 AVC je tehnologija koja je omogućavala digitalni video u skoro svakoj oblasti, koja nije prethodno bila pokrivena sa H.262/MPEG-2 standardom. Sada se koristi za mnoge primene, uključujući emitovanje HD TV signala putem satelita, kabla ili zemaljskih sistema prenosa, akviziciju digitalnog sadržaja, Internet i mobilni video, Blu-ray diskove itd. Međutim, povećavanje raznolikosti servisa, rastuća popularnost HD videa i pojavljivanje formata većih od HD formata (4Kx2K, 8Kx4K rezolucije) stvaraju još veće potrebe za kodnom efikasnošću superiornijom u odnosu na H.264/MPEG-4 AVC mogućnosti. Šta više, saobraćaj koji prouzrokuju video aplikacije namenjene mobilnim uređajima i tablet računarima, kao i potreba za prenosom servisa poput videa na zahtev, nameću ozbiljne izazove današnjoj mreži. HEVC je dizajniran kako bi se najpre pobrinuo za sve postojeće primene H.264/MPEG-4 AVC kao i da se naročito fokusira na dva ključna problema: povećanu video rezoluciju i povećano korišćenje arhitektura za paralelno procesiranje.

NAPOMENA:

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



Slika 1 - Tipični HEVC koder i dekodera (elementi dekodera su ošeceni sivom bojom)

3.1 Sloj za kodovanje videa

Sloj za kodovanje videa kod HEVC standarda koristi isti hibridni pristup (intra predikciju slike i 2-D transformaciono kodovanje) koji se koristi u svim standardima za kompresiju videa od H.261 standarda.

Slika 1 prikazuje blok dijagram hibridnog video koda, koji proizvodi binarni niz podataka koji odgovara HEVC standardu. Svaka slika se deli na regione u obliku blokova i to sa tačnim deljenjem blokova koje se prenosi do dekodera.

Prva slika video sekvence (i prva slika na svakoj slučajno odabranoj tački u video sekvenci) koduje se koristeći intra predikciono kodovanje slike. Za sve preostale slike sekvence ili između slučajno odabranih tačaka, koriste se, za većinu blokova, modeli vremenskih predikcionih kodovanja.

Proces kodovanja, za predikciju između slika, sastoji se od biranja podataka pomeraja koja se sastoje od odabrane referentne slike i vektora pomeraja (motion vector MV) koji se koriste za predikciju odbiraka svakog bloka.

Razlika između originalnog bloka i njegove predikcije transformiše se linearnom prostornom transformacijom. Transformacioni koeficijenti se onda skaliraju, kvantizuju, entropijski koduju i prenose zajedno za predikcionom informacijom.

Koder duplira dekodерову procesirajuću petlju (videti sivo ošecene blokove na slici 1) tako da će oboje generisati identične predikcije za naredne podatke. Stoga se kvantizovani transformacioni koeficijenti konstruišu inverznim skaliranjem i onda se inverzno transformišu kako bi se dobila dekodovana aproksimacija preostalog signala.

Ostatak se zatim dodaje predikciji, a rezultat tog dodavanja može biti unet u jedan od dva filtra kako bi se potisnuli artefakti koji nastaju usled blokovskog procesiranja i kvantizacije.

Konačni izgled slike (koja predstavlja duplikat izlaza iz dekodera) čuva se u baferu za dekodovanu sliku i koristi se za predikciju narednih slika.

3.2 Arhitektura sintakse visokog nivoa

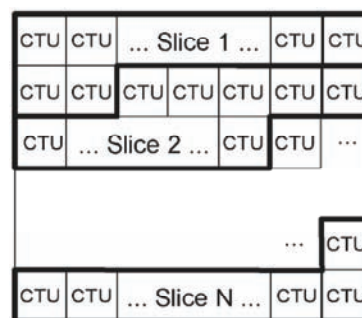
1) Slajs - Slajs je struktura podataka koja se može dekodovati nezavisno od drugih slajseva iste slike, u smislu entropijskog kodovanja, predikcije signala, i ostatka rekonstrukcije signala.

Slajs može biti cela slika ili region slike. Slajs je zapravo red kodnih stabala poređanih jedan pored drugog. Jedna od glavnih svrha slajsa jeste resinhronizacija u slučaju gubitka podataka.

Svaka slika u HEVC-u deli se na jedan ili više slajseva. Svaki slajs je nezavisan od drugih u smislu da se informacija, nošena u jednom slajsu, koduje bez zavisnosti od podataka od drugih slajseva unutar iste slike.

Slajs se sastoji od jednog ili više slajs segmenata, gde se prvi segment slajsa naziva nezavisni segment zato što je nezavisan od drugih slajs segmenata.

Sledeći slajs segment, ako postoji, naziva se zavisni slajs segment zato što zavisi od prethodnog. Izgled slajsa prikazan je na slici 2.



Slika 2 - Izgled slajsa

2) Tiles - Opcija da se slika podeli na pravougaone regione naziva se tiles. Glavna svrha tiles jeste povećanje mogućnosti paralelnog kodovanja. Tiles su nezavisno dekodabilni regioni slike koji se koduju sa nekom zajedničkom informacijom u zaglavlju.

Tiles mogu dodatno da se koriste za svrhu slučajnog pristupa lokalnim regionima video slika. Tipična konfiguracije tiles neke slike sastoji se od izdvojene slike u pravougaone regione sa približno jednakim brojem struktura kodnih stabala u svakom tiles. Podela slike na tiles prikazana je na slici 3.

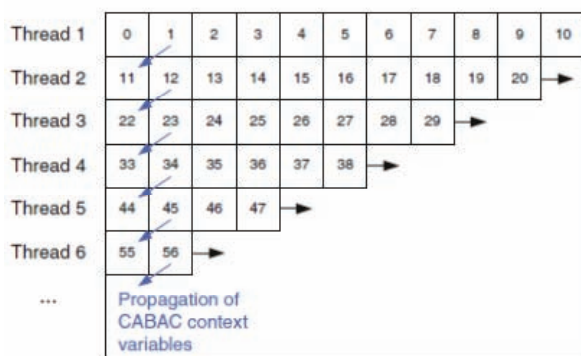
0	1	2	3	12	13	14	21	22	23	24
4	5	6	7	15	16	17	25	26	27	28
8	9	10	11	18	19	20	29	30	31	32
33	34	35	36	41	42	43	47	48	49	50
37	38	39	40	44	45	46	51	52	53	54
55	56	57	58	63	64	65	69	70	71	72
59	60	61	62	66	67	68	73	74	75	76

Slika 3 - Slika podeljena na 9 tiles

3) Wavefront paralelno procesiranje - Kada je omogućeno procesiranje, slajs se deli na redove kodnih stabala. Korišćenjem WPP svaka nit dekodera procesira jedan red kodnog stabla slike.

Prvi red se procesira na običan način, drugi red može da počne da se procesira nakon što se samo dva kodna stabla (engl. CTU - coding tree unit) procesiraju u prvom redu. Drugim rečima nit koja procesira prethodni red kodnih stabala mora da je procesirala dva kodna stabla više nego nit koja procesira trenutni red kodnih stabala.

Zbog toga nije omogućeno svim nitima da započnu procesiranje redova kodnih stabala istovremeno. Jedan problem WPP jeste što se dekodovanje neće završiti u isto vreme na kraju svakog reda. Može se desiti da sporo kodno stablo u redu kodnih stabala odugovlači u procesiranju sledećih redova kodnih stabala. Primer WPP procesiranja može se videti na slici broj 4.



Slika 4 - WPP procesiranje redova kodnih stabala

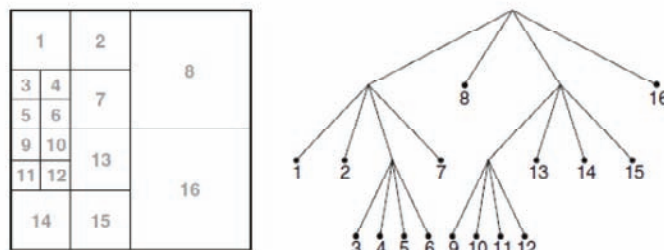
3.3 HEVC tehnike video kodovanja

Slika se deli na jedinice kodnih stabala (engl. CTU - coding tree units), od kojih svaka jedinica kodnog stabla sadrži lumentni CTB blok i hrominentni CTB blok. Lumentni CTB blok je veličine $L \times L$ odbiraka a hrominentni CTB blok pokriva $L/2 \times L/2$ odbiraka za svaki od dve hrominentne komponente.

Vrednost L može biti jednaka 16, 32 ili 64. Kod starijih standarda korišćena je fiksna veličina 16×16 lumentnih odbiraka. HEVC podržava promenljive veličine CTB blokova. Podrška za većim CTB-ovima nego u

prethodnim standardima je naročito korisna kada se koduje video sadržaj visoke rezolucije. Lumentni CTB i dva hrominentna CB zajedno formiraju CTU, osnovnu jedinicu za procesiranje.

Blokovi naznačeni kao lumentni i hrominentni CTB blokovi mogu se direktno koristiti kao CB blokovi ili se mogu dalje deliti na višestruke CB blokove. Podela se postiže koristeći strukturu stabla kao što je prikazano na slici 5.



Slika 5 - Primer podele 64×64 CTB bloka na CB blokove veličina od 8×8 do 32×32

Proces podele uz pomoć kvaternarnog stabla odvija se sve dok veličina CB bloka ne dođe do minimuma koji je dozvoljen.

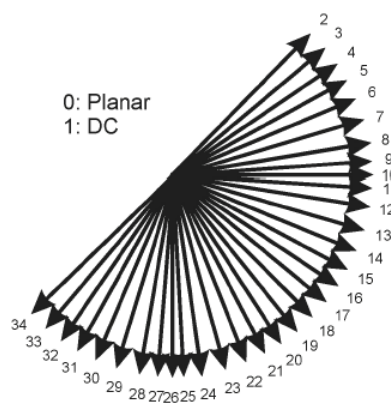
3.4 Intra predikcija slike

HEVC intra predikcione metode mogu se klasifikovati u dve kategorije. Metode ugaone predikcije formiraju prvu kategoriju i daju kodeku mogućnost da tačno modeluje strukture sa usmerenim ivicama. Metode u drugoj kategoriji, planarna predikcija i DC predikcija, omogućavaju procenu glatkog sadržaja slike.

Ukupan broj intra predikcionih režima koje HEVC podržava je 35 i prikazani su na slici 6. Svi HEVC intra predikcioni režimi koriste referentne odbirke od prethodno rekonstruisanih blokova. Intra predikcija slike funkcioniše prema veličini TB bloka.

Kod intra ugaone predikcije uglovi se dizajniraju kako bi postigli gušću pokrivenost za horizontalne i vertikalne uglove. Svaki transformacioni blok predviđa se od susednih blokova. Gledaju se blokovi sa leva, levo dole, gore i gore desno od trenutnog bloka.

Kod intra-dc predikcije koriste se odbirci sa leva i iznad blokova za koje se vrši predikcija, a kod planarne predikcije koriste se vrednosti dve linearne predikcije uz pomoć četiri referentna ugaona odbirka.



Slika 6 - Pravci intra predikcije

3.5 Inter predikcija

Inter predikcija je predikcija kod koje se koriste vrednosti iz prethodno dekodovanih slika. Za nju su nam neophodne referentne slike i vektori pomeraja. Jedan od problema koji se ovde može javiti vezan je upravo za vektore pomeraja.

Naime, kada su pomeraji jednaki tačno rastojanjima između piksela onda se lako mogu odrediti. Međutim u praksi se često dešava da pomeraji nisu jednaki rastojanjima između piksela.

HEVC u tu svrhu koristi razlomačku interpolaciju. Koriste se filtri sa 8 tačaka u odzivu za dobijanje polu pozicija odbiraka i filtri sa 7 tačaka u odzivu za dobijanje četvrt pozicija odbiraka.

3.6 HEVC transformacija

Transformacija se primenjuje na sliku razlike. Slika se deli na blokove veličina $N \times N$, a potom se svaki blok šalje kao ulaz dvodimenzionalne transformacije.

Transformacija se primenjuje kao jednodimenzionalna na svakom redu i koloni posebno. HEVC standard podržava jezgra transformacionih matrica veličina 4×4 , 8×8 , 16×16 i 32×32 .

Elementi transformacionih matrica izvedeni su iz aproksimiranih skaliranih DCT baznih funkcija. Za blok veličine 4×4 koristi se celobrojna transformacija izvedena iz DST.

Matrica za takvu transformaciju prikazana je na slici 7.

$$H = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}.$$

Slika 7 - Matrica za blok 4×4

3.7 In-Loop filtri

HEVC standard navodi dva in-loop filtra, deblokirajući filtar (Deblocking filter - DBF) i SAO (sample adaptive offset). Ovi filtri se primenjuju u kodujućim i dekodujućim petljama, nakon inverzne kvantizacije i pre čuvanja slike u baferu za dekodovanu sliku. Deblokirajući filtar se prvi primenjuje.

On slabi diskontinuitete na granicama predikcionih i transformacionih blokova. Drugi filtar, SAO, primenjuje se na izlazu deblokirajućeg filtra i on dalje poboljšava kvalitet dekodovane slike tako što smanjuje artefakte i promene u intenzitetima odbiraka u nekim regionima slike. Najvažnija prednost in-loop filtra je poboljšani subjektivni kvalitet rekonstruisane slike. Dodatno, koristeći filtre u dekodujućoj petlji poboljšava se kvalitet referentnih slika a time i efikasnost kopresije.

4. PRAKTIČNA IMPLEMENTACIJA KODOVANJA

Praktična implementacija kodovanja izvršena je uz pomoć programa Handbrake. Handbrake je jedan od, trenutno retkih, programa koji ima ugrađene kodere za H.264 MPEG-4 AVC, ali i najvažniji koder u ovom radu H.265 HEVC.

Program Handbrake je vrlo lak za korišćenje i nudi veliki broj opcija prilikom kodovanja. Korišćen je video rezolucije 1920×1080 piksela i veličine je 317 MB.

Nakon kodovanja videa sa H.264 koderom, dobio se video koji je imao 524 MB.

Kada se originalni video koduje sa H.265 koderom dobija se video koji ima 394 MB. Kvalitet videa koji je kodovan sa H.265 koderom bolji je nego onaj koji je kodovan sa H.264. Nakon toga izvršeno je kodovanje originalnog videa sa H.265 koderom i deblokirajućim filtrom. Rezultat je video koji ima veličinu 272 MB odnosno manje nego originalni video. Pokazalo se da takav video ima kvalitet koji je lošiji od videa dobijenog sa H.264 koderom.

5. ZAKLJUČAK

U eri sve veće potražnje za kvalitetnim video sadržajima visoke rezolucije jasno je da je neophodno razvijati alate koji će moći da izađu na kraj sa takvim problemom. Jedan od njih je svakako standard H.265 koji je izvesno uspešniji od svog prethodnika H.264. Naravno, standard je relativno nov i predstoji još dosta testiranja, upoređivanja, možda i nekih izmena, ali već prvi rezultati govore da on radi svoj posao odnosno omogućava veću kodnu efikasnost od svog prethodnika.

6. LITERATURA

- [1] dr Željen Trpovski, "Digitalna televizija", skripta sa predavanja, Novi Sad, 2015
- [2] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", Decembar 2012
- [3] Jens-Rainer Ohm, Gary J. Sullivan, Heiko Schwarz, Thiow Keng Tan, Thomas Wiegand, "Comparasion of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC)", Decembar 2012
- [4] Vivienne Sze, Madhukar Budagavi, Gary J. Sullivan, "High Efficiency Video Coding (HEVC)", Švajcarska, 2014
- [5] <https://handbrake.fr/> (Preuzeto u oktobru 2015)
- [6] <https://www.youtube.com/watch?v=6v2L2UGZJAM> (Preuzeto u oktobru 2015)
- [7] <https://hevc.hhi.fraunhofer.de/> (Preuzeto u septembru 2015)

Kratka biografija:



Milan Meljanac rođen je u Zrenjaninu 1991. godine. Upisao je smer Energetika, Elektronika i Telekomunikacije na Fakultetu tehničkih nauka Univerziteta u Novom Sadu 2010. godine. Diplomirao je 2014. godine i iste godine upisao master studije, takođe na Fakultetu tehničkih nauka.



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

ЈЕДНО РЈЕШЕЊЕ КОМПОНЕНТЕ ЗА УПРАВЉАЊЕ ДИСТРИБУИРАНИМ АКВИЗИЦИОНО УПРАВЉАЧКИМ СЕРВЕРИМА**ONE SOLUTION OF COMPONENT FOR MANAGING DISTRIBUTED SCADA FRONT-END PROCESSORS**

Драган Млађеновић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратки садржај – Тема овог рада је имплементација компоненте координатора дистрибуираним аквизиционо управљачким серверима (SCADA FEP) у оквиру система за напредно управљање дистрибуцијом електричне енергије (ADMS).

Abstract – The theme of this paper is the implementation of coordinator component for distributed acquisition and control servers (SCADA FEP) within the system for advanced energy distribution management (ADMS).

Кључне ријечи: SCADA, FEP, ADMS

1. УВОД

Тренд развоја електроенергетских система у правцу “smart grid” интегралног управљања као један од предуслова има покривање дистрибутивног дјела мреже SCADA телеметријом све до самих потрошача. Аквизиција овако великог обима података намеће изузетне захтјеве ка свим компонентама SCADA система, у погледу комуникационе пропусности, меморијске и процесорске моћи.

Као једно од могућих рјешења овог проблема, овај рад предлаже архитектуру SCADA система која се заснива на дистрибуцији функционалности везане за аквизицију и обраду података у виду компоненти које су лоциране ближе извору мјерних величина. У овом раду таква компонента се назива дистрибуирани аквизиционо управљачки сервер или SCADA FEP (енг. Front-End Processor). У предложеној архитектури FEP сервер шаље обрађене резултате аквизиције мјерних величина једној централној тачки која носи назив координатор. Поред примарне улоге управљања аквизицијом координатор је такође задужен за расподелу конфигурације и управљачких акција подређеним FEP серверима. Тема овог рада је управо имплементација координатора дистрибуираним аквизиционо управљачким серверима као и његова интеграција у оквиру једног комерцијалног ADMS система за напредно управљање дистрибуцијом електричне енергије.

Треба напоменути да је софтверско рјешење за SCADA FEP сервер базирано на редуктованом аквизиционо управљачком блоку dAub школског аквизиционо управљачког система dScada [1].

НАПОМЕНА:

Овај рад је проистекао из мастер рада чији ментор је био др Бранислав Атлагић, ванр. проф.

Имплементација самог сервера није покривена овим радом и сви детаљи везани за њу се могу пронаћи у раду „Једно рјешење дистрибуираног аквизиционог сервера и интеграција са ADMS софтвером“ [2], који заједно са овим радом чини цјелину. Читалац у том раду такође може пронаћи више детаља везаних за командну спрегу између координатора и ADMS система, који су у овом раду описани само из угла координатора.

2. ТЕОРИЈСКЕ ОСНОВЕ**2.1. Smart Grid**

Smart Grid је електроенергетска мрежа, која користи дигиталне и друге напредне технологије, за надгледање и управљање транспорта електричне енергије од свих извора до крајњих потрошача који имају различите захтјеве у потрошњи електричне енергије. Користећи информационе и комуникационе технологије, Smart Grid систем прикупља податке и аутоматски дјелује на основу информација о производњи и потрошњи, у циљу унапређивања ефикасности, економичности и одрживости производње и дистрибуције електричне енергије. [3][4][5].

2.2. ADMS

Један од првих корака у аутоматизација smart grid мреже је уградња дигиталних кућних бројила, али и покривање дистрибутивних мрежа SCADA телеметријом. Како тренутно већи дио дистрибутивне мреже није покривен SCADA телеметријом, SCADA рјешења се у овом сегменту проширују са DMS (енг. Distribution Management System) софтвером који се бави прорачуном недостајућих мјерења и оптимизацијом стања цијелог система. DMS рјешења се додатно проширују са OMS (енг. Outage Management System) софтвером који омогућава предикцију и праћење отказа на мрежи и проналажење њихових узрока, као и рестаурацију система после отказа. Заједно ове три компоненте чине јединствену софтверску платформу за напредно управљање електродистрибутивном мрежом, ADMS (енг. Advanced Distribution Management System).

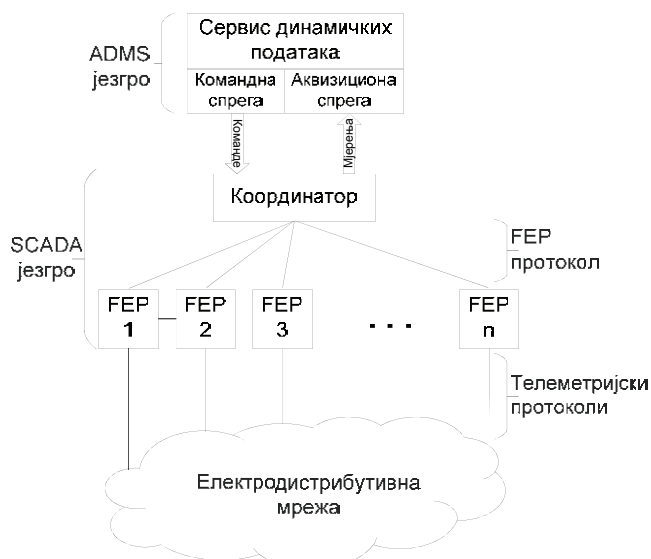
2.3. SCADA системи

Основни циљ аквизиционо управљачких система је обезбјеђивање ефикасног надзора и управљања над произвољним физичким процесом коришћењем дигиталних рачунарских компоненти и програмске подршке. Овакви системи најчешће се називају SCADA (енг. Supervisory Control and Data Acquisition) системима и обично подразумевају скуп

намјенских, просторно дистрибуираних рачунарских модула чији је заједнички циљ да остваре функције надзора и управљања физичким процесом у реалном времену. Основним задатком једног аквизиционо управљачког система сматра се циклична аквизиција дигитализованих одмјерака физичких величина, које одређују стање физичког процеса.

3. АРХИТЕКТУРА РЈЕШЕЊА

Већ је истакнуто да *Smart Grid* захтијева аквизицију велике количине података. Комплексност електро-дистрибутивне мреже повлачи за собом високе захтјеве у погледу физичке архитектуре и програмске подршке аквизиционо управљачких система. Квалитет рада ADMS софтвера директно зависи од перформанси SCADA система при аквизицији процесних контролера постављених у електро-дистрибутивној мрежи.



Слика 1. Преглед архитектуре рјешења

На слици 1 је представљен концепт архитектуре SCADA система, који је прилагођен за потребе надзора и контроле електродистрибутивних мрежа, и његова интеграција са ADMS софтвером.

3.1 Координатор

Координатор је централна компонента задужена за управљање FEP серверима. Реализован је као динамичка библиотека која је преко аквизиционе и командне спреге укључена у сервис динамичких података. Основни задаци координатора су дистрибуирање појединачних конфигурација свим подређеним FEP серверима, надгледање статуса FEP сервера, пријем обрађених процесних података од стране FEP сервера и њихово прослеђивање преко аквизиционе спреге, као и пријем команди преко командне спреге и њихова дистрибуција одредишним FEP серверима.

У својој конфигурацији координатор садржи комуникационе адресе свих FEP сервера, као и појединачне конфигурације за сваки FEP сервер које укључују дефиниције процесних контролера, њихових комуникационих параметара и дефиниције њима придружених процесних улаза/излаза. Координатор је задужен за дистрибуцију појединачних конфигурација одговарајућим FEP серверима.

Комуникација између координатора и FEP сервера је имплементирана по узору на клијент-сервер модел, посредством FEP протокола. FEP протокол је бинарни апликативни протокол базиран на интернет пакету протокола, осмишљен првенствено са циљем да се минимизује обим података који је потребно размјенити између координатора и FEP сервера како би се смањило притисак на комуникациону инфраструктуру између њих. Пошто је координатор „свјестан“ комуникационих адреса свих FEP сервера он иницира комуникацију, па је имплементиран као клијент.

FEP као серверска компонента је по свом покретању у пасивном режиму рада, чека док координатор не успостави везу са њим. По успостављању везе, координатор прослеђује FEP серверу његову конфигурацију, на основу које FEP креира и пуни своју базу података за рад у реалном времену. Након тога, FEP врши одговарајућу иницијализацију и припрема аквизицију процесних података. Током цијелокупног времена комуникације координатор аутоматски провјерава да ли су подређени FEP сервери активни, као и статус комуникационе везе између њих. Уколико уочи испад неког од FEP сервера или комуникационе инфраструктуре, координатор ће периодично покушавати да обнови везу са FEP сервером.

Аквизициони циклус FEP сервера не почиње док он не прими сигналну поруку од координатора. По пријему сигналне поруке FEP сервер започиње аквизицију процесних података. Након завршетка првог аквизиционог циклуса FEP сервер шаље координатору аквизициону поруку која садржи комплетно стање своје базе података, тј. садржи стања свих процесних контролера и њихових процесних промјенљивих на основу којих се врши поравнање стања између FEP сервера и сервиса динамичких података (енг. *integrity update*). Након тога FEP шаље аквизиционе поруке са стањем само оних процесних промјенљивих за које је детектована промјена.

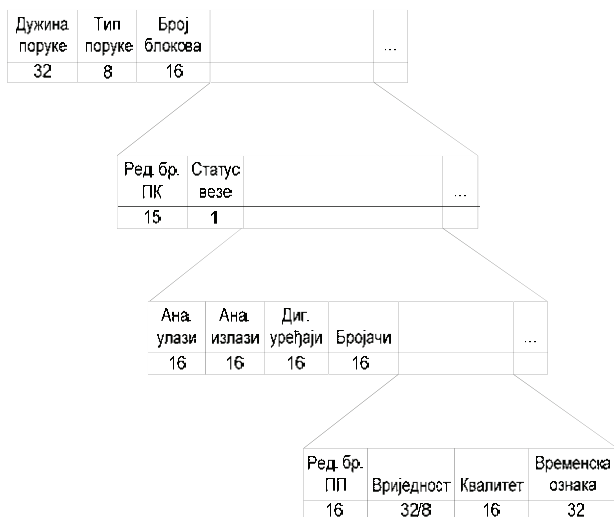
Координатор прима команде од ADMS софтвера преко командне спреге, било да су издате од стране дежурног оператора у контролном центру или су аутоматски генерисане у циљу оптимизације мреже. На основу своје конфигурације, координатор распоређује команде оним FEP серверима који су одговорни за дио мреже гдје се послане команде требају извршити.

4. ИМПЛЕМЕНТАЦИЈА

4.1 FEP протокол

Због потенцијалне географске удаљености између координатора и FEP сервера протокол за комуникацију између њих ја базиран на интернат инфраструктури, тачније TCP/IP групи протокола. Аквизиционе поруке садрже стања процених промјенљивих груписаних на основу типа и процесног контролера коме припадају. Формат аквизиционе поруке је приказан на слици 2. Порука на слици је подијељена на сегменте чије су дужине изражене у битима.

Порука започиње стандардним 32-битним заглављем које одређује дужину преосталог дијела поруке у бајтима, након чега слиједи 8-битно поље које идентификује тип аквизиционе поруке.



Слика 2 - Формат аквизиционе поруке

Постоје два типа аквизиционих порука. ACQ_UI поруке се шаљу након првог аквизиционог циклуса на FEP серверу и садрже стања свих процесних контролера и процесних промјенљивих које су у надлежности FEP сервера. У наставку се размењују ACQ_EX поруке које садрже само промене стања процесних промјенљивих. Следећих 16 бита након ознаке поруке представљају број блокова. Сваки блок преноси стање једног процесног контролера и њему придружених процесних промјенљивих. Блок почиње са редним бројем процесног контролера (16 бита) у конфигурационом моделу FEP сервера, што ограничава максимални број процесних контролера по FEP серверу на 32768. Следећи бит представља индикатор комуникационог статуса везе између процесног контролера и FEP сервера (0 - веза је активна, 1 - веза је прекинута). Уколико је вриједност индикатора 1 то представља крај текућег блока. У супротном, блок се наставља са четири 16-битне вриједности које представљају број стања аналогних улаза, излаза, дигиталних уређаја и бројача која се налазе у текућем блоку респективно. Након тога слиједе сегменти од којих сваки преноси стање једне процесне промјенљиве. Сегменти су груписани по типу процесне промјенљиве а групе прате редослед већ поменути четири вриједности. Сваки сегмент садржи следећа поља:

1. Редни број процесне промјенљиве у оквиру процесног контролера коме припада у оквиру групе промјенљивих истог типа (16 бита).
2. Вриједност која може бити изражена у инжењерским јединицама као 32-битни број у покретном зарезу дефинисан стандардом IEEE 754, у случају аналогних улаза/излаза и бројача; или као 8-битна вриједност чија битска поља представљају стање дигиталног уређаја.
3. 16-битно поље чији појединачни бити представљају ознаке квалитета процесне промјенљиве.
4. Цјелобројна 32-битна вриједност која означава временски тренутак детекције промјене процесне величине изражене у UNIX формату, броју протеклих секунди од почетка 1970. године.

4.2 Аквизициона спрега

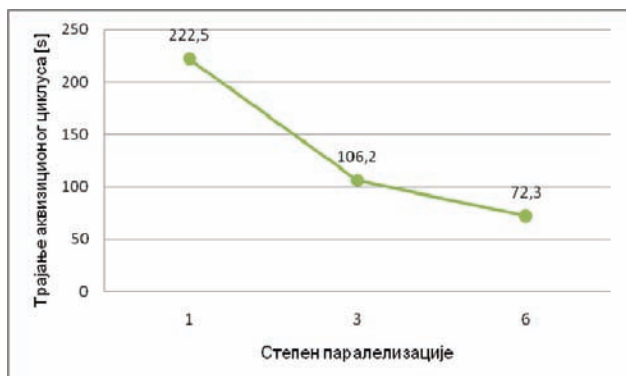
Аквизициона спрега представља прикључак сервиса динамичких података који омогућава упис промјена пристиглих са FEP сервера преко координатора у ADMS базу података модела динамике, одакле се даље пропaгирају компонентама заинтересованим за стање мреже. Модел динамике се заснива на концепту тачке која представља мјерни или извршни уређај постављен на терену. За сваку тачку се веже сигнал, где је сигнал еквивалент процесне промјенљиве. Аквизициона спрега мапира процесне промјенљиве на сигнале. Да би се убрзао упис у базу креира се меморијска табела која мапира четворку, коју чине редни број FEP сервера, редни број процесног контролера, редни број и тип процесне промјенљиве, на јединствени идентификатор сигнала. За сваки FEP сервер се креира посебан ред аквизиционих догађаја и посебна обрађивачка нит. Тако се догађаји са различитих FEP сервера могу обрађивати конкурентно уз очување редоследа догађаја на нивоу једног FEP сервера. Када се приликом обраде пристиглих промјена детектује прекид везе са процесним контролером, из локалне конфигурације координатора се преузимају све процесне промјенљиве које припадају датом контролеру и за њихове еквиваленте сигнале уписујемо заставицу квалитета *BrokenLink* (прекид везе са процесним контролером). Када се деси прекид везе са читавим FEP сервером, ова операција се понавља за сваки процесни контролер који му припада.

5. ТЕСТИРАЊЕ

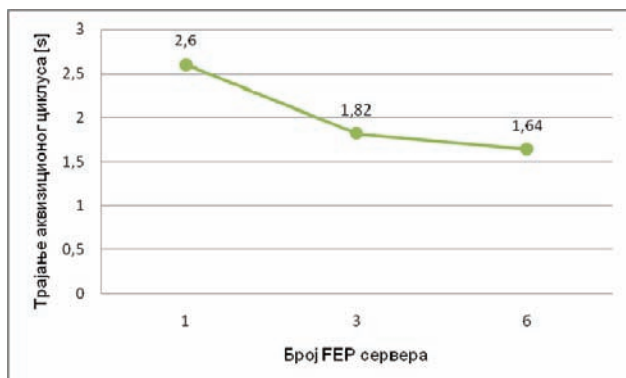
Комплетно програмско рјешење, које је интегрисано са ADMS системом, је тестирано над подацима из електродистрибутивне мреже града Новог Сада. Поред потврде коректности саме имплементације, вршено је и поређење перформанси у односу на комерцијално SCADA рјешење које се већ користи у склопу истог ADMS система.

Тестно окружење се састоји од два рачунара која су повезана локалном мрежном инфраструктуром. Први рачунар има процесор *Intel Core i7-3820 3.60GHz*, 24GB RAM меморије и *Windows 7 Enterprise 64-bit* оперативни систем. На овом рачунару је инсталиран ADMS софтвер. На другом рачунару су покренути симулатори процесних контролера у трансформаторским подстанцима. Симулатори су подешени да користе MODBUS TCP протокол, варијанту небалансираног MODBUS протокола који се користи у *Ethernet* окружењу. За потребе тестирања овог рјешења, на првом рачунару је подешен координатор и покренути FEP сервери као одвојени процеси. У другом случају на овом рачунару је подешена и покренута комерцијална SCADA. Тестирање овог рјешења је вршено над конфигурацијама за један, три и шест FEP сервера. На сличан начин, приликом тестирања комерцијалног SCADA рјешења, које подржава подешавање нивоа паралелизације, тестови су спроведени у случајевима када је SCADA паралелизована на један, три и шест нивоа. Праћено је вријеме пуног аквизиционог циклуса и вријеме извршавања појединачних команди. На сликама 3 и 4 се налазе графици који приказују однос трајања пуног

аквизиционог циклуса (у секундама) комерцијалног SCADA рјешења и рјешења заснованог на FEP серверима у односу на ниво паралелизације и број FEP сервера, респективно.



Слика 3 - Однос трајања аквизиције и нивоа паралелизације комерцијалног SCADA рјешења



Слика 4 - Однос трајања аквизиције и броја FEP сервера

Као што се може видјети из претходних графика, убрзање пуне аквизиције у случају рада са шест FEP сервера у односу на комерцијално SCADA рјешење износи приближно 44 пута.

Што се тиче извршавања команди, мјерено је вријеме које протекне од тренутка када ADMS пошаље команду преко коадне спреге, те све док резултат командовања не дође на ADMS преко аквизиционе спреге. Комерцијална SCADA (у случају паралелизације на шест нивоа) читав овај круг затвори за 2.9 секунди, док рјешењу које користи шест FEP сервера за то треба 0.47 секунди, што значи да је командовање брже нешто више од 6 пута. Просјечном брзином извршавања команди од 0.47 секунди би нпр. било могуће промијенити стање преко петнаест хиљада прекидача у року од два сата за потребе редукције вршних оптерећења у потрошњи електричне енергије (енг. *demand response*).

У случају рада са шест FEP сервера, вријеме које протекне док промјена појединачне процесне величине са FEP-а дође до ADMS софтвера износи 5 до 10 милисекунди. По извјештају америчког министарства за енергетику, ово вријеме задовољава брзину детектовања промјене на терену приликом управљања електродистрибутивним мрежама [5].

7. ЗАКЉУЧАК

У оквиру овог мастер рада је имплементирана компонента координатора SCADA FEP серверима у виду динамичке библиотеке базиране на .NET радном оквиру. Библиотека је интегрисана у сервис за руковање динамичким подацима у оквиру постојећег ADMS система ослањајући се на тачке проширења које дати сервис омогућава. Развијен је апликативни протокол за комуникацију са FEP сервером. На самом крају је извршено тестирање на којем је ово рјешење показало добре перформансе што оправдава даљи истраживачки рад у овом правцу.

Приказано рјешење само по себи није комплетно али представља полазну основу за даљи развој. Тренутни недостаци и могући даљи правци развоја:

- Имплементација подршке за динамичку расподелу аквизиционих послова између FEP сервера на основу података о њиховом оптерећењу.
- Подршка за расподелу аквизиционих послова једног FEP сервера на остале, уколико се догоди трајни испад неког од њих.
- Увођење механизма сигурности и аутентификације страна у протокол за комуникацију са FEP сервером.
- Увођење помоћне компоненте које ће омогућити системском администратору да ручно прекида и започиње сесије ка FEP серверу независно од алгорита за поновну успоставу сесије.
- Подршка за динамичку реконфигурацију којом би се омогућила промјена конфигурације координатора за вријеме рада система како би се повећала његова расположивост избегавајући ситуације у којима морамо приступити рестарту сервиса динамичких података да бисмо учитали нову конфигурацију

8. ЛИТЕРАТУРА

- [1] Бранислав Атлагић, Софтвер са критичним одзивом у електроенергетским системима, Нови Сад 2014,
- [2] Слободан Вулић, Једно рјешење дистрибуираног аквизиционог сервера и интеграција са ADMS софтвером, мастер рад, 2015,
- [3] International Energy Agency, Technology Roadmap - Smart Grids, 2011,
- [4] Vincenzo Giordano, Ettore Bompard, Assessing Smart Grid Benefits and Impacts: EU and U.S. Initiatives, 2012,
- [5] Department of Energy, Communications Requirements of Smart Grid Technologies, October 2010

Кратка биографија:

Драган Млађеновић рођен је 1991. године у Тузли. Факултет техничких наука у Новом Саду уписао је 2010. године, смјер Рачунарство и аутоматика. На истом факултету је 2014. уписао мастер академске студије смјер Електроенергетски софтверски инжењеринг.

ИМПЛЕМЕНТАЦИЈА ТЕКСТУАЛНЕ СИНТАКСЕ ЗА EUIS DSL**IMPLEMENTATION OF A TEXTUAL SYNTAX FOR EUIS DSL**

Зоран Тркуља, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У раду је представљено креирање текстуалне синтаксе за језик EUIS DSL у оквиру Кроки алата употребом Бакус-Наурове форме BNF (Bacus-Naur Form), проширене Бакус-Наурове форме EBNF (Extended Bacus-Naur Form) и ANTLR (Another Tool For Language Recognition) алата.

Abstract – This paper presents implementation of textual syntax for EUIS DSL within Kroki tool by using Bacus-Naur Form, Extended Bacus-Naur Form and ANTLR (Another Tool For Language Recognition) tool.

Кључне речи: Кроки, BNF, EBNF, ANTLR, синтакса, парсирање, таб функционалност

1. УВОД

У овом раду биће описано креирање текстуалне синтаксе за EUIS (Enterprise User Interface Specification) DSL (Domain Specific Language) за потребе Кроки алата. За дефинисање синтаксе користиће се метасинтаксе, при чему ће најзаступљеније бити Бакус-Наурова форма (Bacus-Naur Form) [1] и проширена Бакус-Наурова форма (Extended Bacus-Naur Form) [2] док ће се за парсирање синтаксе користити ANTLR (Another Tool For Language Recognition) [3]. Коришћење креиране синтаксе биће приказано на примерима у оквиру конзолног дела Кроки алата. Кроки (фр. *croquis* - скица) алат [4] је намењен за интерактивни развој пословних апликација базираних на скицама.

Конзолни део Кроки алата треба да се тако прилагоди да омогући брзо креирање модела пословних апликација, као и снимање, учитавање и парсирање историје команди у оквиру развијеног језика.

2. ОПИС КОРИШЋЕНИХ ТЕХНОЛОГИЈА**2.1. Метасинтакса**

Главна намена граматике је једнозначно задавање синтаксе, тако да се провера синтаксне исправности појединих језичких елемената може вршити формалним поступком који се може аутоматизовати имплементацијом преводиоца. За ту сврху математички приказ граматике није довољно погодан јер се елементи програма (константе, идентификатори, делови наредби, целе наредбе итд.) тешко могу представити помоћу појединачних симбола, а да та представа буде јасна. Такође, сродне продукције погодно је груписати, опет у циљу концизности и разумљивости.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији је ментор била др Гордана Милосављевић.

На пример, за приказивање синтаксног правила облика *нетерминал* $\rightarrow$ *цифра* неопходно би било написати десет посебних продукција које би сасвим личиле једна на другу што би опис учинило гломазним. Коначно, конститутивне граматике, поред примене у аутоматској синтаксној анализи, користе се у уџбеницима или приручницима за навођење свих дозвољених облика елемената неког језика за шта је уобичајена математичка нотација неадекватна јер јој недостају многи симболи.

Због тога, за практично представљање граматике користе се друга средства. Постоји већи број оваквих система, тзв. метасинтаксе од којих издвајамо Бакус-Наурову форму и проширену Бакус-Наурову форму.

Бакус-Наурова форма (BNF) је конвенција за записивање правила контекстно-слободних језика. Проширена Бакус-Наурова форма (EBNF) додаје одређене синтактичке изразе БНФ нотацији и омогућава једноставнији запис граматике.

БНФ нотација је типична метасинтакса која се састоји од метасимбола, метаоперација и формула. Као и друге метасинтаксе, једноставна је, јер треба да допринесе концизности и разумљивости задавања синтаксе програмског језика, а не обрнуто. Елементи БНФ нотације су сложени објекти (нетерминали), терминали, додела, ексклузивна дисјункција и репетиција.

ЕБНФ нотација је изведена је из основне БНФ нотације. Уз нека проширења и измене обухвата све конструкције Бакус-Наурове форме. ЕБНФ има исту моћ изражавања као и БНФ са тим што су у њој извршене измене које доприносе читљивости записа правила. ЕБНФ обухвата све конструкције БНФ-а уз следеће измене:

1. Нетерминали се наводе без посебних ограничавања $<$ и $>$; дакле, уместо $<\text{слово}>$ у ЕБНФ се одговарајући нетерминал наводи само именом: *слово*.
2. Терминали се, у сврху разликовања од нетерминала, пишу под знацима навода, нпр. „0“ „7“ „if“ „A“
3. Уместо знака $::=$ за метарелацију доделе користи се знак једнакости $=$.
4. Конструкција $\{ \text{метаизраз} \}$ значи појављивање метаизраза 0 или више пута (код БНФ подразумевало се најмање једнпут).
5. Конструкција $[\text{метаизраз}]$ значи појављивање 0 или једанпут (тј. у питању је опциони метаизраз).
6. Уведене су мале заграде „(…)“ за груписање делова метаизраза.
7. Формула се обавезно завршава тачком.

2.2. ANTLR

ANTLR или *Another Tool For Language Recognition*, је генератор парсера који подржава LL(*) парсирање. ANTLR је наследик *Purdue Compiler Construction Tool Set (PCCTS)*, створен 1989. године, а и даље је у активном развоју. Особа која се стара о развоју ANTLR-а је професор *Therence Parr* са Универзитета Сан Франциско.

ANTLR као улаз узима граматiku одређеног језика и генерише изворни код преводиоца за тај језик. Језик се дефинише коришћењем контекстно независне граматике изражене проширеном Бакус-Науровом формом (ENFB).

ANTLR може да генерише лексере (*lexers*), парсере, стабло парсера и комбиноване лексере-парсере (*lexer-parsers*). Парсери аутоматски генеришу апстрактно синтаксно стабло (*abstract syntax trees*) које касније може бити обрађено са *tree parser*. ANTLR пружа конзистентну нотацију за навођење лексера, парсер и *tree parser*-има. Ово је у супротности са осталим парсер/лексер генераторима и додаје лакоћу коришћења алата.

ANTLR парсери користе технологију названу *Adaptive LL*()* или *ALL*()* ("all star") коју су развили професори *Therence Parr* и *Sam Harwell*. *ALL*()* је наставак *LL*()* технологије из треће верзије ANTLR-а која извршава граматичку анализу динамички.

2.3. Кроки алат

Кроки (фр. *croquis* - скица), је алат намењен за интерактивни развој пословних апликација базираних на скицама.

Кроки имплементира конкретну синтаксу EUIS (*Enterprise User Interface Specification*) DSL-а (*Domain Specific Language*), која је пројектована тако да се омогући моделовање корисничког интерфејса на „природан“ начин. Кроки подржава скицирање више врста форми, поштујући стандард који дефинише њихов изглед и функционалност.

EUIS DSL је имплементиран у виду UML профила [5], који је даље коришћен при изради Кроки алата. Профил поседује бројне стереотипе који представљају проширене метакласе *Element*, *Class*, *Property*, *Operation*, *Parameter*, *Constraint* и *Package* из пакета *UML::Kernel*, и реализовани су у виду интерфејса који декларишу методе за прибављање и подешавање вредности обележја која су дефинисана унутар класа елемената UML профила.

Основни елемент корисничког интерфејса представља стереотип *VisibleElement*, који проширује метакласу *Element*. Видљивим елементима придружује се компонента корисничког интерфејса (текстуално поље, дугме, *comboBox*, итд.) и натпис.

Видљиве класе проширују метакласу *Class*, репрезентујући класе које се пресликавају на панеле различите врсте.

Стандардни панел увек се придружује некој перзистентној класи, док *parent-child* панел омогућује хијерархијску организацију других стандардних и *parent-child* панела.

Видљива обележја проширују метакласу *Property*, омогућавајући пресликавање атрибута видљивих класа на компоненте корисничког интерфејса. Видљиви крајеви асоцијације проширују метакласу *Property* и могу се применити на видљиво навигабилно обележје које припада бинарној асоцијацији успостављеној између две видљиве класе. Дефинише однос између панела придруженог класи која је власник обележја и панела који припада класи на другом крају асоцијације. При имплементацији алата, како би се увела ова функционалност, креирана је класа *VisibleAssociationEnd*. Конкретна природа односа између панела дефинисана је врстом примењеног стереотипа, те се уочавају *Next*, *Zoom* и *Hierarchy*. Све класе које имплементирају видљиве крајеве асоцијације поседују обележја која одређују активациони и циљни панели, на основу чега се може закључити о томе који су панели повезани и да ли се ради о бидерекционој вези.

3. КРЕИРАЊЕ ТЕКСТУАЛНЕ СИНТАКСЕ ЗА EUIS DSL

3.1. Креирање програмског језика

Пошто се сматра да ће се програмски језик Кроки тренутно користити само за унос команди у конзолном делу Кроки алата, креирање синтаксе биће базирано на већ постојеће функционалности и потребе које Кроки алат већ поседује.

Као основне команде које ће се извршавати је креирање новог пројекта, пакета и панела, стандардног и *parent-child*. Команда које следи наредбу креирања стандардног панела и *parent-child* панела јесте креирање компоненти везано за сваки од панела.

Тако се за стандардни панел вежу компоненте: панел (*Panel*), поља за уноса текста (*Textfield* и *Textarea*), *ComboBox*, *Checkbox*, *Report*, *Transaction* и *Link*, док се за панел *parent-child* веже компонента *Hierarchy*. *Combозoom* компонента се веже за стандардни панел и она се трансформише из компоненте *ComboBox*.

Синтаксу се дели на две компоненте:

1. Лексичка структура. Лексичка структура одређује како се низови карактера у тексту програма издвајају и категоришу у токене (знакове).
2. Синтаксна структура. Синтаксна структура језика одређује како се токени групишу у сложеније структуре. Синтакса структура често се референцира као граматика.

Као основу за креирање синтаксе и неки почетни ниво реализације синтаксе узет је скуп команди које ће бити имплементирани у складу са већ имплементираним акцијама Кроки алата, тако да ће се првенствено приступити синтаксној структури. При томе команде на које је стављен највећи акценат су команда за креирање пројекта, пакета, стандардног панела, *parent-child* панела, компоненти (овде се мисли на компоненте стандардног панела), *Combозoom* компоненте као трансформацију компоненте

Combobox и компоненте *Hierarchy* (ова компонента је валидна само ако се користи са панелом *parent-child*).

Све наведене команде креирања могу се сврстати у једну команду. Скуп команди креирања биће назван *makeCommand*. Користећи ЕБНФ нотацију ову команду тј. скуп команди дефинишемо као:

```
makeCommand ::= makeProject | makePackage |  
makeStdPanel | makeParentChildPanel | makeComponen  
| makeCombozoom | makeHierarchy ;
```

У команди *makeCommand* знак | означава алтернативу, тако да позивом команде *makeCommand* тај позив се односи на све команде које су наведене у тој команди. На слици 3.1 приказан је део синтаксе на којој се виде команде за креирање елемената Кроки алата.

```
makeCommand ::= makeProject | makePackage | makeStdPanel | makeParentChildPanel |  
makeComponent | makeCombozoom | makeHierarchy ;  
makeProject ::= make, project, "", elementName, "" ;  
makePackage ::= make, package, "", elementName, "", in, "", elementName, [{"", elementName}], "" ;  
makeStdPanel ::= make, std-panel, "", elementName, "", in, "", elementName, [{"", elementName}], [{"",  
components, "": ""] ;  
makeParentChildPanel ::= make, pc-panel, "", elementName, "", in, "", elementName, [{"",  
elementName}], "" ;  
makeComponent ::= make, component, [{"", components, "": ""] ? in, "", elementName, [{"",  
elementName}], "" ;  
makeCombozoom ::= make, combozoomComponent, in, "", elementName, [{"", elementName}], "" ;  
makeHierarchy ::= make, hierarchyComponent, in, "", elementName, [{"", elementName}], "" ;
```

Слика 3.1 Приказ синтаксе команди креирања елемената Кроки алата

Приказ наведених команди се сматра као почетни ниво, тако да се даље приступа дефинисању сложенијих елемената команди. Овде се првенствено мисли на компоненте стандардног панела.

Стандардни панел садржи компоненте као што су *Panel*, *Textfield*, *Textarea*, *Combobox*, *Checkbox*, *Report*, *Transaction*, *Link* и *Combozoom*.

Део синтаксе који садржи компоненте стандардног панела приказан је на слици 3.2

```
panelComponent ::= panel, Label, elementName, [Visible, trueFalseElement],  
[BackgroundColor, elementName], [ForegroundColor, elementName], [FontColor, elementName],  
[BorderColor, elementName], [Orientation, orientationElement], [Alignment, alignmentElement] ;  
commonComponentAttributes ::= [Visible, trueFalseElement], [BackgroundColor, elementName],  
[ForegroundColor, elementName], [FontColor, elementName], [DisplayFormat, elementName],  
[Mandatory, trueFalseElement], [Representative, trueFalseElement], [AutoGo, trueFalseElement],  
[Disabled, trueFalseElement], [DefaultValue, elementName], [LabelTextCode, trueFalseElement],  
[ColumnName, elementName], [PersistentType, number], [Length, number], [Precision, number]  
 ;  
textfieldComponent ::= textfield, Label, elementName, DataType, textfieldDataType,  
[commonComponentAttributes] ;  
textareaComponent ::= textarea, Label, elementName, [commonComponentAttributes] ;  
comboboxComponent ::= combobox, Label, elementName, [commonComponentAttributes],  
[{Values, elementName}] ;
```

Слика 3.2 Приказ синтаксе креирања компоненти стандардног панела

Панел *parent-child* садржи само *Hierarchy* која се дефинише као на слици 3.3

```
hierarchyComponent ::= hierarchy, Label, elementName, [Visible, trueFalseElement],  
[BackgroundColor, elementName], [ForegroundColor, elementName], [FontColor, elementName],  
[Add, trueFalseElement], [Update, trueFalseElement], [Copy, trueFalseElement],  
[Delete, trueFalseElement], [Search, trueFalseElement], [ChangeMode, trueFalseElement],  
[DataNavigation, trueFalseElement], [Mandatory, trueFalseElement],  
[DefaultViewMode, defaultViewModes], [DefaultOperationMode, defaultOperationModes],  
[ConfigureDelete, trueFalseElement], [SayAddMode, trueFalseElement],  
[GoToLastAdded, trueFalseElement], [DataFilter, elementName], [SortBy, elementName],  
[ActivationPanel, elementName], [TargetPanel, elementName], [AppliedToPanel, elementName],  
[HierarchyPanel, elementName] ;
```

Слика 3.3 Приказ синтаксе *Hierarchy* компоненте *parent-child* панела

Сваки од атрибута који је део компоненте дефинисан је променљивом. У командама креирање компоненти атрибуту почињу првим великим словом.

Дефинисање променљивих приказано је на слици 3.4

```
letter ::= "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" |  
"R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" |  
"l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z" ;  
whiteSpace ::= " " ;  
elementName ::= letter, { letter | number | whiteSpace } ;  
trueFalseElement ::= True | False ;  
orientationElement ::= Horizontal | Vertical | Free ;  
alignmentElement ::= Left | Right | Central ;  
textfieldDataType ::= String | Integer | Long | BigDecimal | Date ;  
persistentTypes ::= Char | Varchar | Text | Integer | Number | Float | Decimal | Boolean | Date | Time |  
DateTime ;  
defaultViewModes ::= TableView | InputParameter ;
```

Слика 3.4 Дефинисање променљивих од атрибута

3.2 Коришћење ANTLR алата за парсирање синтаксе

Синтаксу у ANTLR делимо на део везан за парсер и лексер. У делу везан за парсер биће смештане команде за креирање компоненти Кроки алата док у делу који је везан за лексер се смештају атрибуту, променљиве атрибута и карактери од којих се састоје променљиве. Команде из парсера означавамо са почетним малим словом док за атрибуте, променљиве атрибута и карактере од којих се састоје променљиве означавамо све са великим словом, на тај начин разликујемо између парсера и лексера. Празна поља, табови и сепаратори који могу да раздвоје ANTLR речне симболе као што су идентификатори, се игноришу. Тако на пример, „FirstName LastName“ се по ANTLR-у представља као низ од два токена а не токен, празно поље и токен. Наредба са којом избегавамо празна поља, табове и сепараторе је *WHITESPACE* : [\t\n\r]+ -> channel(HIDDEN) ; Синтакса програмског језика Кроки у ANTLR фајлу изгледа као на слици 3.5. Ово је иначе само део кода.

```
// PARSE  
makeCommand : (makeProject | makePackage | makeStdPanel | makeParentChildPanel | makeComponent  
| makeCombozoom | makeHierarchy)+ ;  
makeProject : MAKEPROJECT "" ELEMENTNAME "" ;  
makePackage : MAKEPACKAGE "" ELEMENTNAME "" IN ELEMENTNAME (" " ELEMENTNAME)? "" ;  
makeStdPanel : MAKESTDPanel "" ELEMENTNAME "" IN ELEMENTNAME (" " ELEMENTNAME)+  
(LBRACKET (COMPONENTS (" ")+)? ) ;  
makeComponent : MAKECOMPONENT LBRACKET (COMPONENTS (" ")+ ? " " IN ELEMENTNAME (" "  
ELEMENTNAME)+ ;  
makeParentChildPanel : MAKEPARENTCHILDPanel "" ELEMENTNAME "" IN ELEMENTNAME (" "  
ELEMENTNAME)+ (LBRACKET (HIERARCHYCOMPONENT (" ")+ ? ) ) ;  
makeCombozoom : MAKECOMBOZOOM IN ELEMENTNAME (" " ELEMENTNAME)+ ;  
makeHierarchy : MAKEHIERARCHY IN ELEMENTNAME (" " ELEMENTNAME)+ ;  
  
// LEXER  
NUMBER : "0".."9" ;  
LETTER : ("A".."Z" | "a".."z")+ ;  
WHITESPACE : (" " | "\t" | "\n" | "\r")+ -> channel(HIDDEN) ; // to skip empty space -> channel(HIDDEN)  
ELEMENTNAME : LETTER ( LETTER | NUMBER )+ ;
```

Слика 3.5 Приказ синтаксе програмског језика Кроки у ANTLR фајлу

3.3 Имплементирање таб функционалности конзолног дела Кроки алата

Конзолни део Кроки алата омогућава лако креирање елемената Кроки алата. Да би се омогућило лакше креирање потребно је смањити брзину куцања и брже креирање елемената. Таб функционалност која је присутна у нпр. програму *Windows Command Prompt* значајно омогућава лакше и брже куцање наредби у том истом програму, тако да би омогућавање таб функционалности у конзолном делу Кроки алата значајно убрзао и олакшао рад.

Куцање наредби зна некада да буде напорно тако да се применом таб функционалности убрзава рад и ово значи да се на тастер таб препознаје која реч тј. команда почела да се куца. Пример таб функцио-
налности је приказан на слици 3.6 у случају када од почетних карактера *ma* креирамо команду *make project* „NewProject“



Слика 3.6 Слика употребе таб функционалности при креирању команде за креирање пројекта

Успешно креирање новог пројекта приказано је на слици 3.7



Слика 3.7 Приказ новокреираног пројекта

На овај начин за све валидне опције које корисник унесе се могу креирати команде.

3.4 Додавање функционалности за снимање и учитавање историје команди конзолног дела Кроки алата и примена Кроки програмског језика

Додавањем *Save history* и *Load history* омогућено је снимање историје унесених команди у конзолном делу Кроки алата, затим учитавање те историје, парсирање и извршавање наредби које су смештене у ту историју. Наредбе из историје упоређујемо са синтаксом програмског језика Кроки и ако су наредбе валидне онда се и извршавају.

4. ЗАКЉУЧАК

У овом раду је описано креирање програмског језика помоћу БНФ и ЕБНФ нотације у комбинацији са *ANTLR* алатом препознавање језика за потребе Кроки алата. Имплементирани су и нове функционалности којим се проширује могућност Кроки алата, са којима се олакшава коришћење самог алата и конзолног дела апликације где су функционалности имплементирани попут таб функционалности. Омогућено је снимање свих унесених наредби у конзолном делу у један нови фајл, при том да се тај фајл може поновно учитати у Кроки алат и извршити наредбе које су унесене у том фајлу. Вођено је рачуна о одабиру алата за препознавање језика при парсирању синтаксе и одабира начина како креирати нови програмски језик тако да свака следећа измена и допуна синтаксе програмског језика омогућава да то буде урађено прилично лако и брзо.

Даљи развој обухватао би праћење развоја Кроки едитора и са тим допуна синтаксе програмског језика уколико би се креирали нови елементи у Кроки алату као и омогућавање таб функционалности за те нове елементе.

5. ЛИТЕРАТУРА

- [1] *Bacus-Naur Form*
https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_Form
- [2] *Extended Bacus-Naur Form*
https://en.wikipedia.org/wiki/Extended_Backus%E2%80%93Naur_Form
- [3] *Another Tool For Language Recognition ANTLR*
<https://en.wikipedia.org/wiki/ANTLR>
- [4] М. Filipović, V. Marsenić, G. Milosavljević, I. Dejanović. Kroki: alat za interaktivni razvoj poslovnih aplikacija baziranih na skicama.
- [5] B. Perišić, G. Milosavljević, I. Dejanović, B. Milosavljević, UML Profile for Specifying User Interfaces of Business Applications. Computer Science and Information Systems, Vol. 8, No. 2, 405-426. (2011)
- [6] Hotomski Petar, Malbaški Dušan, Matematička logika i principi programiranja, Tehnički fakultet "Mihajlo Pupin", Zrenjanin 2006.
- [7] Terence Parr, *The Definitive ANTLR 4 Reference*, 2012.

Кратка биографија:

Зоран Тркуља је рођен у Карловцу 1987. године. Факлутет техничких наука уписао је 2006. године. Бечелор рад из области Електротехнике и рачунарства – Рачунарске науке и информатика одбранио је 2013. године. Мастер рад из исте области одбранио је 2015. године.

TEHNIKE OPTIMIZACIJE WEB STRANICA, MODERNI ALGORITMI ZA RANGIRANJE WEB SAJTOVA**TECHNIQUES FOR WEB PAGE OPTIMIZATION, MODERN ALGORITHMS FOR WEBSITE RANKING**

Miodrag Popović, Željen Trpovski, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu opisana je optimizacija web stranica (sajtova) za internet pretraživače. Cilj optimizacije web sajtova jeste što bolja pozicioniranost na web pretraživačima za određene ključne reči - upite. Optimizaciju webstranica možemo podeliti na unutrašnju (on - page) i na spoljašnju optimizaciju. Unutrašnja optimizacija predstavlja skup pravila po kojima se kreira web stranica, kako bi se što bolje pozicionirala za određene ključne reči na web pretraživačima. Pravila unutrašnje optimizacije se primenjuju na samoj web stranici. Spoljašnja optimizacije predstavlja niz postupaka pomoću kojih se podiže pozicija sajta na web pretraživačima za određene upite. Spoljašnja optimizacija se realizuje bez promena na samoj stranici, odnosno "izvan stranice".

Abstract – This paper describes the optimization of websites (sites) for internet search engines. The purpose of websites optimization is better positioning on web search for specific keywords - inquiries. Website optimization can be divided into internal (on - page) and external optimization. Internal optimization is a set of rules by which to create a web page in order to better position for certain keywords on web search. Rules of internal optimizations are applied to the content of the web page. External optimization is a series of procedures by which raises the position of the site on web search engines for certain queries. External optimization is implemented without changes to the page, or "off the page".

Ključne reči: Optimizacija web stranice, Ključne reči, Unutrašnja optimizacije web stranice, Spoljašnja optimizacija web stranice.

1. UVOD

Internet je postao nezamenjivo sredstvo sadašnjice, najvažnije sredstvo za komunikaciju, trgovinu, marketing, zabavu, kao i mnoge druge delatnosti. Internet je dobio ovoliki značaj tek pojavom internet pretraživača i od tada je počeo da se razvija "brzinom svetlosti". Internet pretraživači su omogućili korisnicima da jednostavno dođu do informacija koje se nalaze na web sajtovima, biranjem ključnih reči i fraza.

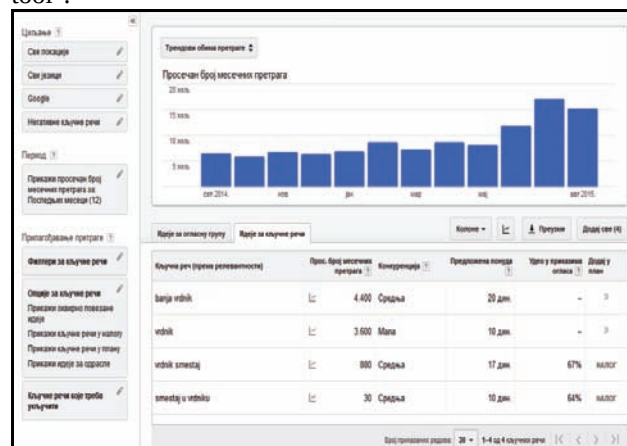
Optimizacija web sajtova za pretraživače predstavlja način spajanja sadržaja sajta sa korisnicima koji žele da

dođu do tog sadržaja. Optimizaciju web sajtova čini skup pravila po kojima se "izgrađuju" web prezentacije kako bi bile što bolje pozicionirane na internet pretraživačima. Optimizacija web sajtova može da se podeli na dve osnovne grupe, a to su unutrašnja i spoljašnja optimizacija web sajtova.

2. INTERNET PRETRAŽIVAČI

Internet pretraživači su web servisi koji na osnovu ključnih reči (upita) koje korisnici unose u njih kao rezultat "izbacuju" najkvalitetnije sajtove čiji je sadržaj usko povezan sa značenjem ključnih reči koje korisnici unose. Najpopularniji internet pretraživači u svetu su Google, Bing i Yahoo. Google pretraživač je najpopularniji i najčešće korišćen internet pretraživač u svetu, kao i u našoj zemlji. Prema nekim statistikama, u čiju verodostojnost nećemo ulaziti, Google pretraživač, kao omiljeni pretraživač, koristi preko 80% korisnika interneta u svetu.

Optimizacija web stranica za pretraživače tesno je povezana sa pozicijom sajta ili stranice na web pretraživačima. Ključne reči predstavljaju reči ili fraze koje korisnici unose u internet pretraživač kako bi došli do željene informacije ili pronašli proizvod ili uslugu koja ih interesuje. Web stranice se optimizuju na ključne reči koje se smatraju bitnim, a koje se biraju logički i pomoću specijalnih alata kao što je "Google keyword tool".



Slika 1. Google keyword tool

3. UNUTRAŠNJA OPTIMIZACIJA

Unutrašnja optimizacija ili kako se još drugačije naziva optimizacija na stranici, predstavlja skup pravila po

NAPOMENA:

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

kojima se jedna stranica piše, odnosno stvara, kako bi ona bila uspešno pozicionirana na pretraživaču.

Svaka web stranica napisana je u osnovnom jeziku za web programiranje koji se naziva HTML (HyperText Markup Language). Dokument, odnosno stranica pisana u ovom jeziku sastoji se od takozvanih tagova (slika 2), a u okviru tih tagova se nalazi tekst, slike, linkovi i sve ostalo što sačinjava jednu stranicu.

Najbitnijim tagovima kroz unutrašnju optimizaciju:

- Optimizacioni ili meta tagovi
- Tagovi za naslov
- Tagovi za formatiranje teksta

```
<a> <b> <blockquote> <br> <button> <cite> <code>  
<div> <dd> <dl> <dt> <em> <fieldset> <font> <h1>  
<h2> <h3> <h4> <hr> <i> <it> <img> <label> <li>  
<ol> <p> <pre> <span> <strong> <table> <tbody>  
<tr> <td> <th> <u> <ul>
```

Slika 2. HTML tagovi

Optimizacioni ili meta tagovi predstavljaju osnov unutrašnje optimizacije. Meta tagovi se ubacuju u zaglavlje stranice, odnosno umeću se u okvir "header" taga.

Postoje tri različita optimizaciona meta taga bitna za unutrašnju optimizaciju i još nekoliko koji su manje bitni. Tri najvažnija optimizaciona meta taga su tag koji predstavlja naslov stranice "title", zatim tag u kojem se navode ključne reči "keywords" i na kraju tag u kojem se navodi opis stranice "description". "Title" tag predstavlja najjače "sredstvo" u unutrašnjoj optimizaciji, posle njega, po važnosti sledi "keywords" tag.

Naslov stranice treba najbolje da opiše sadržaj stranice pa je samim tim i algoritam pretraživača takav da uzima u obzir naslov stranice pri određivanju pozicije stranice za određene ključne reči. Može se reći da je naslov treća najbitnija stvar pri optimizaciji stranice posle naziva stranice i ključnih reči (misli se na ključne reči u okviru "keywords" taga).

Postoji veliki broj tagova za formatiranje teksta od kojih su neki važniji, a neki su manje važni sa aspekta unutrašnje optimizacije.

Najpoznatiji i najčešće upotrebljavani tagovi za isticanje ključnih reči su tagovi za podebljan tekst, tag za pisanje kosog teksta i tag za pisanje podvučenog teksta. Za pisanje podebljanog teksta koriste se dva najpoznatija taga. To su tagovi "b" - bold i "strong". Za kosi tekst postoji jedan samo jedan tag i to je tag "i" - italic, a za podvučeni tekst se koristi tag "u" - underline.

3.1. Optimizacija stranica za prikaz na mobilnim uređajima

Izrada sajta za mobilne telefone i tablet računare je trend koji osvaja svet. Sve više poseta sajtovima se beleži sa mobilnih uređaja i telet računara. Po zvaničnoj Google statistici oko 40 % poseta sajtovima beleži se preko mobilnih uređaja. Postoji dva načina prilagođavanja sajta za mobilne uređaje. Prvi je detekcija da li korisnik koristi

mobilni telefon prilikom posete sajta i preusmeravanje na mobilnu verziju sajta. Drugi način jeste da se omogući da sajt sam podesi prikaz ekrana u zavisnosti od toga da li je u pitanju poseta sajtu preko desktop računara ili mobilnog uređaja - "responsive design" (slika 3.). Uticaj optimizacije sajtova za mobilne uređaje na broj poseta i rangiranje sajta na pretraživačima je ogroman, samom činjenicom da oko 40% pretraga dolazi sa mobilnih uređaja. Sajtovi koji nisu optimizovani biće prikazani na dnu pretrage na web pretraživačima mobilnih uređaja.



Slika 3. Primer "responsive design" sajta

Prilagođavanje sajta tač uređajima je tesno povezano sa prilagođavanjem sajta za prikaz na mobilnim uređajima. Pored toga što je važno da na mobilnim uređajima čiji su ekrani sitniji i niže rezolucije, veličina slova bude dovoljno velika kako bi tekst bio čitljiv, jedan od problema sa kojima se susreću korisnici "tač" uređaja jeste kretanje kroz sajt, koje zna da bude veoma otežano ako stranice nisu projektovane tako da ovaj problem bude prevaziđen.

4. SPOLJAŠNJA OPTIMIZACIJA

Spoljašnja optimizacija predstavlja niz aktivnosti pomoću kojih se podiže pozicija sajta ili stranice na pretraživačima za određene ključne fraze, odnosno upite. Spoljašnja optimizacija je složen i dugoročan postupak. Razlika spoljašnje optimizacije u odnosu na unutrašnju jeste ta što se kod spoljašnje optimizacije ne menja ništa na samom sajtu.

4.1. Izbor domena kao pred-optimizacija

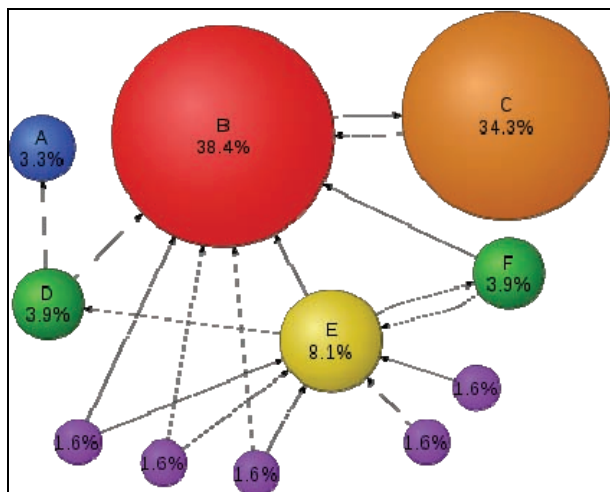
Da bi optimizacija sajta bila uspešna i da bi se uopšte isplatilo raditi kvalitetnu optimizaciju potrebno je izabrati odrovarajući domen. Izbor domena je zapravo najbitnija stvar za optimizaciju sajtova. Suština izbora domena jeste ta da izabrani domen treba da sadrži najbitnije ključne reči za naš sajt. Najbitnija stvar jeste da sam domen najbolje opisuje delatnost sajta, odnosno da najbitnije ključne reči budu sadržane u samom domenu.

4.2. Linkovanje

Veoma važnu ulogu u pozicioniranju sajta u rezultatima pretrage na neke bitne ključne reči ima broj i kvalitet linkova koji sa drugih (tuđih) sajtova vode ka našem sajtu. Postavljanje našeg linka na druge sajtove se naziva

linkovanje ili SEO izgradnja. Link ka našem sajtu na nekom drugom sajtu može biti postavljen na različita mesta kao što su podstranice sajta, zatim različita mesta na početnoj stranici sajta kao što su futer, link u tekstu i drugo.

Najbolji i najveći efekat ćemo postići ako link bude postavljen na početnu stranicu nekog drugog sajta. Postoji dva načina postavljanja linka, ili dva tipa linka od koji svaki ima svoje prednosti i mane. To su tekstualni linkovi i linkovi sa slikom koji se nazivaju baneri.



Slika 4 - Šematski prikaz linkovanja

Algoritam pretraživača je takav da uzima u obzir sličnost sajtova koji se međusobno linkuju, ili pak jedan linkuje drugog, svejedno. Što su sajtovi tematski sličniji to je efekat linkovanja veći. Ako su sajtovi skroz različiti, i tematski nemaju nikakve veze jedan sa drugim, efekat će biti najmanji.

Ako želimo sajt da "probijemo" na bolju poziciju za određenu ključnu reč, linkovi ka našem sajtu treba da sadrže upravo tu reč. Najveći efekat je ako je tekst link baš ta ključna reč koja je bitna.

4.3. Zloupotrebe prilikom linkovanja

Pre nekoliko godina broj linkova je bio jedan od glavnih "sastojaka" za SEO izgradnju, odnosno jedna od najbitnijih stvari koja je uticala na pozicioniranje sajta na internet pretraživačima. Kao što to obično biva mnogi su se setili da to i zloupotrebe.

Razmena linkova je bila "glavni hobi" onih koji su se bavili ovom delatnošću, razvili su se mnogi portali, ili kako se to popularno nazivalo "katalozi linkova", čija je jedina funkcija bila razmena linkova u cilju podizanja Google Page Rank-a, a zatim bi se na istim "tako jakim sajtovima", koji su u nekim slučajevima imali PR5 ili PR6, prodavalo mesto za link.

Pošto se svi internet pretraživači, među kojima je prvi Google, trude da u rezultatima pretrage korisnicima ponude sajtove sa što kvalitetnijim sadržajem, kako bi korisnički doživljaj pri pretraživanju bio što bolji i kvalitetniji, ovaj način razmišljanja, odnosno algoritam rangiranja na osnovu što većeg broja linkova, koji nemaju nikakvog smisla, morao je biti ukinut, a sajtovi koji su profitirali, odnosno postigli nezasluženo mesto u pretrazi, morali su biti sankcionisani.

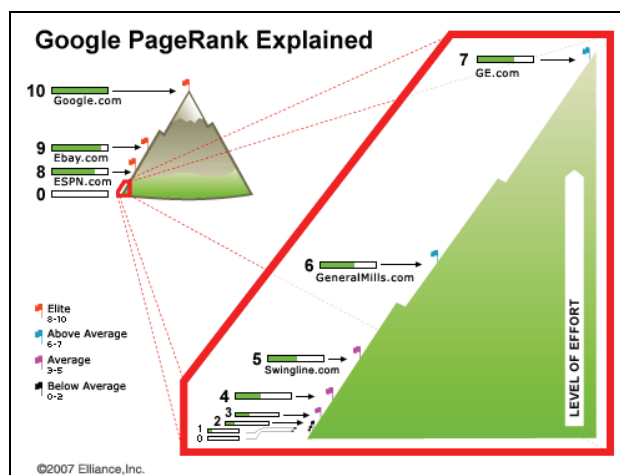
Algoritam koji je kaznio mnoge sajtove je sproveden pod nazivom "Pingvin" algoritam, a mnogi sajtovi se još oporavljaju od ovog algoritma.



Slika 5 - Zanimljiv način prikaza "Pingvin" algoritma

4.4. Google Page Rank - odraz jačine stranice

Google je razvio svoj pokazatelj "jačine" sajta koji se naziva Google Page Rank. Google Page Rank može imati vrednost od nula do deset, sa tim da je vrednost nula najmanja vrednost za najslabije sajtove i imaju je svi novi sajtovi koji su tek napravljeni. Google Page Rank zavisi od više faktora, ali najviše zavisi od broja i kvaliteta dolaznih linkova, odnosno linkova koji su postavljeni na drugim sajtovima i vode ka našem sajtu. Google Page Rang nije linearna funkcija, već logaritamska funkcija. Ovo je bitno jer je na primer razmak između ranga 3 i ranga 4, neuporedivo manji nego razmak između ranga 4 i ranga 5. Jednostavnije rečeno, neuporedivo je lakše doći do na primer ranga 4 nego do ranga 5.



Slika 6 - Logaritamska funkcija Google PR

5. FAKTORI KOJI UTIČU NA RANGIRANJE SAJTA

Pored mnogih faktora koje smo opisali u okviru poglavlja koja su se bavila unutrašnjom i spoljašnjom optimizacijom, postoje još neki faktori koji se lakše ili teže mogu svrstati u jednu od ove dve grupe, a ipak utiču na

rangiranje sajta na internet pretraživačima kao što je Google.

5.1. Uticaj web servera na rangiranje sajta na pretraživaču

Web server je računar koji je povezan na internet, a na kojem se čuvaju svi podaci koji su sastavni deo jednog sajta (fajlovi i baze podataka). Jednom sajtu se pristupa, tako što se pristupa web serveru na kojem se on nalazi. Jedan od faktora koji poprilično utiče na rangiranje sajta na web pretraživačima, jeste brzina učitavanja elemenata sajta, odnosno brzina učitavanja stranica i njenih sastavnih delova (slike, video i drugo). Što je brzina učitavanja elemenata sajta veća, doživljaj korisnika koji pregledaju sajt je kvalitetniji, pa su i pretraživači skloni da ovakve sajtove nagrade boljom pozicijom u odnosu na sajtove koji se teško otvaraju i imaju slične probleme, što je i logično.

5.2. Uticaj lokacije servera na rangiranje sajta na pretraživaču

Jedan od faktora koji utiče na poziciju sajta ili stranice na web pretraživaču, doduše ne u velikoj meri, jeste i fizička lokacija web servera na kojoj se sam sajt nalazi. Suština algoritma jeste da pri rangiranju sajtova, za upite koji stižu sa određene lokacije, prednost imaju oni sajtovi koji se nalaze najbliže lokaciji sa koje stižu upiti. U praksi, za korisnike koji pretražuju internet iz Srbije, prioritet imaju sajtovi koji se nalaze na serverima koji su locirani u Srbiji.

5.3. Uticaj društvenih mreža na poziciju sajta

Jedan od faktora koji u poslednje vreme sve više utiče na poziciju sajta u rezultatima pretrage na svim web pretraživačima jeste i aktivnost sajta na društvenim mrežama. Aktivnost sajta na društvenim mrežama na poziciju sajta može da utiče direktno, ali i indirektno. Skoro sve društvene mreže kako što su Facebook, Google društveni nalog ili Twitter razvile su "dugme" koje se postavlja na sajt, kako bi korisnici lakše podelili sadržaj sajta sa drugima na društvenim mrežama. Direktan uticaj aktivnosti sajta na društvenim mrežama na pozicioniranje sajta se ogleda u povećanju broja povratnih linkova, dok se indirektni uticaj ogleda u povećanju broja poseta, koje opet pozitivno utiču na različite načine.

6. ZAKLJUČAK

U poslednje vreme najpopularniji vid marketinga postao je internet marketing. Danas gotovo sve firme i pojedinci koji se bave nekom delatnošću imaju svoju internet stranicu preko koje prezentuju, odnosno reklamiraju svoje proizvode i usluge.

Isto tako mnogi svoje oglase postavljaju na specijalizovane sajtove koji se bave reklamiranjem usluga i proizvoda neke određene delatnosti.

Suština internet reklamiranja svodi se na pretraživanje interneta uz pomoć internet pretraživača od kojih je najpoznatiji Google pretraživač koji u svetu koristi 80-90 % ljudi, a u Srbiji je taj procenat još veći.

Kvalitetnu reklamu koju će videti najveći broj ljudi, što je i najbitnije u marketingu imaće oni koji su dobro pozicionirani na pretraživaču na određene bitne ključne reči koje veliki broj ljudi upisuje u pretraživač kako bi došli do informacije koja ih interesuje.

Ovde dolazi do izražaja optimizacija web stranica za pretraživače jer će najbolje optimizovani sajtovi zauzimati najkvalitetnije pozicije na pretraživačima i time pridobiti najveći broj posetilaca, odnosno klijenata.

Zaključujemo da je optimizacija web stranica za pretraživače jedna od najbitnijih stavki u internet marketingu, a samim tim i u marketingu uopšte jer je internet danas najviše korišćeno marketinško sredstvo.

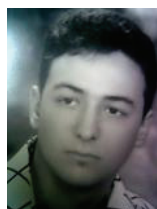
7. LITERATURA

[1] Jerri L. Ledford: "Search engine optimization, bible", Wiley Publishinh 2007.

[2] Jaimie Sirovich i Christian Darie: "SEO with PHP", Wiley Publishinh 2007.

[3] Dave Chaffey: "SEO Best practice guide 2007", E-consultancy.com 2007

Kratka biografija:



Miodrag Popović rođen je u Beogradu 1987. god. Master rad odbranio je 2015. 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.

SIMULACIJA SISTEMA ZA KODOVANJE I MODULACIJU DVB-S2 NA PREDAJNOJ STRANI**SIMULATION OF DVB-S2 TRANSMISSION SIDE CODING AND MODULATION**

Đorđe Jarac, Željens Trpovski, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu opisan je jedan metod realizacije sistema za kodovanje i modulisanje na predajnoj strani primenom standarda za satelitsko emitovanje, DVB – S2. Realizovani su: LDPC koder i modulatori koji obavljaju funkciju APSK modulisanja ulaznog signala. U svrhu realizacije sistema upotrebljene su funkcionalnosti programskih jezika C++ i Matlab. Celokupni model sistema je moguće pokrenuti iz Simulink modela u kojem se obavljaju operacije formiranja informacione sekvence, kodovanja Reed Solomon i LDPC kodom, modulisanja, simuliranja AWGN kanala i ilustrovanja konstelacionog dijagrama emitovanog signala.

Abstract – In this paper the one method of implementation system for coding and modulation at the transmitter side, using the standard for satellite broadcasting DVB – S2 is described. Also, this paper describes implementation of LDPC encoder and APSK modulators. In order to realize the system, different functionalities of the programming languages C++ and Matlab were used. The entire system can be turned on through Simulink model consisting of blocks performing operations of forming information sequence, Reed Solomon and LDPC coding, modulation, simulation of AWGN channel and illustration of constellation diagram of the transmitted signal.

Ključne reči: DVB – S2, Reed Solomon, LDPC, APSK modulacija, kodovi za ispravljanje grešaka, generatorska matrica, matrica provere parnosti.

1. UVOD

Satelitski komunikacioni sistemi pružaju mogućnost komunikacije praktično u svim tačkama naše planete i na taj način upotpunjuju globalnu komunikacionu uslugu koja je dostupna stanovništvu širom sveta.

Tri glavna načina komunikacije (*mobile, broadcast* i *data communications*) moguće je ostvariti putem satelitskih komunikacionih sistema i oni često predstavljaju jedino rešenje za ostvarenje komunikacije, posebno u slučajevima kada je neophodno pružiti komunikacionu uslugu u retko naseljenim područjima, pomorskim putevima, naftnim platformama, itd.

Tokom svog prostiranja, satelitski signal je izložen delovanju šumova koji postoje u komunikacionom kanalu. Delovanje šumova, kao i osobine samih prenosnih kanala predstavljaju veliki izazov ostvarenju pouzdanih i globalnih komunikacija.

2. SATELITSKI KOMUNIKACIONI SISTEMI

Kada govorimo o satelitskim komunikacijama, izdvajamo tri glavna elementa koji pružaju uslugu globalne komunikacije putem satelita: predajna zemaljska stanica, satelit kao osnovni element komunikacije i prijemna zemaljska stanica ili više njih. Svaki satelit poseduje svoju orbitu, tj. putanju koju prelazi tokom praćenja glavnog tela. Ukoliko se orbite posmatraju u zavisnosti od visine u odnosu na Zemlju, definišu se tri vrste orbita: geostacionarna orbita (GEO orbita), srednja zemaljska orbita (MEO orbita) i niska zemaljska orbita (LEO orbita).

U satelitskim telekomunikacijama, podrazumeva se da komunikacija počinje slanjem signala ka satelitu, a nastavlja se njegovom retransmisijom ka jednoj ili više prijemnih stanica na Zemlji. Satelit ne obavlja samo klasično reflektovanje i pojačanje signala, već i mnoge druge složenije funkcije poput dodatnih modulacionih postupaka, detekcije, filtriranja, itd. Praktično, na satelitu postoje i prijemnik i predajnik koji zajedno čine strukturu koja se naziva transponder [1].

Kada se govori o satelitskom prenosu i signalima uvek postoje dve deonice duž kojih se pomenuti signali prostiru i nailaze na razne ometajuće faktore. U svakom satelitskom sistemu, posebna pažnja se posvećuje pripremi signala za uzlaznu i silaznu deonicu prostiranja. Na svom putu od zemaljskog predajnika do satelitskog prijemnika signal se prostire duž tzv. uzlazne deonice. Po dolasku do satelita, signal se dodatno pojava i obrađuje na samom satelitu, u čemu posebnu ulogu ima tzv. TWT¹ cev. Nakon obrade na satelitu signal se sa satelitskog predajnika šalje ka zemaljskom prijemniku i time prelazi tzv. silaznu deonicu.

3. DRUGA GENERACIJA STANDARDA U SATELITSKOJ TELEVIZIJI

DVB - S2, predstavlja drugu generaciju standarda za satelitsko emitovanje, formiranu 2003. godine od strane DVB² projekta. U drugoj generaciji standarda primetne su značajne promene u načinima modulacije signala, a takođe, koriste se benefiti modernih tehnika kodovanja

NAPOMENA:

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

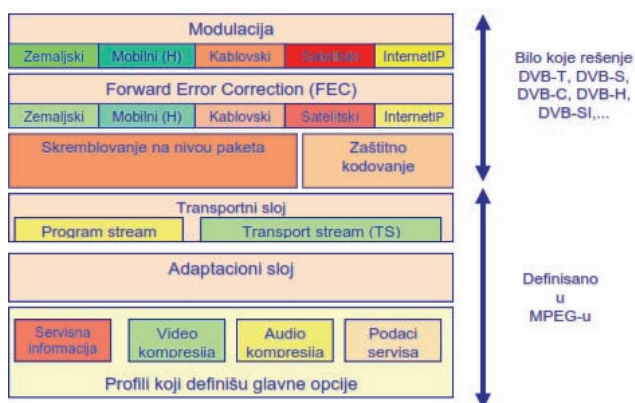
¹ Traveling wave tube - specijalna vrsta talasovoda na satelitu.

² Digital video broadcasting.

čijim primenama se nastoji umanjiti uticaj šumova duž propagacionih deonica. Glavna razlika u pogledu servisa koji se pružaju današnjim korisnicima digitalnog satelitskog prenosa, a koju uvodi DVB - S2 standard, jeste u stvari dodavanje usluga IP saobraćaja. DVB S2 sistemi predstavljaju skup opreme (algoritama i hardvera), koja prihvata dolazeće signale poruke iz jednog ili više MPEG multipleksera i nakon toga ih koduje zaštitnim kodovima i moduliše.

Proizvod delovanja DVB S2 sistema jeste DVB paket koji sadrži korisničke, kao i dodatne pomoćne signale. U postupku formiranja DVB paketa, izvorni koder predstavlja koder MPEG tipa. Transportni sloj služi za formiranje programskog i transportnog toka podataka. Programski tok se koristi za direktno snimanje i arhivaciju, dok transportni služi za distribuciju krajnjim korisnicima.

Ovim je komprimovana video sekvenca upakovana i spremna za dalju obradu, pre samog slanja. Signal je nakon toga neophodno zaštititi nekim od metoda zaštitnog kodovanja, a zatim i modulisati. Kodovanje i modulacija su određeni DVB S2 standardom i prilagođeni području primene signala. Na slici 1. prikazani su MPEG i DVB slojevi i procesi [2].



Slika 1. MPEG i DVB slojevi i procesi [2]

Cilj celokupne procedure na DVB S2 predajniku jeste transformacija poslate MPEG sekvence u tzv. BB frame (baseband frame), tj. okvir podataka koji se radio talasima šalje ka određenoj [3]. Ovakav BB frame, predstavlja okvir podataka koji odlazi na fizički sloj u OSI referentnom modelu. Pre nego što se nad signalom na predajnoj strani obavi modulacija i prilagođavanje za slanje na fizičkom sloju, obavlja se FEC kodovanje. Rezultat ovog postupka jeste formiranje FEC okvira podataka (FEC frame).

Prilikom formiranja FEC frame - a obavlja se spoljno i unutrašnje kodovanje, korišćenjem BCH ili Reed Solomon, odnosno LDPC koda. Uobičajene dužine FEC frame - ova iznose 16.200 bita za kratki, odnosno 64.800 bita za dugi frame. Različite varijante kodnih brzina se primenjuju u različitim prilikama (vremenski uslovi na deonicama signala, različite primene signala). Osim PSK modulacije, standard DVB - S2 primenjuje i APSK modulaciju, pri kojoj dolazi do modifikovanja faze i amplitude nosioca.

Dve verzije APSK modulacije su prisutne kod DVB - S2 standarda: 16 i 32 APSK, dok unapređeni DVB - S2x predlaže i APSK modulacije višeg reda sa 64, 128 i 256 simbola. Glavna unapređenja koja odlikuju DVB S2x

standard ogledaju se u povećanju dostupnih MODCOD³ - ova, umanjenoj roll - off⁴ faktora i uvođenju viših modulacionih šema do 256 simbola. Namera ovog proširenja standarda jeste povećanje obima primena satelitskih telekomunikacionih sistema, kao i njihovo jače pozicioniranje na tržištu IP servisa i RTV difuzije.

4. FEC KODOVANJE U FVB – S2 STANDARDU

D - arni linearni kod (n, m) , predstavlja m - dimenzioni potprostor vektorskog prostora $GF(D)^n$ n - dimenzionih vektora u $GF(D)$. U praksi je čest slučaj da je $D = 2$ pa se ovakvi kodovi nazivaju binarni linearni kodovi. Važno svojstvo linearnih kodova jeste da bilo koja linearna kombinacija kodnih reči predstavlja takođe kodnu reč. Ukoliko se pretpostavi binarni linearni kod, onda je broj kodnih reči koje mu pripadaju jednak D^m .

Prilikom formiranja kodne reči, dolazi do množenja vektora informacionih simbola sa karakterističnom matricom, generatorskom matricom, koja u potpunosti karakteriše linearni kod. Dimenzije generatorske matrice iznose $m \times n$. Generatorska matrica na osnovu koje se formira sistematski kod ima sledeću karakterističnu formu (slika 2.):

$$G = [I_m P] = \begin{bmatrix} 1 & 0 & \dots & 0 & p_{1,1} & \dots & p_{1,n-m} \\ 0 & 1 & \dots & 0 & p_{2,1} & \dots & p_{2,n-m} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & p_{m,1} & \dots & p_{m,n-m} \end{bmatrix}$$

Slika 2. Opšti oblik generatorske matrice

Osim na osnovu generatorske, svaki linearni kod može biti opisan i na osnovu matrice provere parnosti (parity check matrix), H . Skup vektora (kodnih reči) koje su ortogonalne na sve kodne reči iz linearnog koda C predstavlja takođe potprostor i naziva se ortogonalni komplement koda C . Generatorska matrica ortogonalnog komplementa linearnog koda C predstavlja matricu provere parnosti koda C i ima dimanzije $n-m \times n$.

LDPC (low density parity check) kodovi predstavljaju kodove koji su dizajnirani kao linearni i binarni blok kodovi. U svojoj doktorskoj disertaciji, teoretičar informacija Robert G. Gallager, dao je opis LDPC koda i formulisao prvi način formiranja matrice provere parnosti. Prema notaciji koju je formulisao Gallager, LDPC kod se zapisuje na sledeći način: $C_{LDPC}(n, \omega_c, \omega_r)$. U ovom zapisu n predstavlja dužinu kodne reči, ω_c predstavlja broj jedinica po jednoj koloni, dok ω_r predstavlja broj jedinica po jednoj vrsti matrice provere parnosti [4]. Matrica provere parnosti, H , zadovoljava uslov da broj jedinica ove matrice mora biti mnogo manji od broja nula, tj. ova matrica se smatra retkom matricom. Linearnim transformacijama (sabiranje i zamena vrsta ili kolona) nad matricom H , moguće je dobiti matricu provere parnosti u sistematskom obliku, od koje je onda moguće formirati i generatorsku matricu za određeni LDPC kod. Nakon što

³ MODCOD - mogućnost odabira odgovarajućih modulacija i kodnih brzina u zavisnosti od aplikacije i stanja u kanalu.

⁴ Merilo zauzetosti opsega učestanosti izvan Nikvistovog opsega.

je matrica H formulisana u novom formatu, generatorska matrica G može biti predstavljena na sledeći način:

$$G = [I_m P^T] \quad (1)$$

RS (Reed Solomon) kodovi predstavljaju linearne ciklične nebinarne kodove. Formiraju se nad poljem $GF(q)$, gde je q stepen prostog broja p , $q=p^m$, m je pozitivan celi broj. Svako konačno polje ima najmanje jedan primitivan element α . Za element α se kaže da je primitivan ukoliko je njegov rang tj. najmanji celi pozitivan broj e takav da je $\alpha^e = 1$ jednak $q-1$. Svi nenulti elementi polja $GF(q)$ mogu se predstaviti kao $q-1$ uzastopnih stepena od α , tj. $\{1, \alpha, \alpha^2, \alpha^3, \dots, \alpha^{q-1} = 1, \alpha^q = \alpha, \dots\}$. Primitivni element α je koren nekog primitivnog nesvodljivog polinoma $p(x)$. Polinom $p(x)$ stepena m nad poljem $GF(q)$ je nesvodljiv nad $GF(q)$ ukoliko nije deljiv bilo kojim drugim polinomom nad $GF(q)$ stepena manjeg od m , ali većeg od nule. Neki nesvodljivi polinomi su i primitivni.

Nesvodljivi polinom $p(x)$ iz $GF(q)$ stepena m naziva se primitivnim ukoliko je najmanji prirodan broj n za koji je $p(x)$ delilac (bez ostatka) polinoma $x^n + 1$, takav da je $n=q-1$. Najčešći način konstrukcije RS kodova jeste pomoću generatorskog polinoma. Generatorski polinom RS koda koji ispravlja t grešaka, ima kao korene, $2t$ uzastopnih stepena nekog primitivnog elementa α .

Prema tome, generatorski polinom ima sledeći oblik: $g(x) = \prod_{j=0}^{2t-1} (x - \alpha^{h+j}) = \sum_{j=0}^{2t-1} g_j x^j$ gde je h celobrojna konstanta, najčešće jednaka 1. Kodna reč (kodni polinom) formiran pomoću generatorskog polinoma predstavlja kodnu reč linearnog, cikličnog blok RS koda. Kodni polinomi predstavljaju umnoške generatorskog polinoma i stoga sadrže sve njegove nule. Polinom poruke se formira na sledeći način: $m(X) = m_0 + m_1 X + \dots + m_{k-1} X^{k-1}$. U nastavku sledi procedura za formiranje kodnog polinoma:

$$X^{n-k} m(X) = q(X)g(X) + p(X), \quad (2)$$

gde su:

$g(X)$ -generatorski polinom,
 $q(X)$ -količnik,
 $p(X)$ -ostatak.

5. PRAKTIČNA REALIZACIJA PREDAJNIKA DVB - S2 STANDARDA

U svrhu realizacije kodera i modulatora predajnika, upotrebljene su funkcionalnosti programskih jezika C++ i Matlab. Formiranje generatorske matrice i proces kodovanja LDPC kodom realizovan je pomoću programskog jezika C++, dok je proces kodovanja Reed Solomon kodom realizovan pomoću RS koda programskog jezika Matlab.

Proizvod (generatorska matrica LDPC koda) realizovan u programskom jeziku C++, inkorporiran je u jedinstveni Simulink model koji sadrži sve komponente neophodne za realizaciju predajnika. Osim dva sistema za kodovanje, realizovani su i blokovi za modulisanje, prvenstveno APSK modulacijama višeg reda koje su karakteristične za DVB - S2 i DVB - S2x standarde. Konačan izlazni signal iz predajnika dodatno je zašumljen propuštanjem kroz Simulink AWGN blok i kao takav grafički je prikazan zajedno sa konstelacionim dijagramom upotrebljene

modulacione šeme. Rezultat pokretanja programa za formiranje generatorske matrice jeste *txt file* koji sadrži navedenu generatorsku matricu LDPC koda. Ovakav *txt* dokument se učitava u Matlab i posebnom procedurom dodatno i u glavni Simulink model, u kome postaje blok za simulaciju generatorske matrice.

Generatorska matrica, G , biva formirana nakon niza linearnih transformacija nad matricom provere parnosti, H , (u ovom radu to je Galagerova forma matrice provere parnosti). U cilju ilustracije prikazani su: izgled početne forme Galagerove matrice H i izgled generatorske matrice nakon primene transformacija napisanih u C++ - u (slike 3. i 4.). Početna forma matrice H sadrži 9 vrsta i 12 kolona.

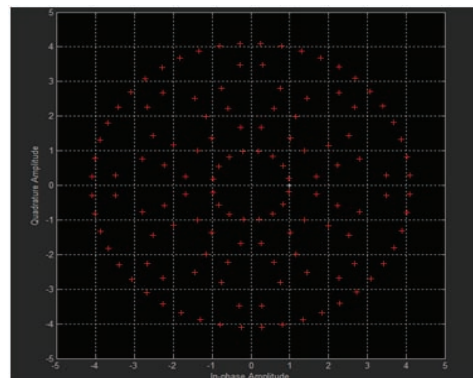
1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Slika 3. Početna forma matrice H

1	0	0	0	0	0	1	0	0	1	1	0
0	1	0	0	0	1	0	0	1	0	1	0
0	0	1	0	0	0	0	1	1	0	0	1
0	0	0	1	0	0	0	1	0	0	1	1
0	0	0	0	1	0	0	1	1	0	0	1

Slika 4. Matrica G , dobijena programski

Blokovi koji obavljaju modulacije realizovani su u programskom jeziku Matlab. U ovom radu funkcijski su realizovane APSK modulacije, koje se koriste u DVB - S2 standardu, kao i u najnovijoj unapređenoj verziji standarda, DVB - S2x. Razvijene su sledeće funkcije: 16 APSK, 32 APSK, 64 APSK na dva načina i 128 APSK. Slika 5. prikazuje izgled konstelacionog dijagrama 128 APSK modulacije.

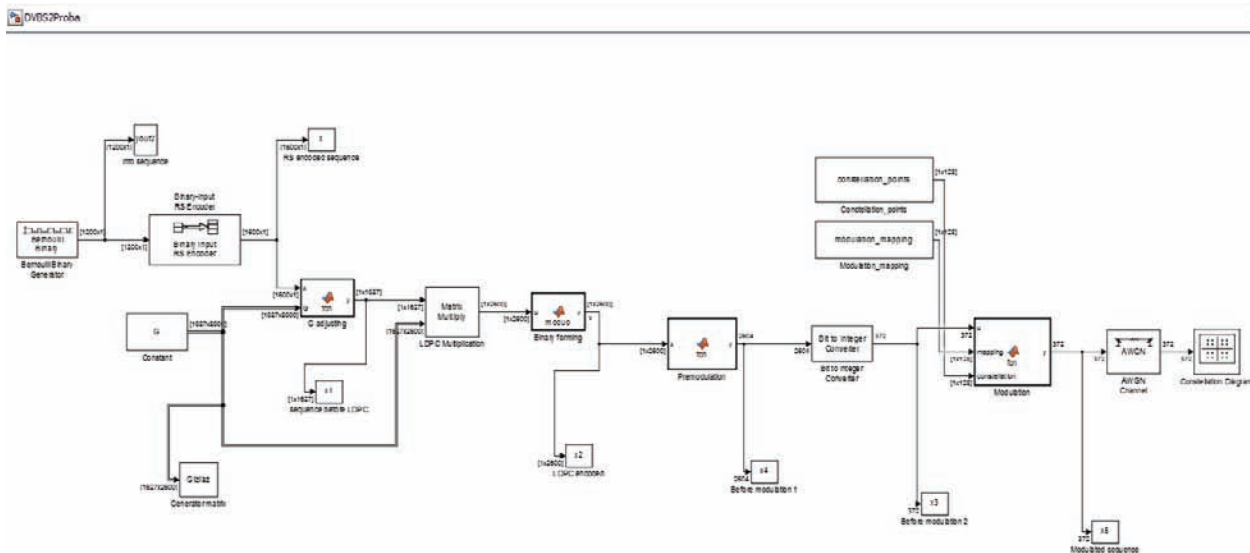


Slika 5. Konstelacioni dijagram 128 APSK

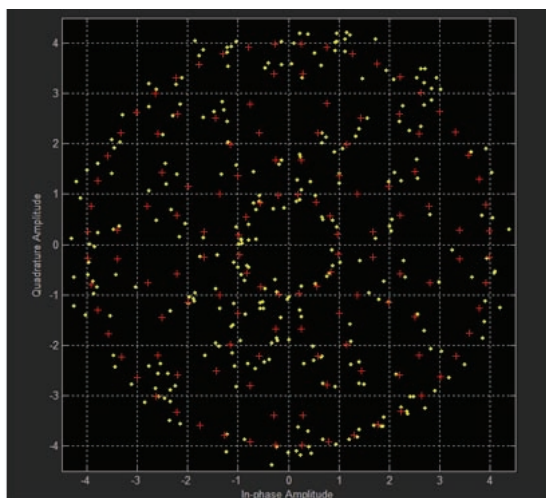
Konačan Simulink model predajnika sadrži još nekoliko korisnički definisanih i ugrađenih funkcija koje omogućavaju nesmetan tok niza bita od izvora do odredišta, tj. od generatora informacione sekvence (*Bernoulli binary generator*) do prikaza modulisanog signala pogođenog šumom na konstelacionom dijagramu. Na slici 6. prikazan je izgled celog modela za slučaj slanja 1200 bita uz primenu RS/LDPC kodovanja i 128 APSK

modulacije. Takođe prikazan je i izgled modulisanog signala (žuto) i referentne konstelacije (crveno) nakon simulacije prolaska duž AWGN kanala (Slika7.). Dimenzije generatorske matrice su 1627×2600 , što znači da informaciona sekvenca LDPC koda (tj. kodna reč koja nakon RS koda dolazi na LDPC koder) treba da sadrži 1627 bita, koji se dalje koduju i modulišu. Ukoliko kodna reč, proizašla iz RS koda sadrži manje bita od zahte-

vanog broja bita informacione sekvence LDPC koda, dolazi do odgovarajućeg dopunjavanja nulama (blok *G adjusting*). Nakon matičnog množenja matrice *G* i informacione sekvence LDPC koda, kao i pretvaranja matičnog proizvoda u proizvod po modulu 2, kodovana sekvenca bita se pretvara u sekvencu celih brojeva (Bit to integer converter) i nakon toga moduliše APSK modulacijom.



Slika 6. Celokupni Simulink model pradjnika u DVB – S2 standardu (RS/LDPC, 128 APSK)



Slika 7. Predajni signal nakon obrade u predajniku

6. ZAKLJUČAK

Cilj ovog master rada bila je praktična realizacija simulacije predajnog kodovanja i modulacije signala po uzoru na arhitekture kodovanja i modulacije u DVB - S2 standardu. Takođe, realizovane su i dodatne APSK modulacione funkcije koje su našle primenu u najnovijem unapređenju DVB – S2 standarda, DVB – S2x. Prilikom izrade odgovarajućih programskih celina korišćene su funkcionalnosti programskih jezika C++ i Matlab.

Celokupan sistem predajnog kodovanja i modulisanja na kraju je prikazan u jedinstvenom Simulink modelu, koji je povezan sa Matlab radnim okruženjem.

7. LITERATURA

- [1] – Praštalo Radojka: Satelitske telekomunikacije, ETF Banja Luka, 2006.
- [2] – Reljin Irini, Reljin Branimir: DVB – Multimedija, Pos Tel 2006.
- [3] – Use of DVB – S2 ETSI standard in High Data Rate TeleMettry for near space – earth transimissions, CCSDS RF & Modulation & Channel Coding Working Groups 2012.
- [4] – Jorge Castineira Moreira, Patrick Guy Farrell: Essentials of error - control coding, Wiley 2006.

Kratka biografija:

Đorđe Jarac rođen je u Foči 1990. god. Diplomski - master rad odbranio je 2015. godine na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacioni sistemi.



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

FORMIRANJE MAPE DUBINE IZ JEDNE SLIKE SINGLE FRAME DEPTH MAP GENERATION

Milana Antin, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Svrha ovog rada je upoznavanje sa tehnikama formiranja mape dubine iz samo jedne slike, na osnovu obeležja dubine uz pomoć obučavanja neuronskih mreža. Razlika u odnosu na dosadašnje metode, je dobijanje mape dubine iz samo jedne slike, a ne iz više uzastopnih slika iste scene, uz pomoć geometrijskih transformacija i zakona binokularnosti. Za procenu kvaliteta mape dubine korišćene su full reference (FR) metričke, za koje postoji referentna.

Abstract – The purpose of this paper is to introduce the techniques of forming depth maps of only one image, based on the depth cues with the help of training neural networks. The difference compared to previous methods, is obtaining depth maps from only one image, not from a sequence of the same scene, with the help of geometric transformations and binocularity law. In assessing the quality depth maps, full references (FR) metrics are used, for which there is a reference.

Ključne reči: depth map, depth cues, artificial neural network

1. UVOD

U osnovi čovekove trodimenzionalne percepcije prostora, leži vizuelni sistem zasnovan na dva oka (binokularna vizija) što je još davne 300. godine p.n.e zaključio i Euklid prilikom svog istraživanja. Svako oko, na određenom rastojanju, stvara sliku prostora iz svog ugla, a mozak obrađuje podatke koji su pristigli sa oba oka, i kombinuje ih u jednu sliku koju čovek doživljava kao trodimenzionalni prostor. Ovakav model, poslužio je kao ideja za razvitak mnogih modernih tehnologija, kao što su 3D televizija, uređaji za snimanje, senzori, roboti i sl., kod kojih postoji stalna težnja za integracijom treće dimenzije u cilju prevazilaženja ograničenja dvodimenzionalnih površina prikaza i postizanja što veće sličnosti sa realnim svetom, čime bi se obogatilo doživljaj krajnjih korisnika.

2. TEORIJSKE OSNOVE

Prvi korak ka proširivanju 2D slike novom dimenzijom jeste analiza dubine prostora, i pronalaženja zakonitosti između rastojanja, položaja i teksture oblika u prostoru kako bi se formirao teorijski model za dalje ispitivanje u praksi. Ono što treba utvrditi jeste na koji način se dubina prostora interpretira na ravnoj 2D slici, i kako se iz te transformacije 3D u 2D mogu izvući sve informacije potrebne za generisanje 3D iz 2D slike.

NAPOMENA:

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

2.1. Obeležja dubine

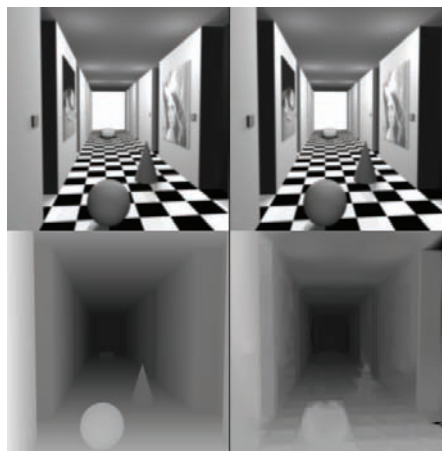
Čovekov vizuelni sistem interpretira dubinu koristeći psihološke i fiziološke osobine. Preduslov nekih fizioloških obeležja jeste prijem informacija sa oba oka (binokularnost), dok neka obeležja dubine mogu da se uoče i ako se gleda samo sa jednim okom (monokularnost).

U fiziološka obeležja spadaju: akomodacija, konvergencija, binokularna paralaksa i monokularna paralaksa pomeraja. U psihološka obeležja dubine spadaju: retinalna veličina slike, linearna perspektiva, gradijent teksture, preklapanje, vazдушna perspektiva, nijanse i senke u slici [4] i defokus, i sva psihološka obeležja su monokularna.

S obzirom da se rad zasniva na dobijanju mape dubine na osnovu jedne slike, korišćena su tri monokularna obeležja: vazдушna perspektiva, linearna perspektiva i defokus (zamućenost).

2.2. Mapa dispariteta i mapa dubine

Najjače obeležje dubine jeste binokularna paralaksa, i ukoliko nema nijednog drugog obeležja, može se proceniti dubina scene samo na osnovu ovog obeležja. U domenu stereoskopije, binokularna paralaksa ispoljava se kao disparitet, a čest je i pojam mapa dispariteta. Ona ukazuje na rastojanje između istih tačaka, ili tzv. prividnog „kretanja“ objekata, u dve slike koje su pomerene za određeni ugao, slika 1. To kretanje definisano je pomoću nijansi sive koje se smenjuju od bele do crne boje, u skladu sa količinom pomeraja. Slične osobine ima mapa dubine, zbog čega su se tokom analize, kao referentne mape dubine za FR (full reference) metričke, koristile mape dispariteta.



Slika 1. Gornje dve slike se razlikuju za mali ugao gledanja, a disparitet je prikazan na slikama dole

2.3. Veštačke neuronske mreže

Veštačke neuronske mreže (eng. Artificial Neural Networks, ANNs), predstavljaju familiju statističkih

modela učenja, po uzoru na biološke nervne sisteme živih bića, koji mogu da aproksimiraju određene funkcije na osnovu velikog broja ulaznih vrednosti, i koje su u većini slučajeva nepoznate. Neke od osnovnih osobina i mogućnosti koje nudi neuronska mreža jesu:

- adaptivnost,
- nelinearnost i
- nivo poverenja za donošenje odluka.

Arhitektura neuronske mreže (skraćeno NM) organizovana je u slojeve sačinjene od jedinica – neurona. Tako imamo prvi ulazni sloj, poslednji izlazni sloj, a svi slojevi između se nazivaju skriveni slojevi i može ih biti više od jednog, ali se takvi modeli NM ne koriste često [5].

2.3.1 Koncept učenja

Najinteresantniji aspekt veštačkih neuronskih mreža jeste sposobnost učenja, odnosno da na osnovu skupa opservacija, od skupa funkcija neuronske mreže F , pronađe onu funkciju $f \in F$ koja će dati optimalno rešenje za dati problem. Daleko najznačajnija neuronska mreža, koja je i u ovom radu upotrebljena jeste mreža sa propagacijom greške unazad (*backpropagation*). U predloženoj metodi, korišćena su dva *backpropagation* algoritma iz grupe algoritama nadgledanog obučavanja: *Levenberg-Marquardt* i *Bayesian Regularization*.

2.3.2 Levenberg-Marquardt backpropagation algoritam

LMA (Levenberg-Marquardt Algorithm) jeste algoritam koji ažurira težinske i bias koeficijente prema LM optimizaciji, i smatra se pseudo-metodom drugog reda za treniranje NM, jer se Hesijanova matrica ne računa, već aproksimira prema Gaus-Njutnovoj metodi [9]. Hesijanova matrica predstavlja kvadratni parcijalni izvod drugog reda od funkcije za koju se računa. Proces ažuriranja težinskih koeficijenata vrši se na sledeći način:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e. \quad (2.1)$$

gde je J Jakobijanova matrica, koja sadrži prve izvode grešaka, a parametar e je vektor grešaka mreže. Za malu vrednost skalara μ , proces se svodi na GNA metod sa aproksimacijom Hesijanove funkcije, dok za slučaj kada je skalar μ veliki, proces se svodi na *Gradient descent* metod [6].

2.3.3 Bayesian Regularization (BR) backpropagation algoritam

BR algoritam podrazumeva ažuriranje vrednosti težinskih i bias koeficijenata po uzoru na LM optimizaciju. Minimizuje se linearna kombinacija kvadrata grešaka i težinskih koeficijenata, tako da se generiše mreža koja će imati sposobnost generalizovanja novih ulaznih podataka. Prema BR, objektivna funkcija je kao u izrazu (2.2)

$$F(\omega) = \alpha E_\omega - \beta E_D. \quad (2.2)$$

gde je E_ω suma kvadrata težinskih koeficijenata, E_D je suma kvadrata grešaka obučenog modela, pri čemu su u BR težinski koeficijenti mreže nasumične vrednosti, pa se raspodela težinskih koeficijenata i trening set smatraju Gausovom raspodelom. Parametri objektivne funkcije $F(\omega)$ iz izraza (2.2), α i β , definišu se uz pomoć Bajesove teoreme, i to na sledeći način:

$$P(\alpha, \beta | D, M) = \frac{P(D | \alpha, \beta, M) P(\alpha, \beta | M)}{P(D | M)}. \quad (2.3)$$

gde je D raspodela težinskih koeficijenata, M predstavlja samu arhitekturu neuronske mreže, $P(D | M)$ je faktor normalizacije, $P(\alpha, \beta | M)$ je uniformna gustina raspodele regularizacionih parametara, a $P(\alpha, \beta | D, M)$ je funkcija izglednosti raspodele D [6].

3. ANALIZA PROBLEMA

Interesovanje naučnika u oblasti 3D TV najviše je raslo u 19. i 20. veku, počev od razvoja prvih stereoskopskih 3D televizijskih sistema pa do prvih snimljenih 3D filmova, koje su gledaoci mogli videti samo u specijalno opremljenim bioskopima, jer je sama tehnologija bila izuzetno zahtevna i skupa, a do sredine prošlog veka televizori nisu bili još uvek zastupljeni u domaćinstvima.

2009. godine pojavljuju se televizori koji prikazuju specijalnu stereoskopsku sliku – anaglif¹. U ovom slučaju posmatrači moraju da nose specijalne 3D naočari koje imaju stakla različite boje, najčešće plavo i crveno, tako da predstavljaju filtere, i propuštanjem odgovarajućih boja svako oko dobija različitu sliku koje se u mozgu obrađuju kao jedna trodimenzionalna slika. Stereoskopska slika, dobija se pomoću sistema kamera koje moraju biti savršeno sinhronizovane, jer i najmanje odstupanje u boji ili poravnanju objekata može potpuno pokvariti 3D efekat kod posmatrača. Uz to, sam postupak je izuzetno zahtevan, a ne može se zanemariti ni cena neophodne opreme. Uspeh 3D televizije ne zavisi samo od tehnologije prenosa i prikaza, već i od samog sadržaja, odnosno njegove dostupnosti u 3D obliku. Postoji velika količina 2D sadržaja koji treba da se konvertuje u 3D oblik,

kako bi se omogućio njihov prikaz u 3D TV a ujedno i nadoknaditi nedostatak stereo sadržaja za potrebe 3D TV.

Postojeće tehnike automatske konverzije 2D u 3D sadržaj, zasnivaju se na principu integracije mape dubine u 2D sliku, pri čemu se mapa dubine dobija na osnovu više uzastopnih frejmova iste scene, i analizom određenih binokularnih i monokularnih obeležja kao što su binokularni disparitet, odnosno paralaksa pomeraja, defokus i linearna perspektiva [7].

4. PREDLOŽENO REŠENJE

Postupkom koji se predlaže, data je jedna od mogućnosti za poboljšanje rezultata dobijenih postojećom metodom [2] automatskog dobijanja mape dubine na osnovu jedne slike. Slično kao u [1], primenjen je princip estimacije vrednosti dubine na ivičnim pikselima, i propagacije tih vrednosti na celu sliku, nakon koje se dobija mapa dubine. Pretpostavka je da se dubina menja postepeno u slici, osim u regionima ivica sa povećanom frekvencijom sadržaja, gde promene dubine dostižu svoj maksimum.

¹ anaglifa je stereoskopska slika sastavljena od dve slike, jedna je u superpoziciji u odnosu na drugu, jedna plave, jedna crvene boje načešće, tako da stvaraju 3D efekat kada se posmatraju kroz 3D naočari

Zbog toga se, u cilju dobijanja pouzdane estimacije dubine u tim regionima, računaju vrednosti dubine na ivicama. Za obeležja dubine, uzeta su tri obeležja – „magla“, vertikalne ivice i zamućenost.

4.1 Opis postupka

Postupak se sastoji iz faze izdvajanja obeležja i faze generisanja mape dubine.

Faza izdvajanja obeležja podrazumeva da se za svaki 4x4 blok piksela formira vektor obeležja, gde je jedan blok predstavljen jednom vrednošću. Blokovi se smiču za jedan piksel, tako da se na kraju dobijaju tri nove matrice, veličine originalne ulazne slike, sa izračunatim vrednostima za tri obeležja.

Obeležje vazdušne perspektive, ili „magla“ izdvaja se pronalaženjem tzv. tamnog kanala, odnosno najmanje vrednosti R, G i B kanala, po obrascu 4.1:

$$I_{dark}(x, y) = \min_{k \in \{R, G, B\}} \left(\min_{x, y \in \Omega} I_k(x, y) \right) \quad (4.1)$$

gde Ω predstavlja blok piksela 4x4, a I_k predstavlja minimalnu vrednost svakog kanala, crvenog, zelenog i plavog, unutar bloka piksela. Obeležje vertikalnih ivica izdvaja se računanjem horizontalnog gradijenta, po obrascu (4.2):

$$\bar{v}(x, y) = \sum_{(x, y) \in \Omega} \frac{v(x, y)}{M} \quad (4.2)$$

gde Ω predstavlja 4x4 blok, M je broj piksela u bloku, a $v(x, y)$ označava vrednosti horizontalnog gradijenta. Za izdvajanje obeležja defokusa, odnosno zamućenosti, korišćena je tehnika estimacije zamućenosti, predstavljena u [2], koja računa drugi izvod Gausovog filtra za određivanje zamućenosti objekata u slici.

Kada se izračunaju tražena obeležja, potrebno je izdvojiti samo vrednosti sa ivičnih piksela. To podrazumeva prvenstveno detekciju i izdvajanje ivica u slici, što se radi Canny-jevom filtrom, sa pragom za ivicu 0.1. Koristeći poznate lokacije ivice, izdvoje se vrednosti ivičnih piksela estimiranih vektora obeležja, i dobijaju se tzv. „retke“ matrice za sva tri obeležja, tj. sve vrednosti matrice su nule, osim piksela ivice, koji imaju vrednost odgovarajućeg obeležja. Ceo postupak se iterativno ponavlja za sve slike skupa, i izdvojena obeležja se dodaju u jedan skup, tako da se na kraju formira trening skup vrednosti sva tri obeležja dubine na ivičnim pikselima, za sve slike iz skupa, koji će predstavljati ulazni skup podataka za sledeću fazu.

Uporedo sa tim, izdvajaju se i referentne vrednosti slika za ivične piksele (vrednosti referentne mape dubine), i dodaju u skup podataka za obuku neuronske mreže. Izlaz iz prve faze su dva skupa podataka za ulazne slike i njima odgovarajuće referentne mape dubine.

Druga faza generisanja mape dubine obuhvata tri koraka: podelu ulaznog skupa podataka na trening i test skup, obučavanje neuronske mreže za dobijanje vrednosti dubine na ivicama, i propagaciju vrednosti dubine na celu sliku kako bi se dobila potpuna mapa dubine. Na osnovu novih ulaznih vrednosti, obučeni model NM će odlučivati o vrednosti dubine na konkretnom ivičnom pikselu. Skup

podataka za treniranje je u obliku $\{\{p_1, t_1\}, \{p_2, t_2\} \dots \{p_n, t_n\}\}$, gde svako p predstavlja vektor obeležja u obliku $\{o_1, o_2, o_3\}$, a t ciljnu, tj. referentnu vrednost dubine dobijenu iz referentne slike. Obučeni model NM testira se na novim vrednostima za koje nije bio obučen, a rezultat je „retka“ matrica vrednosti dubine. Izabrani algoritmi koji realizuju obučavanje neuronske mreže jesu *Levenberg-Marquardt* i *Bayesian Regularization* algoritam. Propagacija vrednosti vrši se uz pomoć interpolacije koja uzima u obzir ivice. (*edge-aware interpolation*) [3]. Postupak propagacije se svodi na optimizacioni problem, odnosno rešavanje linearnog sistema jednačina u kojem figuriše „retka“ matrica.

$$(L + \lambda D_s) \alpha = \lambda s. \quad (4.3)$$

gde α predstavlja popunjenu mapu dubine, s je „retka“ matrica, λ je ponder između oštine „retke“ matrice, i „ispeglanosti“ krajnje slike dubine, D_s predstavlja dijagonalnu matricu, čiji su elementi sa glavne dijagonale 1 za piksele iz „retke“ matrice, a 0 za ostale piksele, i L predstavlja Laplasovu matricu matiranja (eng. *matting*). Sa obzirom na to da su vektori obeležja bazirani na blokovima piksela, ovo može uzrokovati neprecizne estimacije dubine na nekim ivičnim pikselima. Zbog toga se, pre estimacije vrednosti dubine, primenjuje bilateralno filtriranje na „retkoj“ matrici.

4.2. Mere za ocenu kvaliteta predloženog rešenja

Za procenu kvaliteta dobijenih mapa dubine, korišćene su subjektivne ocene, u kombinaciji sa objektivnim merama, i to za slučaj kada za rezultatnu sliku postoji i referentna slika (*full-reference* metrike):

- 1) Srednja kvadratna greška - MSE (*Mean Squared Error*),
- 2) Vršni odnos šuma i signala - PSNR (*Peak Signal-to-Noise Ratio*),
- 3) Normalizovana kros-korelacija - NCC (*Normalized Cross-Correlation*),
- 4) Prosečna razlika - AD (*Average Difference*),
- 5) Strukturni sadržaj - SC (*Structural Content*),
- 6) Maksimalna razlika - MD (*Maximum Difference*),
- 7) Normalizovana apsolutna greška - NAE (*Normalized Absolute Error*) i
- 8) Strukturna sličnost – SSIM (*Structural Similarity*).

5. ISPITIVANJA I REZULTATI

Tokom istraživanja, korišćeno je MATLAB (R2014) programsko okruženje. Za obučavanje neuronske mreže, korišćene su postojeće MATLAB funkcije iz alata *Neural Network Toolbox-a*, za *Levenberg-Marquardt* algoritam funkcija je *trainlm* a za *Bayesian Regularization* algoritam, funkcija je *trainbr*. Prilikom testiranja, varirane su vrednosti broja neurona u skrivenom sloju NM (3, 6 i 10 neurona), kako bi se utvrdilo ponašanje i performanse NM u zavisnosti od tog broja.

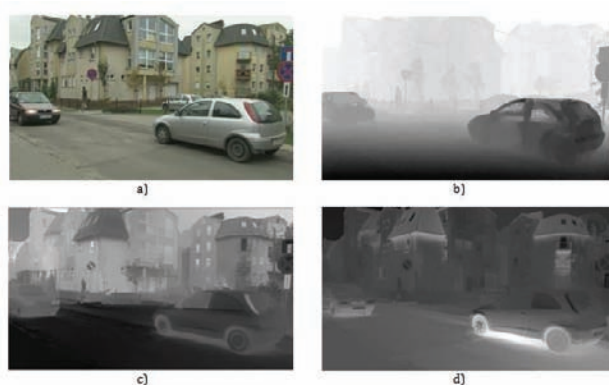
5.1. Rezultati

Radi bolje preglednosti, rezultati su dati kao uporedni pregled na slikama 2. i 3.: a) originalna slika, b) referentna mapa dubine, c) dobijena mapa dubine predloženom metodom, i d) mapa dubine dobijena metodom [2]. Metod [2] pravi greške pri proceni dubine što se najbolje vidi iz slike 2., a u slici 3. posebno je uočljiv problem

pogrešne klasifikacije senke. Sa druge strane, predložena metoda je dala vidno bolje rezultate ukoliko se uporede sa odgovarajućim mapama dubine.



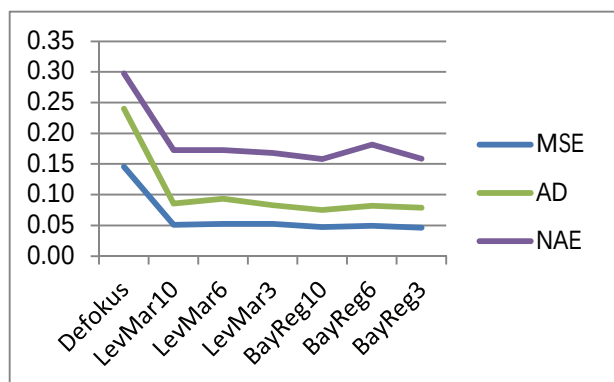
Slika 2. Poređenje rezultata sa rezultatima iz [2]



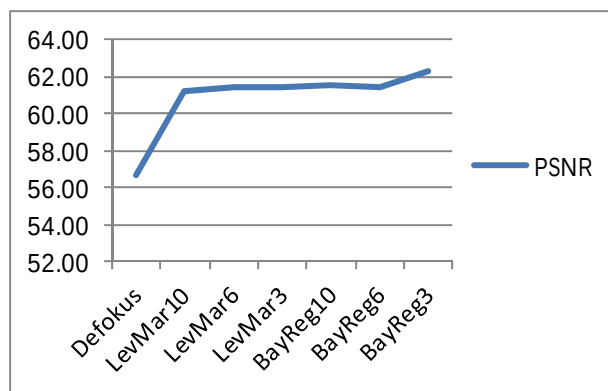
Slika 3. Poređenje rezultata sa rezultatima iz [2]

Na slikama 4. i 5. dat je grafički prikaz srednjih vrednosti metrika za metodu defokusa [2] i predloženu metodu, za sve vrednosti broja neurona u skrivenom sloju. Najveću osetljivost na razlike dobijenih rezultata u poređenju sa referentnim slikama, pokazale su metrike MSE, AD i NAE i PSNR.

Slabija osetljivost metrika za dobijene mape dubine se vidi za NCC, MD i SSIM, međutim i u tom slučaju se mogu uočiti nešto bolji rezultati za mape dubine na osnovu predložene metode. Strukturni sadržaj (SC) je jedina metrika koja je dala nešto lošije rezultate za metodu opisanu u ovom radu u odnosu na metodu.



Slika 4. MSE, AD i NAE metrike



Slika 5. PSNR metrika

6. ZAKLJUČAK

Iz priloženih grafika, uočljivo je poboljšanje vrednosti pojedinih metrika za predloženu metodu, u odnosu na metodu [2] koja koristi samo obeležje defikusa za dobijanje mape dubine. Analizirane objektivne metrike nisu sasvim odgovarajuće za procenu kvaliteta mape dubine, ali se zato iz datih slika rezultata, na osnovu subjektivne ocene može uvideti bolji kvalitet dobijenih mapa dubine. Postoji nekoliko ideja za dalju optimizaciju algoritma i poboljšanje i rezultata, a ovaj rad predstavlja doprinos postojećim rešenjima i jedan novi aspekt sa kog se posmatra problem generisanja mape dubine.

7. LITERATURA

- [1] Wafa, A., Nasiopoulos, P., & Leung, V. C. (2015, May) "Automatic Real-time 2D-to-3D Conversion for scenic views", IEEE Quality of Multimedia Experience (QoMEX), 2015
- [2] Zhuo, S., & Sim, T. (2011, March) "Defocus map estimation from a single image". Pattern Recognition, 44(9), pp. 1852-1858.
- [3] Levin, A., Lischinski, D., & Weiss, Y. (2008, February). "A closed-form solution to nature image matting". Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30(2), pp. 228-242.
- [4] http://www.hitl.washington.edu/projects/knowledge_base/virtual-worlds/EVE/III.A.1.c.DepthCues.html
- [5] <http://solair.eunet.rs/~ilicv/neuro.html>
- [6] Baghirli, O. (2015, June) "Comparison of Lavenberg-Marquardt, Scaled conjugate gradient and Bayesian Regularization backpropagation algorithms for multistep ahead wind speed forecasting using multilayer perceptron feedforward neural network", Upsalla University
- [8] https://en.wikipedia.org/wiki/2D_to_3D_conversion
- [9] Foresee, F. D. & Hagan M. T. (1997, June) "Gauss-Newton approximation to Bayesian Learning", Proceedings of the 1997 International Joint Conference on Neural Networks, Vol. 3, pp.1930-1935

Kratka biografija:



Milana Antin rođena je u Zrenjaninu 1989. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Obrada signala, odbranila je 2015.god. Trenutno je zaposlena u istraživačko-razvojnom institutu RT-RK u Novom Sadu.

A solution of an ontology used as a data model for an Internet content provider

Stefan Pijetlović, Ilija Bašičević, *Member, IEEE*, Đorđe Kovačević, Nikola Teslić, *Member, IEEE*

Abstract — this paper describes a usage of ontologies for describing the data model for a cloud based Internet content provider. The system uses the events watched on the TV along with the user profiles as input in order to search for the content which can be of interest to the user. This content originates from both the digital video broadcasting (DVB) stream and also from the Internet and is mapped to the ontology regardless of its origin. All the acquired content is then filtered and presented to the end user based on his or her profile.

Keywords — content providing, digital television, Internet, ontology.

I. INTRODUCTION

WE are witnessing the rapid expansion of the Internet which has the potential to become the most dominant media in the future. The amount of information present on the Internet is enormous and is growing every day. Also, according to various online surveys and studies, the number of people using Internet for educational and entertaining purposes is in the increase. But even so, television still remains the most widespread media in most parts of the world.

In the past few years, a new trend appeared – connection of the expanding digital television with Internet on devices such as set-top boxes and smart TVs. One of the issues however is the large variety of protocols used, the data format is very different and when it comes to metadata, the amount of it varies greatly depending on the provider of this information.

To solve this issue, we have developed an ontology to model all the important information about the content items that we need. All the data coming from different sources is mapped to this ontology and then, filtered out, delivered to the end users.

The paper is organized as follows: first, an introduction to ontologies is given, followed by a formal/theoretical solution of the problem stated before. Afterwards, an

implementation is presented, along with evaluation and a conclusion with future plans.

II. ONTOLOGY

The term ontology originally comes from philosophy where it represents a study of the nature of being along with its general, constitutive and fundamental properties. Similarly, in computer and information science, ontologies are used to formally express knowledge within a domain. One of the first definitions of the term ontology was given by Tom Gruber in his paper [1] where he sees ontologies as a way of describing the terms within a system and the various connections and relations which coexist with the terms. The knowledge within one particular domain is represented by a hierarchy of classes, types, properties and their relations.

The question that comes to mind is – why would anybody want to develop an ontology in the first place? Some of the answers given in [2] include: knowledge analysis, classification and categorization, reuse of knowledge in similar systems or integration into other systems, separation of the domain from operational knowledge and so on. The development of an ontology of a particular system is usually not the goal by itself. Rather, it is one of the steps in designing quality software.

In computer science, ontologies are most frequently used in the areas of Artificial Intelligence (AI), Semantic Web, biomedicine, managing libraries, etc. During the 70s, developers in the field of AI have discovered that the modeling of knowledge is of vital significance in large and powerful AI systems. More recently, ontologies became interesting for the concept of Semantic Web. This is a movement led by the World Wide Web Consortium (W3C) whose cause is to transform the current state of the web into a web of data which the machines can understand and process. In his paper [3] Henry Kim predicts that the Semantic Web is the future and that the development of the concept depends mostly on the development of the numerous ontologies which are the fundamental parts of the project. Nowadays, they are being used in the areas of multimedia and quality of service (QoS) [4], [5] and are becoming a part of the architecture of these systems.

Due to the fact that the Internet and digital television domains are enormous, we have to make some kind of classification and categorization in order to separate the necessary information from the unnecessary. By doing so, we create a hierarchy where the most general properties are at the top, and the more specific properties are “below”.

This project was partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia under grant number TR32031, year 2014.

Stefan Pijetlović, Fakultet tehničkih nauka, Trg Dositeja Obradovića, 21000 Novi Sad, Srbija, e-mail: stefan.pijetlovic@rt-rk.com

Ilija Bašičević, Fakultet tehničkih nauka, Trg Dositeja Obradovića, 21000 Novi Sad, Srbija, e-mail: ilija.basicevic@rt-rk.com

Đorđe Kovačević, Fakultet tehničkih nauka, Trg Dositeja Obradovića, 21000 Novi Sad, Srbija, e-mail: djordje.lj.kovacevic@rt-rk.com

Nikola Teslić, Fakultet tehničkih nauka, Trg Dositeja Obradovića, 21000 Novi Sad, Srbija, e-mail: nikola.teslic@rt-rk.com

NAPOMENA:

a) Ovaj rad proistekao je iz master rada Stefana Pijetlovića. Mentor je bio prof. dr Nikola Teslić.

b) Rad je prethodno publikovan na konferenciji TELFOR, Beograd, novembar 2013.

The components which make up a particular ontology are: classes, data properties (name of the term, its relevant properties), object properties (relations between different terms) and constraints (stating which relations are forbidden). Every single class represents one term in the knowledge domain. In our solution the class named *Content* represents all the digital content which can be delivered to the end user, be it an event from the stream, a teletext page or a video clip or article from the Internet. Every single event or webpage is an instance of some child class of the class *Content* but the instances themselves are not a part of the ontology. The ontology is only a description of the system, terms and their relations. Combined with all the individuals (instances) this description makes up the knowledge base.

III. FORMAL SOLUTION

As mentioned before, the domains of digital television and especially Internet are vast and in order for our system to work properly, we have to create a usable subset of all information available.

Our system's purpose is to provide metadata which is appropriate to the currently watched events and to recommend content from the stream and Internet that might be interesting to the user. The main issue covered by this paper is how to model the content items which come from two disjunctive sources and treat them as equals while at the same time use as much useful information possible. The initial solution [6], before the ontology was introduced, mainly relied on the data which was broadcasted such as start and end time, genre, short and long description etc. There was a strict difference between the DTV events which were the "main" content items and the other web content which was considered additional. For the classification of events, the main criteria were the genres. However, this type of classification proved to be too general and a wide range of content from the web could not fit into the genres provided by the DVB standard [7]. To cope with this problem, the concept of tags was introduced. Tags represent words or terms which are in some way related to the events and allow more specific categorization and easier searching of all the events related to the same tag. This did not fix the problem completely, it just introduced another which is the number of tags needed – too little means the tags are not specific enough and we have not made any difference. On the other hand, too many tags mean that the events will fall into several categories and the maintenance of the tags becomes another issue. And so, a new concept had to be introduced. Based on the ontology of the online knowledge base Freebase [8] we have developed our own ontology, although a lot less complex.

There is no single root class for our ontology, except the generic class *Thing* which is the default parent of all classes similar to the *Object* class in Java. Instead, there are two branches which shape up the structure of our system. One is the class called *Content* which represents all forms of content which can be delivered to the end user, and the other is the *Metadata* class representing all

metadata (data about data) associated with the content items.

The *Content* class has some data properties which are common for all types of content such as URI, content name, description, language, parental rate and so on. Other more specific data properties are applied on the children classes. The children classes are called and represent *TimeRestrictedContent* and *TimeUnrestrictedContent*. These restrictions mainly apply to the contents which have a duration, such as a video clip or an audio track, and if they are scheduled (live TV or radio) or can be accessed at any point of time (for example YouTube clips). An example of time unrestricted content items are web pages like Wikipedia articles.

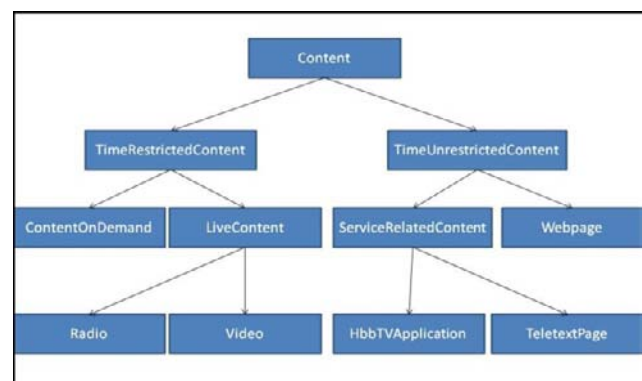


Figure 1 – Content class and its successors

The *Metadata* class has a lot of children classes which represent specific real life terms such as *Person*, *Book*, *Event*, *Location*, *Movie*, *Sport* and many more. There is an important object property named *About* which models relationships between different entities and content items and tells us whether a certain content item is about a specific person, book, movie and so on. The *Person* class has a lot of subclasses, each representing a profession. If we only know the person's name and age, that does not give us a lot of useful information, due to the fact that all people have a name and age. We could link this person with all other people of the same name or age but that does not give us any semantic value. But if we know that this person is an actor and he acted in several movies that the user liked, we can then recommend all the movies where this person appeared. The relations actor acts in a movie is described using object properties and different classes have different object properties. Using this type of structure, we can connect many terms which could not be connected in the old version of the system.

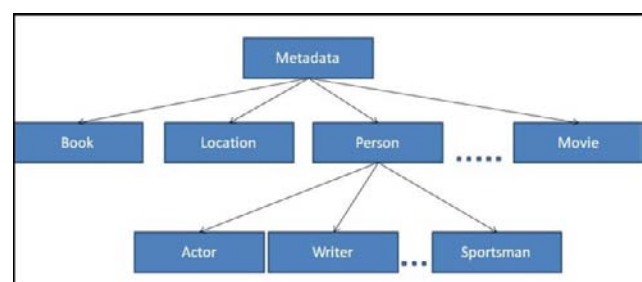


Figure 2 – Metadata class and its successors

IV. IMPLEMENTATION

The ontology was modeled using the Protégé 4.3 ontology development tool [9]. It can be filled with individuals in order to create a knowledge base. The base can be accessed through a Java API (Application Programming Interface). There is also an option to automatically generate the Java classes based on the ontology model but this is also not effective because in order to access an object of certain classes, we have to cast the object several times. In order to get the most out of the knowledge base, it should be stored into graph based databases. Due to historic reasons which mainly include an already developed infrastructure, the database we use is relational. These are the main reasons why we rewrote the ontology using Enterprise Java technologies. The classes are based on those modeled in the ontology with several changes mainly to improve the performance.

The ontology spreads across several modules in the system architecture, although the biggest part is concentrated in two. The *CloudEngineComponents* is a library where all the entities and connections are defined. It contains the *ContentItem* class which is the replica of the *Content* class from the formal solution. The other module is called *KnowledgeBase* and is dedicated to the metadata classes and communication with Freebase.

The *CloudEngineComponents* module handles the user actions, user profiles, content recommendation and many more. The *ContentItem* class is fundamental for our system. Most of the data properties are replicated as fields, such as URI, name, description, language. It has an enumeration which separates various content types and so we have content on demand, live video/radio, webpage, teletext page and an HbbTV application. The *ContentMetadata* class has the minimal set of properties which are stored in the database. These properties include an ID, entity name, the source of information for the entity and the data source key used to quickly fetch all the information about the entity. The purpose of this class is to link an entity that shows up in the description of the content item to a particular online database and to pull as many connections as possible to other related entities. All these connections are treated differently based on the user's profile. The *TimeRestriction* class is present to model the duration and scheduled program. It is a member in the *ContentItem* class and is present in order to reduce the number of classes and to avoid generic classes where this is not needed. There is also a filter class used to search the database by specific criteria.

The *KnowledgeBase* module is divided into two packages: one which has the API towards other knowledge bases and the second which contains the entities. The API consists of two methods – one is responsible for searching the entity and the other for requesting it.

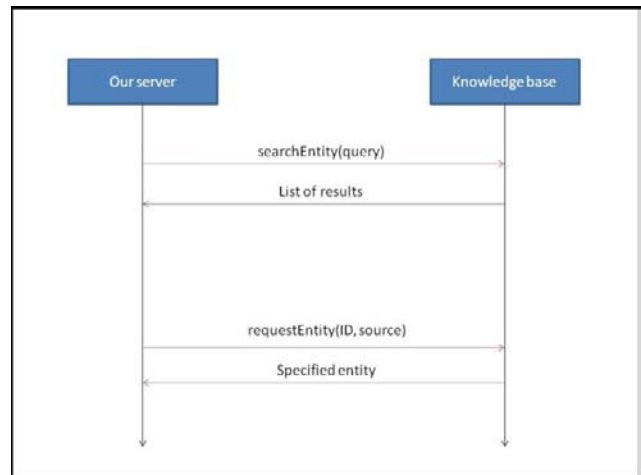


Figure 3 – Example of communication between our server and the knowledge bases

When we recognize a name of a person, movie or location in the description of the content item using natural language processing [10], we send queries to the knowledge bases and then map the top results to some of the entities in our system. Once we define a certain entity, it is saved in the database along with its data source and data source key. Next time the name of this entity turns up in the description we use the *requestEntity* method directly. The query represents an independent class which has several other properties besides the name of the entity, such as the preferred language, type of the entity (in case we have two or more people with the same name but are famous for different things) and so on. All the classes from the *Metadata* hierarchy have their Java counterparts which are abstract. These classes have only the core properties (a list of movies and awards for actors for example). For each knowledge base there is a set of child classes which are concrete realizations and have fields depending on what information the base has to offer. In case of Freebase, we have a *FreebaseActor* or *FreebaseMovie* class which only use a small subset of information which we think might be interesting to the user.

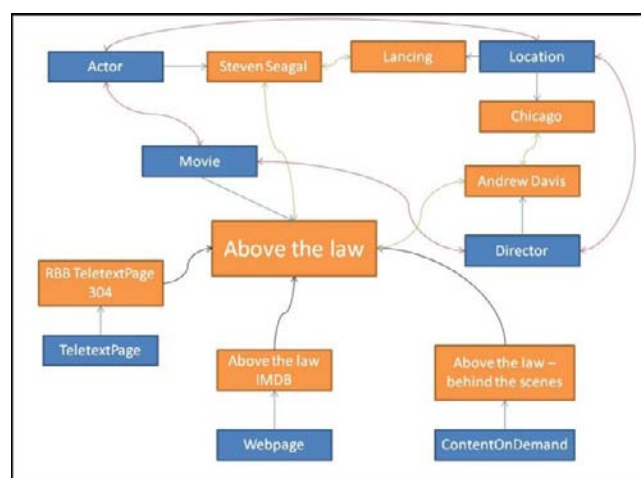


Figure 4 – Example of complex relations which can be formed when using and ontology

V. EVALUATION

Testing the ontology itself is not possible due to the fact that it spreads across many modules of the system and relies on components outside the system such as various websites, the clients, network capacity etc. And so, in order to verify the ontology logic, we used the FaCT++ reasoner [11].

Reasoners are software components which take the rules defined in the ontologies and knowledge bases and provide RDF models. For our needs, we only used the reasoner's ability to check the logical consistency of the system. Other capabilities include logical optimizations using the Boolean algebra for the purpose of generating a more compact model. We did not need these optimizations due to the fact that we implemented the whole model in Java. Since we got the model with no errors, we can presume that there is no illogic.

The main contribution of the ontology is the semantic richness, precise modeling and better search possibilities. In Table 1 we can see the difference in the number of classes used to describe the system.

TABLE 1: COMPARISON OF THE NUMBER OF CLASSES IN THE THEORETICAL AND IMPLEMENTED SOLUTION

Number of classes	Theoretical solution	Implemented solution
Content hierarchy (Content/ContentItem class and it's successors)	7	5
Metadata hierarchy (Metadata/Entity class and it's successors)	15	16

In order to take up less space in the database we reduced the number of classes which describe the content items. The concrete metadata hierarchy is equivalent to the theoretical solution with an addition of the *Result* class. The sole purpose of this class is to provide a faster mapping of the search results. It plays an important role in minimizing unnecessary communication but has no semantic value and therefore was not covered by the ontology in the first place.

VI. CONCLUSION

In the paper we presented one usage of ontologies for describing the data model for an Internet content provider. We presented the theoretical solution to the problem of managing content which comes from various sources and also given an insight into the implementation. We have shown the advantages of using an ontology based model through complex searches which can be realized. We note that this project is still a proof of concept and that is why there are no numeric results considering the response time, scalability and other performance issues.

As for future plans, the ontology itself will be modified and improved as time goes by. Place for improvement is present in the natural language analysis module and the recommendation systems.

REFERENCES

- [1] Tom Gruber (1995.) "Toward Principles for the Design of Ontologies used for Knowledge sharing", International Journal of Human-Computer Studies
- [2] Natalya F. Noy, Deborah L. McGuinness, "Ontology Development: A guide to creating your first ontology" http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html loaded 24.09.2014.
- [3] Henry Kim, "Predicting how ontologies for the Semantic Web will evolve", Communications of the ACM February 2002/Vol. 45, No. 2 H. Lim, *Signal Detection*. New York: Springer-Verlag, 1985, ch. 4.
- [4] Myriam Lamolle, Chan Leduc, Ludovic Menet, "An Ontology-driven Architecture Approach for Open Multimedia Service Oriented Architectures", 2010. Third International Conference on Communication Theory, Reliability, and Quality of Service
- [5] Jorge Gómez-Montalvo, Ernesto Exposito, "A Semantic Approach to User-based QoS Provision for Multimedia Services in Home Networks", 2010. Third International Conference on Communication Theory, Reliability, and Quality of Service
- [6] S. Pijetlovic, N. Jovanovic, N. Jovanov, S. Ocovaj, "One solution of a system for data acquisition and storage from the digital television transport stream and its exposure to the clients", 21st Telecommunications Forum TELFOR, Belgrade, November 2013.
- [7] TSI Digital Video Broadcasting (DVB), Specification for service information (SI) in DVB systems: http://www.etsi.org/deliver/etsi_en/300400_300499/300468/01.11.01_60/en_300468v011101p.pdf loaded 24.09.2014.
- [8] Freebase, <https://www.freebase.com/> loaded 24.09.2014.
- [9] Protégé, http://en.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_%28software%29 loaded 24.09.2014.
- [10] A. Beserminji, M. Knežević, G. Stupar, V. Peković, "One implementation of search feature in an Internet based EPG provider", The 4th IEEE International Conference on Consumer Electronics, Berlin, September 2014.
- [11] Dmitry Tsarkov, Ian Horrocks, "FaCT++ Description Logic Reasoner: System Description", School of Computer Science, University of Manchester, UK <http://www.cs.ox.ac.uk/ian.horrocks/Publications/download/2006/TSHo06a.pdf> loaded 24.09.2014.

UGRADNJA ALATA ZA KONTINUALNU INTEGRACIJU U OKRUŽENJE VISUAL STUDIO**INTEGRATION OF CI TOOLS INTO THE VISUAL STUDIO ENVIROMENT**

Dunja Torbić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je opisana ugradnja alata za kontinualnu integraciju u **Visual Studio** razvojno okruženje putem **Visual Studio** paketa. Objašnjena je realizacija paketa, njegovo distribuiranje, ažuriranje i servisiranje. U okruženje se integrišu sledeći alati: **night build**, **check warnigns**, **sonar** analiza, jedinično testiranje, testiranje komponenti i testiranje performansi. Ovaj paket je razvijen korišćenjem C# programskog jezika u okruženju **Visual Studio**.

Abstract – This paper describes basic concepts and technologies used for extending Visual Studio Integrated Devopment Enviroment by Continuous Integration tools. Following tools are integrated into the enviroment: **night build**, **check warnigns**, **sonar** analisys, unit testing, component testing, and performance testing. This software module is developed using C# programming language in Visual Studio Enviroment and Visual Studio Package technology.

Ključne reči: **Visual Studio** paket, kontinualna integracija

1. UVOD

Kontinualna integracija predstavlja praksu u razvoju softvera po kojoj programeri više puta na dan spajaju svoje radne kopije u jedan centralni repozitorijum. Ovom praksom se u velikoj meri izbegavaju konflikti pri spajanju radnih kopija i štedi vreme na njihovo razrešavanje [1].

Cilj kontinualne integracije jeste da se u svakom trenutku, kada se preuzme poslednji kod sa repozitororijuma, on uvek uspešno bilduje i uspešno prolazi sve testove. Kako bi se ovaj cilj postigao, potrebno je pre svake integracije proveriti da li radna kopija ispunjava zadate standarde i da li se njenim spajanjem kvalitet zajedničke celine neće narušiti.

Postoji nekoliko prolaza kroz koje radna kopija mora proći pre integrisanja. Sledeći alati se koriste u ciklusu kontinualne integracije i predstvaljaju njen prvi prolaz:

- **night build**,
- **check warnings**,
- **sonar** analiza,
- jedinično testiranje,

- testiranje komponenti i
- testiranje performansi.

Navedeni alati se pokreću van okruženja u kome se programira, tako što se parametrima navodi koje akcije treba da se izvrše i nad kojim projektom. Rezultati izvršenja alata su logovi koje je potrebno ručno pretražiti. To od programera zahteva dodatno vreme i napor, a neophodna je i obuka za korišćenje alata.

Uzevši u obzir učestalost korišćenja ovih alata, vreme potrebno za prosleđivanje parametara i pretraživanje logova nije zanemarljivo, i potrebno je redukovati ga.

Kako bi se ovaj problem rešio, potrebno je integrisati navedene alate u okruženje u kome se programira (**Visual Studio** okruženje), tako da se oni pokreću na isti način kao i već postojeći **Visual Studio** alati, da se rezultati iz logova formatiraju i predstavljaju čitljivo unutar okruženja i da na svaku lokaciju iz njih može da se pristupi automatski putem navigacije.

Cilj ovog rada je da se iskoriste mogućnosti proširivanja **Visual Studio** okruženja, kako bi se u njega ugradili **Continuous Integration** alati.

2. KREIRANJE PROŠIRENJA VISUAL STUDIO OKRUŽENJA ZA KONTINUALNU INTEGRACIJU

Iako je broj ugrađenih opcija ograničen i ponekad nedovoljan programerima, **Visual Studio** je alat koji se može proširiti. **Microsoft** je kreirao mehanizme, koji omogućuju programerima da lako ugrade nove opcije, koje su im potrebne u **Visual Studio** okruženje [2].

Postoji nekoliko načina na koji se **Visual Studio 2013** programsko okruženje može proširiti.

Ako je potrebno integrisati veliku komponentu u **Visual Studio**, kreiraju se **Visual Studio** paketi (**VSPackages**). **Visual Studio** paketi predstavljaju centralnu gradivnu jedinicu **Visual Studio** okruženja, tj. i sam **Visual Studio** je izgrađen većinom od kolekcije paketa. Pomoću paketa mogu se kreirati novi elementi interfejsa (meniji, komande, prozori, itd), ali se takođe može ugrađivati i u postojeće elemente [3]. Postoji set objekata na osnovu kojih se **Visual Studio** okruženje može proširivati. Model, kojim je predstavljena hijerarhija ovih objekata, zove se automatizovani model. Objekti automazivanog modela kontolišu glavne aspekte okruženja. Tako Visual Studio integrisano razvojno okruženje predstavlja **DTE** objekat. On je na vrhu hijerarhije automatizovanog modela.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Hajduković, red.prof.

Pomoću **DTE** objekta se pristupa svim objektima u okruženju, potrebnim za kreiranje paketa.

Ako je potrebno kreirati aplikaciju zasnovanu na **Visual Studio** tehnologiji, kreira se **Visual Studio Shell Isolated** aplikacija, koja se takođe zasniva na **Visual Studio** paketima.

Još jedan mehanizam za proširivanje **Visual Studio** okruženja zasnovan je na **Managed Extensibility Framework-u (MEF)**, i omogućava proširivanje i prilagođavanje **Visual Studio** editora.

Pored navedenih mehanizama postoji i **Add-in**, koji je prisutan još od prvih verzija **Visual Studio** okruženja i predstavlja sličan mehanizam **Visual Studio** paketu. Međutim **Add-in** predstavlja eksterni deo **Visual Studio** razvojnog okruženja, dok su paketi potpuno integrisani u njega i učitavaju se kao njegov deo. Takođe, **Add-in** mehanizam zastareva i ukinut u novijim verzijama **Visual Studio**-a.

Uzimajući u obzir domen problema, kao i verziju okruženja koja se proširuje, za njegovo rešenje je izabran **VSPackages** mehanizam, koji na najbolji način omogućuje ugradnju alata za kontinualnu integraciju u **Visual Studio 2013**. Ovim mehanizmom podržana su sva potrebna svojstva: omogućeno je kreiranje novih menija, kreiranje novih prozora i integrisanje u postojeće, kontrolisanje stanja korisničkog interfejsa, kao i dodavanje novih opcija.

Visual Studio paket za kontinualnu integraciju ugrađuje u **Visual Studio** okruženje komande za pokretanje **Continuous Integration** alata, i prikaz njihovih rezultata [4]. Ovi alati se pokreću putem ugrađene **CI Tools** stavke menija na glavnom meniju, prikazane na slici 1, i na kontekstnom meniju nad projektima ili **solution**-om [5]. **CI Tools** stavka menija sadrži tri grupe komandi: prva grupa obuhvata komande nad učitanim **solution**-om, druga komande nad selektovanim projektima, a treća sadrži samo jednu komandu, i to prečicu do opcija vezanih za ove alate. Komandama za akcije nad **solution**-om se može pristupiti i preko akceleratora.



Slika 1. Stavke menija za kontinualnu integraciju

Proces izvršavanja komandi vrši se na sledeći način:

1. Preuzimaju se projekti, folderi koji sadrže projekte ili **solution** nad kojim je komanda izabrana [6]. Na osnovu

njihovog naziva, putanje na disku do njih, i alata koji je izabran, kreiraju se parametri za naredbu koja treba da se izvrši. Za neke alate potrebno je pri kreiranju parametara uzeti u obzir i aktivnu **solution** konfiguraciju, kao i parametre definisane u **Options** prozoru. Na osnovu svih ovih parametara generiše se naredba, ili više njih.

2. Generisana naredba/naredbe se prosleđuju menadžeru procesa, koji ih jednu po jednu prosleđuje **Command Prompt**-u. **Command Prompt** tom naredbom pokreće određeni alat za kontinualnu integraciju. Ove akcije su pozadinske i nisu vidljive korisniku **Visual Studio** okruženja. Od trenutka kad se alat pokrene, onemogućuje se pokretanje ostalih alata kontinualne integracije i **Visual Studio build** alata. Takođe, dok je pokrenut **build** alat, alati kontinualne integracije su onemogućeni.

3. Menadžer procesa zatim sinhrono preuzima rezultate akcija iz **Command Prompt**-a, logova, fizičkih lokacija, i prikazuje ih korisniku putem stavke **Output** prozora sa imenom pokrenutog alata, **Error List** prozora, i novog **Test Report** prozora. Takođe, menadžer procesa prekida izvršavanje akcije, ukoliko je korisnik poslao komadu za prekid, i obaveštava korisnika da je proces prekinut. Načini na koji su rezultati predstavljeni različiti su za razne alate kontinualne integracije. Ono što je za sve alate zajedničko jeste ispis na **Output** prozoru, koji sinhrono obaveštava korisnika šta alat trenutno radi.

2.1. Integrisanje **night build** alata u **Visual Studio** okruženje

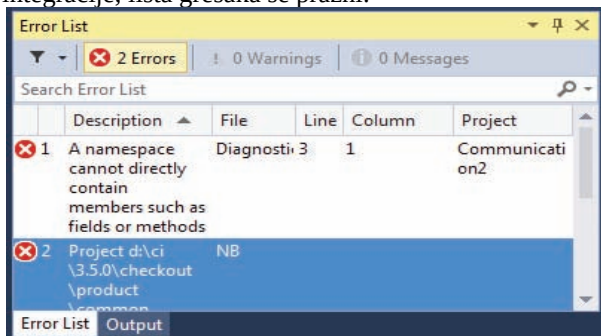
Night build predstavlja alat za bildovanje koda. Pre pokretanja **night build** komandi nad **solution**-om mogu se podesiti parametri, koji određuju koje tipove projekata je potrebno uključiti u **night build** (projekte sa produkcionom kodom, projekte jediničnog testiranja, testiranja komponenti ili testiranja performansi). Kreirana je stranica **Options** prozora, u kojoj se ovi parametri mogu podesiti. Podrazumevano je da svi tipovi projekata budu uključeni u **night build**, ali to korisnik može izmeniti. U menijima za kontinualnu integraciju su implementirane prečice do stranice **Options** prozora sa ovim podešavanjima.

Pri pokretanju **night build** komande generiše se naredba na osnovu parametara iz **Options** prozora, odabranih projekata ili **Solution**-a, i aktivne **Solution** konfiguracije. Zatim počinje ispis na **Output** prozoru o izvršavanju akcija.

Nakon izvršenja **night build** alata, generišu se dva log fajla. Jedan predstavlja spisak **MSBuild** grešaka, a drugi spisak konfiguracionih grešaka. Pre ugrađivanja, potrebno je iz loga preuzeti jednu po jednu grešku i isparsirati je, kako bi se definisalo šta je njen opis i gde se greška nalazi. Zbog različitih formata, ova dva tipa grešaka su integrisana u **Error List** prozor na dva različita načina. Kod **MSBuild** grešaka je navedena lokacija greške, a kod konfiguracionih ona nije dostupna, već je naveden alat koji je prijavio grešku, odnosno njegovi inicijali (**NB**), kao što je prikazano na slici 2.

Integracija grešaka u **Visual Studio** okruženje implementirana je u **Error List** prozor. Klikom na grešku, otvara se fajl, i navigira do linije i kolone gde se greška nalazi. Ako je **Visual Studio** već prijavio istu grešku koja se javila i prilikom izvršavanja **night build** alata, ona se

neće upisati u prozor, kako ne bi postojale dve iste greške u listi. Takođe, ako **Visual Studio** krene da prijavi grešku koju je **night build** alat već prijavio, duplikat se briše iz liste. Prilikom svakog pokretanja komandi kontinualne integracije, lista grešaka se prazni.



Slika 2. Integrisane greške u **Error List** prozoru

2.2. Integrisanje **check warnings** alata u **Visual Studio** okruženje

Check Warnings je alat kojim se proverava da li postoje upozorenja u prethodno izbuildovanom projektu ili **Solution**-u. Ovaj alat preuzme iz logova identifikacioni broj poslednjeg izvršenog **night build**-a. Zatim proverava nad koliko projekata i nad kojim repozitijumima je **night build** izvršen i nad njima proveriti koji je broj generisanih upozorenja, a zatim sve ove podatke ispiše na **Output** prozoru.

2.3. Integrisanje alata za testiranje u **Visual Studio** okruženje

U kontinualnoj integraciji, potrebno je primeniti nekoliko tipova testiranja:

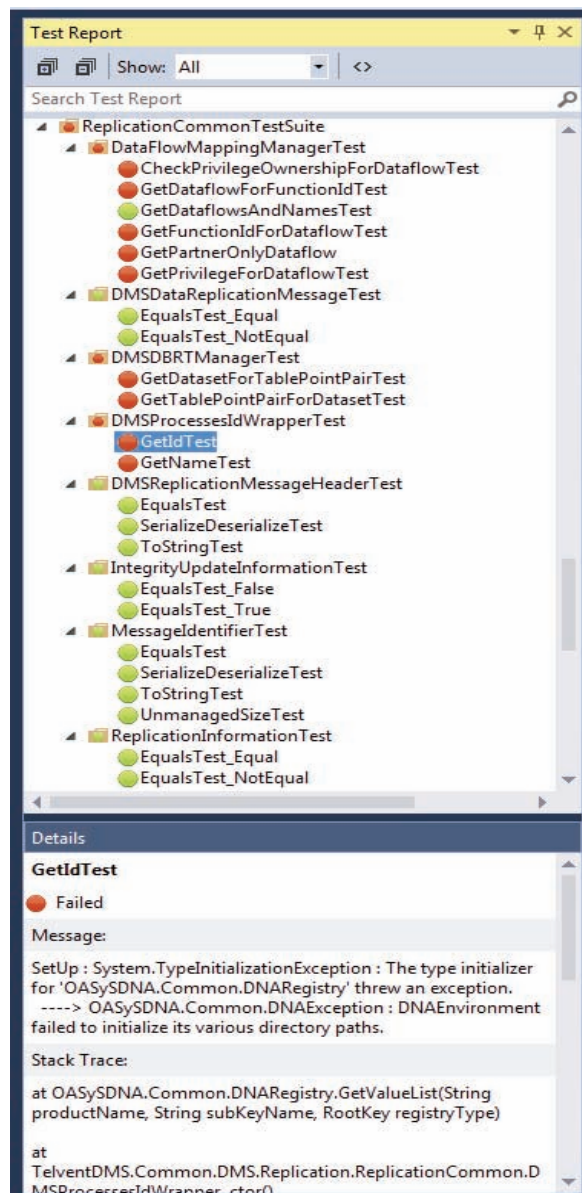
- **jedinično testiranje** - proces u kome se najmanji delovi koda, koji se mogu testirati, individualno testiraju, kako bi se utvrdilo da li funkcionišu onako kako je predviđeno,
- **testiranje komponenti** – u kom se pronalaze defekti u modulima i verifikuje se funkcionisanje softvera, i

- **testiranje performansi** - koje predstavlja proveru efektivnosti i brzine izvršenja koda. Svako od navedenih testiranja pokreće se pomoću zasebnog alata, a svaki alat prikaže rezultate na isti način, tako što se ispišu izvršeni testovi, i informacije o njima (uspešnost, poruku, lokaciju testa). Zbog istog formata rezultata kod svih vrsta testova, oni su integrisani na isti način u **Visual Studio** okruženje.

Integrisani su tako da se, nakon pokretanja komande za testiranje, paralelno ispisuju obaveštenja o izvršavanju akcija u **Output** prozoru i pojavljuju izvršeni testovi u **Test Report** prozoru. **Test Report** prozor je kreiran u paketu i inicijalizuje se svaki put kad se komanda za testiranje pokrene.

Alati za testiranje za svaki testni projekat kreiraju poseban log u vidu **XML** dokumenta. U trenutku kad se test izvrši, rezultat izvršenja se upisuje u log. Zatim ga **Visual Studio** paket za kontinualnu integraciju preuzme iz loga i ispiše u **Test Report** prozoru, prikazanom na slici 3, kao

hijerarhiju testova, prikazanu u vidu stabla. U stablu su testovi predstavljeni kao listovi. Testovi koji su se uspešno izvršili prikazani su zelenom bojom, a oni koji nisu crvenom.



Slika 3. Prozor za prikaz izvršenih testova

Selektovanjem testa ili grupe testova u stablu, u **Details** delu prozora ispisuju se njegovi/njeni detalji.

Nad stablom su, radi lakšeg pregleda i pretraživanja testova, implementirane sledeće operacije:

- **Expand All** - kojom se svi čvorovi proširuju,
- **Collapse All** - kojom se svi čvorovi skupljaju,
- **Show: All, Failed Only, Succeeded Only** - kojom se filtriraju testovi po uspešnosti,
- **View Code** – opcija postoji samo ako je selektovan test, i klikom na nju se otvara fajl gde se test nalazi i navigira do test metode, i
- **Search** – opcija za pretraživanje testova i grupe testova.

2.4. Integriranje alata za sonar analizu u Visual Studio okruženje

Postoji određen skup pravila za stil pisanja koda, kojih se potrebno pridržavati, jer u suprotnom dolazi do greški, obaranja performansi, loše čitljivog koda ili teško održivog koda. **Sonar** analiza pronalazi mesta u kodu gde su ova pravila narušena, i prijavljuje ih kao problem.

Za razliku od drugih alata, gde se u **Output** prozoru opisuje samo izvršavanje akcija, kod **sonar** analize se, pored opisa, navode i linkovi do stranica sa detaljnim izveštajem. Alat za **sonar** analizu, na kraju njenog izvršavanja generiše stranice sa interaktivnim sadržajem, koji opisuje da li su pravila narušena, koja, gde i kako. Putanje do ovih dokumenata se ispisuju u vidu hiperlinka, nakon izvršenja analize, i omogućavaju korisniku da ih otvori unutar okruženja.

Na stranicama je prikazano koliko novih problema se javilo, a koliko ih postoji da su već razrešeni. Takođe su prikazane definicije pravila, i na koliko mesta je svako od njih narušeno. Korisniku je omogućeno da filtrira pogled: da budu prikazana samo nova narušavanja ili sva, ili da budu prikazana samo ona kojim se narušilo određeno pravilo.

3. OPIS DISTRIBUIRANJA, AŽURIRANJA I SERVISIRANJA VISUAL STUDIO PAKETA ZA KONTINUALNU INTEGRACIJU

Visual Studio paketi se mogu distribuirati na dva načina:

- direktnim prosleđivanjem fajla koji nastaje kao rezultat bildovanja paketa, ili
- njegovim postavljanjem na server, tako što se kreira privatna galerija u kojoj se paket objavi ili se objavi javno na **Visual Studio Gallery**.

Direktno prosleđivanje fajla nije zadovoljavajuće rešenje, jer se u tom slučaju ne vode evidencije o trenutnim i dostupnim verzijama paketa, a potrebno da se paket dalje razvija i objavljuju se njegove nove verzije. A pošto je kreirani paket za kontinualnu integraciju privatna, potrebno je distribuirati ga putem privatne galerije **Visual Studio** proširenja [7]. Za hostovanje privatne galerije, potrebno je kreirati **atom feed**. **Atom feed** predstavlja stranicu, u kojoj su definisana dostupna proširenja. Definisani su osnovni parametri proširenja: broj verzije, naziv, id paketa, putanja do paketa, vreme kreiranja, vreme poslednje izmene, itd. Zatim je pomoću **IIS Manager**-a kreiran i podignut sajt, koji predstavlja galeriju i sadrži paket za kontinualnu integraciju i **atom.html** stranicu.

Da bi se paket instalirao na korisničko **Visual Studio** okruženje, prvo je potrebno konfigurisati novu galeriju, a zatim sa nje instalirati željeni paket.

Kada se kreira nova verzija paketa, ažurira se **atom feed** i postavlja novi datum poslednje izmene, kao i broj verzije. Na osnovu parametara koji predstavljaju verziju, datum kreiranja i ažuriranja **atom feed**-a, korisničko **Visual Studio** okruženje prepoznaje da li postoji nova verzija paketa. Ako postoji, **Visual Studio** će ponuditi korisniku da je ažurira. Prilikom svakog ažuriranja neophodno je ponovo pokrenuti okruženje, kako bi se inicijalizovali novi prozori, komande, meniji.

Visual Studio okruženje nudi opciju **Remove** pomoću koje je moguće izbrisati paket, i opciju **Disable/Enable**, pomoću koje se paket može onemogućiti, odnosno omogućiti.

4. ZAKLJUČAK

U ovom radu je predstavljena **VSPackage** tehnologija na konkretnom primeru ugrađivanja alata za kontinualnu integraciju u **Visual Studio** okruženje. U okruženje su integrirani sledeći alati: **night build**, **check warnings**, **sonar** analiza, jedinično testiranje, testiranje komponenti i testiranje performansi. Implementirane su komande za alate, ugrađeni ispisi u **Output** prozor, greške su integrisane u **Error List**, a opcije u **Options** prozor. Pored navedenog, kreiran je i novi prozor za izveštaj o izvršenim testovima. Formatirani su rezultati izvršenja alata i implementirane navigacije. Prosleđivanje parametara je automatizovano, tako da ne može doći do greške prilikom njihovog navođenja.

Pravci daljeg razvoja mogli bi biti:

- kreiranje scenarija u kojima će biti definisan redosled naredbi kontinualne integracije, koje će se puštati jedna za drugom, tako da korisnik ne mora da pokreće jednu po jednu naredbu, nego pusti odjednom grupu naredbi da se izvrše,
- provera pokrivenosti koda testovima,
- realizacija dodatnih opcija koje postoje u alatima i njihovo ugrađivanje u **Options** prozor (kopiranje logova, definisanje koje projekte iz **solution**-a ne treba uključiti u akciju kontinualne integracije...) i
- implementacija **real-time** prepoznavanja narušenih pravila **sonar** analize.

5. LITERATURA

- [1] Paul M. Duvall: *Continuous Integration: Improving Software Quality and Reducing Risks*, 2007, Addison-Wesley
- [2] Keyvan Nayyeri : *Professional Visual Studio Extensibility*, 2008, Wiley Publishing Inc.
- [3] <https://msdn.microsoft.com>, VSPackages
- [4] <http://dotneteers.net>, VS 2010 Package Development
- [5] <https://searchcode.com/codesearch/view/895116>, GUIDs and IDs in Visual Studio
- [6] <http://www.wlwlicious.com/2011/03/29/ervdte-getting-all-projects-html>, ErvDTE : Getting all projects (the SolutionFolder PITA)
- [7] <http://blogs.msdn.com>, Hosting a Private Visual Studio Gallery

Kratka biografija:



Dunja Torbić je rođena 1990. godine u Šapcu. Završila je Šabačku gimnaziju 2009. godine. Fakultet tehničkih nauka u Novom Sadu je upisala 2009. godine. Ispunila je sve obaveze i položila je sve ispite predviđene studijskim programom.

SINHRONIZACIJA PODATAKA IZMEĐU SQL SERVERA I SQL COMPACT SERVERA

DATA SYNCHRONISATION BETWEEN SQL SERVER AND SQL COMPACT SERVER

Nebojša Bobić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Tema ovog rada je proučavanje sinhronizacije podataka između centralne baze podataka i udaljenih uređaja koji svoje podatke smeštaju u lokalne baze podataka. U tu svrhu ispitane su mogućnosti alata **Microsoft Synchronization Framework** i njegove performanse.

Abstract – This article explores data synchronisation between central database and remote units which collect data in local databases. For this purpose this work analyse Microsoft Synchronization Framework and its performance.

Ključne reči: sinhronizacija, baza podataka, udaljeni uređaji, **SQL server**, **SQL Compact server**

1. UVOD

Cilj rada je da demonstrira na praktičnom primeru sinhronizaciju podataka između **SQL** servera i **SQL Compact** servera uz pomoć **Microsoft Synchronization Framework**-a. Pored toga rad se osvrće i na performanse postignute upotrebom **Microsoft Synchronization Framework**-a.

U ovom radu je predstavljen problem sinhronizacije podataka između centralne baze i udaljenih uređaja, prikazana je struktura sistema od interesa, opisane su pojedinačne uloge komponenti sistema i njihova uloga. Nakon toga sledi opis **Microsoft Synchronization Framework**-a, njegove primene kao i prednosti i novine koje uvodi u proces sinhronizacije.

U opisu rešenja nalazi se detaljno objašnjenje koncepta sinhronizacije podataka između **SQL** servera i **SQL Compact** servera kao i konkretni koraci u postupku implementacije rešenja. Pored toga navedeni su rezultati merenja performansi sinhronizacije baza podataka.

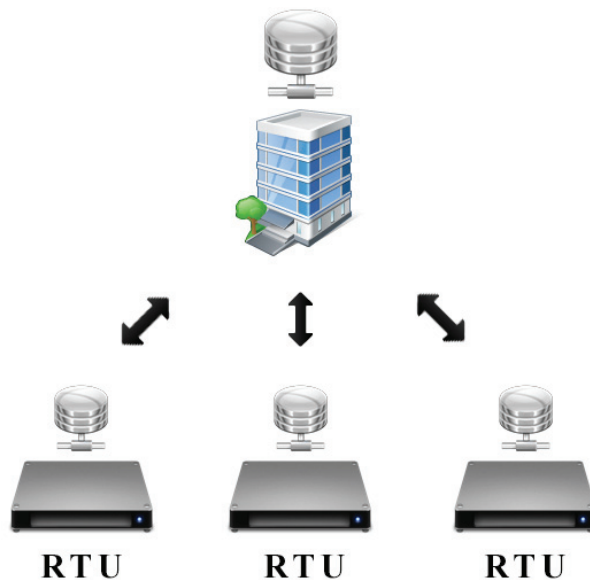
Na samom kraju nalazi se rezime implementiranog rešenja.

2. OPIS REŠAVANOG PROBLEMA

U sistemima koji upravljaju elektroenergetskim sistemima od ključnog je značaja prikupljanje podataka, njihovo čuvanje i analiza. Budući da elektroenergetski sistemi predstavljaju kompleksne sisteme sa kritičnom misijom, obim podataka je veoma velik. Pored toga komponente elektroenergetskih sistema se nalaze na različitim, često veoma udaljenim, geografskim lokacijama, što unosi još jednu dimenziju u kompleksnost sistema.

Budući da je za analizu potrebno da podaci budu centralizovani u jedinstvenu bazu podataka, potrebno je obezbediti adekvatnu sinhronizaciju između udaljenih baza podataka koje čuvaju lokalne podatke i centralne baze podataka.

Ovaj rad se bavi problemom sinhronizacije podataka između **Remote Terminal Unit** (RTU), koji se nalaze na udaljenim lokacijama i mere temperaturu i poziciju regulatora napona, i centralne baze podataka, koja sadrži podatke za sve RTU-ove. RTU-ovi imaju ulogu da u određenim vremenskim intervalima vrše merenja na regulatorima napona i merene podatke upisuju u lokalnu bazu podataka. Ukoliko uzmemo u obzir da jedan elektroenergetski sistem sadrži mnogo RTU-ova i da se merenja vrše u prilično kratkim vremenskim intervalima (uglavnom na svakih 15 minuta), jasno je kakav bi problem nastao ukoliko bi svaki uređaj podatke upisivao direktno u centralnu bazu podataka. Upravo je to razlog da se podaci prvo upisuju u lokalne baze podataka i zatim se vrši sinhronizacija sa centralnom bazom podataka. Pored toga izmene podataka su moguće i u centralnoj bazi podataka, tako da se sinhronizacija odvija u oba smera. Slika 1. grafički predstavlja učesnike opisanog problema.



Slika 1. Komunikacija RTU-ova i centralne baze podataka

3. MICROSOFT SYNCHRONIZATION FRAMEWORK

Microsoft Synchronization Framework predstavlja sveobuhvatnu platformu koja omogućava sinhronizaciju između aplikacija, servisa i uređaja. Programerima

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miroslav Hajduković, red.prof.

omogućuje da izgrade sistem za integraciju bilo koje aplikacije sa bilo kojom vrstom podataka, nezavisno od načina komunikacije. Ključni aspekt **Synchronization Framework**-a predstavlja mogućnost razvoja namenskih provajdera. Provajderi omogućavaju bilo kojem izvoru podataka da učestvuje u procesu sinhronizacije.

Microsoft Synchronization Framework predstavlja elegantno rešenje za razvoj **Occasionally Connected Application** (OCA). OCA predstavlja aplikacije koje se izvršavaju na lokalnim uređajima, a koriste podatke centralne baze podataka. OCA ne pristupa centralnoj bazi podataka direktno, već se podaci, neophodni za funkcionisanje aplikacije, čuvaju lokalno na samom uređaju. U cilju da lokalna baza podataka bude napunjena podacima, OCA mora da implementira određeni vid sinhronizacije. Sinhronizacija podataka se sastoji iz periodičnog uzimanja podataka iz lokalne baze i sinhronizacije sa centralnom bazom podataka. Prednost rešenja baziranog na sinhronizaciji ogleda se u tome što aplikacija nema potrebu za stalnom konekcijom, kako bi pristupala podacima. Bitna prednost ovakvih aplikacije jesto to što aplikacija ne zavisi od postojanja konekcije sa centralnom bazom podataka. Takođe bitno je napomenuti da kod OCA aplikacija centralna baza podataka trpi znatno manje opterećenje, budući da se centralnoj bazi podataka ređe pristupa, nego što je slučaj kod standardnih aplikacija.

3.1. Sinhronizacija baza podataka

Provajder za sinhronizaciju baza podataka koji je sastavni deo **Microsoft Synchronization Framework**-a, podržava sinhronizaciju između **ADO.NET** baza podataka. Bilo koja baza podataka, koja koristi ovaj provajder, takođe može razmenjivati podatke sa drugim izvorima podataka koji su podržani od strane **Microsoft Synchronization Framework**-a, kao što su **web** servisi ili sistemi datoteka.

3.1.1. Praćenje promena

Kako bi sinhronizacija bila efikasna, neophodno je implementirati način za praćenje promena. Praćenje promena predstavlja mogućnost pružanja liste izmena (dodavanje, izmene i brisanje) koje su se dogodile u bazi podataka između dve tačke određene u vremenu. Kao primer može se uzeti udaljena aplikacija, koji se konektuje na centralnu bazu podataka, sa ciljem da pošalje izmene nastale između poslednje sinhronizacije i sadašnjeg trenutka. Ukoliko praćenje promena ne bi postojalo, korisnik bi morao da preuzme sve podatke sa centralne baze i zatim ih spoji sa izmenama nastalim na lokalnoj bazi podataka.

U cilju praćenja promena **Microsoft Sync Framework** koristi **change tracking**. **Change tracking** predstavlja ugrađeni mehanizam **SQL** servera, koji prati promene nad tabelama od interesa.

3.1.2. Održavanje podataka

Tabele koje su neophodne za praćenje podataka obično veoma brzo rastu, što utiče na zauzeće memorijskog prostora kao i na performanse upita koji se izvršavaju nad njima. Sledeći logičan korak jeste određivanje koje praćene promene mogu biti uklonjene. Čišćenje podataka, bazirano na poslednjem vremenu sinhronizacije, sa korisnicima je teško, budući da uređaj možda još nije

sinhronizovan ili je pak korisnik prestao da ga koristi. Ove činjenice otežavaju administratorima da odluče kada svi su svi korisnici preuzeli sve neophodne izmene kako bi mogli da očiste podatke o praćenju promena.

3.1.3. Detektovanje i rešavanje konflikata

Konflikti su još jedan problem vezan za OCA. Konflikt će se dogoditi ukoliko dve ili više baza podataka promene isti podatak, a zaim proces sinhronizacije pokuša da izvrši sinhronizaciju. Kao primer može se uzeti RTU u kojem se promeni podatak vezan za neko merenje. Ukoliko se desi da taj isti podatak bude promenjen u centralnoj bazi podataka prilikom sinhronizacije, dogodiće se konflikt. Postoji više načina da se ovaj problem reši. Jedno od rešenja jeste da se poslednja izmena uzme kao relevantna. Alternativa tome jeste da se uzme izmena koja ima veći prioritet.

Microsoft Synchronization Framework ima mogućnost detektovanja i razrešavanja konflikta mimo **SQL** servera, što pojednostavljuje detektovanje.

3.1.4. Prioriteti prilikom razmene podataka

Prilikom razvoja OCA treba voditi računa i o optimizaciji procesa usled nedovoljno dobrih performansi mrežne konekcije. Jedan od dobrih načina optimizacije jeste postavljanje prioriteta podataka. Upotrebom prioriteta postiže se da podaci koji su kritični ili važni budu, pre predmet sinhronizacije od manje važnih podataka, odnosno podataka sa manjim prioritetom. Ako uzmemo za primer scenario gde korisnici imaju pristup konekciji čije su performanse ograničene, postavljanjem prioriteta omogućavamo da se podaci, koji su važniji, sinhronizuju odmah dok manje važni podaci dolaze na red tek kasnije.

3.1.5. Različite topologije sinhronizacije

Microsoft Synchronization Framework omogućava fleksibilnost u pogledu topologije sistema koji je predmet sinhronizacije. Proces sinhronizacije se može odvijati između lokalnih i centralne baze podataka. Takođe ne postoji prepreka da se sinhronizacija odvija i između lokalnih baza.

3.1.6. Različite topologije sinhronizacije

Microsoft Synchronization Framework povećava sigurnost aplikacije koja zavisi od sinhronizacije. Na strani uređaja, **SQL Compact** server pruža mogućnost enkripcije baze podataka, kao i autentifikacije. Sa stanovišta sinhronizacije, **Microsoft Synchronization Framework** ima mogućnost enkripcije podataka, koji se prenose između baza podataka. Sa strane servera **SQL** server takođe nudi mogućnost autentifikacije prilikom razmene podataka [1].

4. OPIS REŠENJA PROBLEMA

Problem sinhronizacije podataka između udaljenih uređaja i centralne baze podataka sistema za upravljanje elektroenergetskim sistemima predstavlja veliki izazov. U cilju prevazilaženja ovog problema u radu se ispituju mogućnosti **Microsoft Synchronization Framework**-a. Budući da elektroenergetski sistem predstavlja sistem sa kritičnom misijom, ispitivanja nije moguće uraditi na

živom sistemu, već je potrebno implementirati sistem koji simulira ponašanje realnog sistema. Upravo na taj način je u ovom radu implementirano rešenje koje ispituje mogućnosti **Microsoft Synchronization Framework**-a.

Ovaj rad se bavi sinhronizacijom podataka između RTU-ova koji se nalaze na udaljenim lokacijama i mere temperaturu i poziciju regulatora napona i centralne baze podataka koja prikuplja podatke o svim RTU-ovima. Osmišljeni sistem se sastoji iz četiri komponente:

- simulator merenja: sadrži logiku koja na zadati vremenski interval bira slučajan broj RTU-ova za koje će se simulacija izvršiti. Nakon odabira RTU-a, vrši se generisanje podataka i ti podaci se upisuju u odgovarajuće lokalne baze podataka;
- simulator merenja na centralnom serveru: sadrži sličnu logiku kao i simulator merenja, razlika je samo u tome što simulator merenja podatke smešta u centralnu bazu podataka, koja se nalazi na **SQL Server**-u.
- orkestrator: zadužen je za sinhronizaciju porataka **SQL** servera i **SQL Compact** servera. Sinhronizacija se izvršava na zahtev korisnika ili na određeni vremenski interval;
- aplikacija za monitoring: ima zadatak da prikaže dostupne RTU-ove. Pored toga aplikacija prikazuje rezultate simulacije kako za centralni server tako i za svaki RTU u realnom vremenu. Pored prikaza, aplikacija nudi mogućnost startovanja i stopiranja simulacija. Još jedan od zadataka jeste pokretanje sinhronizacije.

Kao sistem za upravljanje bazama podataka centralna baza koristi **SQL** server dok lokalne baze podataka koriste **SQL Compact** server.

4.1. RTU model

RTU predstavlja udaljeni uređaj koji služi za upravljanje određenim aktuatorom ili mernim uređajem. U kontekstu ovog rada RTU predstavlja udaljeni uređaj koji vrši prikupljanje konkretnih merenja sa mernog uređaja i persistuje ih u lokalnu bazu podataka. U radu se prepoznaju dve vrste RTU-ova:

- RTU za očitavanje pozicije regulatora napona.
- RTU za očitavanje temperature regulatora napona.

Iako dva opisana RTU-a imaju različite namene, oni su srodni po prirodi posla koji obavljaju. Iz tog razloga skup zajedničkih osobina za ove dve vrste RTU-a modelovan je kroz apstraktnu klasu koja sadrži podatke o id-u RTU-a, id-u regulatora napona od interesa, konekciji ka lokalnoj bazi podataka i informaciju o samom tipu RTU-a. Pored toga model RTU-u je sadužen za kreiranje lokalne baze podataka i implementaciju upisivanja izmerenih vrednosti u istu.

4.2. Tok simulacije

Simulacija se inicira putem korisničkog interfejsa aplikacije za monitoring. Nakon pokretanja simulacije simulator servera na zadati vremenski interval generiše merenja za slučajno izabran broj RTU-ova. Izabrani RTU-

ovi prihvataju generisana merenja i upisuju ih u lokalne baze podataka. Paralelno, simulator merenja na centralnom serveru upisuje podatke u centralnu bazu podataka. Generisani podaci, koji se upisuju kako u lokalne baze podataka tako i u centralnu bazu, prate se putem aplikacije za monitoring (Slika 2). Simulacija se odvija na ovaj način sve dok se eksplicitno ne zaustavi, putem korisničkog interfejsa aplikacije za monitoring.



Slika 2. Prikaz simuliranih podataka u aplikaciji za monitoring

4.3. Tok sinhronizacije

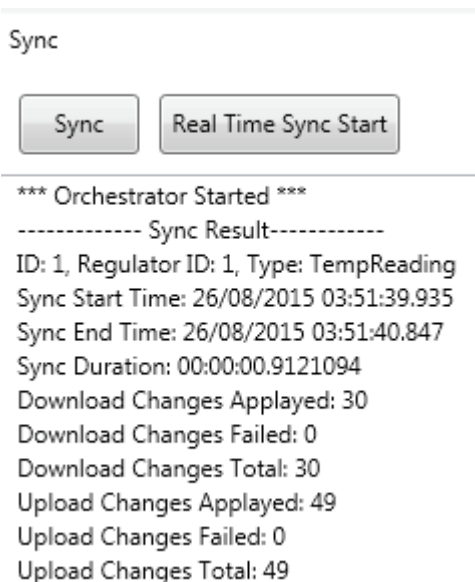
Najvažniji deo implementacije sinhronizacije predstavlja orkestrator. Orkestrator predstavlja servis koji nudi mogućnost prijavljivanja i odjavljivanja RTU-a za sinhronizaciju i naravno samu sinhronizaciju. Pre nego što je uopšte moguće izvršiti sinhronizaciju, potrebno je izvršiti podešavanja nad centralnom bazom, koja se ogledaju u uključivanju praćenja promena za tabele od interesa. Pored toga potrebno je podesiti i filtere, tako da lokalne baze prilikom sinhronizacije sa centralnom bazom podataka, dobiju samo informacije koje ih zanimaju. Na taj način postiže se da lokalna baza poseduje podatke samo za RTU od interesa, dok centralna baza podataka sadrži podatke za sve RTU-ove. Proces sinhronizacije implementiran je putem **Microsoft Synchronization Framework** koji se stara kako o sinhronizaciji, tako i o podešavanju centralne i lokalnih baza podataka, koje je neophodno za sinhronizaciju.

Prvi korak sinhronizacije predstavlja prijavljivanje RTU-ova za sinhronizaciju. Nakon prijave, RTU-a orkestrator treba da izvrši podešavanja nad lokalnom bazom RTU-a, kako bi omogućio proces sinhronizacije. Podešavanja se sastoje od kreiranja odgovarajuće sesije između servera i klijenta, kao i uvođenja novih tabela u lokalnu bazu, koje omogućavaju praćenje promena.

Nakon iniciranja sinhronizacije putem korisničkog interfejsa orkestrator asinhrono inicira sinhronizaciju između svakog prijavljenog RTU-a i centralne baze podataka.

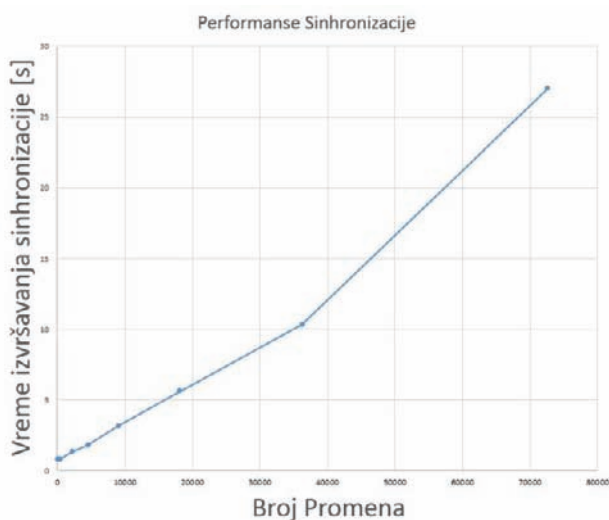
Nakon izvršene sinhronizacije, aplikacija za monitoring prikazuje rezultate sinhronizacije, koji se sastoje od statistike vezane za sinhronizaciju centralne baze podataka sa konkretnim RTU-om. Statistika prikazuje vreme početka sinhronizacije, vreme završetka sinhronizacije, trajanje sinhronizacije i podatke vezane za broj razmenjenih promena.

Primer prikaza rezultata sinhronizacije prikazan je na slici 3.



Slika 3. Prikaz statistike izvršene sinhronizacije u aplikaciji za monitoring

4.4. Performanse sinhronizacije



Slika 4. Grafik zavisnosti vremena izvršavanja sinhronizacije od broja promena

Bitna stavka svakog sistema za sinhronizaciju jeste vreme sinhronizacije. U posmatranom slučaju performanse igraju značajnu ulogu, budući da se promene na samom sistemu veoma učestale i da broj samih RTU-ova može biti velik. Sistem na kojem su merene performanse sastoji

se iz četiri RTU-a i jedne centralne baze podataka. Svaki RTU je beležio između 40-50 promena po sekundi i odprilike isti broj izmena je bio na centralnom serveru.

Slika 4. prikazuje vreme potrebno za sinhronizaciju celokupnog sistema, u zavisnosti od broja promena. Merenja su izvršena tako što se simulacija pušta dok se ne postigne odgovarajući broj promena. Za svaki broj promena izvršena su po tri merenja i kao rezultujuće vreme je uzet prosek ta tri merenja.

Ono što se sa grafika na slici 4. može zaključiti jeste da vreme potrebno za izvršavanje gotovo linearno zavisi od broja promena u sistemu. Ovakav rezultat je očekivan i pogodan za uspostavljanje sinhronizacije u realnom vremenu, budući da se može predvideti koliko će sinhronizacija trajati i koliko ju je često potrebno izvršavati kako bi podaci između uređaja i centralne baze podataka bili sinhronizovani.

5. ZAKLJUČAK

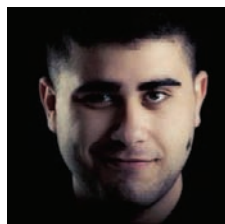
Nakon svega prikazanog može se zaključiti da **Microsoft Synchronization Framework** predstavlja veoma moćan alat za sinhronizaciju podataka. Pri tome je veoma fleksibilan i jednostavan za implementaciju. Njegove performanse su dovoljno dobre da mogu zadovoljiti kompleksan sistem.

Ono što bi bilo interesantno za dalje istraživanje jeste implementacija sistema koji bi, na osnovu analize broja promena i vremena sinhronizacije, mogao da donosi odluku koliko često da se izvršava sinhronizacija u realnom vremenu, a sve sa ciljem postizanja boljih performansi sinhronizacije.

6. LITERATURA

- [1] Microsoft Corporation, *Introduction to Microsoft Sync Framework*, 2006,
<https://msdn.microsoft.com/enus/sync/bb821992.aspx>

Kratka biografija:



Nebojša Bobić rođen je u Slavonskoj Požegi 1991. godine. Diplomski rad iz oblasti Elektrotehnike i računarstva odbranio je 2014. godine na Fakultetu tehničkih nauka. Na Fakultetu tehničkih nauka master rad iz oblasti Elektroenergetski softverski inženjering odbranio je 2015. godine.

ROBOKID**ROBOKID**Ilija Čordašić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu opisana je aplikacija RoboKid, njena funkcionalnost, način i razlog zbog koga je nastala. Takođe se govori i o učenju mladih programiranju, navode se i opisuju već postojeće aplikacije ove vrste, kao i tehnologije koje su korišćene radi stvaranja aplikacije RoboKid, kao što su jezici specifični za domen, TextX, Python i PyCharm.

Abstract – This paper describes the application RoboKid, its function and the way and reason it was made. There is also talk about teaching programming to the young, applications of this kind are listed and described, as well as technologies that were used to make the application RoboKid, such as domain specific languages, TextX, Python and PyCharm.

Ključne reči: JSD, DSL, TextX, Python, Robot, Igra, Edukativno

1. UVOD

U današnje vreme programiranje je među najtraženijim strukama na tržištu, i taj trend ne jenjava. Danas je za veliki broj poslova neophodan računar i makar osnovno znanje rukovanjem istim. Neki čak i predviđaju da će se osim znanja korišćenja računara od radnika očekivati da znaju i da programiraju. U svakom slučaju, programiranje je traženo i popularno do te mere da se danas gleda kako ga učiti od malih nogu [1].

RoboKid je edukativna aplikacija za učenje programera početnika programiranju na zabavan način.

RoboKid za cilj ima da bude edukativan i zabavan. Naime, ovo je program koji kroz igru uči mlade početnike najosnovnije koncepte programiranja. Programski jezik koji se koristi u ovom programu je krajnje jednostavan ali igrajući se sa njim stiču se programerske navike. Naravno, ostavljena je otvorena mogućnost da se ovaj program u budućnosti nadogradi tako da jezik koji se koristi može sadržati kompleksnije elemente, pa se čak može i igrati sa nekim velikim jezicima kao što su C++, Java ili Python.

RoboKid je igra koja zvuči jednostavno: kontroliše putanju robota koji se šeta po tabli. Ali trik je u tome što robota ne kontrolišete putem džojstika ili strelica na tastaturi, nego pomoću koda. Naime, morate koristiti specifični programski jezik i isprogramirati njegovu putanju, pritom morate izbegavati mine, zaobilaziti ili pomerati prepreke, i u slučaju da je izlaz zaključan, otključati ga.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Igor Dejanović, docent.

Da bi robota doveli do cilja morate ne samo smisliti kojom putanjom da mu naredite da se kreće i sa kojim elementima da stupi u kontakt, već se morate postarati da su ta naređenja sintaksno i semantički ispravna.

Ali to nije sve, kontrolisanje robota je samo jedan deo programa. Drugi deo se odnosi na izgradnju nivoa koji robot mora da pređe, odnosno postavljanje elemenata na tablu. To se takođe radi sa specifičnim programskim jezikom. Taj zadatak može biti učitelj, ali može i učenik takođe da sam pravi svoje nivoe igre, pa da ga rešava ili da nekom drugom da ga reši.

1.1. Inspiracija

Ovaj program je i počeo kao nadogradnja već postojećeg, krajnje jednostavnog primera „Robot“ koji demonstrira neke funkcionalnosti TextX-a. Za stvaranje RoboKid-a, u obliku u kome je sada, sam kao inspiraciju imao online igru „Code Combat“ koja se može naći na codecombat.com [2].

Još jedan interesantan projekat za učenje programiranju je Scratch koji je projekat Lifelong Kindergarten Group MIT-ove Media Lab. Distribuirao se besplatno na internetu [3].

1.2. Cilj rada

Lično mislim da je problem sa programima kao što je Scratch to da korisnik ne piše kod sam nego je kod već napisan i nalazi se u šarenim blokovima koje korisnik mora samo da poređa. To je možda dobro rešenje za veoma mlade korisnike, ali ako želite da naučite da programirate, morate sami pisati kod.

Zbog toga mi se više sviđa Code Combat, jer u njemu se korisnik susreće sa činjenicom da on sam mora da se postara da kod bude pravilno napisan, tj. u Code Combat-u korisnik se susreće sa konceptom sintaksnih grešaka.

Moj cilj je bio da napravim program koji će biti prelaz između Scratch-a i Code Combat-a, zbog toga je kod programskog jezika koji se koristi u RoboKid-u tako jednostavan, ali je ipak kod koji treba da se piše a ne slaže.

Takođe, cilj programa je bio i da se pokaže šta se sa TextX-om može napraviti. Programski jezici RoboKid-a su pravljeni uspomoc TextX-a, i otvorena je mogućnost da se u budućim rešenjima mogu koristiti programski jezici koji mogu biti napravljeni da budu kompleksniji ili jednostavniji.

1.3. Postojeća rešenja

Code Monkey je interaktivna online igra koju mogu koristiti deca od devet godina pa nadalje i koja se može naći na www.playcodemonkey.com. Njegovi stvaraoci stalno prave nove nivoe i probleme i dodaju je igri [4].

Na prvi pogled ova igra je sličnija Code Combat-u. Umesto junaka epske fantastike koji se bore u tamnicima, ovde se kontroliše majmun koji skuplja banane. Ali razlika je u tome da se koristi jezik specifičan za domen koji je jednostavniji od jezika opšte namene koji se koriste u Code Combat-u.

Guido van Robot, ili GvR, je programski jezik i interaktivna aplikacija čiji je cilj da se početnici upoznaju sa fundamentalnim osnovama programiranja. Radi na Windows-u, Macintosh-u i GNU/Linux-u [5].

Ovaj program je veoma sličan RoboKid-u, jer se radi o igri gde se kontroliše robot koji se kreće po tabli uz pomoć jezika specifičnog za domen. Postoje i prepreke u vidu zidova i interaktivni elementi kao što su beeper-i koje robot može da podigne, nosi sa sobom i spusti. Ali za razliku od RoboKid-a, ovde je jezik malo kompleksniji jer se od korisnika traži da definiše klase i koristi petlje.

Programski jezik Guido van Robot je osmislio Jeffrey Elkner i nastao iz dva jezika-roditelja, Karel the Robot, koji je osmislio Richard Pattis, i Python-a [6].

2. KORIŠĆENE TEHNOLOGIJE

2.1. Jezici specifični za domen

Jezik specifičan za domen - JSD (Domain specific language - DSL) je programski jezik koji daje notaciju skrojenu aplikacionom domenu i bazira se na relevantnim konceptima i osobinama tog domena, to jest, jezik koji je prilagođen na određeni domen problema [7].

Dobro dizajniran JSD je onaj koji je baziran na dobrom razumevanju dobro definisanog domena za koji je namenjen i pokriva ga na odgovarajući način [7] [8]. Možemo dakle zaključiti da su JSD-ovi, kao programski jezici, veoma ograničeni jer je opseg JSD-a ograničen njegovim domenom [9].

Po načinu izrade, postoje dve vrste JSD-a: eksterni i interni. Kao i svaki softverski jezik i JSD se sastoji od, apstraktne sintakse, jedne ili više konkretnih sintaksi i semantike [8].

2.2. Python

Python je interpretiran programski jezik visokog nivoa sa dinamičkom semantikom koji je veoma popularan za brz razvoj programa, kao i jezik za skriptovanje i povezivanje [10]. Stvorio ga je Guido van Rossum 1990 godine a danas je jedan od najpopularnijih jezika [11]. Stavljjen je akcenat na brzinu i efikasnost programiranja i čitkost koda, jednostavan je da se koristi i da se nauči [10]. Postoji sveobuhvatna i veoma razvijena standardna biblioteka koja se može koristiti na svim platformama [10]. Python podržava modele i pakete tako da ohrabruje modularnost i ponovnu upotrebu koda [10].

2.3. TextX

TextX je metajezik koji služi za specifikaciju jezika specifičnih za domen (Domain specific languages - DSL). Inspirisan je Xtext-om, ali za razliku od Xtext-a koji radi za Java platformu, TextX radi za Python platformu [12].

TextX će od jednostavnog opisa jezika napraviti meta-model u formi Python klase. Takođe će napraviti parser koji će parsirati iskaze na vašem jeziku i automatski napraviti graf Python objekata (tj. model) u skladu sa meta-modelom [12].

2.3. PyCharm

PyCharm je Integrirani Razvojni Alat (Integrated Development Environment - IDE) koji se koristi za programiranje u programskom jeziku Python. Proizvodi ga češka kompanija JetBrains. Može se koristiti na Windows, Mac OS i Linux operativnim sistemima. To je open source program koji se može skinuti sa GitHub-a. Pošto je open source može se nadograđivati i poboljšavati od strane zajednice [13].

PyCharm ima sve što bi očekivali od IDE-a, prozore i toolbarove neophodne za pisanje koda i kontrolu i navigaciju nad projektom, mehanizme za detekciju sintaksnih grešaka (kao i ponudu mogućih rešenja), mehanizme za menjanje (refactoring) elemenata za ceo projekat, Python debugger, mogućnos građenja (build) i pokretanja (run) programa, kao i podršku rada na web frameworku kao što su Django, web2py i Flask [13] [14] [15].

3. APLIKACIJA ROBOKID

Aplikacija RoboKid je Python aplikacija čiji je osnovni deo Python fajl base.py koji se sastoji iz nekoliko klasa od kojih svaka izvršava neku funkciju aplikacije ili elementa igre. Takođe u sklopu aplikacije su i spoljni resursi, slike i fajlovi u kojima se nalaze definicije jezika specifičnih za domen rađeni po TextX platformi i ostali pomoćni fajlovi.

3.1. JSD i program za nivo

Opis JSD-a za nivo se nalazi u fajlu level.tx. Prema opisu, programi za nivo počinju sa ključnom rečju 'begin' i završava se sa ključnom rečju 'end', a sve što je između je uvučeno. U telu programa za nivo se definišu elementi koji će se nalaziti na tabli. Svaki od njih je definisan uz pomoć ključne reči koja je naziv elementa i integera koji su koordinate.

Ako pokušate da napišete program bez 'begin' i 'end' ili ne uvučete kod između – program će podići grešku. Ako pokušate da definišete element koji ne postoji – program će podići grešku. Ako pokušate da zadate koordinate koje nisu integer ili ne stavite zarez između brojeva ili pokušate uneti brojeve koji su manji od nula (0) ili veći od devet (9) – program će podići grešku. Ako pokušate da stavite dva elementa na isto mesto, odnosno date im iste koordinate – program će podići grešku.

3.2. JSD i program za robota

Opis JSD-a za robota se nalazi u fajlu robot.tx. Prema opisu, programi za robota počinju sa ključnom rečju 'begin' i završava se sa ključnom rečju 'end', a sve što je između je uvučeno. U telu programa za robota se definiše kretanje robota po tabli i njegovo delovanje na elemente na koje naiđe. Svaki od koraka je definisan sa jednom od četiri moguće ključne reči koje definišu pravac (up, down, left, right) i sa vrednošću integer koji definiše za koliko polja će se pomeriti. Bitno je znati da se integer može izostaviti, u slučaju da ga ne navedete podrazumeva se vrednost jedan (1). Takođe morate imati na umu da ne možete reći robotu da odšeta van tabele, ako to uradite – program će podići grešku. Takođe treba obratiti pažnju da robot možda neće izvršiti ovu putanju, to sve zavisi od elemenata koji mu se nađu na putu.

Komanda za delovanje se zove „interact“ i navodi se posle koraka koji će robota dovesti do polja na kome je element.

Ako pokušate da napišete program bez 'begin' i 'end' ili ne uvučete kod između – program će podići grešku. Ako pokušate da definišete element koji ne postoji – program će podići grešku. Ako pokušate da zadate koordinate koje nisu integer ili ne stavite zarez između brojeva ili pokušate uneti brojeve koji su manji od nula (0) ili veći od devet (9) – program će podići grešku. Ako pokušate da stavite dva elementa na isto mesto, odnosno date im iste koordinate – program će podići grešku.

3.3. Elementi

Postoji šest elemenata koji se pojavljuju na tabli, a to su:

Robot Kid – je osnovni element programa. Njegova funkcija je da se kreće po tabli i deluje na druge elemente. Robot Kid mora biti ubačen u svaki nivo, ako se napravi nivo bez robota program će vas obavestiti kada ga učitate. Moguće je imati jedan i samo jedan Robot Kid po nivou.

Pomerajuća cigla – je jedan od prepreka koje mogu da se stave u program. Robot Kid ne može proći kroz Pomerajuću ciglu, ali zato može da je pomeri za jedno polje tako što dođe na polje na kome se nalazi i izvrši komandu interact koja će tada pometiti ciglu za jedno polje u pravcu suprotnom od onoga is kojeg je Robot Kid došao. Ako se ne izvrši komanda interact kada Robot Kid dođe do polja na kome je cigla, pomeranje će biti neuspešno i Robot Kid će se zaustaviti na polju tik istic pred cigle. Može se staviti nijedna, jedna ili više Pomerajućih cigli u svaki nivo.

Nepomerajuća cigla – je još jedna prepreka kao i Pomerajuća cigla kroz koju Robot Kid ne može da prođe, ali za razliku od Pomerajuće cigle, Nepomerajuću ciglu Robot Kid ne može pomeriti. Može se staviti nijedna, jedna ili više Nepomerajućih cigli u svaki nivo.

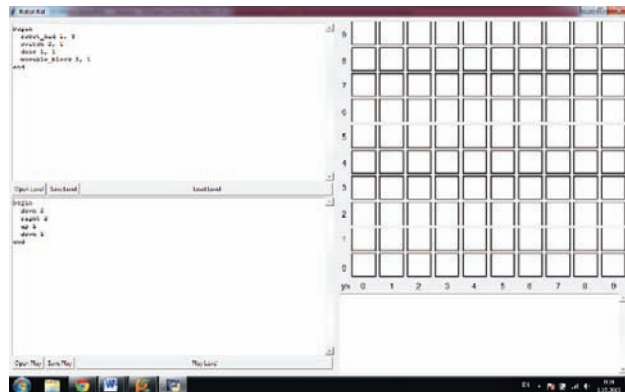
Prekidač – je element nad kojim Robot Kid može da deluje, tačnije on ga okida. Kada Robot Kid dođe na polje na kome je prekidač on sa komandom interact može da ka okine. Prekidač tada promeni boju iz crvene u zelenu, što je znak da je okinut. Kada je Prekidač okinut Vrata, koja su dotad bila zaključana se otključavaju. Za razliku od Pomerajuće cigle i Nepomerajuće cigle Robot Kid može proći kroz polje na kome je Prekidač pa čak i staj na njega a da pritom ne mora delovati na njega. Može biti nijedan ili samo jedan Prekidač u nivou.

Vrata – su element koji je krajnji cilj nivoa, to jest cilj Svakog nivoa je da Robot Kid prođe kroz vrata. Ako postoji Prekidač u nivou vrata će biti zaključana i biće crvene boje, ali ako nema Prekidača u nivou ili ako je Prekidač okinut Vrata će biti otključana i biće zelene boje. Da bi Robot Kid prošao kroz vrata on mora da stane na polje na kome se vrata nalaze, ako su Vrata otključana, Robot Kid će ih otvoriti i program će vas obavestiti da ste prošli kroz Vrata. Nivo se tada završava i sve komande koje su zadate a idu posle koraka u kome je Robot Kid prošao kroz vrata se neće izvršiti. Mogu biti nijedna ili samo jedna vrata u nivou. Iako su Vrata cilj svakog nivoa, može se napraviti nivo i bez njih da bi se napravio, na primer, nivo na kome se vežba delovanje nad samo jednim elementom ili kretanje Robot Kid-a.

Mina – je još jedna prepreka, ali za razliku od Pomerajuće cigle i Nepomerajuće cigle, Robot Kid može preći preko nje. Ali takođe za razliku od Prekidača i Vrata, kada jednom Robot Kid pređe preko nje on će eksplodirati i nivo će se završiti.

3.4. GUI

Grafički interfejs ove aplikacije je jedan prozor veličine ekrana.



Slika 1. GUI aplikacije RoboKid

Kao što vidite na slici, on se sastoji od tri tekstualna polja, šest dugmadi, i tabele sa 10x10 polja.

Prvo tekstualno polje (ono gore levo) je mesto gde se piše kod za nivo, tj. za stvaranje nivoa.

Kod takođe može i da se učita uspomoć dugmeta „Open Level“ koje otvara dijalog za otvaranje fajla iz foldera „levels“. Kada se fajl otvori njegov sadržaj će biti ispisan u tekstualnom polju.

Takođe, kod može i da se sačuva uspomoć dugmeta „Save Level“ koji otvara dijalog za čuvanje fajlova.

Dugme „Load Level“ učitava nivo u tabelu, odnosno iscrta elemente iz koda u tabelu grafički prikazujući lokaciju svakog elementa. Ovo dugme se takođe ponaša i kao reset dugme. O tome kako sve ovo funkcioniše biće reči kasnije.

Drugo tekstualno polje (dole levo na slici) je mesto gde se piše kod za robota, tj. komanda za robota.

Funkcije dugmadi „Open Play“ i „Save Play“ su iste kao i „Open Level“ i „Save Level“, samo što je u pitanju učitavanje i čuvanje koda za robota u donje tekstualno polje.

Pritiskom dugmeta „Play Level“ se izvršava kod robota, i na tabeli ćete tada videti kako robot korak po korak izvršava zadatke koje ste mu zadali. Imajte na umu da ako pritisnete dugme „Play Level“ pre nego što ste pritisnuli dugme „Load Level“ ništa se neće desiti, jer koja je svrha prikazati rešenje nivoa ako sam nivo nije prikazan.

Ako niste zadovoljni rešenjem možete resetovati nivo ponovnim pritiskom dugmeta „Load Level“, prepraviti kod za robota i pokrenuti ga ponovo sa „Play Level“.

I na kraju, poslednje tekstualno polje (dole desno na slici) ispisuje informacije o tome šta se dešava. Takođe, ovde se prikazuju informacije o greškama.

3.5. Funkcionalnost

Program se sastoji iz više komponenti. Glavni program se nalazi u Python fajlu `base.py`. Ostale komponente se nalaze u folderima.

U folderima „levels“ i „robots“ se čuvaju nivoi i rešenja nivoa, tj. fajlovi sa specifičnim kodom za stvaranje nivoa za rešavanje i fajlovi sa kodom za robota koji su rešenja. Ovi fajlovi se, u suštini, mogu čuvati bilo gde, ali je preporučljivo, radi lakšeg snalaženja, da se čuvaju u ovim folderima. Program će zato automatski otvoriti te foldere kada korisnik reši da sačuva neki kod.

U folderima „levels graphics“ i „robots graphics“ se čuvaju .dot grafički fajlovi kodova iz „levels“ i „robots“ respektivno. Program automatski pravi grafičke fajlove i smešta ih u te foldere kada korisnik sačuva kod.

U folderu „app files“ se nalaze fajlovi neophodni za pravilno funkcionisanje programa.

Fajl `base.py` se sastoji iz:

- klase `Application` koja generiše i osvežava GUI,
- klase `Board` u kojoj se određuje izgled tabele za svaki korak,
- klase `Level` koja obrađuje program za nivo,
- klase `Robot` koji obrađuje program za robota,
- klase `Program` koja vrši iteracije za svaki korak robota i proverava da li je robot naleteo na neki element,
- klase `Obstacle` koja kalkuliše da li je robot naleteo na prepreku, i
- klasa `UnmovableBlock`, `MovableBlock`, `Switch`, `Door` i `Mine`, koje kalkulišu robotovu interakciju sa elementima respektabilno.

4. ZAKLJUČAK

Napomenuli smo koliko je važno učenje mladih programiranju, tako da su ovakvi edukativni programi korisni. Važno je da se prave programi koje mogu koristiti svi, od dece do odraslih i od početnika do onih koji već imaju nekog iskustva. Tu imaju prednost programi koji koriste jezike specifične za domen jer se kompleksnost jezika može prilagođavati nivou stručnosti učenika. Jednostavni JSD-ovi se mogu koristiti za učenje dece i početnika dok se kompleksniji JSD-ovi mogu koristiti za starije i one sa većim iskustvom. Aplikacija `RoboKid` je napravljena tako da se, sa malim izmenama, mogu koristiti tekstualni JSD-ovi bilo koje kompleksnosti, čak i jezici opšte namene.

5. LITERATURA

- [1] 7 Apps for Teaching Children Coding Skills <http://www.edutopia.org/blog/7-apps-teaching-children-coding-anna-adam>, Accessed 2015 Oct 22.
- [2] Why CodeCombat? <https://codecombat.com/about>, Accessed 2015 Oct 22.
- [3] About Scratch <https://scratch.mit.edu/about>, Accessed 2015 Oct 22.
- [4] Code Monkey <https://www.playcodemonkey.com/about>, Accessed 2015 Oct 22.
- [5] Guido van Robot <http://gvr.sourceforge.net>, Accessed 2015 Oct 22.
- [6] Programming with GvR <http://gvr.sourceforge.net/lessons/rfrank/>, Accessed 2015 Oct 22.
- [7] Van Deursen, Arie, and Paul Klint. "Domain-specific language design requires feature descriptions." *CIT. Journal of computing and information technology* 10.1 (2002): 1-17.
- [8] Igor Dejanović, Prilog metodama brzog razvoja softvera na bazi proširivih jezičkih specifikacija, University of Novi Sad, Faculty of Technical Science, Novi Sad, 2011
- [9] Martin Fowler, Rebecca Parsons. "Domain-specific languages" Addison-Wesley Professional 9.24 (2010): 1-640
- [10] What is Python? Executive Summary <https://www.python.org/doc/essays/blurb>, Accessed 2015 Oct 23.
- [11] Uvod u programiranje i programski jezik Python <http://www.slideshare.net/amarkalaba/uvod-u-programiranje-i-programski-jezik-python>, Accessed 2015 Oct 16.
- [12] TextX, <http://igordejanovic.net/textX/>, Accessed 2015 Oct 17.
- [13] JetBrains Strikes Python Developers with PyCharm 1.0 IDE <http://www.eweek.com/c/a/Application-Development/JetBrains-Strikes-Python-Developers-with-PyCharm-10-IDE-304127>, Accessed 2015 Oct 22.
- [14] Creating Web2Py Project <https://www.jetbrains.com/pycharm/help/creating-web2py-project.html>, Accessed 2015 Oct 22.
- [15] Creating Flask Project <https://www.jetbrains.com/pycharm/help/creating-flask-project.html>, Accessed 2015 Oct 22.

Kratka biografija:



Ilija Čordašić rođen je u Zrenjaninu 12. Avgusta 1990. god. Osnovne akademske studije je završio u na tehničkom fakultetu »Mihailo Pupin« u Zrenjaninu 2013. god. na smeru Softversko inženjerstvo. Master studije je upisao na Fakultetu tehničkih nauka u Novom Sadu na smer Elektrotehnika i računarstvo 2013. god. Master rad je odbranio 2015. god.

**IZBOR FASADNE OBLOGE STAMBENO-POSLOVNOG OBJEKTA U NOVOM SADU
NA BAZI ANALIZE TROŠKOVA****THE SELECTION OF FACADE CLADDING FOR RESIDENTIAL AND COMERCIAL
BUILDING IN NOVI SAD BASED ON COSTS ANALYSIS**

Olinka Franić, Jasmina Dražić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – GRAĐEVINARSTVO

Kratak sadržaj – U radu je prikazan postupak izbora optimalnog varijantnog rešenja fasadne obloge za ventilirajuće fasade na osnovu ekonomskih kriterijuma i vremena ugradnje. Optimizacija se bazira na inicijalnim i eksploatacionim troškovima (LCC analiza).

Abstract – This paper present a method for selecting optimal type of facade cladding for ventilated facade based on economic criteria and installation time. The optimization is based on initial and exploitation cost (LCC Analysis).

Ključne reči: ventilirajuće fasade, fasadne obloge, troškovi, LCC analiza, vreme ugradnje

1. UVOD

Fasada je jedan od najvećih i najvažnijih elemenata objekta u estetskom i tehničkom smislu. Osnovna uloga fasade je da zaštiti unutrašnjost objekta, kao i samu konstrukciju od atmosferskih uticaja, prenese spoljašnje opterećenje na noseću konstrukciju i da objektu da vizuelni identitet. Iako je fasada prvo što se primeti na objektu, ona ne treba da zadovolji samo estetsku funkciju. Pored estetske funkcije koju ima, dobro projektovana fasada mora objektu obezbediti osvetljenje, ventilaciju, komfor za korisnike, mora biti pogodna za održavanje, laka za eventualne popravke oštećenja.

Do devedesetih godina XX veka, u razvijenim zemljama se do 60 % budžeta predviđenog za građevinarstvo trošilo na održavanje i popravke, a odabir rešenja vršio se uglavnom na osnovu potrebnih inicijalnih ulaganja koja su dobijana iz predmera i predračuna radova. Razvojem adekvatnih metoda moguće je u analizu uključiti "buduće troškove" tj. troškove životnog ciklusa koji nastaju u toku eksploatacije objekta.

Izbor fasadne obloge za ventilirajuću fasadu u ovom radu, urađen je za stambeno poslovnu zgradu u Novom Sadu, sa ciljem da se za predložena varijantna rešenja, na bazi analize troškova i vremena potrebnog za ugradnju, odabere optimalno rešenje.

2. VENTILIRAJUĆE FASADE

Ventilirajuće fasade spadaju u tip savremenijih fasada. Dominantan su način oblaganja poslovnih objekata, a u poslednje vreme se sve češće koriste i na rezidencijalnim objektima.

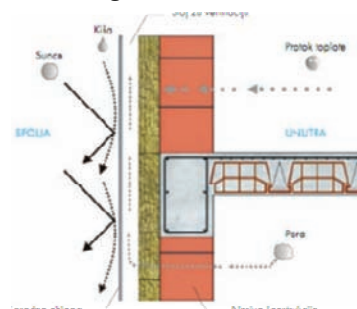
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Jasmina Dražić, red.prof.

Osnovna razlika u odnosu na druge tipove fasada je u postojanju sloja vazduha za ventilaciju koji razdvaja spoljašnju oblogu od termoizolacije.

Elementi ventilirajućih fasada su (slika 1):

- noseća podkonstrukcija, vertikalni i/ili horizontalni aluminijumski profili,
- termoizolacija,
- ventilirajući sloj vazduha i
- završna obloga.



Slika 1. Osnovni elementi ventilirajućih fasada

2.1. Aluminijumska podkonstrukcija

Uloga podkonstrukcije je prihvatanje težine svih elemenata fasade, kao i prenošenje svih statičkih i dinamičkih uticaja od spoljašnjih dejstava na fasadu, a posebno uticaje vetra. Ona mora omogućiti izravnavanje devijacija i neravnina na postojećim zidovima. Zavisno od vrste i kačenja završne obloge formira se podkonstrukcija koja se najčešće sastoji od aluminijumskih kotvi, T profila, L profila, aluminijumskih kutija. Razlikuju se dve osnovne grupe podkonstrukcije prema obliku fasadnih ploča, a to su panelni i kasetni sistemi.

2.2. Termoizolacioni sloj

Kao termoizolacija kod ventilirajućih fasada se koristi polutvrdo presovana mineralna vuna. Mineralna vuna se deli na kamenu i staklenu mineralnu vunu. Kamena mineralna vuna se proizvodi od kamena vulkanskog porekla, dolomita, dijabaza i bazalta, sa dodatkom krečnjaka, krečnjaka sa primesama dolomita ili gline da bi se pospešio proces topljenja. Treća komponenta je šljaka. Staklena mineralna vuna se pravi od staklene šljake i peska procesom koji je sličan procesu pravljenja kamene mineralne vune. Debljina ploča termoizolacije se dimenzioniše prema rezultatima termičkog proračuna zavisno od klimatskih uslova i zahteva investitora, kao i obaveze poštovanja energetske efikasnosti objekta. Termoizolacija se može postavljati i u dva sloja pri čemu se mora voditi računa da ploče drugog sloja moraju pokriti fuge prvog sloja. Dozvoljeni razmak između ploča je 2mm, a ukoliko

je veći mora se ispuniti istim materijalom. Ploče za termoizolaciju se postavljaju tesno jedna uz drugu, bez zazora između njih i moraju se mehanički pričvrstiti, da bi se postigao efekat pritiskanja ploča na zid (slika 2).



Slika 2. Postavljanje termoizolacije

2.3. Sloj vazduha za ventilaciju

Izolacione komponente (ploče za termoizolaciju, protiv-požarna izolacija) su strukturalno razdvojene slojem vazduha od spoljašnje obloge, koja sprečava prodor atmosferskih padavina u objekat. Vodena para iz samih elemenata konstrukcije, kao i ona nastala upotrebom prostorija u objektu se izvodi iz objekta u ovaj sloj, i tu se zahvaljujući strujanju vazduha suši. Vlažni spoljašnji zidovi se suše u najkraćem mogućem roku. Time se postiže priyatna i zdrava mikroklima u prostorijama. Ovo je posebno naglašeno u zimskom periodu kada se vodena para kreće od toplijih ka hladnijim slojevima.

U letnjem periodu, ovaj sloj vazduha štiti objekat od prebrzog i preteranog zagrevanja. Kako se pod dejstvom sunčeve toplote zagreva spoljašnja obloga tako se zagreva i vazduh u ventilacionom kanalu. Ovako zagrejan vazduh se penje na gore stvarajući 'efekat dimnjaka' i izlazi kroz ventilacione otvore van konstrukcije koji se nalaze uglavnom ispod prozora. Da bi se ovaj efekat povećao, spojnice između ploča spoljašnje obloge se mogu zaptivati raznim sredstvima (gumom, specijalnim smolama...) i tako se povećava 'cug' u ovom sloju. Ulazak svežeg vazduha je po pravilu omogućen kroz specijalne ventilacione rešetke koje se nalaze u donjem delu fasade i iznad otvora objekta (prozori). I u donjim i u gornjim ventilacionim otvorima mogu se ugrađivati ventilatori, kojima se može u potpunosti kontrolisati razmena vazduha u ventilacionom kanalu. Da bi se sprečio ulazak insekata svi otvori veći od 10mm moraju se zatvoriti zaštitnom rešetkom.

2.4. Završna obloga

Za završnu oblogu ventilirajućih fasada na tržištu postoji širok spektar raznovrsnih materijala. Zahvaljujući tome, arhitektonsko oblikovanje i dizajniranje fasade dolazi do punog izražaja. Princip funkcionisanja ventilirajuće fasade se ne menja bez obzira na to da li je završna obloga od lakih ili teških materijala. Uloga završne obloge je da štiti objekat od atmosferskih uticaja i da uticaje vetra preko podkonstrukcije prenese na noseću konstrukciju. Na aluminijumsku podkonstrukciju obloga se može montirati na više načina, zavisno od tipa obloge i toga da li je kačenje vidljivo ili nije sa lica fasade. Dekorativna završna obloga može biti od: stakla, mermera, granita, granitne amike, alubonda, vlaknasto-cementnih ploča,

drveta, laminama, HPL kompakt ploča, keramike, armirano betonskih ploča, bakra, titan-cinka.

3. VARIJANTNA REŠENJA

Izbor optimalnog tipa fasadne obloge urađen je za stambeno-poslovni objekat u Novom Sadu, spratnosti Su+Pr+4+Pot., projektovanom prema važećim propisima za taj tip objekta. Konstrukcijski sistem je armiranobetonski skelet koga čine grede i stubovi ukrućeni AB platnima. Objekat je fundiran na temeljnoj ploči. Unutrašnji zidovi, zidani blokom i šupljom opekom u produžnom malteru, su debljine 25, 20 i 12 cm.

Strukturu ventilirajuće fasade čine:

- aluminijumska podkonstrukcija tipa Rivet. Sistem čine osnovni noseći vertikalni T profili dimenzija 100x60x3 mm, raspoređeni modularno na rastojanju od 600 mm. Spoj vertikale sa konstrukcijom objekta ostvaren je preko aluminijumskog ankeri podesivog po svim osama. Ankeri su konstruisani tako da se mogu koristiti pri rastojanjima između zida i spoljne ravni obloge od 60 do 170mm.
- termoizolacija od staklene mineralne vune, jednostruko kaširana na staklenom voalu, debljine d=12,0 cm.

U radu su razmatrana tri varijantna rešenja fasadne obloge za predloženu ventilirajuću fasadu:

- **Varijantno rešenje I** – je fasadna obloga od granitnih pločica italijanskog proizvođača Laminam, iz kolekcije Collection u kombinaciji boja Fumo i Perla, debljine 3mm.

Granitne pločice, porcelanski laminat, dobijen je mokrim brušenjem sirovih glinovitih materijala (granita i metamorfizisa) - feldspata i keramičkih pigmenata. Ovaj materijal se sabija i sinteruje na 1200°C. Materijal je lagan (7 kg/m²), savršeno ravan, otporan na hemijske uticaje, mraz, visoke temperature, bakterije, buđ i UV zračenje. Na slici 3, je prikazan primer objekta sa ventilirajućom fasadom obloženom granitnim pločicama.



Slika 3. Fasadna obloga - granitna keramika

- **Varijantno rešenje II** – je fasadna obloga od Alubonda debljine 4 mm, gustine 1,37 g/cm³, težine 5,48 kg/m², termoprovodljivosti 0,15-0,19 W/mK, zvučne izolacije 26 dB. Odabrana boja je Metallic Silver i Metallic Red.

Alubond je aluminijumski kompozitni panel kojeg čine dva aluminijumska lima debljine 0,5 mm sa polietilenskim jezgrom, integrisanim u sendvič panel procesom kontinuirane koekstruzije. Alubond paneli imaju termičku dilataciju od 2,7mm/m pri temperaturnoj razlici od 100°C,

a samo 1 mm/m za temperaturnu razliku 50°C, značajno redukuje nivo buke (24-27 dB), male je težine (4,55 kg/m² do 7,34 kg/m²), 1,6 puta je lakši od aluminijuma i 3,4 puta lakši od čelika. Primer objekta sa ventilirajućom fasadom obloženom Alubondom, prikazan je na slici 4.



Slika 4. Fasadna obloga - Alubond

- **Varijantno rešenje III** – je fasadna obloga od vlaknasto-cementnih ploča Eternit tipa Natura debljine 8mm, bojena u masi sa završnom obradom akrilnim bezbojnim premazom koji je otporan na atmosferske uticaje. Površina ploča je glatka, matirana sa ravnomerno vidljivom strukturom vlaknastog cementa. Fabričke dimenzije ploča su 2500 x 1250 x 8 mm.

Vlaknasti cement je kompozitni materijal koji se pravi od cementa, peska, celuloze i vode. Otporan je na atmosferske uticaje, mehaničke udarce i abrazivna dejstva, mikroorganizme i bakterije, česte promene i ekstremne temperature (do -60°C), kiseline i soli, postojan na hemijske agense i atmosferska zagađenja, vatrootporan (A2). Na slici 5 je prikazan primer objekta sa ventilirajućom fasadom obloženom vlakno-cementnim pločama.



Slika 5. Fasadna obloga - vlakno-cementne ploče

4. ANALIZA TROŠKOVA I VREMENA UGRADNJE ZA FASADU

Analiza za ceo objekat je obuhvatila analizu troškova nabavke i ugradnje, analizu troškova održavanja i demontaže, analizu troškova životnog ciklusa i analizu vremena potrebnog za izvođenje fasade.

Pod troškovima nabavke fasadne obloge se podrazumevaju troškovi proizvodnje i transporta do gradilišta, i oni su preuzeti od proizvođača. Troškovi ugradnje se izračunavaju iz građevinskih normativa za odgovarajuće građevinske radove, gde se pored troškova materijala obračunavaju i troškovi radne snage [1].

Na ovaj način se dobijaju informacije o troškovima koje se tiču same izgradnje objekta, dok o “budućim troškovima” nemamo nikakve informacije. Budući troškovi tj. troškovi životnog ciklusa su svi oni troškovi koji nastaju u toku eksploatacije objekta; troškovi održavanje, troškovi demontaže i troškovi odlaganja otpadnog materijala nakon životnog ciklusa [2].

Održavanje fasadnih obloga za ventilirajuće fasade se obavlja po unapred određenoj dinamici u cilju produžavanja životnog veka obloga. Zavisno od tipa obloge, čišćenje se obavlja vodom ili vodom sa dodatkom sredstva za čišćenje.

U analizu ekonomskih kriterijuma, korišćena je i LCC analiza, koja je proširila aspekt posmatranja troškova na period eksploatacije, tj. troškove nakon završetka objekta. LCC analiza se obračunava i izražava u različitim formatima, a najčešći format je neto sadašnja vrednost (NSV).

Prema Flannagan-u i ostalima [3], neto sadašnja vrednost se izražava u obliku:

$$NPV = \sum_{t=0}^T \frac{C_t}{(1+r)^t} \quad (1)$$

gde su:

C_t - ukupni procenjeni troškovi kroz određeni period,

r - diskontna stopa,

T - posmatrani period analize

U realnosti, vrednost investicije može biti narušena efektima inflacije, te uzimajući nju u obzir, sadašnji nivo investicije biće uvećan. Faktor izmene (modifikacije) naziva se neto diskontna stopa inflacije (nett of inflation discount rate):

$$ndr = \left[\left(\frac{1 + \text{kamatna stopa \%}}{1 + \text{inflacija \%}} \right) - 1 \right] \quad (2)$$

Proračun troškova životnog ciklusa (LCC analiza) obuhvatio je period od 20 godina, a osim dužine eksploatacionog perioda, uzete su u obzir i vrednosti kamatne stope i stopa inflacije za posmatrani vremenski period. Proračun je urađen korišćenjem programa *Mathcad 14* [4].

U tabeli 1 date su rekapitulacije svih troškova, za sva tri varijantna rešenja, a u tabeli 2 dana potrebno za izvođenje fasade.

Tabela 1. Ekonomski kriterijumi

Varijantna rešenja	Troškovi nabavke i ugradnje [€]	Troškovi održavanja i demont. [€]	Troškovi prema LCC analizi [€]
I	72789,94	24719,56	116272,8052
II	102934,44	15675,57	114233,1338
III	107506,66	18991,85	130919,6409

Tabela 2. Vreme potrebno za izvođenje fasade

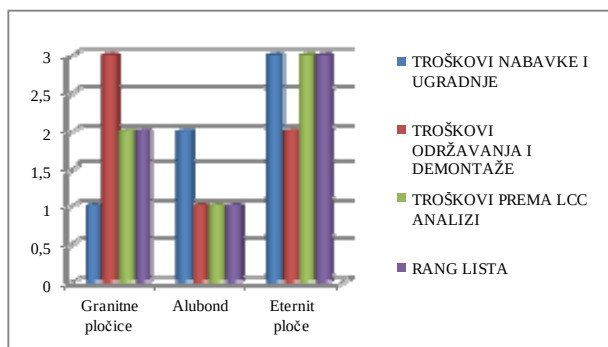
Varijantna rešenja fasadne obloge	Dana potrebno za ugradnju
I	28
II	33
II	34

5. IZBOR OPTIMALNOG TIPFA FASADNE OBLOGE ZA VENTILIRAJUĆU FASADU

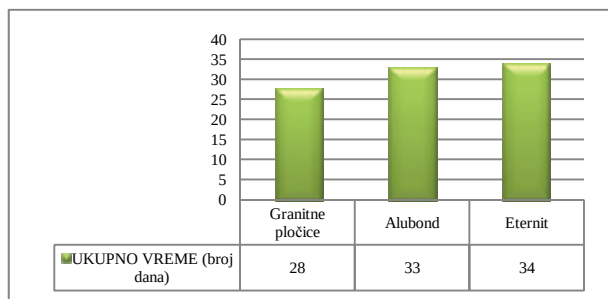
Za izbor optimalnog tipa fasadne obloge, metodom rangiranja, uzete su u obzir sve vrste troškova, troškovi

nabavke i ugradnje, troškovi održavanja i demontaže, troškovi prema LCC analizi i vreme ugradnje fasade.

Na slici 6, grafički je prikazano rangiranje varijanti fasadne obloge prema troškovima, a na slici 7, prema vremenu ugradnje.



Slika 6. Rangiranje fasadne obloge na bazi troškova



Slika 7. Rangiranje na osnovu vremena potrebnog za ugradnju

Pojedinačno i konačno rangiranje, po svim kriterijumima, na bazi troškova i vremena ugradnje, prikazano je u tabeli 3, a grafički na slikama 8 i 9.

Tabela 3. Rangiranje troškova i vremena ugradnje – konačna rang lista

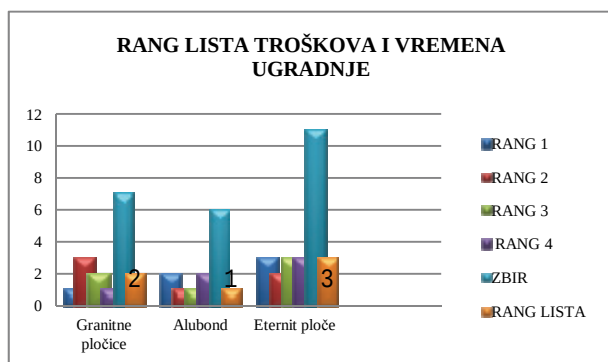
	Rang 1.	Rang 2.	Rang 3.	Rang 4.	zbir	R list.
V-1	1	3	2	1	7	2
V-2	2	1	1	2	6	1
V-3	3	2	3	3	11	3

Rang 1 - Rangiranje troškova proizvodnje i ugradnje

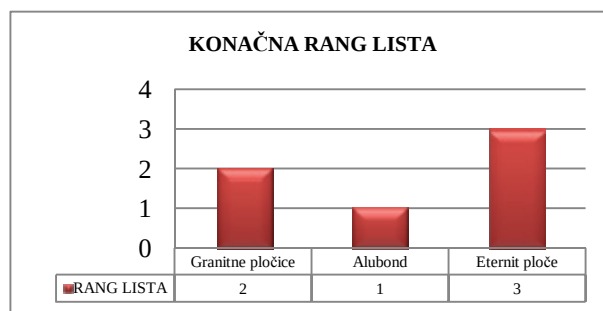
Rang 2 - Rangiranje troškova održavanja i demontaže

Rang 3 - Rangiranje troškova prema LCC analizi

Rang 4 - Rangiranje prema vremenu ugradnje



Slika 8. Rang lista troškova i vremena ugradnje



Slika 9. Konačna rang lista

6. ZAKLJUČAK

Predmet ovog rada je bio izbor optimalnog tipa fasadne obloge za ventilirajuću fasadu za stambeno-poslovni objekat u Novom Sadu. U radu je opisan postupak ugradnje i osnovne karakteristike fasada, za odabrana varijantna rešenja: varijantno rešenje I - fasadna obloga od granitne keramike, varijantno rešenje II - fasadna obloga od Alubonda i varijantno rešenje III - fasadna obloga od vlaknasto-cementnih ploča. Za svaku varijantu sračunate su tri vrste troškova: troškovi nabavke i ugradnje, troškovi održavanja i demontaže i troškovi životnog ciklusa, kao i vreme potrebno za izvođenje fasade. Na osnovu sve četiri analize, optimalan izbor fasadne obloge za ventilirajuću fasadu je: varijantno rešenje II - fasadna obloga od Alubonda.

Ovim radom je ukazano i na mogućnost i potrebu sagledavanja troškova sa šireg aspekta, analize inicijalnih, ali i troškova životnog ciklusa. Izbor optimalnog tipa fasadne obloge, baziran je na većem broju kriterijuma, a način sagledavanja ekonomskog pokazatelja doprinosi kvalitetu konačnih investicionih odluka.

7. LITERATURA

- [1] *Normativi i standardi rada u građevinarstvu-visokogradnja, građevinski radovi-1*, Građevinska knjiga, Beograd, (2004).
- [2] Beker, I., Stanivuković D. *Logistika*, FTN, Novi Sad;
- [3] Bull, J. (1993) *Life Cycle Cost for construction*, New Castle upon Tyne.
- [4] Aleksić, I.: *Izbor optimalne varijante podne obloge metodom višekriterijumske optimizacije na bazi troškova*, Diplomski-master rad, Fakultet tehničkih nauka, Novi Sad, 2011.

Kratka biografija:



Olinka Franić rođena je u Jajcu, Bosna i Hercegovina 1980. Srednju građevinsku školu »Jovan Vukanović«, odsek niskogradnja, završila je u Novom Sadu. Master rad na Fakultetu tehničkih nauka u Novom Sadu odbranila je 2015. .



Jasmina Dražić rođena je u Novom Miloševu 1958.god. Doktorirala je na Fakultetu tehničkih nauka 2005.god., a od 2015.god. je u zvanju redovnog profesora. Oblasť Zgradarstvo–građevinske i arhitektonske konstrukcije.

**PROJEKAT DELIMIČNO UKOPANOG REZERVOARA, ČELIČNE NADSTREŠNICE I
ANALIZA SEIZMIKE NA DELIMIČNO UKOPANI REZERVOAR****PROJECT OF PARTIALLY BURIED TANK, STEEL CANOPY AND SEIZMIC
ANALYSIS ON PARTIALLY BURIED TANK**Bojan Poletanović, *Fakultet tehničkih nauka, Novi Sad***Oblast – GRAĐEVINARSTVO**

Kratak sadržaj – Rad se sastoji iz dve celine koje su međusobno povezane. Pod prvim delom podrazumevan je proračun delimično ukopanog rezervoara i čelične nadstrešnice iznad njega, dok drugi deo podrazumeva detaljnu seizmičku analizu delimično ukopanih rezervoara.

Abstract – Thesis contains two parts which are connected. First part includes complete structural design of tank and canopy, while the second part includes detailed seismic analysis on partly buried tank.

Cljučne reči: Delimično ukopan rezervoar, čelična nadstrešnica, statički proračun, dinamički proračun, seizmička analiza...

1. UVOD

Projektovan je delimično ukupan rezervoar za potrebe prijema uljnih muljeva. Iznad rezervoara, kao zaklon projektovana je čelična nadstrešnica. Objekat se nalazi u Beočinu.

Rad sadrži:

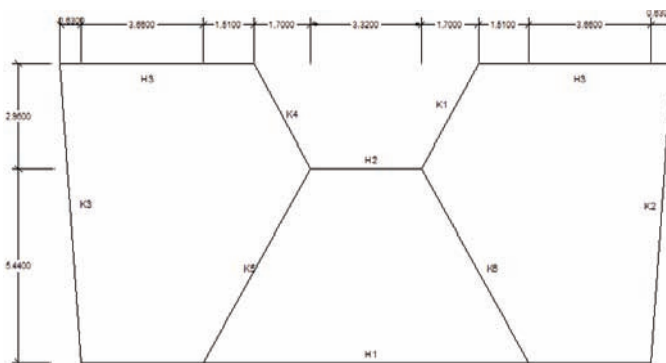
- tekstualnu dokumentaciju
- grafičku dokumentaciju
- numeričku dokumentaciju

2. OPIS PROJEKTA**2.1 Arhitektonsko rešenje i konstruktivni sistem**

Rezervoar je predviđen za skladištenje uljnih muljeva zepreminske težine 10 kN/m^3 . Projektovan je kao armiranobetonski. Temeljna ploča je debljine 40 cm i armirana je rebrastom armaturom RA 400/500 u dva pravca. Zidovi rezervoara su debljine 25 cm i takođe armirani armaturom RA 400/500. Uslov pri proračunu rezervoara je takođe bio da bude nepropusan, tj. sa prislinama manjim od 0,1 mm. Rezervoar je ukopan u zemlju 1,6 m, a 0,8 m se nalazi iznad zemlje, pa mu je ukupna dubina 2,4 m. Rezervoar je podeljen u tri komore, zapremina $V_1 = 91,6 \text{ m}^3$, $V_2 = 66,2 \text{ m}^3$ i $V_3 = 91,6 \text{ m}^3$, pa je ukupna zapremina rezervoara $V = 199,5 \text{ m}^3$. Zbog skladištenja agresivnih materijala u rezervoar, potrebno je posebnu pažnju posvetiti hidroizolovanju rezervoara.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Đorđe Ladinović.



Slika 1. Dispozicija bunkera

Nadstrešnica je projektovana kao čelična konstrukcija. Iznad postojećih rezervoara nalaziće se čelična nadstrešnica. Nadstrešnica ima dimenzije u osnovi $22,0 \times 19,2 \text{ m}$ sa razmakom glavnih osa nosive čelične konstrukcije od 4,80 m. Svetla visina konstrukcije u unutrašnjosti hale je 9,814 m. Nadstrešnica je otvorenog tipa sa fasadnim trapezastim limom postavljenim samo sa jedne strane, u osi H_2.

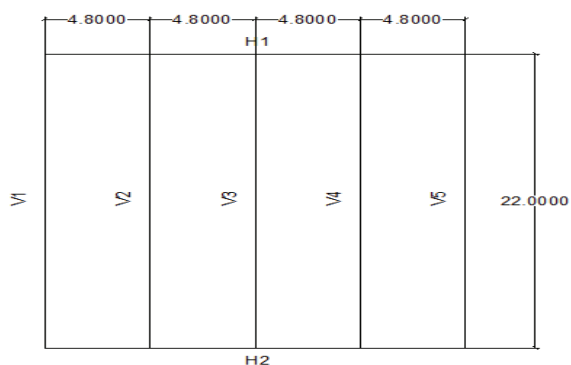
Nagib krova je dvostran (12 %), krov je obložen, takođe, trapezastim limom. Pod hale je postavljen na kotu cca 80,85 mm. Glavnu nosivu konstrukciju čine pet rešetkastih nosača na stubovima IPB 280 postavljeni na osovinskom razmaku od 4,80 m. Rešetkasti nosači se sastoje iz gornjeg pojasa sa dvostranim nagibom od 12 %, izrađenog od HOP profila 200x100x6,5, donjeg pojasa izrađenog takođe od HOP profila 200x100x6,5.

Vertikale i dijagonale rešetke izrađene su od profila HOP 100x60x5. Nosači fasade u osi H_2 su HOP 100x100x5, postavljeni na razmaku približno 1,50 m. Rožnjače su postavljene na osovinskom razmaku od 1,55 m, a izrađene su od profila [220].

Sve veze u čeličnoj konstrukciji se izvode u zavarenoj izradi. Nadstrešnica je ukrućena spregovima za prihvatanje uticaja vetra u krovnoj ravni na svim stranama krovne ravni.

Radi ukrućenja postavljana su još 4 IPB 180 stuba, koji su povezani riglama za glavne stubove i ukrućeni spregovima.

Svi spregovi su od kružnog čeličnog poprečnog preseka, $D = 30 \text{ mm}$.



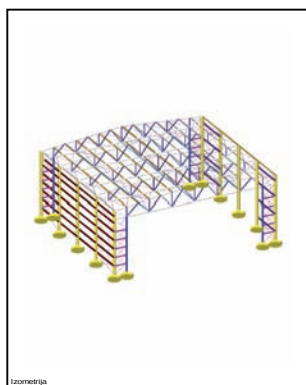
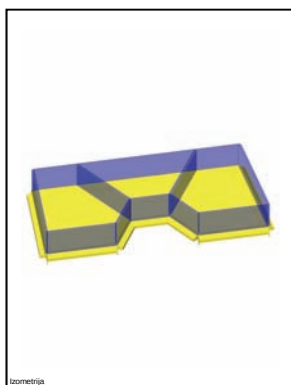
Slika 2. Dispozicija nadstrešnice

2.3. Analiza opterećenja

Stalno opterećenje čine težina konstrukcije i dodatno stalno opterećenje od limova. Sopstvena težina nosivih elemenata se proračunava softverom. Opterećenje od tla i dodatnog korisnog opterećenja se proračunava na osnovu važećih domaćih pravilnika za karakteristično opterećenje. Opterećenje od mulja se proračunava prema važećim domaćim standardima za opterećenje konstrukcije tečnostima. Opterećenje snegom se proračunava prema važećim domaćim pravilnicima. Opterećenje vetrom je sračunato prema važećim domaćim propisima za vetar. Seizmičko opterećenje se izračunava pomoću softvera, Tower 7.0, koji nudi opciju seizmičkog proračuna prema važećem domaćem standardu i naneto je odgovarajuće opterećenje od vode i tla usled seizmičkog dejstva.

2.4. Statički i dinamički proračun

Konstrukcija je modelirana prostorno u programskom paketu Tower 7.0. Veza objekta i podloge kod rezervoara je modelirana pomoću elastičnih opruga po Vinklerovom modelu.



Sl. 3. 3D model rezervoara

Sl. 4. 3D model nadstrešnice

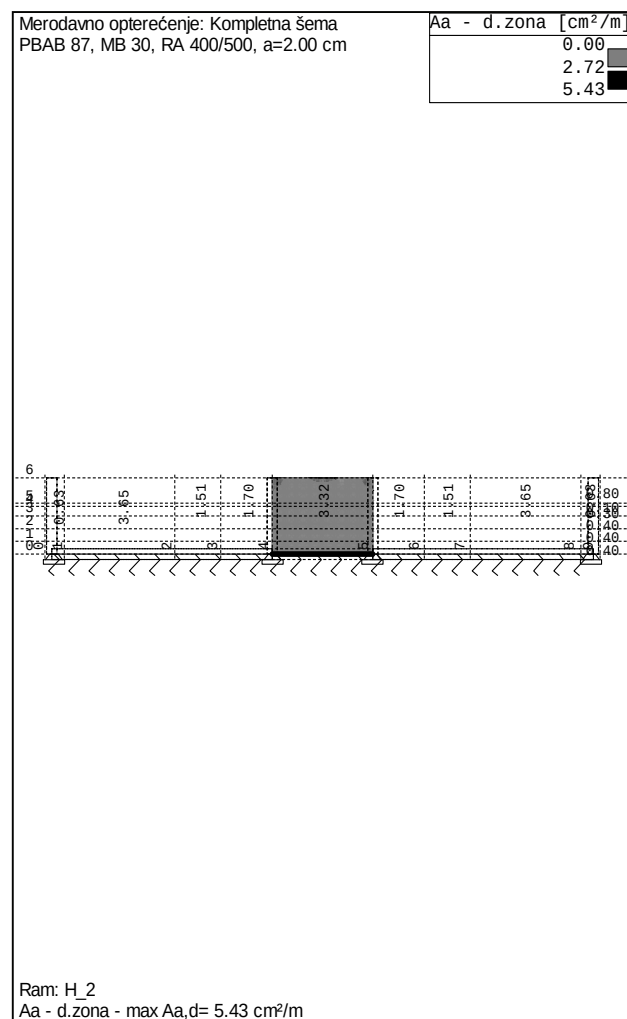
Nakon sprovedene modalne analize zgrade, pristupa se definisanju parametara za proračun seizmičkih sila. Programski paket nudi opciju seizmičkog proračuna prema JUS-u, preko Multi modalne analize.

2.5. Kontrola napona i pomeranja

Kod rezervoara je izvršena kontrola napona u tlu i sleganje temelja. Kontaktni naponi u tlu su manji od dozvoljenih. Kod nadstrešnice su vrednosti napona manje od dopuštenih, a takođe su zadovoljeni uslovi za kontrolu stabilnosti. Konstrukcija nadstrešnice je ukružena spregovima.

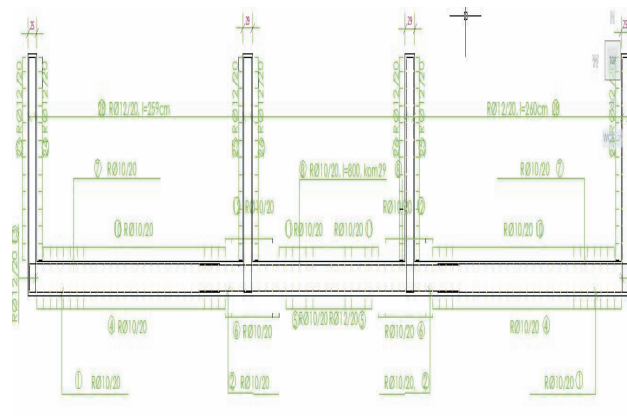
2.6. Dimenzionisanje

Nakon proračuna koji je sproveden u Tower 7, dobijena je potrebna armatura u ploči i zidovima rezervoara. Najveći uticaji se javljaju u zidu rama H_2.



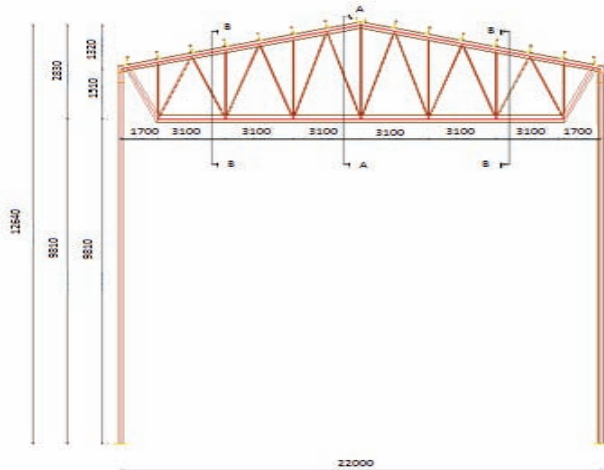
Slika 5. Prikaz potrebe za armaturom u zidu rama H_2

Dodatni uslov, pored graničnog stanja nosivosti, bio je u proračun prslina, koji je zahtevao da u rezervoaru u kojem se skladišti agresivni materijal prslina bude manje od 0,1 mm. Usvojena armatura u zidovima je RΦ12/20 i u gornjoj i u donjoj zoni. Usvojena armatura u donjoj ploči je RΦ10/20 i u donjoj i u gornjoj zoni.



Slika 6. Prikaz armature kroz poprečni presek rezervoara

Nakon provere kontrole napona i kontrole stabilnosti linijskih elemenata kod čelične nadstrešnice usvojeni su sledeći poprečni presezi: glavni stubovi IPB280, rešetka kao glavni noseći elementi na stubovima profila pojasnih štapova HOP 200x100x6,5, dok su dijagonale i vertikale HOP 100x100x5. Rožnjače, kao sekundarni nosači su usvojeni [220].



Slika 7. Čelični ram u osi V_2, V_3 i V_4

3. ANALIZA SEIZMIČKOG DEJSTVA NA DELIMIČNO UKOPAN REZERVOAR

Seizmički proračun inženjerskih objekata vrši se pomoću metode spektralne analize i/ili metode dinamičke analize. U zavisnosti od kategorije objekata vrši se određena analiza. Za objekte „van kategorije“ kao obavezna se mora izvršiti metoda dinamičke analize ali se zahteva i proračun po metodi spektralne analize. Za objekte koji spadaju u „objekte i kategorije“ smatra se da je dovoljno da se seizmički proračun izvrši pomoću metode spektralne analize.

Metodom spektralne analize se dobijaju seimičke sile na sledeći način: $S_{ik} = K_s * \beta_i * \eta_{ik} * \varphi * G_k$

Metodom dinamičke analize vrši se određivanje ponašanje konstrukcije u elastičnom i neelastičnom području rada za vremenske istorije ubrzanja tla očekivanih zemljotresa na lokaciji objekta.

Pored inercijalnih sila koje se javljaju usled težine same konstrukcije, kod ukopanih ili delimično ukopanih rezervoara, usled seizmike moraju se analizirati i dopunski uticaji koji se javljaju.

1. usled tečnosti u rezervoaru (hidrodinamički pritisak) koji se superponira sa statičkim pritiskom tečnosti na rezervoar i
2. usled tla koji deluje na dati rezervoar (ukoliko se radi o elastičnim deformacija tla (kako se obično i dešava) dobijeno seizmičko dejstvo tla se superponira sa njegovim, odgovarajućim statičkim dejstvom).

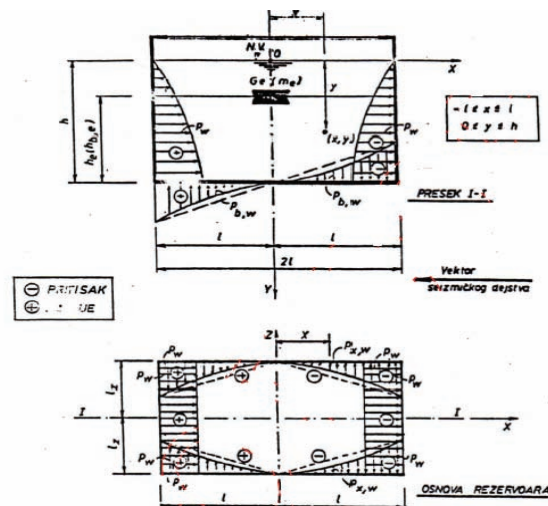
Proračun seizmičkih, inercijalnih sila usled težine same konstrukcije izračunat je pomoću softvera Tower7.

Prilikom zemljotresa, dolazi do podrhtavanja tla i kretanja tečnosti u rezervoaru. Upravo prilikom kretanja tečnosti javljaju se dve vrste pritiska:

1. impulsivni
2. konvektivni.

Impulsivni pritisak potiče od težine tečnosti koja se pretpostavlja da zajedno se kreće sa konstrukcijom, posmatra se kao da je kruto vezana za rezervoar.

Konvektivni pritisak potiče od dela tečnosti koji na samoj površini tečnosti menja visinu usled podrhtavanja konstrukcije. Obično se računa da bi se utvrdilo koliko će tečnost da pokvasi zid. Prema domaćim propisima, dovoljno je tačno računati hidrodinamički pritisak sa zanemarenjem konvektivnog dela (ukupni hidrodinamički pritisak tečnosti računati kao impulsivni pritisak).



Slika 8. Raspored seizmičkih pritiska (zatezanja) kod pravougaonih rezervoara.

Opterećenje zidova pravougaonog rezervoara koji se nalaze ortogonalno od pravca dejstva zemljotresa se računaju kao opterećeni konstantnim pritiskom (ili zatezanjem) i računaju se preko sledećeg obrasca:

$$p_w = K_s * \gamma_w * h * \sqrt{3} * \left[\frac{y}{h} - \frac{1}{2} * \left(\frac{y}{h} \right)^2 \right] * \tanh \left(\sqrt{3} * \frac{L}{h} \right)$$

Dok je opterećenje zidova koji se nalaze paralelno dejstvu zemljotresa računaju prema sledećem obrascu:

$$p_w = K_s * \gamma_w * h * \sqrt{3} * \left[\frac{y}{h} - \frac{1}{2} * \left(\frac{y}{h} \right)^2 \right] * \frac{\sinh \left(\sqrt{3} * \frac{x}{h} \right)}{\cosh \left(\sqrt{3} * \frac{L}{h} \right)}$$

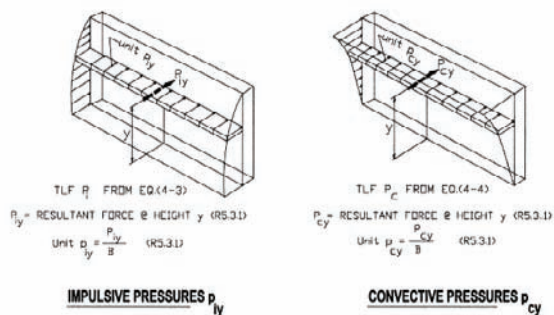
Hidrodinamički, impulsivni pritisak (zatezanje) dna pravougaonog rezervoara pretpostavlja se da deluje po čitavom dnu rezervoara i da mu se intenzitet menja sa koordinatom x. Računa se prema sledećem obrascu:

$$p_{b,w} = \frac{1}{2} * K_s * \gamma_w * h * \sqrt{3} * \frac{\sinh \left(\sqrt{3} * \frac{x}{h} \right)}{\cosh \left(\sqrt{3} * \frac{L}{h} \right)}$$

Prema američkim propisima hidrodinamičko opterećenje se može predstaviti pomoću impulsivnog i konvektivnog dela. Impulsivni i konvektivni deo, prema važećim američkim propisima, se sračunavaju na sledeći način:

$$P_i = Z * S * I * C_i * \frac{W_i}{R_{wi}} ;$$

$$P_c = Z * S * I * C_c * \frac{W_c}{R_{wc}}$$



Slika 9. Grafički prikaz impulsivnih i konvektivnih pritisaka, ACI

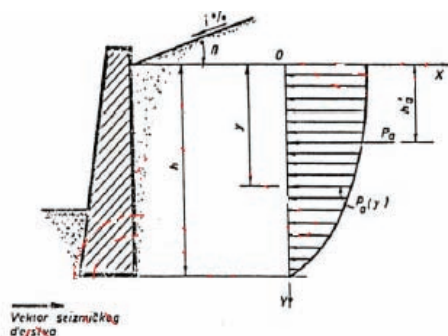
Pri proračunu seizmičkog pritiska tla pored seizmičkih sila usled inercijalnih sila od težine konstrukcije moraju se uzeti u obzir i dodatni seizmički uticaji usled aktivnog ili pasivnog stanja tla. Usled određivanja seizmičkih sila tla mogu se posmatrati dva slučaja:

1. kada su nastupile **elastične** deformacije u tlu
2. kada su nastupile **plastične** deformacije u tlu

Kada su nastupile elastične deformacije u tlu aktivni pritisak tla se superponira sa statičkim pritiskom tla i računa se prema sledećem obrascu:

$$p_a(y) = K_s * \psi * \gamma_z * h * R(y, \beta)$$

$$R(y, \beta) = \left\{ 1 - \frac{1}{4} * \left(\frac{y}{h} \right) * \left[10 * \left(\frac{y}{h} \right)^2 - 9 * \frac{y}{h} + 3 \right] \right\} + \left(1 - \frac{y}{h} \right) * \tan \beta$$



Slika 10. Seizmički, aktivni pritisak tla (elastične deformacije)

4. ZAKLJUČAK

Dobijeno opterećenje prema našem pravilniku je nešto manje od američkih propisa (prema kojima se dobijaju veći uticaji u pločama). Po našim propisima konvektivni pritisci se zanemaruju. Međutim, hidrodinamički pritisci se obično kod rezervoara i ne pojavljuju u najkritičnijoj kombinaciji, jer statičko opterećenje od zemlje i dodatno korisno opterećenje ili opterećenje od uskladištenog materijala kombinovani sa stalnim opterećenjem daju kritičnu kombinaciju opterećenja.

5. LITERATURA

- * Pravilnik o tehničkim normativima za projektovanje i proračun inženjerskih objekata u seizmički aktivnim područjima
- * Grupa autora, Beton i armirani beton prema BAB87, knjiga 1, Univerzitetska štampa, Beograd, 2000.
- * Grupa autora, Beton i armirani beton prema BAB87, knjiga 2, Univerzitetska štampa, Beograd, 2000.
- * Ž. Radosavljević, D. Bajić, Armirani beton 3, Građevinska knjiga, Beograd, 2007.
- * S. Stevanović, Fundiranje I, Naučna knjiga, Beograd, 1989.
- * <http://www.radimpex.rs>. – uputstvo za primenu Tower7

Kratka biografija:



Bojan Poletanović, rođen u Vrbasu 1990. god. Oktobra 2009. god. upisuje odsek za Građevinarstvo na Fakultetu tehničkih nauka u Novom Sadu. Septembra 2014. god. stiče zvanje diplomiranog inženjera građevinarstva. Master rad na Fakultetu tehničkih nauka iz oblasti Seizmička analiza konstrukcija odbranio je u oktobru 2015. godine.

**PROCENA STANJA, KONTROLNI PRORAČUN I SANACIJA KONSTRUKCIJE
SUŠARE U MESTU GAJDOBRA****ASSESSMENT OF CURRENT CONDITION, CONTROL CALCULATION AND
REHABILITATION OF STRUCTURE OF DRIER BUILDING IN PLACE GAJDOBRA**

Jelena Joksović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – GRAĐEVINARSTVO

Kratak sadržaj – Zadatak ovog rada je podeljen na teorijski i stručni deo. U teorijskom delu je obradjena tema: „Silosi“, gde su razmatrana sledeća poglavlja: istorija silosa, podela silosa, vrste protoka pri pražnjenju materijala, proračun prema SRPS ISO 11697:1966 i oštećenja kod silosa. Stručni deo se odnosi na procenu stanja sušare za hmelj, kontrolni proračun postojećeg stanja konstrukcije, kontrolni proračun stanja konstrukcije usled novog opterećenja-pšenice i sanacije konstrukcije. Izvršen je detaljni vizuelni pregled konstrukcije sušare na osnovu čega je izvršena procena stanja elemenata konstrukcije. Detaljno su opisani i kroz fotografije prikazani defekti i oštećenja, nakon čega je tabelarno prikazana njihova zastupljenost na konstrukciji. Nakon procene stanja urađen je kontrolni proračun konstrukcije za postojeće stanje i za slučaj delovanja opterećenja od pšenice. Provereni su statički uticaji, izvršene proračunske kontrole i kontrolisana količina računski potrebne armature. Izvršena je konstrukcijska i nekonstrukcijska sanacija objekta.

Abstract - The aim of this work is divided into theoretical and expert part. The theoretical part deals with topic: "Silos" where following chapters are discussed: history of the silo, types of silos, types of flow during discharge of materials, calculation according to SRPS ISO 11697:1966 and damage in the silos. Expert part refers to the assessment of current condition of drier building for hops, control calculation of the current condition of the structure, control calculation of condition of the structure due to the new load-wheat and rehabilitation of the structure. A detailed visual inspection of structure of drier building is performed, based on which an assessment of current condition of structural elements is made. Defects and damage are described in detail with additional photos, with their distribution along structure shown in table. After assessment of current condition control calculation of structure for the current condition and in case of load of wheat has been made. Static effects are checked, calculation control and check of amount of structural design required reinforcement were performed. Structural and non-structural rehabilitation of the building are made.

Ključne reči: silosi, sušara, defekti, oštećenja, procena stanja, proračun, sanacija.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila prof. dr Mirjana Malešev.

1. PRIKAZ ISTORIJSKOG DELA – SILOSI**1.1. Istorijski pregled**

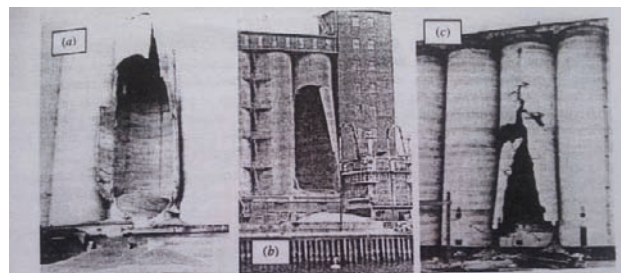
Za smeštaj zrnastih materijala su, u davnoj prošlosti, korišćene silo-jame, pletene košare, i glinene posude. Reč silos je starokeltskog porekla, na grčkom je siros, na španskom silo i označava jamu za smeštaj žita. Prvi silosi su, ustvari, bile iskopane ili izdubljene jame u tlu, u stenskoj masi (Slika1/2.).



Slika 1. Silo-jama

1.2. Problemi kod betonskih i metalnih silosa od opterećenja zrnastim materijalom

Kod ovih konstrukcija postoji rizik od pojave manjih ili većih oštećenja, koja se obično javljaju u vidu pojave prslina na zidovima AB ili PB ćelija, ili izbočavanja limova kod metalnih silosa (Slika 2.)



Slika 2. Rušenje zida ćelije

Kod metalnih silosa, javlja se poseban problem izbočavanja tankozidnih metalnih zidova ćelija.



Slika 3. Oštećenja metalnih silosa usled protočnih problema

1.3. Fundiranje silosa

Kod fundiranja AB silosa se mogu javiti različiti problemi pri eksploataciji, usled rotacije temelja, prevelikog sleganja ili diferencijalnih sleganja.



Slika 4.. Obaranje silosa usled loma tla.

2. PROCENA STANJA SUŠARE

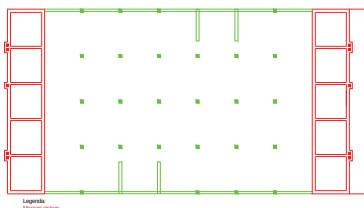
2.1. Tehnički opis

Konstruktivni sistem zgrade

Konstruktivni sistem objekta je mešoviti – skeletni i masivni:

- Noseću konstrukciju skeletnog sistema formiraju pune armiranobetonske ploče oslonjene tačkasto na stubove. Opterećenje sa ploče prihvataju stubovi preko proširenja- kapitela. Kapiteli su izvedeni četvrtastog oblika, za prihvatanje većih raspona i opterećenja, i kako bi zadovoljili osnovni problem ovih tavanica obezbeđenje tavanice konstrukcije od proboja. Po obodu pečurkasta tavanica je oslonjena takođe tačkasto, i to je ostvareno preko polu kapitela, odnosno četvrt kapitela u uglovima. Stubovi su dimenzija 35x35, 30x30 i 25x25 cm. Ispuna je rađena standardnom opekam. Debljina zidova je 25 cm.
- Masivni sistem čine zidovi rađeni standardnom opekam, debljine 38cm i 25cm.

Na slici 5. dat je šematski prikaz konstruktivnog sistema objekta.



Slika 5. Šematski prikaz konstruktivnog sistema objekta

2.2. Procena stanja

Pregledani su sledeći elementi:

- AB stubovi i kapiteli
- Pečurkaste tavanice
- AB serklaži
- Stepenište
- Zidovi
- AB Nadstrešnice

Defekti registrovani na konstrukciji:

- Nedovoljna debljina zaštitnog sloja betona
- Ostaci drvene oplata
- Nepravilno ugrađena armatura

- Tragovi oplata

Najzastupljeniji defekt je tragovi oplata i javlja se na svim pločama.



Slika 6. Tragovi oplata na pločama

Karakteristična oštećenja na konstrukciji:

- Ljuskanje završnog premaza (kreča)
 - Površinsko odvajanje i otpadanje maltera
 - Mrežaste prsline u malteru
 - Obijene ivice stubova (Slika 9.)
 - Vidljiva armatura
 - Površinska korozija armature
 - Otpadanje ivica betona usled korozije armature
 - Prsline
 - Rastvaranje i ispiranje $\text{Ca}(\text{OH})_2$
 - Biološka korozija
 - Lokalno drobljenje betona oko naknadno izvedenih otvora
 - Nepravilno izvedeni otvori (Slika 8.)
 - Krunjenje betona
 - Odvaljivanje i otpadanje maltera (Slika 7.)
 - Prsline u uglovima otvora
 - Pukotina u zidanom zidu
 - Oštećenje zida
 - Krunjenje i otpadanje sloja maltera
 - Krunjenje betona usled zamrzavanja i odmrzavanja
- Otpadanje maltera (Slika 7.), Nepravilno izvedeni otvori (Slika 8.) i odvaljene ivice betonskih stubova (Slika 9.) su najzastupljenija oštećenja.



Slika 7. Otpadanje maltera



Slika 8. Nepravilno izvedeni otvori

2.3. Nedestruktivna ispitivanja

Od nedestruktivnih metoda ispitivanja korištene su: kuckanje čekićem, metoda sklerometra (određivanje površinske tvrdoće betona), određivanje dubine karbonatizovanog sloja betona – kolorimetrijska metoda.

Kolorimetrijskom metodom je utvrđeno da je beton zahvaćen procesom karbonatizacije, a sklerometrom je dobijena čvrstoća betona pri pritisku.



Slika 9. Odvaljena ivica betona

3. KONTROLNI PRORAČUN KONSTRUKCIJE

U vreme kada je objekat projektovan osnovna namena mu je bila skladištenje hmelja.

Shodno zahtevim koje su nametali uticaji od ovog opterećenja izvršeno je dimenzionisanje skladišta. Pošto se skladište više ne koristi za čuvanje hmelja, odlučeno je da se ispita mogućnost skladištenja nekog drugog materijala. Proračun konstrukcije će se obaviti za opterećenje od pšenice, koja je znatno teža od hmelja ($\gamma_{z,hmelj} = 2 \text{ kN/m}^2$, $\gamma_{z,pšenica} = 7.8 \text{ kN/m}^2$). Dalje će se vršiti poređenje uticaja dobijenih od novog opterećenja sa postojećim stanjem, dimenzionisanje i usvajanje dodatne armature ukoliko se javi potreba za njom.

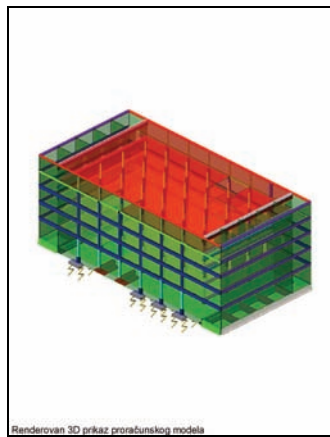
3.1. Analiza opterećenja

- Stalno opterećenje-sopstvena težina nosećih elemenata, težina krovnog pokrivača
- Opterećenje od snega: $1 \text{ (kN/m}^2 \text{)}$ (privremeni tehnički propisi za opterećenje zgrada-opterećenje snegom Sl.II list SFRJ 61/48);
- Korisno opterećenje: korisno opterećenje (definisano prema propisima JUS.U.C7.121/1988)
- Opterećenje od vetra: (važeći standardi SRPS U.C7.110., SRPS U.C7.111, SRPS U.C7.112)
- Seizmičko dejstvo.

3.2. Modeliranje konstrukcije:

Konstrukcija je modelirana u softverskom paketu Tower 6 koji je baziran na metodi konačnih elemenata. Svi uticaji u konstrukciji određeni su na osnovu proračunskog modela koji dovoljno realno predstavlja stvarnu konstrukciju i njeno ponašanje pod dejstvima. Medjuspratne tavanice, temeljna i krovna ploča su modelirani kao površinski elementi, dok su grede i stubovi modelirani kao linijski elementi. Mehaničke karakteristike kao što su računska čvrstoća betona na pritisak pri savijanju, modul elastičnosti, Poisson-ov koeficijent ili koeficijent temperaturnog širenja su određeni kvalitetom betona, tj. njegovom markom. Geometrijske karakteristike su pridružene elementima usvajajući bruto betonske preseke, zanemarujući doprinos armature.

Prilikom modeliranja konstrukcije, za model tla je usvojena Vinklerova podloga. Stepenišna konstrukcija objekta modelirana je zajedno sa glavnom konstrukcijom ali će zbog preglednosti biti prikazano odvojeno.



Slika 10. Renderovan 3D prikaz proračunskog modela

3.3. Proračunske kontrole

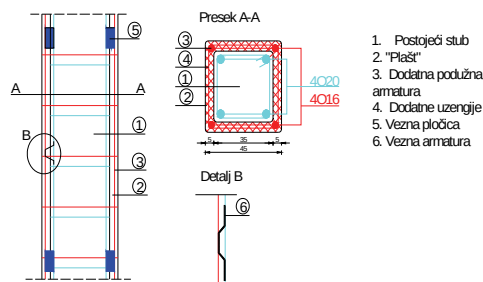
Izvršene su sledeće proračunske kontrole:

- Kontrola napona u tlu
- Kontrola napona u stubovima
- Kontrola horizontalnog pomeranja vrha zgrade

4. SANACIJA OBJEKTA

4.1. Konstrukcijska sanacija stubova

Sanacija AB stubova izvršiće se ojačanjem osnovnog preseka novim betonskim plaštom i ugrađivanjem dodatne armature odgovarajućih prečnika prema potrebi. Oko stubova na kojima je potrebno izvršiti ojačanje oformiće se AB plašt odgovarajuće debljine i u njega smestiti nova armatura i uzengije. Na postojećim stubovima, koje je potrebno sanirati, izvršiće se štemovanjem uklanjanje zaštitnog sloja betona do armature pri čemu će se obrazovati pojasevi sa jasno vidljivom armaturom. U ovim oštemovanim zonama izvršiće se varenje čeličnih pločica debljine $d = 6.00 \text{ mm}$ i visine $h = 10.00 \text{ cm}$ za postojeće šipke podužne armature stuba. Postojeće šipke podužne armature i armaturene šipke betonskog plašta će se dodatno povezati veznom armaturom, na svakih 1 m . Za betoniranje plašta koristiće se trofrakcijski, polimer modifikovani beton.



Slika 11. Ojačavanje stubova dodavanjem novog betonskog plašta i dodatne armature

4.2. Nekonstrukcijska sanacija stubova

Reprofilisanje reparaturnim malterom trošnih ispućalih i odvaljenih ivica stubova je postupak koji se obavlja na većini elemenata, pa će biti opisan u nastavku:

- Uklanjanje ručnim štemovanjem ispućalih, labavih delova betona armiranobetonskih stubova.
- Štemovanjem treba formirati otvor pravilnog geometrijskog oblika, bez isklinjavanja. Ova

operacija se uglavnom odnosi na na ispucale i trošne ivice stubova.

- Otkrivanje šipke armature najmanje po 5 cm u oba smjera radi provjere stanja šipki. Tokom postupka uklanjanja betona u zoni armature treba voditi računa da se ne izazivaju vibracije armatura, ne naruši adhezija između armature i preostalog betona, kao i da ne dođe do pojave novih oštećenja,
- Čišćenje "oslobođenih" delova šipki glavne armature od tragova korozije sve do metalnog sjaja i cementnog kamena žičanim četkama.
- Čišćenje odštemovanih zona betona od prašine i labavih delova pomoću komprimovanog vazduha,
- Nanošenje Sika Mono Top-910 na očišćenu armaturu kao zaštitu od korozije i vezu starog i novog betona,
- Postavljanje jednostrane oplata
- Popunjavanje odštemovanih delova reperturnim malterom Sika Mono Top-412N (karakteristike date u prilogu). U slučaju da je prilikom štemovanja uklonjen veći deo betona od predviđenog, mora se izvršiti reprofiliisanje do dobijanja projektovanog oblika stuba.
- Radi sprečavanja naglog isušivanja maltera i pojave prslina, u prvih 24 časa nakon nanošenja reparaturnog maltera potrebna je intenzivna nega kvašenjem vodom.

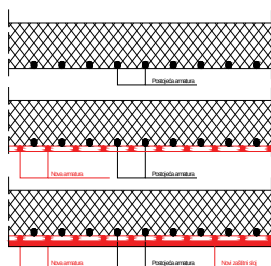


Slika 12. Ivice koje treba odštemati i očistiti armaturu

4.3. Konstrukcijska sanacija ploča

Sanacija će se izvršiti dodavanjem računski potrebne armature u donjoj zoni na naznačenim mestima. Na ostatku ploče, gde se ne javljaju potrebe za dodatnom armaturom dodaje se konstruktivna armatura kako bi se izjednačila debljina ploče.

Usvaja se mrežasta armatura Q335 (Ø6/15).



Slika 13. Primjer ojačanja donje zone ploče

4.4. Nekonstrukcijska sanacija ploča

Na ploči postoji veći broj otvora koji su potrebni za tehnološki proces, ali su nepravilno izvedeni i potrebno ih

je zaštititi postavljanjem karbonskih traka, pa je u nastavku dat opis ovog postupka:



Slika 14. Položaj karbonskih traka

- Trošan malter oko otvora sa Slike 8. je potrebno ukloniti ručnim štemovanjem,
- Armaturu koja viri iz otvora je potrebno skratiti,
- Reprofilisanje ivica otvora sa reperturnim malterom Sika Mono Top 412N,
- Obrada površina na koje se lepe karbonske trake- tehnikom peskarenja, takođe treba proveriti da i postoji odgovarajući stepen adhezije između betona i lepka pull-off tehnikom,
- Postavljanje karbonskih traka u podužnom pravcu oko otvora, kao što je prikazano na Slika 8, lepljenje pomoću odgovarajućih epoksidnih lepkova.

5. LITERATURA

1. Grupa autora: BETON I ARMIRANI BETON prema BAB 87, knjiga 1, univerzitetska štampa, Beograd, 2000.
2. Grupa autora: BETON I ARMIRANI BETON prema BAB 87, knjiga 2, univerzitetska štampa, Beograd, 2000.
3. V. Radonjanin, M. Malešev: MATERIJAL SA PREDAVANJA – TRAJNOST I PROCENA STANJA BETONSKIH KONSTRUKCIJA
4. V. Radonjanin, M. Malešev: MATERIJAL SA PREDAVANJA – SANACIJA BETONSKIH KONSTRUKCIJA
5. V. Radonjanin, M. Malešev: MATERIJAL SA PREDAVANJA – SANACIJA ZIDANIH KONSTRUKCIJA
6. M. Tatomirović: DEJSTVA USKLADIŠTENOG ZRNASTOG MATERIJALA NA ARMIRANOBETONSKE CILINDRIČNE ČELIJE SILOSA
7. Internet adrese: <http://srb.sika.com/>

Kratka biografija:



Jelena Joksović, rođena je u Novom Sadu 1990. god. Oktobra 2009. god. upisuje se Odsek za građevinarstvo na Fakultetu tehničkih nauka u Novom Sadu. Jula 2014. god. stiče zvanje diplomiranog inženjera građevinarstva. Master rad na Fakultetu tehničkih nauka iz oblasti Procena stanja i sanacija građevinskih objekata odbranila je u oktobru 2015. godine.

ДОГРАДЊА, КОНСТРУКЦИЈСКА И ЕНЕРГЕТСКА САНАЦИЈА ЗГРАДЕ ОСНОВНЕ ШКОЛЕ У НАСЕЉУ ГРБАВИЦА У БРЧКОМ У БРЧКОМ**UPGRADING, STRUCTURAL AND ENERGY REHABILITATION OF THE BUILDING OF ELEMENTARY SCHOOL IN GRBAVICA, BRČKO**

Александра Ђурић, Факултет техничких наука, Нови Сад

Област – ГРАЂЕВИНАРСТВО

Кратак садржај – У овом раду приказана је процена стања, доградња и ојачање АБ носеће конструкције школског објекта у Брчком, спратности Пр+С1. Теоријски део рада се односи на енергетску ефикасност у зградарству. У стручном делу рада приказани су основни технички подаци о објекту. Затим је на основу детаљног прегледа зграде дата оцена стања са аспекта енергетске ефикасности, носивости, стабилности и трајности. После адекватне процјене и описа постојећег стања конструкције, урађен је пројекат доградње једног спрата на постојећу конструкцију објекта. Формиран је нови просторни модел за постојеће и дограђено стање са новим додатним оптерећењем од дограђеног спрата. Сprovedена је статичка и динамичка анализа са провјером напона у стубовима и померања врха конструкције услјед хоризонталних дејстава. На крају рада предложене су мере за ојачање одређеног броја елемената носеће конструкције предметне зграде.

Abstract – This paper presents an assessment, upgrade and strengthening of RC structure of the primary school in Brčko, consisting of ground floor and one floor. The theoretical part of this paper refers to energy efficiency in buildings. In the professional part of paper basic technical data about this object are given. Further, an assessment is made based on detailed survey of the building in terms of energy efficiency, carrying capacity, stability and durability. After assessment and description of the existing state of the structure, a project of upgrading of one floor is made. After upgrading a new model of both existing and upgraded state is made, with an additional load from upgraded floor. The static and dynamic analyses are made to check the stress in columns and movement of the top of the structure due to horizontal actions are calculated, also. At the end of this paper some measures for strengthening several elements of bearing structures of this building are suggested.

Кључне речи: процена стања, energetska efikasnost, доградња, санације, "плашт" – бетонски омотач, анкери

1. УВОД

Рад се састоји од две међусобно независне целине. Први дио рада представља теоријско – истраживачки део са темом "Енергетска ефикасност у зградарству", а други део је везан за процену стања, енергетску ефикасност, доградњу и санацију основне школе у Брчком.

2. ЕНЕРГЕТСКА ЕФИКАСНОСТ У ЗГРАДАРСТВУ**2.1. Увод**

Енергетска ефикасност подразумева низ мера које се спроводе како би се на што бољи начин искористишћавала енергија и то тако: да се користи што мање енергије за исти ефекат, да се смањи негативан утицај на животну средину и ублаже климатске промене, те да се у зградарству постигне што боља удобност простора у којем се енергија користи.

Главни циљ енергетске ефикасности у зградарству је успоставити механизме који ће трајно смањити енергетске потребе при пројектовању, изградњи и коришћењу нових зграда, као и реконструкцији постојећих, те уклонити баријере увођењу мера енергетске ефикасности у постојећи и нови стамбени и нестамбени фонд зграда.

Предност енергетски ефикасне градње се огледа у:

- Финансијској уштеди кроз издатке за грејање, хлађење и електричну енергију.
- Угодније и квалитетније становање, те дужи животни век зграде.
- У доприносу заштите околине и смањењу емисије штетних гасова у околину, као и глобалним климатским променама.

2.2. Историја

Нагли развој тржишта, економије, продор и утицај капитала и медија с једне стране и невероватна експанзија градње у нашој земљи последњих неколико година, остављају траг на архитектонским реализацијама које су често упитног квалитета и без енергетског концепта.

Прва кућа која је грађена да би испунила захтев смањене потребе за енергијом је кућа ДТХ "нула енергије". Кућу је пројектовао проф. Вагн Корсгаард 1973. године у Копенхагену.

НАПОМЕНА:

Овај рад је произтекао из мастер рада чији ментор је била проф. др Мирјана Малешев.



Слика 1. – ДТХ кућа "нула енергије" у Копенхагену

2.3. Енергетска ефикасност код нас и у региону

Године 2010. ступила је на снагу нова ЕУ директива о енергетским карактеристикама зграда. Европска унија се обавезала да ће смањити своју годишњу потрошњу примарне енергије за 20 посто до 2020. године. Смањење енергетске потрошње и елиминисање енергетских губитака спадају у главне циљеве енергетске политике Европске уније и стога је ен. ефикасност (ЕЕ) у срцу европске стратегије Европа до 2020. у погледу мудрог, одрживог и свеобухватног раста и преласка на економију која почива на ефикасном коришћењу ресурса.

2.4. Пројектовање према правилнику о енергетској ефикасности

При пројектовању зграда основни принципи физике зграде, према правилнику, су:

- Топлотна заштита
- Топлотна акумулативност и
- Дифузији водене паре

Правилником је у два рачунска нивоа дефинисана топлотна проводљивост објекта. Први ниво подразумева провере појединачних грађевинских конструкција док други ниво посматра конструкцију као целину. Смер простирања топлоте је одређен на основу тога да провођење топлоте увек иде од топлијег ка хладнијем месту (други закон термодинамике). Могућа су три начина преношења топлоте: зрачењем (радијацијом), струјањем (конвекцијом) и спровођењем (кондукцијом).

2.5. Принципи нискоенергетске изградње објеката

Нискоенергетска градња објеката захтева да се још током пројектовања мора водити рачуна о што квалитетнијем постизању и испуњењу следећих битнијих критеријума:

- Квалитетно постављање и формирање објекта у односу на окружење (конфигурација терена, еко системи и постојећи грађевински садржаји у околини)
- Компактан волумен зграде,
- Материјализација и уређење околног припадајућег терена,
- Предвиђање квалитетног распореда зеленила, врста и групација које могу додатно заштитити, односно формирати потребне амбијенталне (еко) целине,

- Избегавање негативних, топлотних мостова и прецизан избор система градње, све грађевинских материјала, као и начина саме уградње,
- Квалитетна термоизолација целог грађевинског омотач, са посебним третирањем и изоловањем све унутрашњих просторија које се ослањају, односно које су усмерене према зонама и просторима који нису грејани, итд.

2.6. Пример изградње нискоенергетске куће код нас

Кућа је саграђена 2010. год. у месту Бачко Градиште, Србија. Грађена је од пасивних елемената (системом термо - блокова).



Слика 2. – Пример нискоенергетске куће код нас

3. СТРУЧНИ ДЕО

3.1 Технички опис

Објекат за који се ради процена стања, а потом и доградња, а за који ће се касније дати и мере санације, представљен је на следећој слици (Слика 3).



Слика 3. - Изглед школског објекта

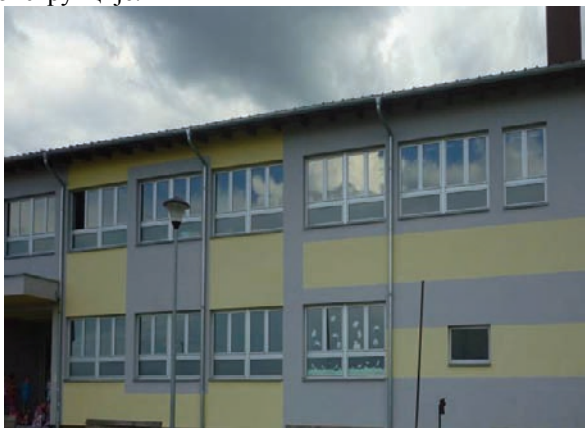
Постојећи објекат је намењен основном образовању, спратности: приземље + један спрат, лоциран је у урбаном насељу Грбавица у Брчком. Просторије у приземљу и на спрату намењене су искључиво за боравак ученика и наставног особља. Спратна висина приземља и спрата је 3.0 m. Вертикална комуникација је остварена једнокраким степеништем. На згради је изведен четвороводни кров, нагиба кровне равни $\alpha = 12^\circ$.

3.2. Конструктивни систем

Конструктивни систем зграде је скелетни без зидова за укрућење. Међусобни растер стубова се разликује у два управна правца. Димензије стубова су 25/25cm а греда 25/28cm, 25/30cm, 25/40cm, 25/48cm, 25/50cm и 25/60cm. Међуспратна таваница је пројектована као плоча носива у једном правцу и ослоњна на ободне греде, димензије плоча су 20 cm и 18 cm. Степениште је једнокрако са плочом дебљине 10cm. Објекат је фундиран на темељним контрагредама 60/50cm, темељним тракама 60/50 cm. Фундирање је извршено на коти – 0.90m.

3.3. Детаљни визуелни преглед

Визуелни преглед је обављен са спољашње и унутрашње стране објекта. На сликама 4 и 5 дат је приказ постојећег стања објекта, где су фотографисани они делови који су били доступни приликом прегледа конструкције.



Слика 4. - Изглед завршне фасаде постојећег објекта



Слика 5. – Изглед противпожарног степеништа

С обзиром да се ради о релативно младом објекту, приликом анализе прегледа елемената конструкције нису уочени значајни дефекти и оштећења на постојећем објекту. Појава прслина и већих оштећења која би озбиљно нарушила интегритет конструкције нису уочена приликом визуелног прегледа. Приликом прегледа пројектно техничке документације утврђено је да у објекту не постоје сеизмичка платна, тако да је неопходно спровести контролу сеизмичке отпорности објекта. Посматрајући конструкцију у целини, доноси се закључак да није угрожена носивост, трајност и употребљивост објекта.

3.4 Енергетска санација

Енергетска санација се односи на увођење нових термоизолационих слојева на спољашњим зидовима, подним плочама и у таванском простору. Дебљина ових слојева је одређена на основу критеријума из Правилника о енергетској ефикасности зграда ("Службени гласник РС" бр. 72/09, 81/09-исправка, 64/10-УС и 24/11).

4. НАДОГРАДЊА

Пројектом је предвиђена доградња од једног спрата на постојећу конструкцију објекта, тако да се добија нови објекта спратности: Пр+2С. Основа додатог спрата је пројектована тако да је по распореду конструкцијских елемената и просторија иста као и претходни спрат. Кровна конструкција је изведена на сличан начин као претходна, задржан је пројектовани четвороводни кров и нагиб $\alpha = 12^\circ$, али је уместо трапезастог лима усвојена тегола као кровни покривач. Конструктивни систем је и даље скелетни, са задржаним димензијама стубова. Међуспратна таваница је „бела таваница“ $d=20\text{cm}$, ослоњена на греде. За вертикалну комуникацију у овом делу зграде изведено је ново степениште. Догређени објекат је променио спратност и висину тако да је неопходно формирати нови модел конструкције и извршити проверу напона у стубовима као и проверу хоризонталног померања врха објекта услед хоризонталних дејстава. Након формирања новог просторног модела, аплицирања постојећег и додатног оптерећења од доградње, спроведена је статичка и динамичка анализа. Дошло се до закључка да су напони у одређеном броју стубова приземља прекорачени. Што се тиче максималног померања врха зграде услед сеизмичког дејства, оно је у допуштеним границама према Правилнику о техничким нормативима за изградњу објекта високоградње у сеизмичким подручјима.

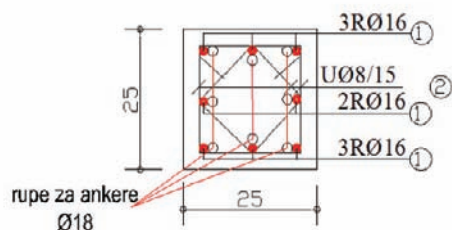
Закључено је да се предвиде конкретне мере санације у виду ојачања одређеног броја стубова приземља у смислу повећања попречног пресека и додавања потребне количине арматуре.

5. ПОВЕЗИВАЊЕ НОВОПРОЈЕКТОВАНЕ И ПОСТОЈЕЋЕ КОНСТРУКЦИЈЕ И ОЈАЧАЊЕ ПОСТОЈЕЋЕ КОНСТРУКЦИЈЕ НЕДОВОЉНЕ НОСИВОСТИ

За новопроектовано стање надоградње потребно је извршити повезивање постојећег и новог дела конструкције. Принцип повезивања састоји се у уграђивању металних анкера у претходно избушене рупе у постојећој армиранобетонској конструкцији. Код оваквог система анкеровања атхезија се остварује помоћу материјала за заливање или ињектирање на бази епоксида. Као анкери у овом случају ће се користити арматурне шипке. Принцип анкеровања састоји се од следећих фаза (слика 6):

- Бушење рупа пречника који је за 2mm већи од пречника арматуре и дубине која одговара дужини сидрења
- Чишћење рупа одговарајућим четкама

- Обеспрашивање помоћу компримованог ваздуха
- Наливање – инјектирање масе у припремљене рупе ради остваривања прионљивости
- Уграђивање анкера
- Постављање пројектоване арматуре елемената конструкције надоградње
- Повезивање анкера и арматуре преклапањем
- Бетонирање новог елемента конструкције у претходно припремљеној оплати

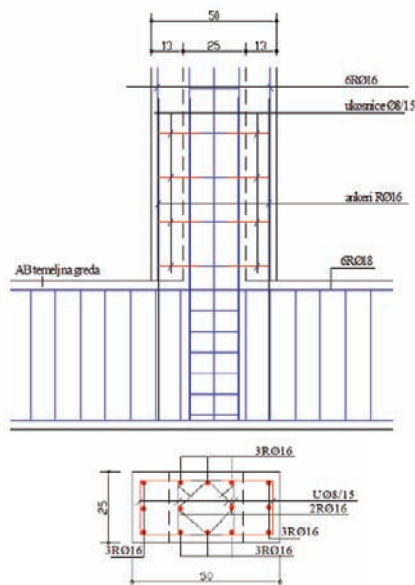


Слика 6. – Положај рупа за анкеровање

Ојачање стубова у приземљу обавиће се на следећи начин: прво је потребно штемовањем уклонити заштитни слој бетона до арматуре у ослоначким зонама постојећег АБ стуба и обавити чишћење преосталог бетона. Пошто је арматура у ослоначким зонама постала видљива може се приступити бушењу рупа за анкере.

Укупна дужина једног анкера је једнака збиру дужине преклапања шипке стуба и дужине сидрења шипке анкера. Санација АБ стубова извршиће се ојачањем основног пресека новим бетонским „плаштом” и уграђивање додатне арматуре одговарајућег пречника према потреби.

Око стубова на којима је потребно извршити ојачање оформиће се АБ омотач одговарајуће дебљине и у њега ће се сместити нова арматура и узенгије (Слика 7). Повезивање постојеће и нове арматуре у АБ „плашту” оствариће се помоћу челичних укосница.



Слика 7. – АБ „плашт” код ојачања стуба

6. ЗАКЉУЧАК

Све већа миграција становништва проузрокована различитим факторима (друштвени, економски, политички, итд.) довела је до потребе за проширењем капацитета стамбених објеката и друштвених установа. Да би се уопште приступило надоградњи неопходна је детаљна анализа и процјена успешности односно неуспешности примене оваквог начина проширења стамбеног простора. У току овог процеса неопходно је препознати и придржавати се дефинисаних фаза које су представљене у овом раду.

7. ЛИТЕРАТУРА

- [1] Збирка Југословенских правилника и стандарда за грађевинске конструкције
- [2] М. Малешев, В. Радоњанин: Скрипта са предавања Услови грађевинске физике, Инжењерска комора Србије, Београд
- [3] Група аутора: Правилник о енергетској ефикасности зграда, Службени гласник РС:061/2011
- [4] Група аутора, *Бетон и армирани бетон*, према БАБ87, књига 1, универзитетска штампа, Београд, 2000.
- [5] Група аутора, *Бетон и армирани бетон*, према БАБ87, књига 2, универзитетска штампа, Београд, 2000.
- [6] З. Брујић, материјал са предавања- *Бетонске конструкције*
- [7] М. Малешев, В. Радоњанин: материјал са предавања- *Праћење. Процена стања и одржавање грађевинских објеката*
- [8] М. Малешев, В. Радоњанин: материјал са предавања- *Материјали и технике санације и заштите грађевинских објеката*
- [9] Група аутора: *Грађевинска физика и материјали*, Београд, 2003.
- [10] Интернет адреса: <http://www.radimpex.co.yu>
Упуство за примјену Tower 6, септембар 2008.
- [11] Интернет адреса: <http://www.core-ns.org>
- [12] <http://srb.sika.com/>
- [13] <http://www.rockwool.com/>
- [14] <http://www.energetska-efikasnost.ba/>
- [15] www.ytong.com/
- [16] www.ecokuce.com
- [17] www.zivotnasredina.com
- [18] www.h-alter.org-ekologija
- [19] www.atma.hr

Кратка биографија:



Александра Ђурић, рођена је у Брчком 1988. године. Мастер рад на Факултету техничких наука из области грађевинарства – модул конструкције – Процјена стања и санације бетонских конструкција, одбранила је 2015. год.

**ANALIZA UTICAJA RIZIKA GRADILIŠTA NA POVREDE NA RADU U
GRAĐEVINARSTVU****ANALYSIS RISKS OF CONSTRUCTION SITE EFFECTS ON INJURY AT WORK IN
CIVIL ENGINEERING**

Sanela Tešić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – GRAĐEVINARSTVO

Kratak sadržaj–Zadatak ovog rada jeste analiza povreda koje su se desile na gradilištu i čiji uzrok je na gradilištu. Za izradu ovog rada korišteni su podaci o prijavljenim povredama za period od 4 godine. Ukupan broj povreda je 719, od kojih se 218 odnosi na gradilište i na organizaciju gradilišta. Analizirani su indirektni i suštinski uzroci povreda, oprema i materijali koji su prouzrokovali povredu, najrizičnije grupe po pitanju starosti, radnog staža, vrste rada i dr. Dobijeni podaci se mogu primjenjivati na sva gradilišta.

Abstract - The subject of this work is analysis injury which are happened on the site and whose cause is on the site. For production of this work have been used data about injury which have reported for period of 4 years. Total number of violations is 719, where 218 of them refer to site and site organization. Indirect and essential causes have been analysed, also equipment and materials which have been caused violation, different groups in terms of age, years of service, type of job and etc. Obtained data can be used for all sites.

Ključne reči: Organizacija i tehnologija građenja, zaštita na radu, povrede na gradilištu, upravljanje rizicima

Key words: Construction management, safety at work, accidents on the construction site, risk management

1. UVOD

Izgradnja građevinskih objekata iziskuje velika materijalna sredstva i angažovanje velikog broja ljudi. S obzirom na angažovanje velikog broja ljudi i postojanje velikog broja potencijalnih opasnosti u građevinarstvu, građevinarstvo je jedna od najrizičnijih delatnosti u pogledu mogućnosti povređivanja i ugrožavanja zdravlja radnika. Povećanju faktora rizika u građevinarstvu doprinosi i sam način zapošljavanja radne snage, uglavnom na određeno vreme, ili sezonski. Plan zaštite na radu sadržan je u Elaboratu o uređenju gradilišta.

Elaborat o uređenju gradilišta obezbeđuje rukovodilac radova uz ovjeru predstavnika investitora, a sadrži: šemu gradilišta (situacioni plan), opis radova i mjere za bezbednost i zdravlje na radu. Sadržaj Elaborata o uređenju gradilišta treba da bude dostupan na gradilištu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Vladimir Mučenski.

2. ZAŠTITA NA RADU**2.1. Zaštita na radu-opšte**

Zaštita na radu je skup tehničkih, zdravstvenih, pravnih, psiholoških, pedagoških i drugih djelatnosti uz pomoć kojih se otkrivaju i otklanjaju opasnosti koje ugrožavaju život i zdravlje osoba na radu i utvrđuju se mere, postupci i pravila kojim bi se otklonile ili smanjile te opasnosti.

Osnovno polazište za organizaciju, uređivanje i sprovođenje zaštite na radu u jednom preduzeću je procena rizika. Pri proceni opasnosti po određenim metodama se utvrđuje postojanje opasnosti, vrsta opasnosti, te opseg opasnosti, a nakon analize opasnosti, utvrđuju se mere za otklanjanje opasnosti i kontrola sprovođenja tih utvrđenih mera.

2.2. Značaj zaštite na radu

Rješavanje problema, odnosno čoveka i radne okoline, jedan je od osnovnih zadataka u prevenciji ozleđa i profesionalnih bolesti.

Za postizanje zadovoljavajućeg nivoa zaštite na radu u građevinarstvu, prije svega treba djelovati na čoveka, odnosno stvoriti pretpostavke da on zna, hoće i može sigurno raditi. Drugi cilj je rešiti problem uređenja radnog prostora, radne okoline u užem i u širem smislu, te sigurnost oruđa za rad i opreme. I treći, ujedno i najvažniji cilj je, postaviti takvu organizaciju u kojoj su nezgode mogu dogoditi samo kao kao posledica delovanja više sile ili protivpravnim delovanjem trećih osoba.

2.3. Povrede na radu

Rad u građevinarstvu je jedna od najrizičnijih delatnosti u pogledu povređivanja i ugrožavanja zdravlja radnika. Srbija je u samom evropskom vrhu po broju povreda na radu, sa godišnjim prosekom od 40 poginulih i oko 1.000 teško povređenih radnika, dok se godišnje evidentira i do 25.000 lakših povreda.

Povreda na radu je svaka povreda izazvana neposrednim i kratkotrajnim, mehaničkim, fizičkim ili hemijskim delovanjem, uzročno vezana za obavljanje poslova na kojima osoba radi. Povreda može biti i uzrokovana naglim promenama položaja tela, njegovim iznenadnim opterećenjem ili drugim promenama fiziološkog stanja organizma. U povredu na radu spada i bolest koja je nastala kao posledica nezgode ili neke više sile tokom rada, kao i povreda nastala na redovnom putu od stana do radnog mesta i obrnuto.

Pored direktnih povreda postoje i tzv.profesionalne bolesti. Profesionalnom bolešću smatra se bolest za koju se dokaže da je posledica delovanja štetnosti u procesu rada i/ili radnoj okolini.

2.3. Zakonsko regulisanje zaštite na radu

Potreba za preduzimanjem određenih aktivnosti radi bezbednosti učesnika u radu javlja se sa prvim oblicima rada, ali se kao organizovana delatnost uspostavlja tek početkom 19. veka.

Za naše okruženje najznačajnija su akta Međunarodne organizacije rada, Svetske zdravstvene organizacije, Evropske unije i drugih evropskih integracija čija je regulativa prihvaćena od strane država članica EU.

Pravo na bezbednost i zaštitu zdravlja, kao osnovno pravo zaposlenih, uređuje se u Republici Srbiji zakonima, podzakonskim aktima, normativima i standardima, kolektivnim ugovorima, aktom poslodavca i ugovorom o radu. Pravo na zaštitu na radu je zagarantovano Ustavom.

3. UPRAVLJANJE RIZICIMA U GRAĐEVINARSTVU

3.1. Osnovne informacije

Rizik na radu se odnosi na mogućnost i na težinu povrede ili obolevanja nastalog kao rezultat izlaganja opasnostima. Rizik je nešto sa čime se suočavamo iz dana u dan. Svaki rizik sa sobom nosi i određenu neizvesnost. Pod neizvesnošću se podrazumeva takva situacija u budućnosti kod koje postoji više varijanti rešenja sa nepoznatom verovatnoćom pojavljivanja.

3.2. Procena rizika

Osnovni cilj procene rizika na radu je zaštita bezbednosti i zdravlja zaposlenih. Procena rizika pomaže u minimalizovanju mogućnosti da zaposleni ili okruženje budu ugroženi tokom aktivnosti u vezi sa procesom rada. Prema zakonima o bezbednosti i zdravlju, svi poslodavci mora da vrše redovnu procenu rizika.

Postoji veliki broj metodologija i alata za procenu rizika. Osnovni dokumenti i standardi koji se koriste za definisanje i procenu rizika u Srbiji jesu Nacionalni pravilnik o proceni rizika, Standard SRPS EN ISO 14121-1:2007-bezbednost mašina-procena rizika na mašinama, Standard OHSAS 18001 i vodič za njegovu primenu, Metoda „Pet koraka“, Preporuka Evropske agencije za procenu rizika iz 1996.godine i Vodič Evropske agencije za BZR za procenu rizika.

3.3. Procena rizika na osnovu preporuke Evropske agencije za procenu rizika

Procena rizika se vrši u pet sledećih koraka:

- Prikupljanje informacija
- Identifikovanje opasnosti
- Procena rizika koji proizilazi iz opasnosti
- Planiranje aktivnosti za otklanjanje ili smanjenje rizika, Pregled procene
- Dokumentovanje procene rizika

4. PLAN ZAŠTITE ZA IZGRADNJU INŽENJERSKIH OBJEKATA

4.1. Uputstvo za izradu plana zaštite

Plan zaštite je potrebno izvršiti u skladu sa podzakonskim aktima Zakona o građenju, a mora uzeti u obzir sve radove koji se izvode.

4.2. Elaborat o uređenju gradilišta

4.2.1. Obezbeđenje gradilišta

Obezbeđenje gradilišta prema okolini propisuje Zakon o građenju, a Plan obezbeđenje gradilišta je sastavni deo Plana zaštite. Planom obezbeđenja se predviđa: označavanje gradilišta, obezbeđenje područja gradilišta, obezbeđenje objekata koji su namenjeni zaštiti okoline, opis bezbednosnih mera u radnom vremenu i van njega i dr. informacije i obaveštenja, zavisno od prirode gradilišta.

4.2.2. Plan tehničkog uređenja gradilišta

Plan uređenja gradilišta izvodi izvođač i mora biti izrađen u skladu sa projektom i sastavni je dio Plana zaštite. Mora da sadrži sve potrebne podatke o komunikacijskim putevima na gradilištu i priključcima gradilišta na javnu infrastrukturu, kao i podatke o skladištima, deponijama, radionicama, kancelarijama za rukovodioce gradilišta, garderobama i prostorima za radnike, te druge podatke koji su važni za siguran i pouzdan rad gradilišta.

Gradilište mora biti sve vreme građenja uređeno tako da je omogućeno nesmetano i bezbedno izvođenje svih radova, da ne postoji opasnost od nastanka povreda i zdravstvenih teškoća radnika i drugih osoba.

5. ANALIZA PODATAKA I PROCENA OPASNOSTI NA GRADILIŠTU

Kako bi se predupredile povrede i svele na najmanju moguću meru, potrebno je izvršiti procenu rizika, a tu procenu je moguće obaviti na osnovu statističkih podataka o povredama koje su se dogodile za istu vrstu poslova. Za procenu rizika u ovom radu korišteni su podaci o broju i težini prijavljenih povreda za period od četiri godine. U posmatranom periodu, broj povreda za visokogradnju je 719, od kojih se 218 odnosi na gradilište i na neadekvatnu organizaciju gradilišta.

5.1. Analiza broja povreda u zavisnosti od starosti radnika

Tabela 1. Broj povreda u zavisnosti od starosti

Starost radnika	18 i 19 god.	od 20 do 24	od 26 do 29	od 30 do 34	od 35 do 39	od 40 do 44	od 45 do 49	od 50 do 54	od 55 do 59	od 60 do 65
Br.povr.	1	14	27	30	26	26	31	39	18	6
Procenat (%)	0,46	6,42	12,38	13,76	11,93	11,93	14,22	17,89	8,26	2,75

5.2. Analiza broja povreda u zavisnosti od školske spreme

Tabela 2. Broj povreda u zavisnosti od školske spreme

Školska sprema	Nepotpuna osnovna škola	Osnovna škola	Srednja škola	Viša škola
Br.povr.	4	94	114	6
Procenat (%)	1,83	43,12	52,29	2,75

5.3. Analiza broja povreda u zavisnosti od radnog staža na poslovima na kojima se povredio

Tabela 3. Broj povreda u zavisnosti od radnog staža

Radni staž	Manje od 1 meseca	1-12 meseci	1-4 god.	5-9 god.	10-19 god.	20-29 god.	30-39 god.
Br. povreda	2	22	48	44	42	39	21
Procenat %	0,92	10,09	22,02	20,18	19,27	17,89	9,63

5.4. Analiza broja povreda u zavisnosti od radnog mesta radnika koji se povredio

Tabela 4. Broj povreda u zavisnosti od radnog mesta

Radno mesto	armirač	betonirac	bravar	čuvar	električar	gipsar	zidar
Br. radnika	6	4	10	2	9	2	21
Procenat (%)	2,75	1,83	4,59	0,92	4,13	0,92	9,63
Radno mesto	limar	monter	stolar	tesar	monter ViK	vozač kamiona	moler-farbar
Br. radnika	2	2	2	33	5	10	2
Procenat (%)	0,92	0,92	0,92	15,14	2,29	4,59	0,92
Radno mesto	Grad. radnik	rukovodilac grad.mašina	Šef gradilišta	Radna mesta gde je došlo do jedne povrede.			
Br. radnika	86	11	3	8			
Procenat (%)	39,45	5,04	1,38	8x0,46=3,68			

5.5. Analiza broja povreda u zavisnosti od meseca u kojem se desila povreda

Tabela 5. Broj povreda u odnosu na mesec

mesec	J	F	M	A	M	J	J	A	S	O	N	D
br.povreda.	5	8	15	14	14	28	21	24	25	19	25	20
Procenat(%)	2,29	3,67	6,88	6,42	6,42	12,84	9,63	11,01	11,47	8,71	11,47	9,17

5.6. Analiza broja povreda u zavisnosti od dana kada se desila povreda

Tabela 6. Broj povreda u odnosu na dan

dan	P	U	S	Č	P	S	N
br. povreda	44	37	51	45	25	15	1
Procenat(%)	20,18	16,97	23,39	20,64	11,47	6,88	0,46

5.7. Analiza broja povreda u zavisnosti od opreme usled čijeg korišćenja je došlo do povrede

Tabela 7. Broj povreda u zavisnosti od opreme

Oprema	Merdev.	Prist. rampa	Radna skela	Rampa za utovar	Razv. ormar	skela	Povrede bez korišćenja opreme
Broj povreda	12	1	6	1	1	5	192
% u odnosu na uk. broj povreda	5,50	0,46	2,75	0,46	0,46	2,29	88,07
% u odnosu na broj povreda usled korišćenja opreme	46,15	3,85	23,08	3,85	3,85	19,23	

5.8. Analiza broja povreda u zavisnosti od materijala koji je prouzrokovao povredu

Tabela 8. Broj povreda u zavisnosti od materijala

materijal	armatura	ekser	ekser u dasci	gredica	kamen	lim
Broj povreda	13	3	40	2	2	3
% u odnosu na uk. broj povreda	5,96	1,38	18,35	0,92	0,92	1,38
% u odnosu na broj povreda usled korišćenja opreme	14,77	3,41	45,45	2,27	2,27	3,41
materijal	staklo	šut	zemlja	Mat.usled čijeg korišćenja je došlo do 1 povrede	Povrede nisu uzrokovane materijalom.	
Broj povreda	2	2	6	15	130	
% u odnosu na uk. broj povreda	0,92	0,92	2,75	6,88	59,63	
% u odnosu na broj povreda usled korišćenja opreme	2,27	2,27	6,82	17,04		

5.9. Analiza veličine povreda u zavisnosti od povređenog dela tela

Tabela 9. Veličina povrede u zavisnosti od povređenog dela tela

	mala	srednja	velika	mala
Glava	6(46,15%)	3(23,08%)	4(30,77%)	13
Oči	/	/	1(100%)	1
Respiratorni sistem	/	3(100%)	/	3
Noge-stopala	28(25,69%)	66(60,55%)	15(13,76%)	109
Ruke-šake	15(44,12%)	10(29,41%)	9(26,47%)	34
Tijelo-koža	2(100%)	/	/	2
Tijelo-torzo	3(16,67%)	6(33,33%)	9(50%)	18
Višestruke povrede	4(10,53%)	14(36,84%)	20(52,63%)	38
Σ	58	102	58	

5.10. Analiza veličine povrede u zavisnosti od radnog staža

Tabela 10. Veličina povrede u zavisnosti od radnog staža

	mala	srednja	velika	Σ
manje od mjesec dana	0	2	0	2
od 1 do 12 mjeseci	7	9	6	22
od 1 do 4 godine	12	27	9	48
od 5 do 9 godina	9	23	12	44
od 10 do 19 godina	10	15	17	42
od 20 do 29 godina	12	19	8	39
od 30 do 39 godina	8	7	6	21
Σ	58	102	58	

5.11. Analiza indirektnih uzroka po vrstama radova

Analizom ulaznih podataka, dobijeni su sledeći rezultati:

- Usled kretanja bez prenosa, za najviše povreda indirektni uzročnik je nečišćenje koridora (36/73=49,31%)
- Usled prenosa, za najviše povreda indirektni uzročnik je nečišćenje koridora (18/35=51,43%)
- Usled tesarskih radova, za najviše povreda indirektni uzročnik je nečišćenje radnog mesta (14/31=45,16%)
- Kod zidarskih radova, za najviše povreda indirektni uzročnik je nečišćenje radnog mjesta (5/11=45,45%)
- Usled rukovanja mehanizacijom, indirektni uzročnik za sve povrede je nečišćenje pristupnog mesta (9/9=100%)
- Usled utovara i istovara, za najviše povreda indirektni uzročnik je nečišćenje pristupnog mesta (6/9=66,66%)
- Kod armiračkih radova, za najviše povreda indirektni uzročnik je nečišćenje radnog mesta (5/8=62,50%)
- Kod betonskih radova za najviše povreda indirektni uzročnik je nečišćenje radnog mesta (4/7=57,14%)
- Kod instalaterskih radova, za najviše povreda indirektni uzročnik je nepravilno postavljanje merdevina (3/7=42,86%)

- Kod zemljanih radova, za najviše povreda indirektni uzročnik je nepravilna obezbijedenost rova ($5/7=71,43\%$)
- Kod bravarskih radova, za najviše povreda indirektni uzročnik je nedovoljno provjetravanje ($3/6=50\%$)

Pored navedenih Analiza, u Master radu se nalaze i Analize koje nisu prikazane u ovom Radu.

6.ZAKLJUČAK

Svaki rad je praćen manjim ili većim opasnostima. Po proceni Međunarodne organizacije rada, 2,2 miliona radnika godišnje umre zbog loših uslova na radnom mestu, odnosno svakih 15 sekundi umre po jedan radnik.

U ovom radu analizirane su povrde koje su se u posmatranom vremenu desile na gradilištu. Za sve osobe koje se nađu na gradilištu, bez obzira da li su radnici preduzeća koje izvodi radove ili su radnici nekih drugih firmi, propisana je obaveza korištenja radne i zaštitne opreme.

Prema Analizi podataka, od 218 povreda koje su se desile na gradilištu, najviše povreda se odnosi na građevinske radnike, 39,45%. Analizom vrste rada u trenutku povrede, došli smo do informacije da se najviše povreda desilo usled kretanja bez prenosa, 33,49%. Ekser u dasci je materijal koji je izazvao najviše povreda, a od opreme, to su merdevine. Ako posmatramo indirektno uzroke, za najviše povreda to je nečišćenje koridora, 27,98%, a od suštinskih uzroka nepostojanje ili loš kvalitet održavanja čistoće gradilišta. Najviše povreda se desilo usled pada na nivo ispod, 33,94% i pada na istom nivou, 32,11%. Što se tiče veličine povreda, najviše je srednjih povreda, 46,79%, a povređenog dela tela, noge-stopala su pretrpele najviše povreda, 50% od ukupnog broja povreda.

Iz svega napred iznetog može se zaključiti da je lista potencijalnih opasnosti na gradilištu velika. Ono što je krucijalno jeste da Investitor i Izvođač radova i lica za bezbednost i zdravlje na radu budu sigurni da nijedan aspekt bezbednosti i zdravlja nije previđen pre nego što posao započne.

7.LITERATURA

- [1] Mučenski. V. (2013), „*Model semikvantitativne procene rizika zaštite na radu za procese izgradnje*“, Novi Sad
- [2] Žderić S. (2010), „*Priručnik za bezbijednost i zdravlje zaposlenih*“
- [3] Pravilnik o zaštiti na radu pri izvođenju građevinskih radova („Sl. glasnik RS“, br.53/97)
- [4] Pravilnik o sadržaju elaborata o uređenju gradilišta („Sl.glasnik RS“, br.31/92)
- [5] Zakon o bezbijednosti i zdravlju na radu („Sl.glasnik RS“, br.101/2005)
- [6] BZR Edukativni centar (2009.), „*Bezbijednost i zdravlje na radu (knjiga 1)*“, Mašinski fakultet u Kragujevcu
- [7] BZR Edukativni centar (2009.), „*Bezbijednost i zdravlje na radu (knjiga 2)*“, Mašinski fakultet u Kragujevcu

Kratka biografija:



Sanela Tešić, rođena u Derventi, 1989. godine. Osnovne studije završila u decembru, 2013. godine, na Fakultetu tehničkih nauka u Novom Sadu, iz oblasti Građevinarstvo, na temu „Projekat tehnologije i organizacije građenja magacina sa poslovnim aneksom, Kač“.

ISPITIVANJE EFEKATA SUDARANJA NA SUSEDNE KONSTRUKCIJE U TOKU ZEMLJOTRESA**AN INVESTIGATION OF THE EFFECTS OF POUNDING TO ADJACENT SEISMICALLY EXCITED BUILDING STRUCTURES**

Dejan Ankić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – GRAĐEVINARSTVO

Kratak sadržaj – U toku zemljotresa može doći do sudaranja između susednih, različitih konstrukcija ako je razmak između njih nedovoljan i ako dođe do njihovog oscilovanja van faze. Sudaranje može dovesti do ozbiljnih oštećenja elemenata, pa i cele konstrukcije. Efekti sudaranja se i dalje istražuju širom sveta, a cilj ovog rada je da analizira jedan manje objašnjen aspekt ove pojave: mogućnost da usled kontakta, povećano opterećenje u stubovima dovede do njihovog sloma.

Abstract – Collision between adjacent non-identical buildings may occur during earthquakes, if the separation between them is insufficient and if they oscillate out of phase. This collision is known as structural pounding. Pounding may cause substantial structural damage of a structure. Impacts of pounding during the major earthquakes around the world are still being examined, and the aim of this thesis is to study one of its' less considered aspects – the possibility of the capacity demand on the columns exposed to pounding becoming so great that they may fail.

Ključne reči: Seizmičko sudaranje, susedne konstrukcije, seizmička analiza, Vremenski zapis zemljotresnog opterećenja.

1. UVOD

Nagli ljudski razvoj, koji je obeležio drugu polovinu XX veka je doveo do jedinstvenog rasta populacije. Potreba za obezbeđenjem dovoljno životnog, poslovnog i drugog prostora, kombinovana sa procesom urbanizacije, je dovela do drastičnog razvoja urbanih gradskih celina. Razvoj gradova je ponekad ličio na stihijski proces, gde su često, usled nedostatka slobodnog građevinskog zemljišta, neodgovarajućeg planiranja ili zbog ekonomske isplativosti, nastajali čitavi blokovi objekata izgrađenih u potpunom ili delimičnom kontaktu. Konstrukcije ovakvih objekata mogu biti posebno ugrožene u toku zemljotresa, jer usled prevelikih spratnih pomeranja može doći do njihovog sudaranja. Ova pojava je u literaturi poznata kao sudaranje konstrukcije (*structural pounding*). Da li će za vreme zemljotresa zaista doći do sudaranja dve susedne konstrukcije, kao i intenziteti udarnih sila i posledice ovih sudara, je uslovljeno brojnim faktorima, a kao glavni su prepoznati razmak između dve konstrukcije, njihove geometrijske i dinamičke karakteristike, kao i jačina

samog zemljotresnog opterećenja kojem su konstrukcije izložene. Većina aspekata sudaranja konstrukcija je do danas razjašnjena, ali oko nekih i dalje ostaju nedoumice. Predmet interesovanja ovog rada je bliže istraživanje i numeričko opisivanje jedne od posledica seizmičkog sudaranja na stabilnost ugrožene konstrukcije, odnosno pojave povećanog opterećenja na nedovoljno otporne stubove, pogotovo kod onih konstrukcija kod kojih prilikom projektovanja nisu važili današnji, strožiji propisi za seizmičko projektovanje.

2. O SUDARANJU

Korišćenjem savremenih tehničkih propisa i dostupnih metoda proračuna grade se konstrukcije koje bez većih problema mogu da izdrže relativno češće i slabije zemljotrese ostankom u svom elastičnom području rada. Za katastrofalne, razorne zemljotrese koji se očekuju jednom u toku eksploatacionog veka konstrukcije, prioritet je da konstrukcija sačuva svoj globalni integritet, dok se dopušta određeni stepen oštećenja sa idejom da se ona kasnije eventualno saniraju. Ipak, u poslednje tri decenije su, prilikom brojnih zemljotresa širom sveta, registrovana neuobičajeno velika oštećenja konstrukcija, koje su ponekad dovodile i do njihovog potpunog kolapsa. Ona su eventualno dovedena u vezu sa pojavom sudaranja susednih konstrukcija. Ova pojava je prvi put primećena i detaljno dokumentovana prilikom zemljotresa koje je 1985. godine pogodio Meksiko Siti.

Meksički zemljotres iz 1985. je imao magnitudu od oko 8.0, sa epicentrom u Srednjeameričkom razlomu, skoro 400 km udaljenog od samog grada. Sa obzirom na ovu relativno veliku udaljenost, razorni efekti koje je imao na Meksiko Siti su iznenadili tadašnju stručnu javnost. Procenom štete i analizom nakon zemljotresa, utvrđen je niz faktora koji su doveli do tolikih razaranja. Pre svega, grad leži na dnu nekadašnjeg jezera, na tlu sastavljenog od mekih glina od vulkanskog pepela i visokim sadržajem vode. Ovakva kombinacija lokalnih geoloških uslova je dovela do toga da seizmički talasi dođu do grada sa neuobičajeno velikom energijom. Visok sadržaj vode u tlu je doveo do pojave likvefakcije i velikih sleganja tla i gubitka stabilnosti temelja. Konačno, utvrđeno je da je među objektima došlo do pojave rezonantnih efekata, kojima su dominantno bile izložene konstrukcije perioda oscilovanja od oko 2.5 s, što je grubo odgovaralo objektima spratnosti šest do petnaest spratova, i kod susednih objekata sa različitim dinamičkim osobinama izazvalo oscilovanja van faze i dovelo do sudaranja kon-

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Đorđe Ladinović, red.prof.

strukcija i znatnih spratnih poprečnih pomeranja [1]. Tako se u preliminarnom izveštaju Rozenbluta i Melija objavljenom naredne godine, tvrdi da se na čak 40% od svih oštećenih konstrukcija javilo sudaranje, a da je kod takvih konstrukcija, sudaranje u 15% slučajeva bilo direktan uzrok potpunog kolapsa [2]. Kasnije je zaključeno da su ove procene ipak bile preterane, ali da je sudaranje zaista bilo uzrok ozbiljnih oštećenja. Kao realniji prikaz stanja se smatra da do sudaranja došlo kod oko 15% objekata koji su pretrpeli ozbiljniju štetu i da je od tih slučajeva, u 20-30% sudaranje direktno izazvalo ozbiljna konstrukcijska oštećenja [3].

Nakon što je posle meksičkog zemljotresa pažnja međunarodne akademske zajednice skrenuta na ovaj problem, sudaranje konstrukcija je praktično redovno beleženo prilikom ozbiljnijih zemljotresa širom sveta. Zaključeno je da su u tom smislu najugroženije konstrukcije objekata zgradarstva u gusto urbanizovanim gradskim celinama, kao i mostovske konstrukcije na nedovoljnom rastojanju.

Savremeni seizmički propisi su prepoznali opasnost od sudaranja konstrukcija, te su kao meru prevencije kod budućih objekata propisali potrebne razmake (*separation gap*) između njih. Ovo se uglavnom svodi na obezbeđivanje tolikog razmaka da pri maksimalnim mogućim spratnim pomeranjima ne dođe do kontakta konstrukcija [4]. Ipak, često pri projektovanju izostaje detaljnije obrađivanje ovog problema, pre svega zbog potrebnog iterativnog proračuna, kao i nedoumica vezanih za obezbeđivanje pravilnog izvođenja ovih spojeva. Takođe, kako je većina postojećih objekata izgrađena pre donošenja ovih propisa, njihove konstrukcije ostaju i dalje ugrožene.

Kao glavni faktori koji mogu dovesti do sudaranja dve konstrukcije su prepoznati nedovoljni razmak između njih, njihove različite dinamičke karakteristike (pre svega krutost) koje će izazvati oscilovanje van faze, i dovoljno veliko seizmičko opterećenje.

U tom smislu, zapisi zemljotresnog opterećenja su podeljeni u tri grupe: na slabe (sa maksimalnim ubrzanjem tla od 0.1 do 0.3g), umerene 0.4-0.6g i jake 0.7-0.9g [5]. Dovoljno velika razlika u krutosti kao neophodan preduslov za izazivanje sudaranja može da bude posledica različite spratnosti dve zgrade, različitih poprečnih preseka elemenata, postojanje npr. zidnih platana za ukrucenje, različitih prostornih rasporeda elemenata i slično.

Razlikuju se dva osnovna slučaja sudaranja, u zavisnosti od spratnih visina susednih konstrukcija. Ako konstrukcije imaju iste spratne visine doći će do udara tavanica jedne o tavanice susedne konstrukcije (*slab-to-slab pounding*). Ako su spratne visine različite, tavanice jedne konstrukcije će udarati u obodne stubove susedne (*slab-to-column pounding*) [6, 7].

Drugi slučaj se generalno smatra opasnijim jer su obodni stubovi tada direktno izloženi udarnim silama koje mogu dovesti do njihovog lokalnog kolapsa. Međutim, i slučaj sudara tavanice o tavanicu može izazvati ozbiljna oštećenja jer će se tavanica pomerati kao kruto telo i posredno na sve stubove na koje se oslanja preneti ista pomeranja, koja se mogu pokazati kao prevelika i izazvati štetu.

3. O ANALIZIRANOM MODELU

U skladu sa prethodno izloženim, predmet ovog rada će biti konstrukcije dva susedna armirano-betonska rama, koji predstavljaju pojednostavljene modele dva stambena objekta, analizirana u ravni, u x-z koordinatnom sistemu. Za analizu je korišćen računarski softver SAP2000. Konstrukcije su izabrane tako da predstavljaju čest primer iz urbanih sredina gde je u postojećoj izgrađenoj sredini došlo do razvoja novog objekta veće spratnosti, ili gde su oba postojeća objekta izgrađena pre usvajanja modernih propisa i čiji elementi možda neće moći da se nose sa pojavom sudaranja. Od opterećenja, konstrukcije su opterećene samo vremenskim zapisima ubrzanja tla, nastalim pri pojedinim zemljotresima (Tabela 1), skaliranih na projektno ubrzanje ispitivane konstrukcije. Nakon proračuna, zabeležena je vremenska promena presečnih sila u elementima rama A, koji su potom detaljnije prikazani za izabrane elemente od interesa.

Tabela 1: Korišćeni zapisi zemljotresnog opterećenja

Naziv zemljotresa	Datum	Maksimalno ubrzanje (g)	Dužina trajanja (s)
Imperial Valley, SAD	19/05/40	0.3124	40.0
Kalamata, Grčka	13/09/86	0.2484	29.25
Crna Gora, Jugoslavija	15/04/79	0.3736	47.84

Predmet analize je desetospratni ram u ravni, ram A, dizajniran u skladu sa Eurokodovima 2 i 8, na srednji nivo duktiliteta i projektno ubrzanje od 0.25g [8]. Ukupna širina rama je 10 m (pruža se preko tri polja dužine 6+4+6 m, i četiri ose A, B, C i D) a visina 30 m (10 etaža spratne visine 3 m). Svi stubovi su kvadratnog poprečnog preseka, čije dimenzije variraju kroz objekat – na prvih šest etaža iznose 40/40 cm za obodne, odnosno 50/50 cm za unutrašnje stubove. Na 7. i 8. etaži obodni stubovi su dimenzija 35/35, a unutrašnji 45/45 cm. Na 9. i 10. etaži obodni stubovi su dimenzija 30/30, a unutrašnji 40/40 cm. Sve grede su pravougaonog poprečnog preseka, na prve četiri etaže dimenzija 25/85 cm, na 5. i 6. etaži 25/80 cm, a na poslednje četiri 20/70 cm.

Pored analiziranog rama A je postavljen petospratni ram B, na rastojanju od 2 m. Konstrukcija rama B je odabrana tako da bude niža od rama A i da ima elemente manje krutosti koji će rezultovati u značajno različitim periodima oscilovanja i oscilovanju van faze u toku zemljotresa. Spratne visine rama B su takođe 3 m, pa će doći do slučaja sudaranja tavanice o tavanicu. Ukupna visina rama B je tako 15 m, a ukupna širina 18 m, jer se ram pruža preko tri polja širine 6 m i definisan je preko četiri vertikalne ose E, F, G, H. Svi stubovi drugog rama su kvadratnog poprečnog preseka i svi su dimenzija 50/50 cm. Sve grede su pravougaonog preseka i sve su dimenzija 30/50 cm. Masa obe konstrukcije je uniformno raspoređena duž elemenata i definisana je njihovim dimenzijama i težinom materijala, u ovom slučaju armiranog betona C20/25, čija je težina definisana kao 25 kN/m³. Prilikom analize, od interesa nije bilo dimenzionisanje elemenata, pa armiranju nije posvećena detaljnija pažnja, već su svi elementi armirani standardnom početnom armaturom od 8RØ9, definisanom u programu. Ovo je

opravdano činjenicom da je glavni cilj analize određivanje presečnih sila u elementima kao i njihovih promena u zavisnosti od razmaka dve konstrukcije. U skladu sa tim, smatrano je da uticaj količine i rasporeda armature u elementima neće značajnije uticati na raspodelu krutosti i time na vrednosti dobijenih sila, jer su od interesa njihove relativne promene, a ne apsolutne vrednosti.

Konstrukcije ramova A i B su u nivoima pet tavaničnih greda spojene *gep* elementima, koji povezuju ivične čvorove rama A sa odgovarajućim ivičnim čvorovima rama B. Oni u fizičkom smislu predstavljaju opruge, koje se aktiviraju samo pri aksijalnoj sili pritiska (odnosno pri kontaktu dve konstrukcije usled sudara – kad nema kontakta opruge su neaktivne). Definisani su svojom krutošću (koja je određena kao 200,000 N/m, odgovara realnoj situaciji nešto fleksibilnijeg diletacionog spoja) i otvorom pomoću koga se lako može definisati i po potrebi varirati rastojanje između ramova, sa obzirom da je ono prilikom matematičkog rešavanja problema u programu definisano upravo preko ovog otvora *gep* elemenata. U ovim elementima će prilikom sudaranja dve konstrukcije doći do razvijanja udarnih sila koje će biti registrovane, i koje će svoju energiju predati elementima u kontaktu, što može dovesti do različitih lokalnih i globalnih oštećenja.



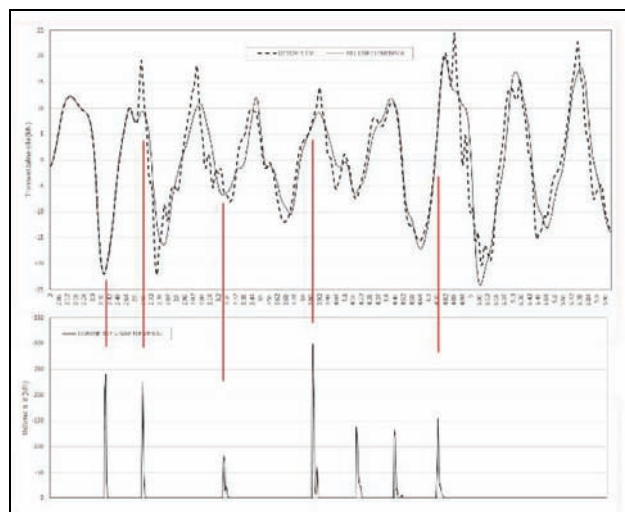
Slika 1: Proračunski model konstrukcija, sa obeleženim i detaljnije analiziranim stubovima

Nakon modelovanja konstrukcija u računarskom programu, na osnovu ulaznih podataka u vidu vremenskog zapisa ubrzanja tla u toku zemljotresa, sprovedena je brza nelinearna dinamička analiza (*fast non-linear dynamic time-history analysis*), koja omogućava prikaz vremenske promene dobijenih rezultata. Otvori *gep* elementa su varirani u određenim koracima da bi se dobila funkcionalna veza promene presečnih sila u elementima sa razmakom između konstrukcija.

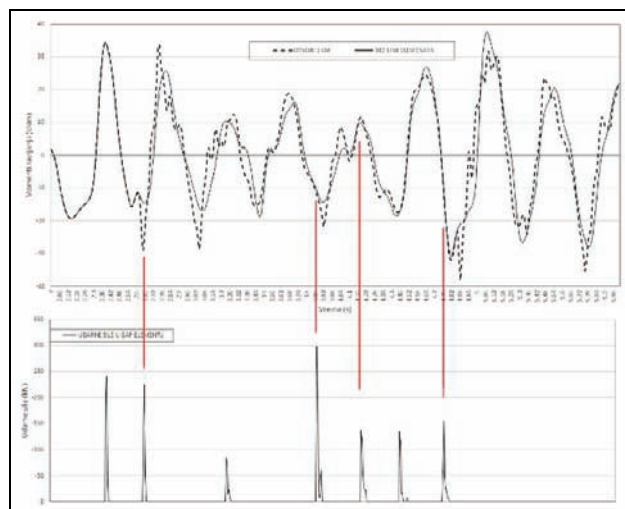
4. ANALIZA REZULTATA

Prvo su prikazani rezultati vremenske promene transversalne sile i momenata savijanja za obodni stub levog, višeg rama A koji se nalazi u osi D na 5. etaži, oznake S-D5. Upoređena su dva osnovna slučaja: kada je otvor dovoljno velik da ne dođe do interakcije dve konstrukcije (0.028 m, odnosno matematički model bez *gep* elemenata), i kada je razmak između konstrukcija jednak 0.01 m (35% kritičnog potrebnog razmaka) koji će izazvati sudaranje. Kao opterećenje je korišćen zapis Imperial Valley zemljotresa, kao najčešće korišćenog u sličnim sprovedenim analizama. Presečne sile su određene za kraj stuba, tj. na njegovoj visini od 3 m. Prikazan je samo period vremena od 2. do 8. sekunde trajanja zapisa, jer se u tom periodu realizuju prvi i maksimalni kontakti, a potom zemljotresno dejstvo po intenzitetu opada i izaziva

male ili nikakve udarne sile. Zabeleženo je da prilikom sudaranja dve konstrukcije dolazi do razvijanja značajnih udarnih sila, maksimalne zabeležene vrednosti od približno 300 kN. Pokazano je da udarne sile imaju znatne poremećajne efekte na tokove promena presečnih sila, na koje u nekim slučajevima mogu delovati blago povoljno, ali najčešće izrazito negativno. Tako je zabeleženo maksimalno povećanje transversalne sile od 113% (sa 9.1 na 19.4 kN) u trenutku 2.63 s. U pojedinim trenucima dolazi do povećanja ove sile u rasponu od 60-90%. Udarne sile slične efekte imaju i na promene momenata savijanja, pa je tako registrovano maksimalno povećanje ove veličine od 102% (sa 16.9 na 34.1 kNm) u trenutku 2.75 s, slike 2 i 3. Sa obzirom na prilično pojednostavljen model konstrukcija, promena apsolutnih vrednosti presečnih sila je od manje važnosti, već je interesantna njihova relativna promena.



Slika 2: Promena transversalne sile u stubu S-D5 i ostvarene udarne sile nastale prilikom kontakta



Slika 3: Promena momenata savijanja u stubu S-D5 upoređeni za slučajeve sudaranja/bez sudaranja i ostvarene udarne sile nastale prilikom kontakta

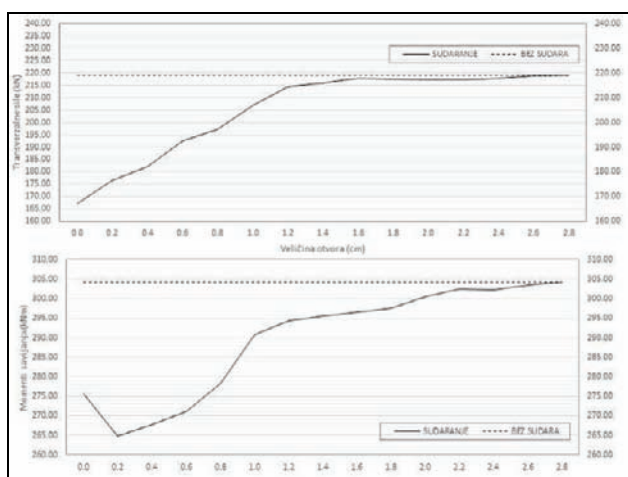
Za kraj su prikazani rezultati koji za cilj imaju objašnjavaње prostornog uticaja sudaranja i udarnih sila na dinamiku promene presečnih sila u stubovima.

Za razliku od prethodnih rezultata koji ovu interakciju posmatraju na lokalnom nivou pojedinih trenutaka

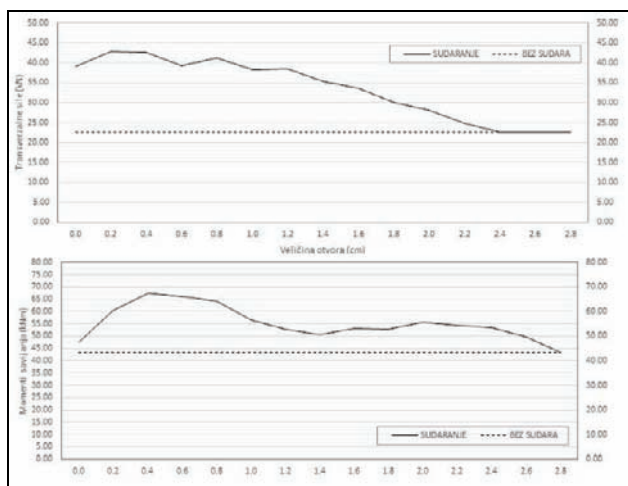
vremena, ovde su za svaku vrednost otvora registrovane samo maksimalne ostvarene vrednosti sila, tj. posmatraju se opšti, globalni efekti sudaranja.

Vrednosti otvora gep elemenata su ovom prilikom varirane malim korakom od 0.2 cm, sve do vrednosti od 2.8 cm, kada do kontakta neće doći, da bi se formirala što tačnija veza promene presečnih sila. Prikazani rezultati se odnose na unutrašnje stubove u osi C, na prvoj i poslednjoj etaži. Registrovano je da kod stuba prve etaže sudaranje ima pozitivne efekte jer dovodi do određenog smanjenja presečnih sila, maksimalne registrovane vrednosti od 20% za transverzalne sile, odnosno 13% za momente savijanja (sl. 4).

Ovi pozitivni efekti su primećeni i u drugim istraživačkim radovima i objašnjavaju se smanjenim spratnim poprečnim pomeranjima nižih tavanica. Oni se, međutim, po visini anuliraju, pa na poslednjoj etaži prelaze u znatne negativne efekte, koji povećavaju transv. sile za maksimalno registrovanih 47%, a momente savijanja za 37% (sl. 5).



Slika 4: Zavisnost promene presečnih sila u stubu S-C1 u funkciji razmaka između konstrukcija. Registrovane promene su povoljnog dejstva.



Slika 5: Zavisnost promene presečnih sila u stubu S-C10 u funkciji razmaka između konstrukcija. Registrovane promene su negativnog dejstva.

5. ZAKLJUČAK

Modelovane su konstrukcije dva rama odabrane po ugledu na realne situacije iz urbanih sredina. Izazvano je njihovo sudaranje primenom zemljotresnog opterećenja. Analizom rezultata je pokazano da preopterećenje postojećih, slabije armiranih stubova ispitivane konstrukcije predstavlja realnu opasnost, jer na lokalnom nivou može da dovede do povećanja presečnih sila i do 100%, a na globalnom nivou i do 50%, te tako značajno utiče na količinu i raspored potrebne armature, ili potrebe za korišćenjem drugih mera za izbegavanje ozbiljnih oštećenja koja mogu da nastanu. Za potpuno rasvetljavanje ispitivanog fenomena je neophodan dalji rad. On pre svega treba da bude usmeren na tačnije određivanje promene presečnih sila korišćenjem komplikovanijih modela i statističku obradu većeg broja dobijenih rezultata. Na taj način bi zaključci bili relevantniji za eventualnu kasniju inkorporaciju u propise i preporuke projektantima seizmičkih konstrukcija.

6. LITERATURA

- [1] D. Aničić, P. Fajfar, B. Petrović, A. Szavits – Nossan, M. Tomažević, "Zemljotresno inženjerstvo: Visokogradnja", Beograd, *Grđevinska knjiga*, 1990.
- [2] E. Rosenblueth, R. Meli, "The 1985 earthquake: causes and effects in Mexico City", *Concrete International*, Vol. 8ACI, pp 23-36, 1986.
- [3] S.A. Anagnostopoulos, C. Karamaneas, "Use of collision shear walls to minimize seismic separation and to protect adjacent buildings from collapse due to earthquake-induced pounding", *Earthquake Engineering & Structural Dynamics*, 37 (12), pp 1371-1388, 2008.
- [4] BS EN1998-1, 4.4.7.2, pp 75.
- [5] S.A. Raheem, "Seismic Pounding between Adjacent Building Structures", *Electronic Journal of Structural Engineering*, Vol 6, pp 66-74, 2006.
- [6] R. Jankowski, "Earthquake-induced pounding between equal heights buildings with substantially different dynamic properties", *Engineering Structures*, Vol 30, pp 2818-2829, 2008.
- [7] C. Karayannis, M.J. Favvata, "Earthquake-induced interaction between adjacent reinforced concrete structures with non-equal heights", *Earthquake Engineering & Structural Dynamics*, 34 (1), pp 1-20, 2005.
- [8] A.J. Kappos, G.G. Penelis, "Earthquake-resistant Concrete Structures", *CRC Press*, 1996.

Kratka biografija:



Dejan Ankić je rođen u Novom Sadu 1990. god. Master rad na FTN iz oblasti Građevinarstva – Seizmička analiza konstrukcija je odbranio 2015. god. U toku studija je u dva navrata bio stipendista Erasmus programa i studirao na Univerzitetu u Limericku, Irska i na City Univerzitetu, London, UK.

ЗАМЕНА РАВНОГ КРОВА КОСИМ КРОВОМ НА ПОСЛОВНОМ ОБЈЕКТУ РАДИ ДОБИЈАЊА НОВОГ КОРИСНОГ ПРОСТОРА**THE REPLACEMENT OF THE FLAT ROOF WITH PITCHED ROOF ON THE BUSINESS BUILDING IN ORDER TO OBTAIN NEW USEABLE AREA**

Моника Сантоши Томић, Факултет техничких наука, Нови Сад

Област – ГРАЂЕВИНАРСТВО

Кратак садржај – У овом раду приказана је надоградња пословног објекта у Сремској Каменици. Први део овог рада представља детаљан технички опис постојеће конструкције. Након извршеног визуелног прегледа конструкције, закључено је да нису угрожени носивост стабилност и трајност конструкције. Надоградња објекта је вршена ради повећања корисног простора. Као основни материјал дограђеног дела изабран је челик. Потом је формиран просторни модел (реконструисаног објекта), спроведена је статичка и динамичка анализа са провером напона у стубовима и тлу. На основу тих података закључено је да након надоградње није потребно вршити измене постојеће конструкције.

Abstract – This paper presents an overstructure project of office building in Sremska Kamenica. The first part of this paper presents a detailed technical description of the existing structure. After the visual inspection of the structure, it was concluded that they were not threatened load stability and durability. Upgrading the building was done in order to increase the useful space. Steel was chosen as the base material of overstructure. After that a spatial model of the reconstructed object is formed static and dynamic analysis to check stresses in columns and in soil was performed. Based on these results after the overstructure is not necessary to make any changes to the existing structure is concluded.

Кључне речи: надоградња, челична конструкција,

1. УВОД

Рад се састоји од две међусобно независне целине, теоријског и стручног дела. Теоријски део рада представља истраживачки део са темом „Нове и старе ознаке за грађевински челик”, а стручни део је везан за надоградњу пословног објекта у Сремској Каменици.

НАПОМЕНА:

Овај рад је проистекао из мастер рада чији ментор је била проф. др Мирјана Малешев.

2.1. ТЕОРИЈСКИ ДЕО – СТАРЕ И НОВЕ ОЗНАКЕ ЗА ГРАЂЕВИНСКИ ЧЕЛИК**2.2. Увод**

Због постојања и употребе великог броја материјала, а како би се олакшало њихово означавање уведени су стандарди. Стандарди су група правила и препорука који могу бити прописани од стране група држава (међународни), затим стандарди који важе у одређеној држави (национални) или интерни (у оквиру једног предузећа).

ISO - International Organization for Standardization (Интернационална организација за стандардизацију) је највећа светска организација за развој и издавање Интернационалних Стандарда. Представља мрежу националних институција за стандардизацију у 160 земаља, са седиштем у Женеви, Швајцарска.

2.3. Историја

Први национални стандард увели су Енглези 1901. године са почетном ознаком **BS**, након Енглеза и друге државе су почеле да уводе своје стандарде: Француска **NF**, Немачка **DIN**, Сједињене Америчке Државе **ASTM**, **EN** европски стандарди итд.

Код нас је први национални стандард је био **JUS**, уведен 1936. године и важио је све до 2003. године када је преименован у **SRPS**, а када су у потпуности прихваћени међународни стандарди са или без превода, ознака наших стандарда је **SRPS ISO**, **SRPS EN** итд.

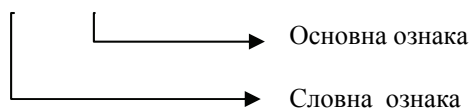
2.4. Материјали металних конструкција

Конструкциони челици спадају у групу угљеничних челика који садрже мање од 0,8% угљеника и називају се још и подетоктоидни челици. Према фазном дијаграму челика при спором хлађењу у овим челицима се формира феритно-перлитна кристална структура. Челици са овом структуром карактерише мања затезна чврстоћа и тврдоћа, као и изразита магнетичности. Поред угљеничних (нелегираних), конструкциони челици могу бити и легирани и они поседују већу затезну чврстоћу и тврдоћу од нелегираних и углавном се користе за одговорније конструкције.

2.5. Означавање челика по JUS стандарду (старо означавање)

Према JUS стандарду JUS C.B9.020:1989 предвиђено је било следеће означавање челика:

Č. XXXX. XX → Допунска ознака



На првом месту се налазила словна ознака која означава врсту материјала. За словну ознаку узимано је прво слово у називу материјала **Челик – Č**.

После словне ознаке иде тачка па следи основна ознака у облику четири броја, након које иде тачка и ако је потребно за дефинисање материјала, следила су још два допунска броја.

Челици према свом хемијском саставу су подељени у две групе:

- челици са негарантованим хемијским саставом и
- челици са гарантованим хемијским саставом, који могу бити угљенични или легирани.

2.6. Систем означавања према SRPS EN 10027-1:2005 (основна ознака)

Ознаке су сврстане у две групе:

1. група челици који се означавају на основу употребе и мханичких или физичких особина,
2. група челици који се означавају на основу хемијског састава, који се дели у четири подгрупе.

2.7 Систем означавања челика према SRPS CR 10260:2005; - додатне ознаке

Додатне ознаке су допуна основној ознаци челика, састоје се из два дела додатне ознаке 1, додатне ознаке 2 и допунске ознаке. Допунска ознака одваја се од додатних ознака знаком „+“. Додатна ознака 2 може да се користи једино ако се користи додатна ознака 1.

3. СТРУЧНИ ДЕО

3.1 Технички опис конструкције

Објект се налази у Улици Војводе Путника бб у Сремској Каменици, број парцеле 4259/1 к.о.

Спратност: Су+Пр+2

Основа објекта: правилна 5x7,3 / 3x8,4m

У сутерену објекта смештен је гаражни простор, простор за истовар и складиштење робе, и склониште. У нивоу сутерена се налази рампа, која обезбеђује пролаз возила са улице до складишта и гаражног простора. Северна страна објекта је делимично укопана до коте ±0,00m.

Приземље је пројектовано као простор за маркет, без преградних зидова, док су типски спратови предвиђени као канцеларијски простори, међусобно одвојени преградним зидовима.

Вертикална комуникација је обезбеђена помоћу двокраког степеништа и лифта.

Кров је решен као раван кров са падом од 1%. Машинска кућица, лифт окно и степениште су покривени равном челичном решеткастом конструкцијом.



Слика 1. Изглед фасаде објекта

3.2. Конструктивни систем

Конструктивни систем зграде је скелетни. Стубови су правоугаоног растерана различитим размацима. Сви стубови су квадратног попречног пресека димензија 60x60 cm.

Фундирање објекта је вршено на печуркастој армирано-бетонској темељној плочи дебљине 40cm са капителима висине 30cm и површине 200x200cm.

Зидови сутерена су армирано-бетонски и то ободни зидови дебљине 30cm, зидови лифт окна и техничког канала 20cm и зидови склоништа 40cm.

Склониште је степена заштите 100 kPa надпритиска. Склониште је пројектовано и прорачунато према Правилнику за склоништа.

Међуспратна конструкција сутерена је армирано-бетонска печуркаста таваница дебљине 30cm са капителима дебљине 20cm површине 200x200cm. Међуспратна плоча склоништа је дебљине 40 cm и ослања се на зидове.

Међуспратне плоче над приземљем, I и II спратом, су пуне армирано-бетонске плоче дебљине 25cm, директно ослоњене на стубове.

Степениште је двокрако, и састоји се од косих степенишних плоча и подеста дебљине 15cm.

3.3. Процена стања

Приликом прегледа елемената конструкције, значајнији дефекти и оштећења нису уочени на постојећем објекту. Унутрашњост објекта је омалтерисана тако да није могуће визуелним прегледом уочити постојање дефеката (бетонска гнезда, мали заштитни слој итд) и имперфекција који би умањили трајност конструкције. Будући да је реч о објекту који није стар, већа оштећења нису ни очекивана.

Посматрајући конструкцију у целини, на основу визуелног прегледа, доноси се закључак да није угрожена носивост, трајност, употребљивост и функционалност постојећег објекта.

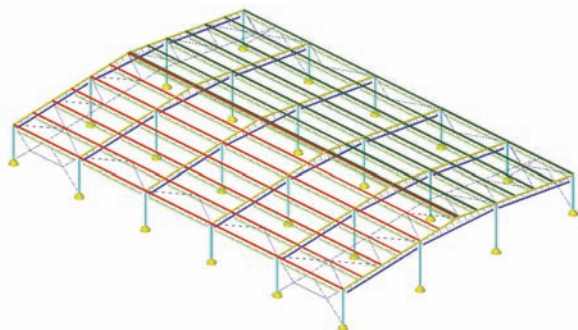
3.4. Измене постојеће конструкције

За постојећи пословни објект, потребно је извести надоградњу једног спрата, који је пројектован тако да је по распореду конструкцијских елемената и просторија исти као и типски спратови.

4. ПРОРАЧУН И МОДЕЛИРАЊЕ НАКОН ИЗМЕНЕ КОНСТРУКЦИЈЕ

Нова кровна конструкција пројектована је као челична решеткаста конструкција директно ослоњена на челичне стубове који по положају одговарају стубовима постојеће конструкције. Све решетке су система просте греде. Рожњаче су типа „R“, док су главни носачи решетке пројектоване од кутијастих профила. Пројектом су предвиђена и додатна укрућења-спрегови по ободу кровне равни и у крајњим пољима у подужном правцу.

Нагиб кровне равни је 10%. Као кровни покривач изабрани су челични сендвич панели од профилисаног лима и испуном од минералне вуне дебљине 80mm. Са доње стране кровне конструкције предвиђено је постављање спуштеног плафона од гипсаних плоча са пратећом металном потконструкцијом која се качи на доњи појас рожњача, према упутству произвођача.



Слика 2. Изглед модела кровне конструкције

4.1 Статички прорачун

Урађен је прво статички прорачун и димензионисање кровне конструкције, а затим је на постојећу конструкцију нането оптерећење од новопроектване кровне конструкције и извршена је контрола напона.

Конструкција је моделирана просторно у програмском пакету Tower 6.0. Веза објекта и подлоге је моделирана помоћу еластичних опруга по Винклеровом моделу. Статички и динамички прорачун спроведен је на моделу код кога су комбиновани линијски и површински елементи. Сви елементи постојеће конструкције су армирани ребрастом арматуром RA 400/500. Марка бетона свих носећих елемената је MB 40.

4.2 Начин аплицирања оптерећења

При уношењу оптерећења од сопствене тежине коришћена је опција Tower-а 6.0 у којој сам генерише сопствену тежину појединих елемената у зависности од припадајућих запремина елемената и врсте материјала елемената. На конструкцију је затим наношено додатно стално оптерећење у вредностима које су дате у поглављу 1.5.3. Оптерећење од фасадних и преградних зидова нането је као линијско, узимајући у обзир и учешће отвора у зидовима.

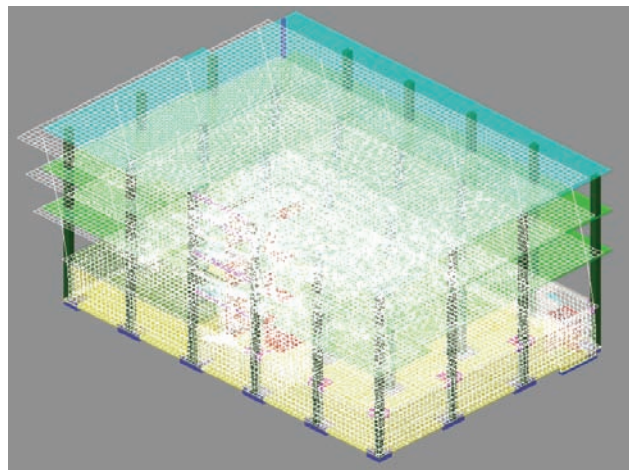
Корисно оптерећење је нането као јединствено површинско оптерећење сагласно стандарду ЈУС У.Ц7.121 (Правилник бр. 07/05-93/131 од 1988-06-03; Службени лист СФРЈ, бр 49/88.

Оптерећење снегом је нането на кровну конструкцију као површинско, а затим је конвертовано у линијско, које делује дуж рожњача. Реакције тачкастих ослонаца које се налазе испод стубова, пренесене су на конструкцију зграде у виду концентрисаног оптерећења.

Оптерећење ветром је нането у виду два различита случаја оптерећења, која се међусобно искључују. Нането је као површинско оптерећење по спољашњим површинама зграде, а затим је конвертовано у линијско које делује на стубове и греде. Оптерећење ветром на кров, нането је као и оптерећење снегом. Одговарајуће сеизмичко оптерећење Tower 6.0 је сам генерисао, након што је урађена модална анализа.

4.3 Динамички прорачун

За динамички прорачун конструкције коришћена је опција програмског пакета Tower 6.0 за модалну анализу. Модална анализа служи за прорачун својствених вредности и облика осциловања конструкције. Овако добијени резултати се касније користе као улазни подаци за сеизмички прорачун конструкције. Да би модална анализа могла бити извршена потребно је формирати модел (дефинише крутост конструкције) и мрежу коначних елемената. Сваком чвору мреже коначних елемената додељује се припадајућа маса, у коју је поред сопствене тежине укључено и додатно стално оптерећење, снег, и половина корисног (претпоставка је да у току деловања земљотреса неће деловати комплетно корисно оптерећење). При прорачуну су биле спречене вертикалне осцилације.



Слика 3. Први тон осциловања конструкције

4.4. Контрола напона у стубовима од експлоатационог оптерећења

Када је извршена замена равнoг крова косим, конструкција постојећег објекта је оптерећена допунским оптерећењем. Такође, висина објекта се повећала што за последицу има повећање сеизмичких сила. Све ове промене обухваћене су приликом формирања новог модела конструкције за постојеће и дограђено стање. Спроведена је нова статичка и динамичка анализа, где су добијене вредности нормалних напона контролисане у складу са допуштеним напонима које прописује Правилник за асеизмичко пројектовање зграда.

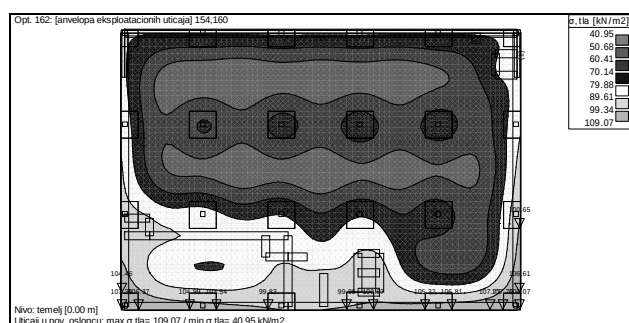
Према Правилнику, а ради обезбеђења дуктилности пресека, напони у стубовима од тоталног експлоатационог оптерећења ограничавају се на 35% чврстоће бетонске призме (0,7 МВ).

На основу добијених резултата се може констатовати да су максимални напони у стубу мањи од допушеног напона, те није потребно вршити ојачање стубова.

Напони у свим стубовима су у дозвољеним границама.

4.5 Контрола напона у тлу

На основу добијених резултата се може констатовати да су максимални напони у тлу мањи од допушеног напона.



Слика 4. Приказ нормалних напона у тлу

6. ЗАКЉУЧАК

Средства која се константно улажу за санацију равних кровова постојећих стамбених и пословних објеката су изузетно велика. Досадашњи начин санације, и поред утрошка великих средстава није дао очекиване резултате, а уједно не представља трајно решење ни у техничком ни у економском погледу. Због тога се сматра економски и технички оправданим замена постојећих равних кровова косим. На овај начин могуће је битно повећати постојећу стамбену и пословну површину изградњом једне или више етажа изнад равних кровова зависно од носивости постојећег објекта. У том случају надограђени део заједно са постојећим објектом мора представљати јединствену грађевинску целину и бити у сагласности са прописима о грађевинском земљишту, планирању и уређењу простора.

7. ЛИТЕРАТУРА

- [1] ЗБИРКА ЈУГОСЛОВЕНСКИХ ПРАВИЛНИКА И СТАНДАРДА ЗА ГРАЂЕВИНСКЕ КОНСТРУКЦИЈЕ:

Југословенски стандард са обавезном применом од 1988-стална оптерећења грађевинских конструкција (JUS U.C7.123).

Југословенски стандард са обавезном применом од 1988– корисна оптерећења стамбених и јавних зграда (JUS U.C7.121).

Југословенски стандард са обавезном применом од 1992– оптерећење ветром (JUS U.C7.110-112).

ПРАВИЛНИК О ТЕХНИЧКИМ НОРМАТИВИМА ЗА ИЗГРАДЊУ ОБЈЕКТА ВИСОКОГРАДЊЕ У СЕИЗМИЧКИМ ПОДРУЧЈИМА.

- [2] Група аутора: БЕТОН И АРМИРАНИ БЕТОН према БАБ 87, књига 1, универзитетска штампа, Београд, 2000.
- [3] Група аутора: БЕТОН И АРМИРАНИ БЕТОН према БАБ 87, књига 2, универзитетска штампа, Београд, 2000.
- [4] З. Брујић: МАТЕРИЈАЛ СА ПРЕДАВАЊА – ВИШЕСПРАТНЕ ЗГРАДЕ
- [5] М. Малешев, В. Радоњанин: МАТЕРИЈАЛ СА ПРЕДАВАЊА – ПРАЋЕЊЕ, ПРОЦЕНА СТАЊА И ОДРЖАВАЊЕ ГРАЂЕВИНСКИХ ОБЈЕКТА
- [6] М. Малешев, В. Радоњанин: МАТЕРИЈАЛ СА ПРЕДАВАЊА – МАТЕРИЈАЛИ И ТЕХНИКЕ САНАЦИЈЕ И ЗАШТИТЕ ГРАЂЕВИНСКИХ ОБЈЕКТА
- [7] Д. Буђевац: МЕТАЛНЕ КОНСТРУКЦИЈЕ У ЗГРАДАРСТВУ, Београд 2009.,
- [8] Б. Зарић, Д. Буђевац, Б. Стипанић: ЧЕЛИЧНЕ КОНСТРУКЦИЈЕ У ГРАЂЕВИНАРСТВУ, Београд, 2004
- [9] ГЛИГОРИЋ, Радојка, Машински елементи [Електронски извор] / Радојка Глигорић- Нови Сад : Пољопривредни факултет, 2015. - (Едиција Основни уџбеник) Начин приступа (УРЛ): пољ.унс.ац.рс/уџбеници/. - са насловног екрана. - Опис заснован на стању на дан: 16.04.2015. - Библиографија. ИСБН 978-86-7520-314-8
- [10] Металургија 01/22/2014 <https://ironlady003.wordpress.com/2014/01/22/celici/>
- [11] Институт за стандардизацију Србије 2010 Institute for Standardization of Serbia <http://www.iss.rs/>
- [12] www.radimpex.co.yu- упутство за примену Tower 6, septembar 2008
- [13] www.knaufinsulation.rs
- [14] www.ytong.com
- [15] <http://www.argus-eng.co.rs/>

Кратка биографија:



Моника Сантоши Томић, рођена је у Новом Саду 1988. године. Мастер рад на Факултету техничких наука из области грађевинарство – Модул конструкције – Процена стања и санација конструкција, одбранила је у 2015. год.

PROCENA STANJA I SANACIJA VIŠESPRATNE ZGRADE U HOPOVSKOJ ULICI U NOVOM SADU NAKON DOGRADNJE**PROJECT OF ASSESSMENT AND REPARATION OF MULTY-STOREY RESIDENTIAL BUILDING IN NOVI SAD**

Исидора Сремачки, Факултет техничких наука, Нови Сад

Област – ГРАЂЕВИНАРСТВО

Кратак садржај – Садржај овог рада огледа се у детаљном прегледу конструкције у виду процене стања и процењивању потребе за санацијом, као и разматрање могућности о доградњи једне етаже на постојећу конструкцију објекта. Објекат је спратности Су+П+2+Пк и налази се у Новом саду у улици Хоповска бр. 2. Рад се састоји из две целине, теоријског и практичног дела. Теоријски део се односи на потребе рационализовања коришћења енергије како код стамбених тако и код пословних објеката и на трајност АБ конструкција кроз класе изложености.

Практични део овог рада говори о основном подацима предметног објекта након чега се детаљно описује стање објекта и разматра потреба за санацијом. С обзиром да се предвиђа доградња, морају се додатно испитати могућности извођења таквог захвата у погледу носивости, стабилности и стања саме конструкције да поднесе ново оптерећење.

Услед недостатка одговарајуће пројектне документације, ради увида у постојећу носивост елемената конструкције, урађен је статички прорачун и димензионисање свих делова конструкције, како основног, тако и дограђеног објекта и направљен просторни модел због анализирања свих утицаја.

Предложене су мере санације оних елемената носеће конструкције за које се показало да немају адекватну носивост да поднесу оптерећења настала доградњом нове етаже.

Abstract – This paper presents an assessment and reparation of multy-storey residential building in Novi Sad consisting of ground floor + basement + 2 floors+ loft. Theoretical part of this paper refers to rationalization of energy use in both residential and in commercial buildings and to durability of RC structures through classes of exposure. On the other hand, practical part of the paper refers to basic technical data about this building and describes in detail the condition of the building and considers the need for reconstruction. Due to lack of proper project documentation, for insight into existing capacity of structural elements the complete static calculations and analyses of all elements in the construction is done.

НАПОМЕНА:

Овај рад је проистекао из мастер рада чији ментор је била Проф. др Мирјана Малешев.

Furthermore, an assessment is made based on detailed survey of the building in terms of carrying capacity, stability and durability. After proper assessment and description of the existing state of the structure, a project of adding one more floor has been made. At the end, there are some suggestions about measures that need to be done to support some elements of bearing structures of the building.

Кључне речи: процена стања, доградња, санације, енергетска ефикасност, класе изложености

1. УВОД

Рад се састоји из две целине. Први део рада представља теоријско – истраживачки део и односи се на анализу утрошка енергије у стамбеним и пословним јединицама и новом начину пројектовања конструкција у смислу продужавања трајности објеката, а други део се односи на процену стања и доградњу објекта у Новом Саду.

ТЕОРИЈСКИ ДЕО**2. ОДРЖИВА ГРАДЊА****2.1. Увод**

Резиденцијални и пословни објекти потроше преко 40% од укупне енергије која се потроши у Европи. Други део енергије потроши се на транспорт и потребе индустрије. Са циљем да се испуне кључни захтеви регулатива о заштити животне средине смањујући при том трошкове енергије чија је цена стално у порасту, од пресудног је значаја да зграде и системи регулисања у њима буду пројектовани тако да буду што ефикаснији.

Бетонске конструкције се пројектују тако да задовоље захтеве граничног стања носивости, трајности и употребљивости, али се сада додаје и утицај конструкције на животну средину који мора да буде у оквиру одговарајућих критеријума. Развој одрживог грађевинарства заснива се на смањењу потрошње енергије и природних сировина, смањењу емисије штетних гасова у ваздух, воду и земљу, смањењу утицаја на климатске промене, смањењу површина земљишта под депонијама, повећању трајности и продужењу експлоатационог века конструкција као и коришћењу секундарних материјала и поновној употреби конструктивних елемената.

Концепт одрживе градње се заснива на пет еко принципа:

- Паметно пројектовање (облик зграде, локација, оријентација, конструкција, изолација,...).
- Употреба еколошких материјала (лако обновљиви материјали, рециклирани, дуготрајни, материјали који нису штетни за животну средину,...).
- Енергетска ефикасност (употреба мање количине енергије за обаљање исте количине посла).
- Рационална потрошња воде (сакупљање кишнице).
- Здрава животна средина (пажљив избор материјала који нису штетни по здравље људи).

Велику већину индустријског отпада чини грађевински отпад. За производњу бетона користе се природне сировине вађењем из земље.

Бетон спада у најјефтиније грађевинске материјале и изражена је сумња да ће добити алтернативу у блиској будућности, што значи да ћемо користити кречњак за цемент и агрегат за бетон у великим количинама с обзиром да је производња бетона на годишњем нивоу износи око 6 милијарди тона.

Од кључне важности је да поведемо рачуна о природи и карактеристикама материјала у нашем окружењу. Треба употребљавати квалитетне, еко материјале. То су материјали који имају изузетне перформансе, дуготрајни су и сировине или сами материјали су лако обновљиви.

Такође треба размишљати о материјалима који настају рециклажом. Тако настали материјали могу иницијално да имају једнако добре перформансе а да у самом процесу рециклаже коштају мање од нових материјала и да омогуће смањење отпада.

2.2. Класе изложености

Деловање средине груписано је у виду класа изложености. Класе изложености које се примјењују зависе од одредби које су важеће на месту употребе бетона.

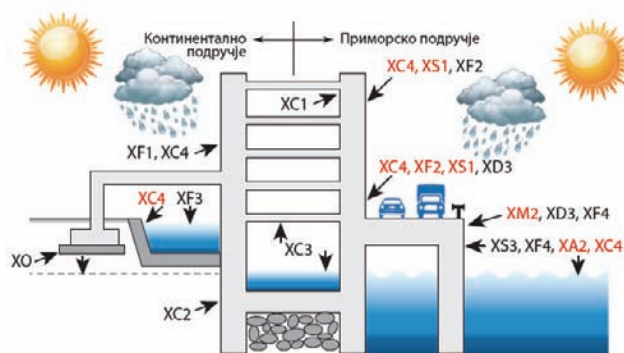
Ова класификација на основу изложености не искључује узимање у обзир посебних услова који постоје на месту употребе бетона или примену заштитних мера, као што су коришћење нерђајућег челика и других метала отпорних на корозију и коришћење заштитних премаза за бетон или арматуру.

Бетон може бити изложен једној или више описаних врста утицаја. Утицаји којима је бетон изложен у складу са тиме могу бити изражени као комбинација класа изложености.

Објекат који је предмет овог рада спада у четврту категорију прорачунског експлоатационог века што подразумева да је предвиђени експлоатациони век конструкције 50 година. Спада у А категорију коришћења површина с обзиром да је предвиђено коришћење објекта за становање и смештај.

Најмањи заштитни слој бетона, утврђује се зависно о класи изложености и начину армирања елемента.

Класе изложености у зависности од окружења и најмање класе чврстоће при притиску бетона за ту класу изложености одређени су одредбом ЕН 206-1 (слика1).



Слика 1. Графички приказ класа изложености бетона

ПРАКТИЧНИ ДЕО

1. ДОГРАДЊА ВИШЕСПРАТНЕ ЗГРАДЕ

1.1 Увод

За постојећи стамбени објекат у улици Хоповска број 2 у Новом Саду спратности Су+П+2+Пк, предвиђено је да се уради процена стања, пројекат доградње и евентуално предлагање одговарајућих мера санације и ојачања конструкције објекта (Слика 2).



Слика 2. Изглед објекта

Спратови зграде су пројектовани као стамбене етаже на којима су заступљени станови различитих структура. Објекат је формиран у скелетном конструктивном систему од армираног бетона, а фасадни зидови са обложени "ЕТИЦС" фасадом.

У објекту постоји укупно 12 стамбених јединица, на свакој етажи се налазе 4 стана. Вертикална комуникација је остварена једнокраким степеништем и лифтом. Објекат је фундиран на армирано-бетонској плочи дебљине 40см. Испод темељне плоче изведена је хидроизолација заштићена неармираним бетоном и заштитним тампон слојем од набијеног шљунка дебљине 20 см. Кров је двоводни са нагибом од 30°.

1.2. Конструктивни систем

Објекат је пројектован у скелетном конструктивном систему. Основни носећи елементи су: АБ стубови, пуне АБ међуспратне плоче, ослоњене директно на стубове и АБ зидна платна за укрућење. Димензије стубова у објекту су различите, тако да су пресеци

стубова следећих димензија: 30/25cm; 45/25cm; 25/25cm. У сутерену су изведени армиранобетонски зидови дебљине 25cm. Међуспратна конструкција је пројектована као плоча дебљине 20cm, директно ослоњена на стубове, са ојачањима код стубова. Терасе су формиране препуштањем међуспратне плоче преко фасадних греда. Ступениште је двокрако са плочом дебљине 12cm. Објект је фундиран на темељној плочи дебљине 40cm.

1.3. Детаљни визуелни преглед

Визуелни преглед објекта је извршен како са спољашње стране, тако и са унутрашње стране у подруму, спратовима, ступеништу и ходницима. С обзиром да се ради о објекту који се завршен 2009.године нису уочена већа оштећења, већ само она неконструктивна. Детаљним визуелним прегледом доступних делова зидова уочена су мања оштећења у виду одвајања и љускања површинског слоја малтера и дисперзивне боје којом су обложене површине зидова ходника и ступенишног простора (слике 3 и 4)



Слика 3. Љуспање боје у ходнику зграде



Слика 4. Одвајање малтера у ходнику

На основу детаљног визуелног прегледа и уочених оштећења на објекту, може се закључити да трајност,

носивост, стабилност објекта, као и његова функционалност нису угрожени.

Сва уочена оштећења су доминантно естетског карактера, али се морају поправити ради спречевања појаве нових или продубљивања тј. погоршања постојећих оштећења. Пукотине, које су регистроване на објекту, су локалног карактера и не угрожавају носивост и стабилност објекта, али би их требало санирати одговарајућим поступцима.

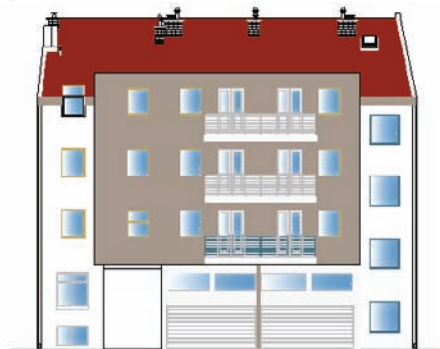
2. ПРОРАЧУН ОСНОВНОГ ОБЈЕКТА

Због недостатка пројектне документације урађен је комплетан статички прорачун објекта пре доградње, ради одређивања расподеле утицаја.

Анализом су обухваћена следећа оптерећења:

1. Стално оптерећење,
2. Оптерећење ветром,
3. Оптерећење снегом и
4. Сеизмичко оптерећење.

На сликама 5 и 6 приказан је изглед објекта пре надоградње.



Слика 5. – Изглед дворашне фасаде



Слика 6. – Изглед уличне фасаде

3. ЗГРАДА ПОСЛЕ ДОГРАДЊЕ

Предвиђена је доградња једне етаже на постојећи објект који се сада састоји од сутерена, приземља, три спрата и поткровља. Конструкција је изведена у скелетном систему, са новом међуспратном конструкцијом у виду пуне бетонске плоче дебљине 20cm, са ступеништем и лифтом уз ивицу објекта.

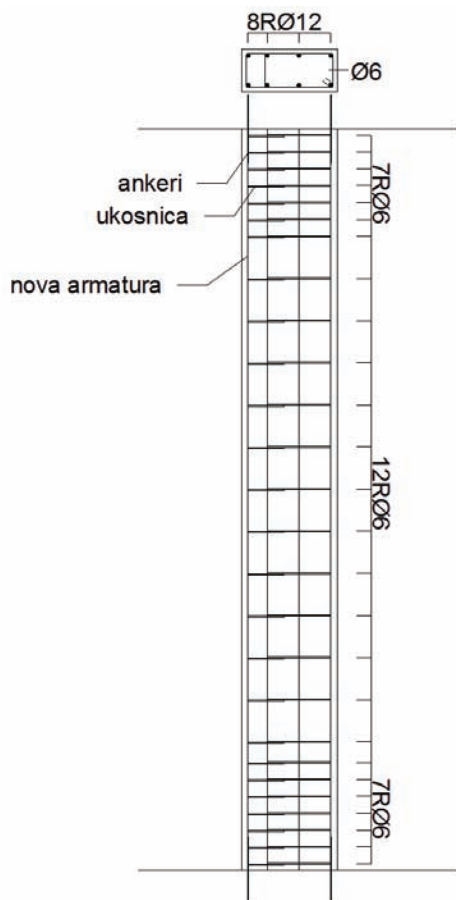
Да би смањили тежину доградње, користиће се лакши материјали тј. уместо црепа, који је био коришћен на објекту, постављаће се тегола. Носећи зидови зидају

се знатно лакшим YTONG Термоблок ПЛУС блоковима димензија 25x62.5x20cm.

Извршен је прорачун дограђене конструкције, провера напона у подрумским зидовима, темељној плочи, као и прорачун напона у тлу услед повећаног оптерећења након надоградње. Прорачун обухвата и сеизмички прорачун и контролу упоредних напона услед сеизмичких сила. Модел је, као и за објекат пре доградње, изведен у софтверу „Tower 6”. Анализа оптерећења и статички прорачун изведени су према важећим SRPS стандардима. Према поменутим стандардима, конструкција је оптерећена сопственом тежином, корисним оптерећењем, оптерећењем од снега, оптерећењем од ветра, као и сеизмичким оптерећењем.

4. ПРЕДЛОГ САНАЦИОНИХ МЕРА ПРИЛИКОМ ДОГРАДЊЕ ОБЈЕКТА И ВЕЗА ПОСТОЈЕЋЕГ ОБЈЕКТА СА ДОГРАДЊОМ

Ради доградње објекта, потребно је да се на поједине елементе конструкције примене одређене санационе мере и да се настави са очувањем стања конструкције. Под санационим мерама се подразумевају захвати на санацији појединих елемената или њихова замена. Потребно је извршити санацију стуба у приземљу тако да се задовољи потребна количина арматуре након доградње што се постиже додавањем две шипке ребрасте арматуре Ø12 (слика 7).



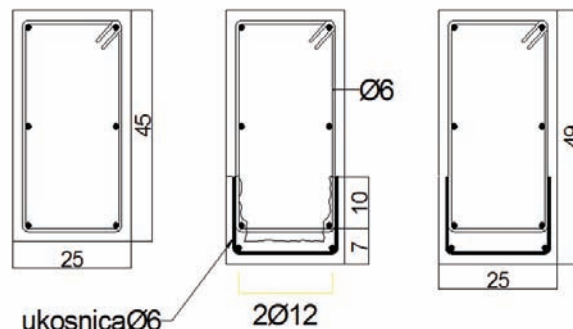
Слика 7. Постављање ојачања у стубу приземља

Да би се стуб повезао са остатком конструкције постављају се анкери који ће вирити како из подне

тако из горње плоче да би се за њих могла заварити нова арматура тј. две шипке Ø12. Око сваке постојеће узенгије мора се оштематити део бетона толико колико је довољно да се нова узенгија у виду “укосница” завари за постојећу узенгију.

Укосница, тј. узенгија додатне арматуре се заварује за постојеће узенгије стуба и то испод или изнад исте, као што је то приказано на слици 8.

Stub S3, 45x25



Слика 8. Скица попречног пресека постављања додатне арматуре у стуб

5. ЗАКЉУЧАК

Упоредивањем утицаја добијених детаљним прорачуном објекта пре и после доградње као и упоређивањем потребне арматуре долази се до података у којим елементима конструкције се јавља потреба за ојачањем или додатном арматуром.

6. ЛИТЕРАТУРА

- [1] М. Малешев, В. Радоњанин: материјал са предавања- *Праћење. Процена стања и одржавање грађевинских објеката*
- [2] М. Малешев, В. Радоњанин: материјал са предавања- *Материјали и технике санације*
- [3] Иван Игњатовић: Санације, реконструкције и одржавање бетонских конструкција
- [4] А.Борковић, М.Томић, Н.Стојковић: Принципи енергетске ефикасности у зградарству
- [5] Интернет адреса : <http://srb.sika.com/>, <http://www.lafarge.rs/>, <http://www.cemex.com/>,

Кратка биографија:



Исидора Сремачки рођена је у Новом Саду 1988. године. Мастер рад на Факултету техничких наука из области грађевинарства – модул конструкције – смер – Санација и процена стања конструкција, одбранила је 2015. год.

HIDRAULIČKA ANALIZA DOGRADNJE GRADSKOG VODOVODNOG SISTEMA U SOMBORU - ZAPADNI SISTEM**HYDRAULIC ANALYSIS OF UPGRADING URBAN WATER SUPPLY SYSTEM IN SOMBOR - WESTERN SYSTEM**

Radovan Marinković, Matija Stipić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – GRAĐEVINARSTVO

Kratak sadržaj – U master radu je urađena hidraulička analiza dogradnje gradskog vodovodnog sistema u Somboru-zapadni sistem. Najpre je urađena hidraulička analiza postojećeg stanja, a zatim i hidraulička analiza novog stanja, kada se na postojeće stanje priključe još tri naselja. Analizirano je i kretanje hlora kroz cevovod i merene su koncentracije u čvornim mestima naselja. Dimenzionisanje vodovodne mreže je rađeno primenom programskog paketa EPANET. Cilj je izvršiti pomenute hidrauličke analize i utvrditi da li u novonastaloj situaciji izvor Jaroš ima kapacitet da obezbedi dovoljnu količinu vode, s obzirom da broj stanovnika u svim naseljima ima tendenciju opadanja i da li je potrebna zamena cevi dovodnog cevovoda. Takođe izvršena je i rekonstrukcija vodovodne mreže u naseljima.

Abstract – The master work is performed hydraulic analysis of upgrading urban water supply system in Sombor-Western system. First he made hydraulic analysis of the current situation, then and hydraulic analysis of the new situation, when the existing state join three other settlements. Moreover, it was the movement of chlorine through the pipeline and the concentrations of the nodal areas of the village. Sizing of water network is done using a software package EPANET. The aim is to carry out the aforementioned hydraulic analysis and determine whether the new situation source Jaros has the capacity to provide a sufficient amount of water, given that the number of inhabitants in all settlements with a declining trend and whether the need to replace pipes of feeder pipeline. Also performed the reconstruction of water supply network in the villages.

Ključne reči: Hidraulička analiza, vodosnabdevanje, protok, pritisak, hidraulički proračun, potrošnja, EPANET

1. UVOD

Predmet proučavanja je hidraulička analiza dogradnje gradskog vodovodnog sistema u Somboru-zapadni sistem. Najpre se vrši hidraulička analiza naselja Stanišić, Svetozar Miletić, Čonoplja i Kljajićevo (sadašnje stanje), a zatim i novog stanja, kada se na postojeće stanje priključe naselja Riđica, Aleksa Šantić i Telečka. Cilj je da se uvidi da li CSV Jaroš sa svojim sadašnjim kapacitetima može da odgovori i potrebama novonastalog

stanja. Cilj je i da se uvidi da li je potrebna zamena postojećeg dovodnog cevovoda, u smislu kapaciteta. Uz to, potrebno je u oba slučaja pravilno dimenzionizati mreže u naseljima, da se dobiju zadovoljavajući pritisci. Takođe, analizom je obuhvaćeno i praćenje hlora kroz cevovod i njegovo merenje u naseljima.

ANALIZA SADAŠNJEG STANJA

(popis stanovništva iz 2002.god.)

2. METODOLOGIJA**2.1. Utvrđivanje broja stanovnika**

U ovim naseljima već duži vremenski period je primećen odliv stanovništva, različitim tempom po dekadama. Kako je 2002.god. radjen popis stanovništva, utvrđen je i broj stanovnika, s kojim se ulazi u proračun, a to je Stanišić 4602 stanovnika, Svetozar Miletić 3169 stanovnika, Kljajićevo 5823 stanovnika i Čonoplja 4359 stanovnika. (Za Čonoplju nije rađena detaljna hidraulička analiza, jer je to prethodnim projektom već urađeno)

2.2. Analiza prosečne potrošnje vode

Realne potrebe stanovništva za vodom ovih naselja nisu poznate i očekivana potrošnja se utvrđuje na osnovu normativa potrošnje datih u vidu tehničkih normi. Njima se utvrđuju prosečne potrošnje različitih kategorija potrošača i očekivane neravnomernosti potrošnje tokom dana i godine.

Tehničke norme korištene u ovom radu su nemačke norme DVGW W 410, koje se uspešno mogu primeniti na našem području. Specifična potrošnja za sva tri naselja je bazirana na procenjenoj specifičnoj potrošnji od 130 L/st.dan čemu su dodati procenjeni gubici. Ovom vrednosti je obuhvaćena javna potrošnja. Gubici su konstantni i uzeti kao vrednost od 20% Qsr.d.

2.3. Časovna neravnomernost

U nedostatku pouzdanih pokazatelja neravnomernosti potrošnje stanovništva maksimalne vrednosti navedenih koeficijenata procenjeni su pomoću empirijskih obrazaca u funkciji broja stanovnika.

Koeficijenti dnevne i časovne neravnomernosti sračunati su na osnovu preporuka tehničkih normi DVGW W 410 po obrascu:

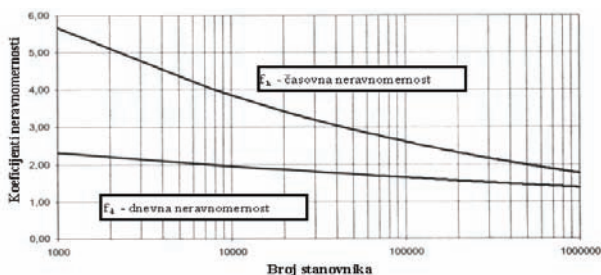
$$fd = 3.9 * E^{-0.0752} \quad (1)$$

$$fh = 18.1 * E^{-0.1682} \quad (2)$$

gde je E broj stanovnika naselja. Funkcionalna zavisnost je data na slici 1.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Matija Stipić, docent.



Slika 1. Koeficijenti neravnomernosti u funkciji br. Stanovnika

Usvojene vrednosti koeficijenata za naselja su: Stanišić $f_d=2.1$, $f_h=4.4$, za Svetozar Miletić $f_d=2.2$, $f_h=4.7$, za Kljajićevo $f_d=2.05$, $f_h=4.3$.

2.4. Protivpožarna potrošnja

Protivpožarna potrošnja je izračunata na osnovu preporuka i u skladu je sa Pravilnikom o tehničkim normativima za hidrantsku mrežu za gašenje požara. Proračun uključuje uvećanu srednju dnevnu potrošnju, gubitke i minimalne protivpožarne protoke. Izračunata je prema sledećem obrascu:

$$Q_{pp}(l/s) = (1.3 \cdot Q_{sr.d}(m^3/d) \cdot 0.08/3.6 + Q_{gub} + Q_{p.min}) \quad (3),$$

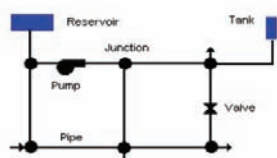
gde je $Q_{p.min} = 10$ L/s.

2.5. Analiza cevnog materijala

Materijal koji je korišćen za izradu cevi je polietilen.

2.6. Modeliranje vodovodnih sistema primenom programskog paketa EPANET

EPANET modelira vodovodni distribucioni sistem kao skup veza spojenih čvorova. Veze predstavljaju cevi, pumpe i kontrolne zatvarače. Čvorovi predstavljaju spojeve, tankove i rezervoare (slika 2).



Slika 2. Fizičke komponente modela

3. HIDRAULIČKI PRORAČUN

3.1. Primenjeni hidraulički model za proračun vodovodne mreže

Osnovne jednačine za određivanje nepoznatih su jednačina kontinuiteta po čvorovima i energetska jednačina za svaki prsten. Cevovod se proračunava kao dugački, zanemaruju se lokalni gubici, a strujanje je jednodimenzionalno stacionarno. Program EPANET zasnovan je na proračunima II metode. Pri rešavanju koeficijenta trenja težilo se obuhvatanju cele oblasti dijagrama $\lambda = \lambda(k/D, Re)$.

3.2. Hidraulički proračun potrošnje vode

Potrebne količine vode za svako naselje izračunate po sledećim izrazima:

$$Q_{sr.d}(l/s) = 130 \text{ l/st/dan} \cdot E / (24 \cdot 3600) + Q_{gub} \quad (4),$$

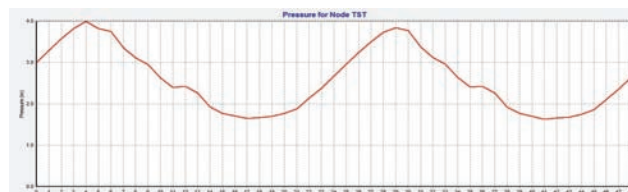
$$Q_{max.d}(l/s) = f_d \cdot (Q_{sr.d} - Q_{gub}) + Q_{gub} \quad (5),$$

$$Q_{max.h}(l/s) = f_h \cdot (Q_{sr.d} - Q_{gub}) + Q_{gub} \quad (6).$$

Na osnovu ovih izraza su dobijene količine potrebne za dimenzionisanje mreže. Vodovodna mreža u naseljima je dimenzionisana na maksimalnu časovnu potrošnju, a dovodni cevovod na maksimalnu časovnu potrošnju.

3.3. Proračun potrebne zapremine rezervoara

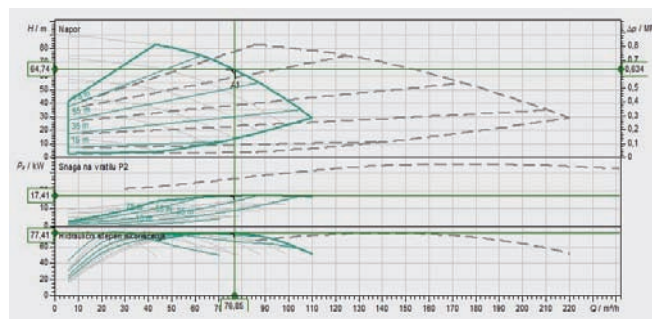
Ukupnu zapreminu rezervoara čine zapremina za izravnavanje dnevne potrošnje i zapremina za protivpožarne potrebe, dok rezervoar u Sv.Miletiću treba da zadovolji i potrebe Stanišića, kao što i rezervoar u Kljajićevu treba da zadovolji potrebe Telečke. Usvojene zapremine rezervoara su: Stanišić 400 m^3 (slika 3), Svetozar Miletić 650 m^3 , a za Kljajićevo 600 m^3 .



Slika 3. Dijagram iskorišćenosti rezervoara u Stanišiću

3.4. Pumpne stanice i izbor pumpe

Pumpne stanice su hidrotehnički objekti čija je uloga da poveća energiju vode koja prolazi kroz pumpnu stanicu. Wilo-vim softverom za selekciju i izbor pumpe prema ulaznim podacima koji su dobijeni hidrauličkim proračunom u softverskom paketu EPANET odabrane su pumpe u naseljima i u CSV Jaroš koja se potiskuje ka rezervoarima u Sv.Miletiću i Kljajićevu. U CSV Jaroš odabrana pumpa je sa karakteristikama tipa Multivert MVIE 7004/2-3/16/E/3-2 (slika 4).



Slika 4. Karakteristike pumpe u CSV Jaroš

U naseljenim mestima su takođe Wilo-vim softverom za odabir pumi prema ulaznim podacima, odabrane odgovarajuće pumpe, dve radne centrifugalne, paralelno vezane i jedna rezervna, koja se uključuje prilikom kvara neke od dve radne.

4. REZULTATI PRORAČUNA

Dominantni prečnik cevi u vodovodnoj mreži u naseljima Stanišić, Svetozar Miletić i Kljajićevo je $\varnothing 100$ mm. Dovodni cevovodi su prečnika $\varnothing 350$ mm, $\varnothing 250$ mm, $\varnothing 200$ mm, a brzine su do oko 1.2 m/s. Brzina u naseljima ne prelazi 1.2 m/s, a u perifernim delovima pada i ispod 0.2 m/s pa se zbog toga nadležnim službama preporučuje redovno održavanje i ispiranje cevi (oko dva puta godišnje). U CSV Jaroš je dodavana koncentracija hlora od 0.5 mg/l i praćeno njegovo kretanje kroz cevovod. Tom prilikom su merene njegove vrednosti u čvornim mestima naselja. Zadovoljavajuće vrednosti su od 0.2-0.5 mg/l. Ove vrednosti su ostvarene u naseljima Kljajićevo i

Svetozar Miletić. U Stanišiću u perifirnim delovima koncentracija hlora pada ispod 0.2 mg/l. Zbog toga je u ovom naselju potrebno vršiti dodatno hlorisanje vode. Prostorije za dohlorisanje se nalaze u sklopu pumpnih stanica i rezervoara. U naselju Stanišić protok i brzine u deonici iza pumpne stanice su najveći u 6 i 13 h, slike 5 i 6.

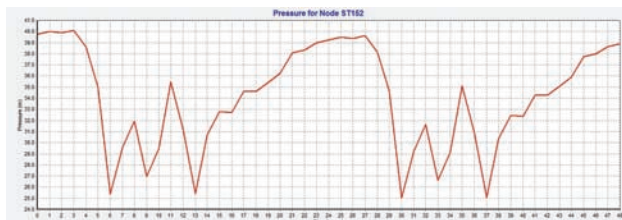


Slika 5. Dijagram promene protoka u deonici ZATS1



Slika 6. Dijagram promene brzina u deonici ZATS1

Pritisci u Stanišiću, na padaju ispod 2.5 bar i na slici 7 je dat dijagram promene u najnepovoljnijem čvoru.



Slika 7. Dijagram promene pritiska u čvoru ST152

U naselju S.Miletić protok i brzine u deonici iza pumpne stanice su najveći u 6h i 9h, a u Kljajićevo u 7h i 13h.

ANALIZA SADAŠNJEG STANJA

(popis stanovništva iz 2015.god.)

5. METODOLOGIJA

5.1. Utvrđivanje broja stanovnika

Kako je 2015. god. rađen novi popis stanovništva, ustanovljeno je da je tendencija opadanja broja stanovnika u ovim naseljima nastavljena. Tako je na osnovu ovog popisa i procene na osnovu podataka za naselja slične veličine u okolini grada Sombor usvojeno da je na kraju planskog perioda broj stanovnika sa kojim se ulazi u proračun za ova naselja sledeći: Čonoplja 3426 stanovnika, Stanišić 3987 st., Svetozar Miletić 2746 st., Kljajićevo 5045 st., Riđica 2011 st., Aleksa Šantić 1770 st., Telečka 1720 st.

5.2. Analiza prosečne potrošnje vode

Analiza prosečne potrošnje se vršila, kao i za analizu potrošnje sadašnjeg stanja.

5.3. Časovna neravnomernost

Za časovnu neravnomernost su korišćeni isti empirijski obrasci, kao i za sadašnje stanje.

Usvojene vrednosti koeficijenata za naselja su:

Stanišić $f_d=2.2$, $f_h=4.55$, Svetozar Miletić $f_d=2.15$, $f_h=4.75$, Kljajićevo $f_d=2.05$, $f_h=4.35$, Čonoplja $f_d=2.15$, $f_h=4.7$, Riđica $f_d=2.15$, $f_h=5.1$, Aleksa Šantić $f_d=2.15$, $f_h=5.2$, Telečka: $f_d=2.2$, $f_h=5.2$.

5.4. Protivpožarna potrošnja

Protivpožarna potrošnja je izračunata kao i za sadašnje stanje, po sledećem obrascu:

$$Q_{pp}(l/s) = (1.3 * Q_{sr.d}(m^3/d)) * 0.08/3.6 + Q_{gub} + Q_{p.min} \quad (7),$$

gde je $Q_{p.min} = 10$ L/s.

5.5. Analiza cevnog materijala

Materijal koji je korišćen za izradu cevi je polietilen.

6. HIDRAULIČKI PRORAČUN

6.1. Primenjeni hidraulički model za proračun vodovodne mreže

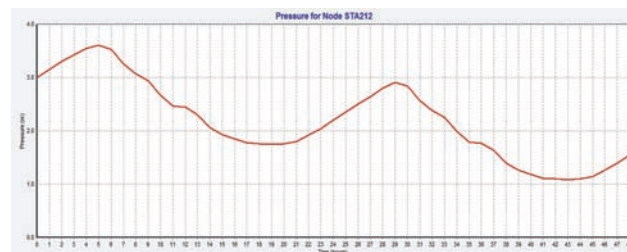
Osnovne jednačine za određivanje nepoznatih su jednačina kontinuiteta po čvorovima i energetska jednačina za svaki prsten. Cevovod se proračunava kao dugački, zanemaruju se lokalni gubici, a strujanje je jednodimenzionalno stacionarno. Program EPANET zasnovan je na proračunima II metode. Pri rešavanju koeficijenta trenja težilo se obuhvatanju cele oblasti dijagrama $\lambda = \lambda(k/D, Re)$.

6.2. Hidraulički proračun potrošnje vode

Potrebne količine vode za svako naselje izračunate po istim formulama kao i potrebne količine vode za naselja za sadašnje stanje. Vodovodna mreža u naseljima je dimenzionisana na maksimalnu časovnu potrošnju, a dovodni cevovod na maksimalnu časovnu potrošnju.

6.3. Proračun potrebne zapremine rezervoara

Ukupnu zapreminu rezervoara čine zapremina za izravnjanje dnevne potrošnje i zapremina za protivpožarne potrebe, dok rezervoar u Sv.Miletiću treba da zadovolji i potrebe Stanišića, a i potrebe naselja Aleksa Šantić. Rezervoar u Stanišiću treba da zadovolji i potrebe stanovništva u Riđici, dok rezervoar u Kljajićevo treba da zadovolji potrebe stanovništva Telečka. Usvojene zapremine rezervoara su: Stanišić 500 m³ (slika 8), Riđica 200 m³, Svetozar Miletić 800 m³, Aleksa Šantić 200 m³, Kljajićevo 500 m³, a za Telečku 200 m³.



Slika 8. Dijagram iskorišćenosti rezervoara u Stanišiću

6.4. Pumpne stanice i izbor pumpe

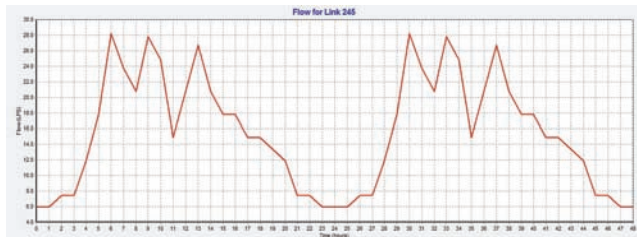
Na isti način i istim softverom su izabrane pumpe i za novo stanje. U CSV Jaroš odabrana je pumpa je sa istim karakteristikama kao i u CSV za sadašnje stanje (slika 4.).

U naseljenim mestima su takođe Wilo-vim softverom za odabir pumi prema ulaznim podacima, odabrane odgovarajuće pumpe, dve radne centrifugalne, paralelno

vezane i jedna rezervna, koja se uključuje prilikom kvara neke od dve radne.

7. REZULTATI PRORAČUNA

Dominantni prečnik cevi u vodovodnoj mreži u naseljima Stanišić, Riđicu, Svetozar Miletić, Aleksi Šantić, Kljajićevo i Telečku je $\varnothing$ 100 mm. Dovodni cevovodi su prečnika $\varnothing$ 350 mm, $\varnothing$ 250 mm, $\varnothing$ 200 mm, $\varnothing$ 160 i $\varnothing$ 140, a brzine su do oko 1.2 m/s. Brzina u naseljima ne prelazi 1.2 m/s, a u perifernim delovima pada i ispod 0.2 m/s pa se zbog toga nadležnim službama preporučuje redovno održavanje i ispiranje cevi (oko dva puta godišnje). U CSV Jaroš je dodavana koncentracija hlora od 0.5 mg/l i praćeno njegovo kretanje kroz cevovod. Tom prilikom su merene njegove vrednosti u čvornim mestima naselja. Zadovoljavajuće vrednosti su od 0.2-0.5 mg/l. Ove vrednosti su ostvarene u naseljima Kljajićevo i Svetozar Miletić. U Stanišiću, Riđici, Aleksi Šantiću i Telečki u perifernim delovima koncentracija hlora pada ispod 0.2 mg/l. Zbog toga je u ovim naseljima potrebno vršiti dodatno hlorisanje vode. Prostorije za dohlorisanje se nalaze u sklopu pumpnih stanica i rezervoara. U naselju Stanišić protok i brzine u deonici iza pumpne stanice su najveći u 6h i 9h, slike 9 i 10.

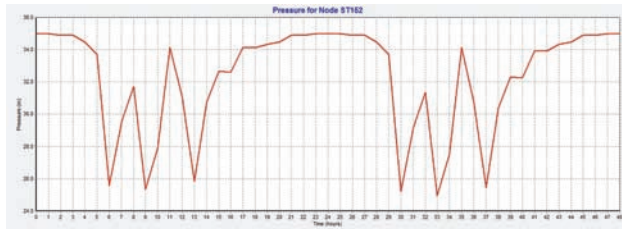


Slika 9. Dijagram promene protoka u deonici 245



Slika 10. Dijagram promene brzina u deonici 245

Pritisci u Stanišiću, na padaju ispod 2.5 bar i na slici 11 je dat dijagram promene u najnepovoljnijem čvoru.



Slika 11. Dijagram promene pritiska u čvoru ST152

U naseljima S.Miletić, Riđica, A.Šantić i Telečka protok i brzine u deonici iza pumpne stanice su najveći u 6h i 9h, a u Kljajićevo u 13h.

8. ZAKLJUČAK

U ovom master radu je obrađena hidraulička analiza dogradnje gradskog vodovodnog sistema u Somboru. Najpre je analizirano trenutno stanje, a zatim je analiziran

problem vodosnabdevanja ukoliko se na postojeći dovodni cevovod pripoje još tri naselja, o čemu je ranije bilo reči. Analizom je obuhvaćeno obezbeđivanje dovoljne količine vode i odgovarajućih pritisaka u mreži, kao i praćenje kretanja hlora kroz cevovod i merenje njegove vrednosti u pojedinim čvornim mestima. Iz ove fabrike voda će se dovodnim cevovodom distribuirati ka ovim naseljima. Potrebni zapreminama rezervoara i pumpnim stanicama obezbeđuje se dovoljna količina vode kao i pritisak u mreži. Iterativnim postupkom hidrauličkog proračuna pomoću programskog paketa EPANET se došlo do adekvatnog načina snabdevanja vodom. Što se pumpi tiče, usvojena je ista u Somboru, za novo, kao i za trenutno stanje, jer su protok i napor koje pumpa treba da ostvari u oba slučaja skoro identični. Ostatak pumpi se usvajao prema potrebama naselja i razlikuju se od naselja do naselja. Rezultati analize dokumentovani su grafičkim i tabelarnim prilogima.

9. LITERATURA

- [1] D. Đurić. Snabdevanje vodom za piće. Banja Luka: Arhitektonsko građevinski fakultet, 2001.
- [2] D. Uzelac. Hidromašinska oprema. Fakultet tehničkih nauka. Novi Sad
- [3] M. Milojević. Snabdevanje vodom i kanalisanje naselja. Građevinski fakultet Beograd, 1987.
- [4] Plan detaljne regulacije naseljenih mesta Bački Monoštor, Kolut i Bački Breg.
- [5] Pravilnik o načinu određivanja i održavanja zone sanitarne zaštite izvorišta vodosnabdevanja (Službeni glasnik Republike Srbije br. 92/08)
- [6] Pravilnik o higijenskoj ispravnosti vode za piće (Službeni glasnik SRJ, br. 42/08)
- [7] Službeni list SFRJ: Pravilnik o tehničkim normativima za hidrantsku mreže za gašenje požara
- [8] M. Stipić, Analiza uticaja protivpožarnih potreba za vodom na tehničke i ekonomske parametre javnih vodovodnih sistema, Doktorska disertacija, Novi Sad, Fakultet tehničkih nauka, 2009 g.
- [9] Tehničke norme „DVGW W 410 potrebe za vodom“ izdalo je nemačko udruženje za vodu i gas, 2008.

Kratka biografija:



Radovan Marinković rođen je u Zrenjaninu 1989. god. Diplomski-master rad na Fakultetu tehničkih nauka iz oblasti Građevinarstvo – Hidrotehnika odbranio je 2015.godine.



Matija Stipić rođen je u Somboru 1964. godine. Doktorirao je na Fakultetu tehničkih nauka 2009. god., a od 2011. ima zvanje docenta. Oblast interesovanja su mu hidraulika i komunalna hidrotehnika.

**RAZVOJ WEB APLIKACIJE ZA ONLINE PRODAJU MOBILNIH TELEFONA
DEVELOPMENT OF AN ONLINE MOBILE PHONE STORE**

Saša Rebović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – U ovom radu je opisan proces izrade veb aplikacije, prezentacija usluga i proizvoda, kao i trgovina datim proizvodima. Takođe se govori o tehnologijama koje su korišćene pri izradi: ASP.NET MVC4, C#, HTML5, CSS3, JavaScript i Razor sintaksa. Govori se i o SQL Server-u koji je korišćen za izradu baze podataka. Realizovano je i plaćanje porudžbina pomoću elektronske platne kartice uz pomoć Authorize.net-a.

Abstract – This paper describes the development process of a web application, the presentation of services, products and of the order process. Also, the technologies used in this project are being explained: ASP.NET MVC4, C#, HTML5, CSS3, JavaScript and Razor syntax. The SQL Server is being used for the database. The payment is done through Authorize.net with payment cards.

Ključne reči: ASP.NET MVC4, C#, HTML5, JavaScript, CSS3, SQL Server, Veb aplikacija

1. UVOD

Prisustvo na internetu u današnje vreme je jedan od presudnih faktora uspeha na tržištu i postiže se uz pomoć veb-sajta [1]. Na taj način firma postaje dostupna svima na svetu, pospešuje prodaju proizvoda i pri tome štedi vreme, jer je sajt dostupan posetiocima na internetu 24h. Današnji veb-sajtovi nisu više statične strane, već su oni sve više aplikacije koji su dizajnirani tako da izvršavaju neke aktivnosti (npr. listaju i prikazuju proizvode iz baze podataka) [2].

U ovom radu predstaviće se takva jedna veb aplikacija koja omogućava registraciju korisnika, dodavanje – brisanje – modifikacija proizvoda od strane administratora i kupovinu određenog proizvoda elektronskim transferom novca.

2. OSNOVNI POJMOVI**2.1 Veb**

Svetska mreža je sistem međusobno povezanih, hipertekstualnih dokumenata koji se nalaze na internetu. Sastavni delovi veb-a su veb server, veb klijent, HTML (eng. HyperText Markup Language – jezik za označavanje hiperteksta), URL (Uniform Resource Locator) i HTTP (HyperText Transfer Protocol) [3].

Uz pomoć internet pregledača, korisnici mogu da gledaju veb stranice koje obično sadrže tekst, slike zvučni i video-zapis.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Gordana Ostojić, vanredni prof.

2.2 Veb Server

Veb server je računar na kojem se nalaze veb stranice i prilikom izrađivanja stranica, da bi one bile dostupne na internetu, potrebno je da se smeste na neki veb server. Veb server bi trebao imati stalnu IP (Internet Protocol) adresu, kako bi mogli registrovati domen i taj domen usmeriti na dotičnu IP adresu. Na taj način posetioci ne moraju pamtiti brojeve (IP) servera već samo adresu [4].

2.3 Veb-sajt i veb aplikacija

Veb-sajt je skup uzajamno povezanih stranica sa nekim sadržajem (slike, snimci i drugi materijalni sadržaj). Oni se koriste na mnogo različitih načina i shodno tome njihova namena je raznolika. Uopšteno gledano veb-sajtovi mogu biti statički ili dinamički (interaktivni).

Statički veb-sajt predstavlja tip veb-sajta čiji sadržaj ne omogućava interaktivnu komunikaciju i odnos sa vlasnicima veb-sajta. Ovakav tip veb-sajtova se uglavnom izrađuje i programira u HTML-u i pomoću CSS-a (eng. Cascading Style Sheet).

Dinamički veb-sajt za razliku od statičkog omogućava promenu svog izgleda i generisanje novog sadržaja u zavisnosti od zahteva i aktivnosti posetilaca veb-sajta. Za kreiranje dinamičkih veb-sajtova koriste se različiti softverski sistemi kao što su ASP (eng. Active Server Pages), Java, ColdFusion, JSP (eng. Java Server Pages) itd. [5].

Veb aplikacija predstavlja bilo koji kompjuterski program koji radi u nekom od postojećih internet pregledača. Osnovna karakteristika veb aplikacije je veoma jednostavno ažuriranje i dodavanje sadržaja. Veb aplikacije koriste veb dokumente napisane u nekom od standardnih formata kao što su HTML i JavaScript, koji su podržani od strane internet pregledača [3].

2.4 Internet pregledač

Internet pregledač (Internet browser, web browser) je program koji korisniku omogućava pregledanje veb stranica i multimedijalnih sadržaja vezanih za njih [3].

Najpopularniji pregledači danas su:

- Mozilla Firefox – besplatni pregledač otvorenog koda
- Google Chrome – Guglov besplatni pregledač, projektovan s' idejom da iskoristi sve mogućnosti Guglovih usluga
- Opera – besplatni pregledač zatvorenog koda firme Opera
- Safari – Eplov besplatni pregledač
- Internet Explorer – besplatni pregledač zatvorenog koda korporacije Microsoft

3. ANALIZA VEB-SAJTOVA I ODREĐIVANJE CILJEVA RADA

Uspešnost sajta zavisi od više kriterijuma. Najvažniji kriterijumi su:

- Dizajn
- Sadržaj
- Navigacija

Dizajn predstavlja sliku firme na internetu i čini proces koji se započinje sagledavanjem koncepta budućeg sajta, planiranjem sajta i na kraju dizajniranjem [1].

Sadržaj je razlog dolaska posetioca na veb-sajt i način na koji je odrađen sadržaj može da utiče na uspeh sajta [1].

Navigacija veb-sajta (npr. navigacioni meni itd.) treba da ima logičan raspored, da ne bi došlo do zabune tokom pregledanja sajta i zbog toga i do napuštanja sajta od strane posetioca [1].

Analizom drugih sajtova moguće je jasno definisati potrebe koje trebaju da se ispune prilikom projektovanja veb aplikacije, kao i sadržaj koji treba da poseduje.

Veb aplikacija treba da ima jasan vizuelni dizajn i navigacija treba da bude jednostavna. Sadržaj treba da je pregledan i dobro definisan. Potrebno je obezbediti registraciju korisnika, mogućnost izmene postojećih podataka od strane korisnika kao i njihovu zaštitu od zloupotrebe naloga ili podataka. Zatim, treba da postoji administratorski nalog koji ima mogućnost provere porudžbina, dodavanje, brisanje i modifikacijom proizvoda. Korisnici moraju posedovati potrošačku korpu za proizvode koji izaberu i koju treba sačuvati u bazi podataka. Nakon odabira proizvoda korisnik treba da se usmeri ka stranici za porudžbine gde se uzimaju podaci za dostavu i bira način plaćanja. U ovom radu je fokus na online plaćanju preko kartice i potrebno je da se odredi način na koji će se to realizovati. Podaci sa kartice moraju tokom obavljanja transakcije biti dobro zaštićeni da ne bi došlo do zloupotrebe. Pored svega toga bi bilo dobro da aplikacija poseduje mogućnost odabira jezika na kom će biti prikazan, u slučaju da maternji jezik korisnika nije srpski.

Pored samog procesa analiziranja i planiranja treba spomenuti i tehnologiju koja se primenjuje u ovom radu, a to je ASP.NET MVC 4. MVC predstavlja softverski patern koji odvaja prikaz informacija od interakcije korisnika sa tim informacijama. Široko je prihvaćen kao patern za aplikacije na internetu, u svim značajnijim programskim jezicima. U osnovi ove arhitekture se nalaze dve centralne ideje a to su:

- Ponovno korišćenje već postojećeg koda
- Jasna raspodela zaduženja među različitim delovima sistema

4. KORIŠĆENE VEB TEHNOLOGIJE

4.1 HTML

HTML predstavlja opisni jezik koji je namenjen samo za opisivanje veb stranica. Pomoću njega se mogu odvojiti elementi kao što su naslovi, paragrafi, citati i slično tome.

Svaki HTML dokument bi trebao da počne sa definicijom tipa dokumenta koji internet pregledaču definiše po kom standardu je dokument napisan [3].

```
<!DOCTYPE html>
```

Ovaj kod znači da je dokument napisan po HTML 5 standardu.

Što se sintakse HTML-a tiče, osnovu predstavljaju tagovi (eng. tag – etikete ili nalepnice) i njihovi atributi [3]. Tagovi su skoro uvek upareni gde prvi tag predstavlja početni a drugi završni tag (<tag> - početni, </tag> - završni).

4.2 CSS

CSS služi za definisanje stilova koji određuju izgled HTML elemenata (font, boje, pozadine itd) [6].



Slika 1. Primer CSS sintakse

Selektor ukazuje na HTML element koji treba da se stilizuje. Deklaracioni blok sadrži jednu ili više deklaracija koji su razdvojeni tačkom i zarezom. Svaka deklaracija se sastoji iz osobine i njene vrednosti, koji su međusobno razdvojeni duplom tačkom [7]. U ovom primeru sintakse (slika 1) se definiše naslov h1 koji će imati plavu boju teksta i veličine fonta 12 piksela.

4.3 JavaScript

JavaScript je programski jezik koji se interpretira, sa objektno orijentisanim mogućnostima [8]. On je najpopularniji skriptni jezik na internetu pomoću kojeg veb stranice postaju dinamičnije i podržavaju ga svi poznatiji internet pregledači [9].

Za izradu programa u JavaScript-u je neophodno osnovno poznavanje HTML-a. Programski kod se ugrađuje u HTML dokument između HTML <script> i </script> tagova (slika 2).

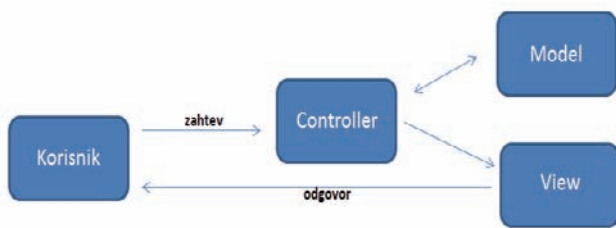
```
<script>
document.getElementById("test").innerHTML = "Moj prvi JavaScript";
</script>
```

Slika 2. Primer JavaScript koda

4.4 MVC Arhitektura

Osnovna odlika ove arhitekture jeste da deli aplikacije na tri različite komponente i definiše njihove međusobne interakcije [3]:

- **Model** – predstavlja stanje sistema koje može promeniti operacije modela
- **View** – obezbeđuje korisniku interfejs pomoću kojeg se unose podaci i pozivaju odgovarajuće operacije koje treba izvršiti nad modelom
- **Controller** – posrednik između view-a i modela koji preuzima korisničke zahteve od view-a, manipuliše njima i prosleđuje ih modelu, pa na osnovu dobijenih podataka od modela, modifikuje view.



Slika 3. MVC arhitektura

4.5 ASP.NET i ASP.NET MVC

ASP.NET je razvojna platforma koja se koristi za izradu dinamičkih veb-sajtova, veb aplikacija i veb servisa. Bazirana je na .NET softverskoj platformi koja uključuje veliki broj gotovih biblioteka kodova za uobičajene probleme u programiranju i pruža jezičku interoperabilnost između nekoliko programskih jezika (C#, VB.NET, itd) pomoću CLR-a (eng. Common Language Runtime).

ASP.NET nudi tri različita modela programiranja za razvoj veb aplikacija i veb-sajtova:

- ASP.NET veb stranice (eng. Web Pages)
- ASP.NET veb forme (eng. Web Forms)
- ASP.NET MVC

ASP.NET MVC model programiranja predstavlja model koji je korišćen u ovom radu. On omogućuje razvoj veb aplikacija i veb-sajtova primenom MVC softverske arhitekture.

4.6 C#

C# (C sharp) je jednostvan jezik, ali veoma izražajan kada je potrebno implementirati moderne programske ideje [8]. Nastao je 2002. godine kao sastavni deo Microsoft-ovog razvojnog orkuženja .NET platforme [3]. Podržava strukturirano, objektno orijentisano programiranje zasnovano na komponentama.

Srž svakog objektno orijentisanog jezika jeste podrška za definisanje klasa i rad sa njima. C# sadrži rezervisane reči (eng. keywords) za deklarisanje novih klasa i njihovih metoda i svojstava, kao i za enkapsuliranje (eng. encapsulation), nasleđivanje (eng. inheritance) i polimorfizam (eng. polymorphism) – tri kamena temeljca objektno orijentisanog programiranja.

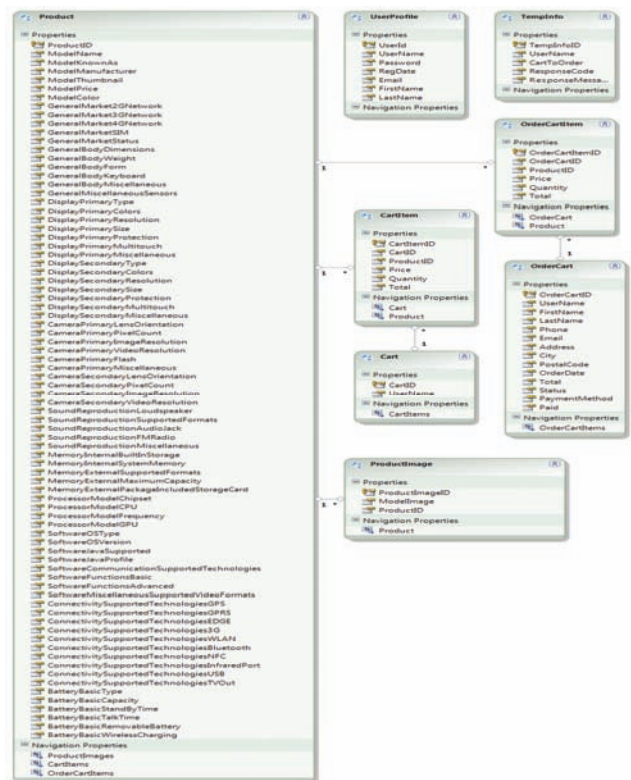
4.7 Razor View Engine

Razor nije programski jezik već markup jezik koji se primenjuje sa serverske strane. Pomoću njega se u veb stranice ugrađuje serversko baziran kod (Visual Basic ili C#) [7]. Razor je objavljen kao sastavni deo ASP.NET MVC 3 i namenjen je za kreiranje veb aplikacija [3].

5. PROJEKTOVANJE I IZRADA VEB APLIKACIJE

5.1 Projektovanje i izrada baze podataka

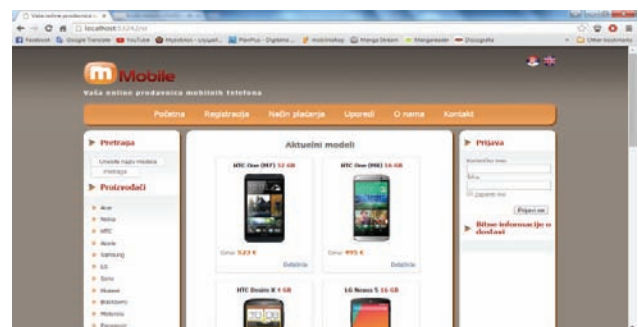
Prvi korak predstavlja najbitniji korak u izradi veb aplikacije, a to je dobro projektovanje same aplikacije i baze podataka za tu aplikaciju. Baza podataka mora da sadrži sve potrebne podatke vezano za korisnika, za proizvode, za korisničku korpu i porudžbinu i to sve mora biti na neki način povezano. Izgled baze podataka za ovu veb aplikaciju je prikazan na slici 4.



Slika 4. Izgled baze za veb aplikaciju

5.2 Projektovanje i izrada veb aplikacije

Sledeći korak predstavlja projektovanje i izradu veb aplikacije, od njenog dizajna pa do funkcionalnosti stranica. Projektovanje počinje od razmatranja potrebnih stranica, šta treba da poseduju i kako će da funkcionišu. Vođeno je računa o tome da sve bude pregledno i jednostavno i da se ne pretrpavaju stranice sa nepotrebnim stvarima. Dizajn aplikacije je jednostavan i ispunjava osnovne zahteve.



Slika 5. Dizajn početne stranice (gornji deo)

Veb aplikacija poseduje jedan zajednički okvir (layout) koji se prikazuje na svim postojećim stranicama veb aplikacije. Na njemu se nalaze najčešće korišćeni elementi za korisnika. Layout poseduje elemente za navigaciju ka stranicama za registraciju, način plaćanja, upoređivanje modela, za više informacija o sajtu i kontakt stranicu. Sa strane su smešteni elementi za pretragu, listu proizvođača telefona, prijavu i bitnije informacije od dostavama. Layout poseduje takođe i elemente za promenu govornog jezika na kom se prikazuju stvari sa sajta. Pošto je ovaj rad baziran na ASP.NET MVC 4, sve stranice su grupisane po svojoj nameni. Svaka od tih grupa pripada nekom od kontrolera radi lakše raspodele funkcija i zadataka koji treba ta neka grupa da obavlja.

Kontroler predstavlja klasu u kojoj se nalaze metode koje su odgovorne za određenu veb stranicu. Te metode se nazivaju *ActionResult*-ovi i oni se pozivaju kad korisnik klikne na određene linkove sa veb stranica. Imena tih *ActionResult*-ova predstavljaju imena stranica sa kojima su povezani. Svaki od tih metoda ide u paru sa return naredbom, gde return naredba poziva određeni View koji treba da se generiše i prikaže korisniku (često se i prosleđuju neki podaci u View). U ovom radu postoji pet kontrolera:

- Home kontroler
- Products kontroler
- Account kontroler
- Cart kontroler
- Order kontroler

Home kontroler upravlja sa Index, About i Contact stranicama. Index metoda uzima osam aktuelnih modela telefona i prosleđuje ih u odgovarajući View za tu metodu. About i Contact metode pozivaju samo View-ove bez ikakvog prosleđivanja podataka.

Products kontroler upravlja sa proizvodima i stranicama vezano za njih – Details, Browse, Create, Edit i Delete. Kada korisnik izabere nekog određenog proizvođača telefona, Browse metoda na osnovu toga vrši filtriranje baze i uzima listu svih telefona datog proizvođača i prosleđuje ih (ograničeno na 16 telefona po stranici, radi preglednosti) u odgovarajući View. Details metoda uzima određeni telefon iz baze (koji je izabran od strane korisnika) i prosleđuje ga u odgovarajući View gde dolazi do detaljnog prikazivanja karakteristika telefona i njegovih slika. Kreiranje novog telefona i njegovo dodavanje u postojeću bazu podataka se vrši preko Create metoda. Izmena postojećeg telefona se vrši tako što se odabrani telefon uzima iz baze podataka pa se prosleđuje u View radi izmene njegovih karakteristika ili slika, od strane korisnika (administratora), pa se zatim ponovo vraća u bazu podataka gde će se sačuvati izmene za dati telefon. Prilikom brisanja telefona kada administrator izabere koji će telefon da izbriše, prvo dolazi do upita da li zaista želi da ga izbriše iz baze pa zatim do njegovog brisanja, ako je potvrđen odgovor.

Account kontroler je kontroler koji upravlja sa korisničkim nalogom. U njemu se nalaze metode koje su zadužene za prijavljivanje i registraciju korisnika.

Cart kontroler je zadužen za korisničku korpu u koju se stavljaju odabrani proizvodi koje korisnik želi da kupi. Ovaj kontroler poseduje metode za prikaz sadržaja korpe (Index), dodavanje proizvoda u korpu (AddToCart), brisanje pojedinačnog proizvoda iz korpe (RemoveFromCart) i brisanje svih proizvoda iz korpe (EmptyCart). OrderController je kontroler koji je zadužen za porudžbine. Plaćanje porudžbine se vrši bankovnom karticom, što je i jedan od zadataka ovog rada.

U ovom radu je odabran Authorize.net kao provajder za obavljanje potrebnih transakcija i on predstavlja jedan od poznatijih provajdera takvih servisa. Komunikacija sa sajtom se vrši preko API-a (Application Programming Interface). Metoda koja je korišćena u ovom radu je DPM metoda. Pomoću ove metode veb aplikacija vlasnika poseduje formu koja će da prosledi podatke kartice korisnika direktno ka sajtu Authorize.net-a, a da ih pri tome ne čuva na serveru, pri čemu automatski sama aplikacija ne podleže nekim uslovima PCI (Payment Card Industry) standarda, a uz to veb aplikacija poseduje veću fleksibilnost tokom kreiranja.

6. ZAKLJUČAK

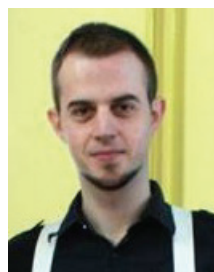
E-trgovina postaje sve zastupljeniji način poslovanja u svetu, u Srbiji je usled problema oko zakonskih propisa taj vid poslovanja u sporom porastu, ali ipak dolazi do razvoja elektronskih prodavnica. One poseduju ipak značajne prednosti za krajnjeg potrošača, od toga da nude lakše i brže mogućnosti dolaženja do informacija, usluga i proizvoda.

Veb aplikacija koja je realizovana u ovom radu ispunjava glavne zadatke koji su definisani na početku rada i prikazan je njen razvoj u ASP.NET MVC-u 4, kao i pogodnosti primene te tehnologije. Naravno, postoji još mnogo načina unapređenja same veb aplikacije tj. njenih mogućnosti. U praksi, na velikom projektu, kao što je ovaj, radi više osoba gde je svaka osoba (ili više osoba) zadužena za jedan deo projekta (npr. dizajn sajta) i na taj način se postiže veća efikasnost, pouzdanost i kvalitet aplikacije.

7. LITERATURA

- [1] <http://www.biznis-akademija.com>
- [2] <http://www.djordjesavic.com>
- [3] <http://www.wikipedia.org>
- [4] <http://wp.pcpsystemsoftware.com>
- [5] <http://www.webdizajn-beograd.com>
- [6] <http://www.popwebdesign.net>
- [7] <http://www.w3schools.com>
- [8] <http://www.mikroknjiga.rs>
- [9] <http://es.elfak.ni.ac.rs>

Kratka biografija:



Saša Rebović rođen je u Mostaru 1985. god. Osnovnu i srednju školu završio je u Šapcu. Master rad na Fakultetu tehničkih nauka iz oblasti Mehatronike – Robotika i automatizacija odbranio je 2015. godine.

PRIMENA HERMITOVOG SPLAJNA ZA PLANIRANJE I PRAĆENJE PUTANJE MOBILNOG ROBOTA

USING THE HERMITE SPLINE FOR PATH PLANNING AND TRACKING CONTROL OF A MOBILE ROBOT

Marko Tikvić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – Rad prezentuje metod planiranja putanje mobilnog robota sa diferencijalnim pogonom zasnovan na Hermitovom splajnu V reda i metod praćenja putanje zasnovan na profilu brzine robota. Prezentovani metod je projektovan tako da omogući kontinualnost izvoda prvog i drugog reda duž cele putanje. Kako bi se robot kretao po željenoj putanji upotrebljena je PID regulacija brzina pogonskih motora robota.

Abstract – The paper present a method for planning a path of a differential drive mobile robot based on 5th order Hermite spline and a path following controller based on robot's velocity profile. The method provided is designed in such manner to provide first and second order derivative continuity during the course of the entire path. In order for robot to follow the desired path a velocity PID controller is used.

Cljučne reči: mobilni robot, planiranje putanje, praćenje putanje, Hermitov splajn, PID regulacija brzine

1. UVOD

Navigacija mobilnih robota sa neholonomnim ograničenjima je veoma važan zadatak u mobilnoj robotici [1] - [4]. U svojoj najjednostavnijoj formi zadatak navigacije se sastoji iz kretanja robota od jedne pozicije do druge, aliu praksi navigacija često obuhvata zahtevne operacije kao što su planiranje putanje kretanja robota, praćenje željene putanje ili pokretnog objekta i izbegavanje statičkih i dinamičkih prepreka [5], [6]. U radu će biti opisane metode koje omogućavaju kretanje robota po lineranim i nelinearnim putanjama uz istovremenu regulaciju pozicije, brzine, ubrzanja i orijentacije robota u tačkama od interesa. Predloženo rešenje je zasnovano na interpolaciji predefinisanih skupa ciljeva koji se sastoje od željenih vrednosti pozicije i orijentacije robota u određenom vremenskom trenutku.

Za interpolaciju tačaka je upotrebljen Hermitov splajn V reda [7] kako bi se obezbedilo da funkcija koja opisuje putanju robota bude dva puta diferencijabilna, odnosno kako bi se obezbedile neprekidne funkcije pravolinijske brzine i ubrzanja robota. U nastavku rada će biti prikazano kako je moguće doći do vrednosti brzina pogonskih motora na osnovu translatorne i ugaone brzine robota. Praćenje putanje je izvršeno regulacijom brzina pogonskih motora upotrebom PID kontrolera.

NAPOMENA:

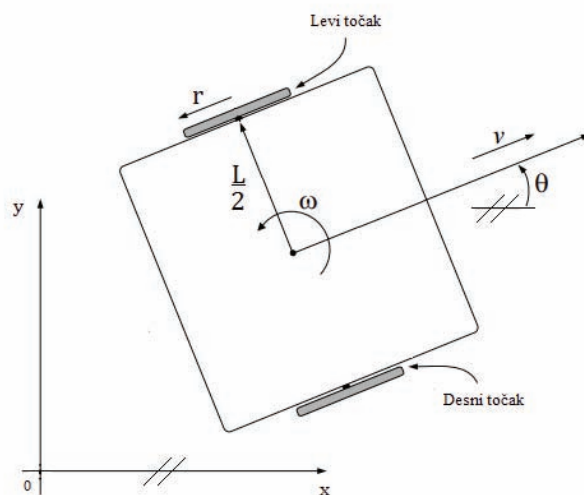
Ovaj rad proistekao je iz master rada čiji mentor je biodoc. dr Mirko Raković.

2. FORMULISANJE ZADATKA

Posmatrajmo robota sa diferencijalnim pogonom u kartezijanskom prostoru kao na slici 1. Zadatak robota je da se kreće u $x-y$ ravni nepokretnog referentnog koordinatnog sistema tako da u predefinisanim željenim vremenskim trenucima dostigne željenu poziciju i orijentaciju. U opštem slučaju imamo skup od $n+1$ takvih željenih vrednosti:

$$P_i = \begin{cases} t_i \\ x_i \\ y_i \\ \theta_i \end{cases}, \quad i = 0 \dots n \quad (1)$$

Gde su (x_i, y_i) i θ_i , respektivno, željena pozicija i orijentacija robota u trenutku t_i . Pored navedenih uslova pozicije i orijentacije, realne situacije zahtevaju glatko kretanje robota duž putanje. Da bi se ovo postiglo potrebno je obezbediti i kontinualnost prvog i drugog izvoda duž cele putanje.



Slika 1. Postavka sistema i oznake

3. PLANIRANJE PUTANJE

Za formiranje putanje, potrebno je izvršiti interpolaciju $n+1$ tačaka opisanih sa (1) uz dodatni uslov da rezultujući interpolant bude dva puta diferencijabilan. Ovaj rad predlaže rešenje primenom polinomijalne interpolacije u obliku Hermitovog splajna.

Da bismo obezbedili kontinualnost prvog i drugog izvoda rezultujućeg interpolanta potreban nam je polinom III reda [8], ali da bismo imali kontrolu nad vrednostima drugog izvoda u zadatim pozicijama potreban nam je polinom V reda [7]. Za interpolaciju $n+1$ tačke potrebno

nam je n takvih polinoma. Kako u polinomu V reda konfiguriše 6 koeficijenata, za pronalaženje rešenja funkcije koja interpolira sve tačke potrebno je pronaći $6n$ takvih koeficijenata.

Vrednosti x i y pozicije robota možemo predstaviti parametarskim jednačinama kao:

$$p_{x,i}(t_j) = a_{x,i}t_j^5 + b_{x,i}t_j^4 + c_{x,i}t_j^3 + d_{x,i}t_j^2 + e_{x,i}t_j + f_{x,i} \quad (2)$$

$$p_{y,i}(t_j) = a_{y,i}t_j^5 + b_{y,i}t_j^4 + c_{y,i}t_j^3 + d_{y,i}t_j^2 + e_{y,i}t_j + f_{y,i} \quad (3)$$

$$t_j \in [t_i, t_{i+1}]$$

$$i = 0 \dots n - 1$$

Da bismo zadovoljili uslov neprekidnosti funkcije potrebno je da se svih n polinoma nastavlja jedan na drugi u čvorovima (zadatim pozicijama) funkcije, odnosno, potrebno je zadovoljiti sledeće uslove:

$$p_{x,i}(t_i) = p_{x,i+1}(t_i) = x_i, \quad (4)$$

$$p_{y,i}(t_i) = p_{y,i+1}(t_i) = y_i \quad (5)$$

$$i = 1 \dots n - 1$$

s tim da su krajevi splajna ograničeni sa:

$$\begin{aligned} p_{x,0}(t_0) &= x_0 \\ p_{y,0}(t_0) &= y_0 \\ p_{x,n-1}(t_n) &= x_n \\ p_{y,n-1}(t_n) &= y_n \end{aligned} \quad (6)$$

Pošto su nam u interesu kontinualnost prvog i drugog izvoda potrebno je zadovoljiti i uslove:

$$p'_{x,i}(t_i) = p'_{x,i+1}(t_i) = x'_i \quad (7)$$

$$p'_{y,i}(t_i) = p'_{y,i+1}(t_i) = y'_i$$

$$p''_{x,i}(t_i) = p''_{x,i+1}(t_i) = x''_i \quad (8)$$

$$p''_{y,i}(t_i) = p''_{y,i+1}(t_i) = y''_i$$

$$i = 0 \dots n - 1$$

gde je

$$p'_{x,i}(t_j) = 5a_{x,i}t_j^4 + 4b_{x,i}t_j^3 + 3c_{x,i}t_j^2 + 2d_{x,i}t_j + e_{x,i} \quad (9)$$

$$p'_{y,i}(t_j) = 5a_{y,i}t_j^4 + 4b_{y,i}t_j^3 + 3c_{y,i}t_j^2 + 2d_{y,i}t_j + e_{y,i} \quad (10)$$

$$p''_{x,i}(t_j) = 20a_{x,i}t_j^3 + 12b_{x,i}t_j^2 + 6c_{x,i}t_j + d_{x,i} \quad (11)$$

$$p''_{y,i}(t_j) = 20a_{y,i}t_j^3 + 12b_{y,i}t_j^2 + 6c_{y,i}t_j + d_{y,i} \quad (12)$$

Sada je potrebno pronaći vrednosti prvih i drugih izvoda u čvorovima splajna kako bismo mogli da započnemo pronalaženje koeficijenata svih polinoma.

Setimo se da nam je jedan od uslova iz (1) orijentacija robota, θ . Orijetacija u $x - y$ koordinatnom sistemu se može predstaviti kao nagib tangente funkcije $f(x, y)$, ali se može predstaviti i preko odnosa nagiba tangenti funkcija $f_x(t)$ i $f_y(t)$.

$$\theta = \text{atan}\left(\frac{f'_x}{f'_y}\right) \quad (13)$$

$$\tan(\theta) = \frac{f'_x}{f'_y}$$

ili u našem slučaju

$$\theta_i = \text{atan}\left(\frac{p'_{x,i}}{p'_{y,i}}\right) \quad (14)$$

$$\tan(\theta_i) = \frac{p'_{x,i}}{p'_{y,i}}$$

Iz (14) se jasno vidi da, ukoliko imamo informaciju o vrednosti prvog izvoda bar jedne od funkcija $p'_{x,i}$ i $p'_{y,i}$, vrlo lako možemo doći do vrednosti izvoda druge funkcije kako bismo zadovoljili uslov za ostvarivanje orijentacije iz (1). S obzirom na to da nam je brzina kretanja robota indirektno propisana parametrima t_i, x_i, y_i , do vrednosti $p'_{x,i}$ ili $p'_{y,i}$ u trenucima t_i možemo doći na sledeći način:

$$x'_i = \sqrt{\frac{(x_i - x_{i-1})^2}{(t_i - t_{i-1})^2}}, \quad (15)$$

$$y'_i = \sqrt{\frac{(y_i - y_{i-1})^2}{(t_i - t_{i-1})^2}} \quad (16)$$

$$x'_0 = x'_n = 0, \quad y'_0 = y'_n = 0 \quad (17)$$

$$i = 1 \dots n - 1$$

Ostalo nam je još da pronađemo vrednosti drugih izvoda kao:

$$x''_i = \sqrt{\frac{(x'_i - x'_{i-1})^2}{(t_i - t_{i-1})^2}} \quad (18)$$

$$y''_i = \sqrt{\frac{(y'_i - y'_{i-1})^2}{(t_i - t_{i-1})^2}} \quad (19)$$

$$x''_0 = x''_n = 0, \quad y''_0 = y''_n = 0 \quad (20)$$

$$i = 1 \dots n - 1$$

Da sumiramo:

- potrebno je interpolirati $n + 1$ poznatih pozicija, uz uslov kontinualnosti prvog i drugog reda rezultujućeg interpolanta,
- za to nam je potrebno n polinoma V reda, odnosno postoji $6n$ nepoznatih koeficijenata koje je potrebno pronaći,
- između (4), (5), (7) i (8) imamo $6(n - 1)$ jednačina na raspolaganju,
- između (6), (16) i (19) imamo 6 raspoloživih jednačina.

Dakle imamo dovoljno raspoloživih jednačina da pronađemo koeficijente svih n polinoma. Ukoliko sistem predstavimo u matičnom obliku dobijamo

$$X_i = T_i K_{x,i}, \quad Y_i = T_i K_{y,i}$$

Gde je

$$X_i = \begin{bmatrix} p_{x,i}(t_i) \\ p_{x,i}(t_{i+1}) \\ p'_{x,i}(t_i) \\ p'_{x,i}(t_{i+1}) \\ p''_{x,i}(t_i) \\ p''_{x,i}(t_{i+1}) \end{bmatrix} = \begin{bmatrix} x_i \\ x_{i+1} \\ x'_i \\ x'_{i+1} \\ x''_i \\ x''_{i+1} \end{bmatrix}$$

$$Y_i = \begin{bmatrix} p_{y,i}(t_i) \\ p_{y,i}(t_{i+1}) \\ p'_{y,i}(t_i) \\ p'_{y,i}(t_{i+1}) \\ p''_{y,i}(t_i) \\ p''_{y,i}(t_{i+1}) \end{bmatrix} = \begin{bmatrix} y_i \\ y_{i+1} \\ y'_i \\ y'_{i+1} \\ y''_i \\ y''_{i+1} \end{bmatrix}$$

$$T_i = \begin{bmatrix} t_i^5 & t_i^4 & t_i^3 & t_i^2 & t_i & 1 \\ t_{i+1}^5 & t_{i+1}^4 & t_{i+1}^3 & t_{i+1}^2 & t_{i+1} & 1 \\ 5t_i^4 & 4t_i^3 & 3t_i^2 & 2t_i & 1 & 0 \\ 5t_{i+1}^4 & 4t_{i+1}^3 & 3t_{i+1}^2 & 2t_{i+1} & 1 & 0 \\ 20t_i^3 & 12t_i^2 & 6t_i & 2 & 0 & 0 \\ 20t_{i+1}^3 & 12t_{i+1}^2 & 6t_{i+1} & 2 & 0 & 0 \end{bmatrix}$$

$$K_{x,i} = \begin{bmatrix} a_{x,i} \\ b_{x,i} \\ c_{x,i} \\ d_{x,i} \\ e_{x,i} \\ f_{x,i} \end{bmatrix}$$

$$K_{y,i} = \begin{bmatrix} a_{y,i} \\ b_{y,i} \\ c_{y,i} \\ d_{y,i} \\ e_{y,i} \\ f_{y,i} \end{bmatrix}$$

$$i = 0 \dots n - 1$$

Odavde možemo lako rešiti matični sistem za $K_{x,i}$ i $K_{y,i}$:

$$K_{x,i} = T_i^{-1} X_i, \quad K_{y,i} = T_i^{-1} Y_i \quad (20)$$

Gde je

$$T_i^{-1} = \frac{adj(T_i)^T}{\det(T_i)}$$

Za određivanje pozicije robota u trenutku $t_x, t_x \in [t_i, t_{i+1}]$ sada je samo potrebno uvrstiti koeficijente $K_{x,i}$ i $K_{y,i}$ u jednačine (2) i (3). Na sličan način dolazimo i do translatorne brzine robota, primenom jednačina (9), (10), (11) i (12), dok se do ugaone brzine robota dolazi pomoću informacija dobijenih iz jednačine (14) i to na sledeći način. Vrednost ugaone brzine robota u trenutku t_x iznosi

$$\omega(t_x) = \frac{\theta(t_x) - \theta(t_x - \Delta t)}{\Delta t}.$$

gde je Δt vreme uzorkovanja pozicije i orijentacije, odnosno vremenski interval između dva uzastopna merenja ovih vrednosti.

4. PRAĆENJE PUTANJE

Da bi se robot kretao po putanji opisanom Hermitovim splajnom potrebno je pronaći zakonitost po kojoj se ugaona i linearna brzina robota preslikavaju na brzine pogonskih motora. Posmatramo robota sa slike 1.

Gde je

- v – pravolinijska brzina robota
- w – ugaona brzina robota
- r – poluprečnik pogonskih točkova
- $\frac{L}{2}$ – polovina rastojanja između pogonskih točkova

Iz klasične mehanike znamo da su brzine centara levog i desnog točka respektivno

$$v_r = v + \omega \frac{L}{2} \quad (21)$$

$$v_l = v - \omega \frac{L}{2} \quad (22)$$

Do brzina pogonskih motora tada dolazimo primenom sledećih jednačina

$$\omega_r = \left(v + \omega \frac{L}{2} \right) \frac{z}{r} \quad (23)$$

$$\omega_l = \left(v - \omega \frac{L}{2} \right) \frac{z}{r} \quad (24)$$

gde je z prenosni odnos između pogonskih točkova i motora.

$$z = \frac{z_t}{z_m}$$

Za regulaciju brzina pogonskih motora upotrebljen je PID regulator opisan sa (25).

$$\omega_{r/l}(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (25)$$

gde je $e(t)$ razlika između referente i stvarne brzine motora, dok su K_p , K_i i K_d proporcionalno, integralno i diferencijalno pojačanje, respektivno. Za određivanje brzine motora upotrebljen je par optičkih enkodera.

5. REZULTATI RADA

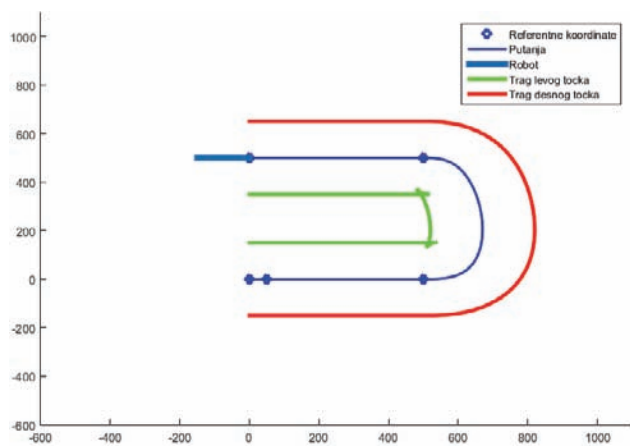
Implementacija algoritama je izvršena na STM32F4VG07 ARM Cortex-M4 mikrokontroleru u C programskom jeziku. Kompletan izvorni kod je dostupan na [9]. Prilikom simulacije¹ pretpostavljen je mobilni robot sa diferencijalnim pogonom i sledećim parametrima: $L = 300 \text{ mm}$, $r = 75 \text{ mm}$ i $z = \frac{1}{2}$. Parametri putanje su prikazani u tabeli 1.

¹ Simulacija je izvršena bez upotrebe PID regulatora usled odsustva dinamičkog modela robota

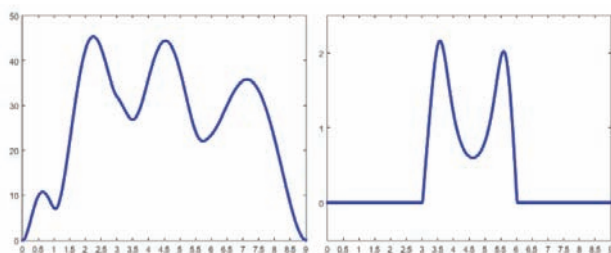
Tabela 1. Željene tačke za interpolaciju

i	t [s]	x [mm]	y [mm]	θ [°]
0	0	0	0	0
1	3	500	0	0
2	6	500	500	180
3	9	0	500	180

Na slici 2. se nalazi rezultatujući interpolant, dok su na slici 3. prikazane pravolinijska i ugaona brzina robota.

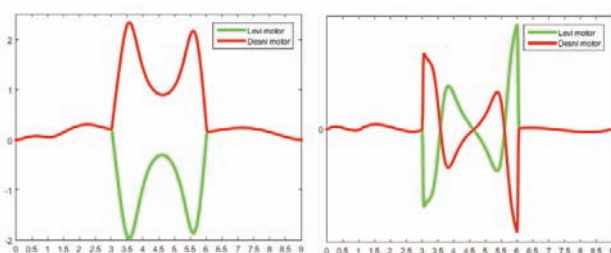


Slika 2. Rezultatujući interpolant (plava) i tragovi levog (zeleni) i desnog (crveni) pogonskog točka



Slika 3. Pravolinijska (levo) i ugaona (desno) brzina robota

Brzine i ubrzanja motora koja odgovaraju kretanju robota prikazanog na slikama 2 i 3 su predstavljena na slici 4.



Slika 4. Brzine (levo) i ubrzanja (desno) levog (zeleni) i desnog (crveni) pogonskog motora

Pored prikazanog kretanja unapred, razvijene metode pokrivaju situacije kretanja u nazad kao i rotaciju robota u mestu.

6. ZAKLJUČAK I DALJI RAD

Sa slika 2., 3. i 4. se može videti da metode opisane u radu pružaju kontinualnost prvog i drugog reda duž cele putanje.

U realnim primenama, mana metode se ogleda u optimizaciji parametara PID regulatora jer se u odsustvu preciznog modela robota postupak svodi na njihovo empirijsko određivanje. Jedan od aspekata daljeg razvoja se ogleda u interfejsu propisivanja putanje robota i može uključivati apstrakciju informacije o vremenskim trenucima na viši nivo, odnosno, potrebno je razviti postupak utvrđivanja vremenskih trenutaka u kojima robot treba da dostigne željene pozicije u pristupu kinematskih ograničenja kretanja.

7. LITERATURA

- [1] Klančar, Gregor, Drago Matko, Sašo Blažič: "Mobile robot control on a reference path" (2005), Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation. IEEE
- [2] Kjargaard, Morten, Nils Axel Andersen, Ole Ravn: "Generic trajectory representation and trajectory following for wheeled robots" (2014), Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE
- [3] Yang, Gil Jin, Byoung Wook Choi: "Robot Velocity Based Path Planning Along Bezier Curve Path" (2013), International Conference on Advanced Science and Technology, ASTL. Vol. 20.
- [4] J S. Wit: „Vector Pursuit Path Tracking for Autonomous Ground vehicles” (2000), Ph.D thesis, University of Florida
- [5] Khatib, Oussama. "Real-time obstacle avoidance for manipulators and mobile robots." The international journal of robotics research 5.1 (1986): 90-98.
- [6] Borenstein, Johann, and Yorem Koren. "Real-time obstacle avoidance for fast mobile robots." Systems, Man and Cybernetics, IEEE Transactions on 19.5 (1989): 1179-1187.
- [7] L. Ridgway Scott: „Numerical Analysis” (2011), Princeton University Press, ISBN-13: 978-0691146867
- [8] D. S. G. Pollock: „A Handbook of Time-series Analysis, Signal Processing and Dynamics” (1999), ISBN: 978-0-12-560990-6
- [9] <https://github.com/markotikvic/malgo-winhost>

Kratka biografija:

Marko Tikvić rođen je 10.9.1991. u Zrenjaninu. Diplomirani inženjera mehatronike je stekao na fakultetu tehničkih nauka na novosadskom univerzitetu 2014. godine, odbranom rada sa temom „Generisanje i praćenje putanje mobilnog robota sa diferencijalnim pogonom primenom polinomijalne interpolacije višeg reda”.

RAZVOJ UPRAVLJAČKOG SISTEMA ZA UPRAVLJANJE STANJIMA U STAMBENIM OBJEKTIMA

DEVELOPMENT OF CONTROLLING SYSTEM FOR CONTROL SITUATION IN RESIDENTIAL BUILDING

Slobodan Mijailović, Fakultet tehničkih nauka, Novi Sad

Oblast – MEHATRONIKA

Kratak sadržaj –U ovom radu je prikazan upravljački sistem za upravljanje stanjima uređaja u stambenim objektima. Uređaji kojima se upravlja su: tri električne sijalice i jedan rashladni uređaj. Upravljanje se vrši preko RS232 komunikacije. Preko nje se upravlja izlaznim stanjima.

Abstract – This paper presents a control system for the devices in residential buildings. Conditions which are managed are three electric lamps and a cooling device. Control is done via RS232 communication.

Ključne reči: upravljački sistem, RS232 komunikacija, Mikrokontroler – PIC16F887, pametna kuća

1. UVOD

Tema ovog rada je razvoj upravljačkog sistema za upravljanje stanjima u stambenim zgradama. Potrebno je analizirati problem razvoja upravljačkog sistema i razviti hardver i softver. Razvojem ovog sistema dobija se upravljanje stanjima sa jednog mesta što dovodi korak bliže razvoju sistema za pametnu kuću. Razvojem tehnologije opada i cena pametnih kuća. U ovom radu će biti objašnjeno i pokazano kako da putem računara upravljamo stanjima koji mogu da uključuju i isključuju aparate i uređaje koji se napajaju električnim naponom iz mreže. Konkretno će biti prikazano kako se vrši upravljanje i izrada potrebne izvršne elektronike i upravljačkog softvera za komunikaciju računara i krajnjeg uređaja pomoću programa napisanog u Microsoft Visual Studio (C#).

2. PRENOS PODATAKA

Prenos podataka predstavlja prenos kodirane informacije. Obično se prenos podataka odnosi na serijski prenos podataka. Nezavisno od toga što se u većini slučajeva podrazumeva serijski prenos podataka, prenos podataka se može ostvariti kao:

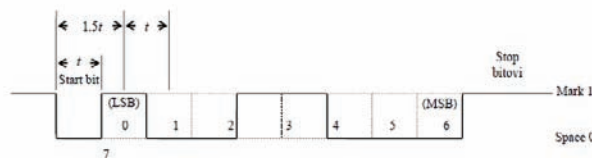
- **paralelni** - kada se prenos većeg broja bitova podataka vrši istovremeno (tipično se ovakav prenos ostvaruje preko sistemske magistrale ili neke druge magistrale za proširenje, tj. preko većeg broja vodova);

- **serijski** – gde se prenos podataka vrši bit po bit preko jedinstvenog voda ili komunikacionog kanala (tipičan primer je komunikaciona linija).

Serijski prenos se sastoji iz dve vrste prenosa:

- **Sinhroni prenos** – prenosi se čitav paket podataka;
- **Asinhroni prenos** – prenosi se karakter po karakter.

Kod asinhronog prenosa se prosleđuju samo podaci koji se menjaju u vremenu. Na slici 1, može se videti talasni oblik kod asinhronog prenosa jednog znaka.

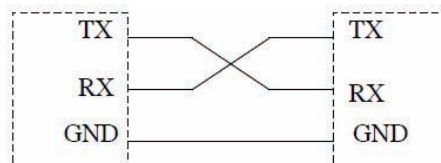


Slika 1. Talasni oblik kod asinhronog prenosa jednog znaka;

Svaki bit se predaje u toku jednog bitskog intervala. Kada se po liniji ne prenose podaci za signal kažemo da je u stanju "marking", što odgovara logičkoj jedinici. Na početku svakog znaka, signal se postavlja na stanje logičke 0 u trajanju od jednog bitskog intervala. Ovaj bit se zove start bit, i on označava vremensku referencu prijemnika. Nakon start bita, u toku svakog narednog bitskog intervala, generiše se po jedan bit podataka, a obično prenos počinje sa LS bitom. Broj bitova podataka može biti 5-8 o čemu postoji dogovor između predajne i prijemne strane. Nakon zadnjeg bita podataka, može slediti bit parnosti. Na kraju sledi predaja jednog ili većeg broja stop bitova.

2.1. RS232 komunikacija

RS232 [1] je asinhrona serijska komunikacija namenjena za kraća rastojanja i manje brzine prenosa. Minimalni broj veza je masa plus jedna linija za svaki smer komunikacije. Komunikacija je dupleksna, što znači da komunikacioni uređaj može istovremeno da šalje i prima podatke. Na slici 2, može se videti tip potpune duplex komunikacije.

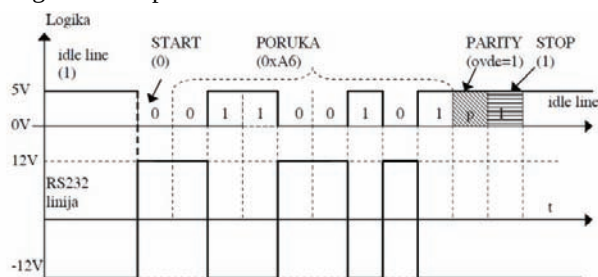


Slika 2. Potpuna dupleks komunikacija između dva uređaja;

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji je mentor bila dr Gordana Ostojić, van. prof.

Standard RS232 je definisan u ANSI (American National Standards Institute) specifikaciji kao interfejs između data terminala (DTE – data terminal equipment) i data kola (DCE – data circuit-terminating equipment), pri čemu se koristi serijski binarni protok podataka. RS232 standard dozvoljava samo dva fizička stanja na liniji za prenos. Prvi naponski nivo je -12V na liniji koji predstavlja ON stanje ili broj 1 ili marker. Drugi naponski nivo je +12V na liniji koji predstavlja OFF ili broj 0 ili prazno mesto. Kada nema transfera poruke, linija je na -12V što predstavlja stanje logičke 1 ili binarni broj 1. Za razliku od neaktivnih -12V, prvi bit u poruci je START bit i uvek je naponskog nivoa +12V, time signalizira početak poruke. Na slici 3, može se videti uobičajni prenos bitova sa logičkim i naponskim nivoima.



Slika 3. Uobičajni prenos bitova sa logičkim i naponskim nivoima;

Za komunikaciju RS232 se najčešće koriste brzine dobijene deljenjem 115200 sa celim brojem, kao na primer 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 i 115200 Bd (Bauda), što označava brzinu broj prenetih bita u sekundi. Da bi komunikacija bila moguća, predajnik i prijemnik moraju biti podešeni na istu brzinu, jer se takt ne prenosi posebno.

3. RAZVOJ HARDVERA I SOFTVERA UPRAVLJAČKOG SISTEMA ZA UPRAVLJANJE STANJIMA U STAMBENIM OBJEKTIMA

Za razvoj hardvera napravljena je jedinstvena električna šema i prateći softver koji se nalazi na računaru preko kojeg se vrši upravljanje. Softverom, aplikacijom, upravlja se preko RS232 komunikacije sa mikrokontrolerom koji dalje kontroliše izlazne releje na kojima se nalaze uređaji kojima se upravlja. U daljem tekstu će biti objašnjeno hardversko i softversko rešenje upravljačkog sistema za upravljanje stanjima u stambenim objektima.

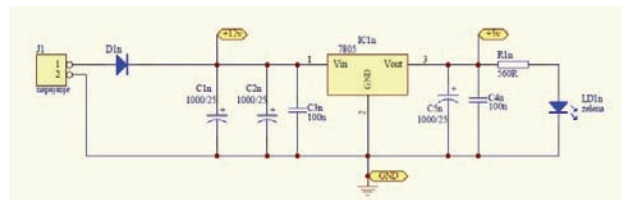
3.1. Hardversko rešenje upravljačkog sistema za upravljanje stanjima u stambenim objektima

Električna šema hardverskog rešenja upravljačkog sistema sastoji se iz sledećih celina:

- Kolo za napajanje;
- Mikrokontroler – PIC16F887;
- Komunikaciono kolo – RS232 komunikacija;
- Displej za indicaciju stanja – WH1602B;
- Izvršna kola;

3.1.1. Kolo za napajanje

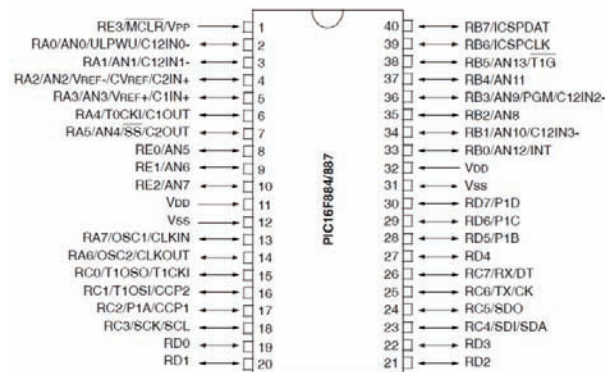
Mikrokontroleri su vrlo osetljivi na nestabilni napon i zbog toga je neophodno stabilizovati napon. U realizaciji ovog projekta korišćeno je integrisano kolo LM7805. Na slici 4. prikazana je električna šema napajanja.



Slika 4. Električna šema za napajanje;

3.1.2. Mikrokontroler – PIC 16F887

Za izradu ovog projekta korišćen je 8 – bitni mikrokontroler [2] firme MicroChip model 16F887. Ovo integrisano kolo ima 40 pinova. Izgled i raspored pinova može se videti na slici 5.



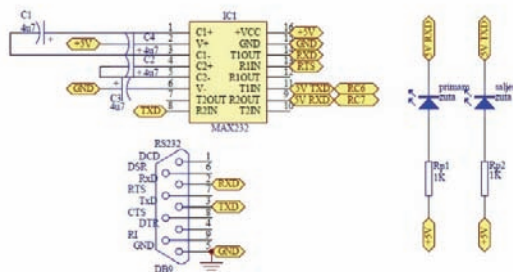
Slika 5. Raspored pinova kod PIC16F887;

Maximalni radni takt mikrokontrolera ide do 20 MHz. Ukupno ima 36 ulazno/izlaznih pinova koji su raspoređeni u 5 portova. U implementaciji rešenja korišćen je kvarc od 8 MHz za radni takt.

3.1.3. Komunikaciono kolo – RS232 komunikacija

Komunikacija između aplikacije koja se nalazi na računaru i mikrokontrolera odvija se preko RS232 porta. Kolo za komunikaciju prikazano je na slici 6.

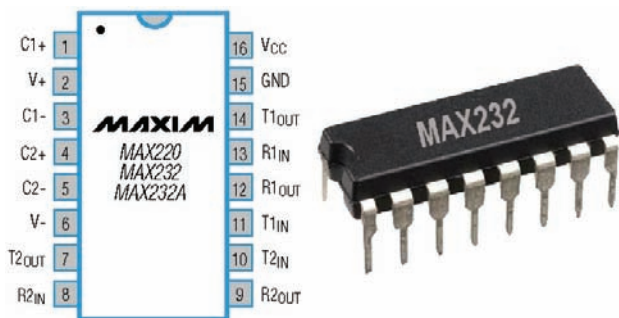
Logički nivoi kod računara i mikrokontrolera se razlikuju i iz tog razloga koristi se integrisano kolo MAX232 za konvertovanje naponskih nivoa, a komunikacija se odvija preko 9 – pinskog konektora.



Slika 6. Izgled šeme kola za komunikaciju;

3.1.3.1. Integrisano kolo MAX232

MAX232 je integrisano kolo koje ima široku primenu u RS232 komunikaciji da konvertuje naponske nivoe za komunikaciju sa TTL uređajima [3]. Izgled i raspored pinova može se videti na slici 7.



Slika 7. Izgled i raspored pinova kod MAX232 kola;

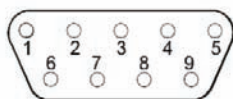
Vrednost napon napajanja (Vcc) je između +4,5V i +5,5V. Naponski nivoi kod RS232 komunikacije su sledeći, logička 0 se nalazi između napona +3V i +15V, a logička 1 između -15V i -3V.

U TTL logici, logička 0 se nalazi između napona 0V i +0,8V, a logička jedinica između +2V i +5V.

I zbog toga je potrebno koristi MAX232. Sastoji se od dva prijemnika (konvertuje potrebne napone od RS232 za TTL logiku) i dva predajnika (konvertuju napone od TTL logike na potrebne nivoe napona sa RS232 komunikaciju).

3.1.3.2. Konektor za komunikaciju – DSUB 9 pin

Konektor koji je konkretno korišćen u izradi ovog projekta je muški 9 – to pinski konektor [4]. Rapored pinova može se videti na slici 8.



Slika 8. Raspored pinova;

U projektu korišćeni su:

- pin 2 - serijski ulaz podataka;
- pin 3 - serijski izlaz podataka;
- pin 5 - masa;

Pomoću njih je ostvarena poptuna dupleks komunikacija.

3.1.4. Displej za indikaciju stanja – WH1602B

U ovom projektu korišćen je LCD displej [5] sa po 16 karaktera u dva reda za vizuelno informisanje o izvršenoj akciji. Komunikacija između displeja i mikrokontrolera odvija se preko pinova porta RB (od RB0 do RB5).

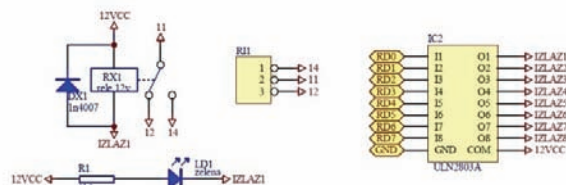
3.1.5. Izvršna kola - releji

U izvršnom kolu korišćeni su releji. Releji su prekidači koji otvaraju i zatvaraju kola elektromehaničkim ili elektronskim putem.

Oni kontrolišu jedno strujno kolo otvaranjem i zatvaranjem kontakta u drugom kolu. Postoje normalno otvoreni (NO – normal open) i normalno zatvoreni kontakti (NC – normal close).

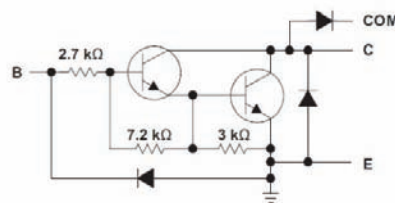
Relejima upravlja mikrokontroler preko porta D. Ukupno ima 8 releja.

Na slici 9. prikazan je jedan kanal.



Slika 9. Prikaz jednog kanala;

Izlazni port D mikrokontrolera povezan je sa integrisanim kolom ULN2803A [6]. ULN2803 je integrisano kolo koje se sastoji od 8 kanal od kojih se svaki sastoji od Darlingtonovog tranzistorskog spoja. Darlingtonov spoj ima bazne otpornike veličine 2,7 KΩ, što omogućava direktan rad sa TTL kolima. Prikaz jednog takvog kanala integrisanog kola ULN2803A dat je na slici 10.

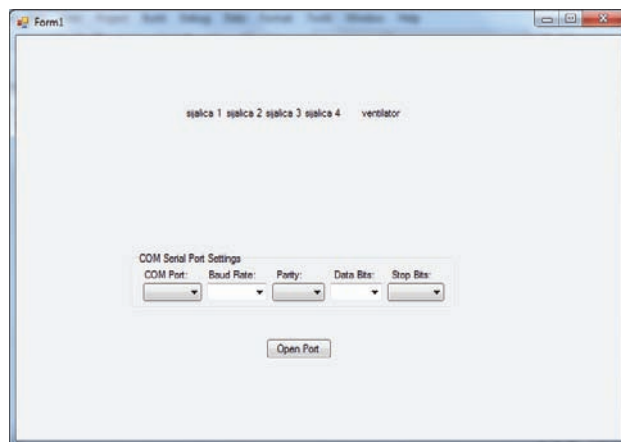


Slika 10. Darlingtonov spoj;

Releji su induktivni potrošači koji prilikom ukidanja napajanja teže da uspostave ravnotežu, tako da se na njegovim krajevima indukuje kontra elektromotorna sila, koju je potrebno kompenzovati jer bi u suprotnom moglo doći do nekog oštećenja na ploči. Zbog toga u projektu su korišćene zamajne diode kod svakog releja.

3.2. Softversko rešenje upravljačkog sistema za upravljanje stanjima u stambenim objektima

Softversko rešenje realizovano je u Microsoft Visual Studio (C#). Aplikacija se nalazi na računaru i preko nje se upravlja izvršnim elementima. Komunikacija između aplikacije i upravljačke ploče odvija se preko RS232 komunikacije, koja komunicira sa mikrokontrolerom i dalje mikrokontroler uključuje i isključuje releje. Startovanjem programa upravljačkog softvera pokreće se aplikacija čiji je izgled dat na slici 11.



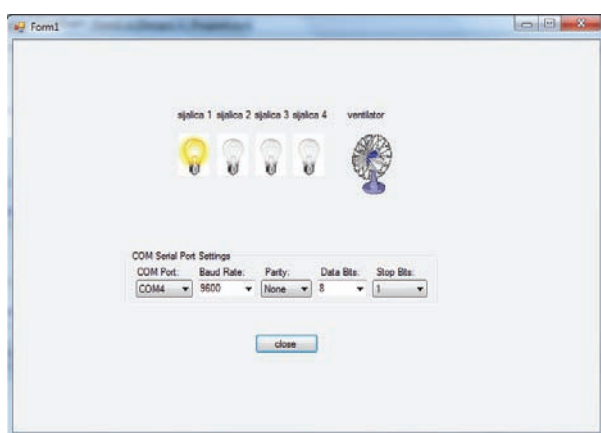
Slika 11. Izgled upravljačkog softvera;

Potrebno je konfigurisati COM port odabirom vrednosti iz padajućih menija. Nakon toga pritisnuti taster "Open Port". Nakon izvršenja ovog dela, novi prozor aplikacije izgleda kao na slici 12.



Slika 12. Izgled prozora aplikacije nakon konektovanja na COM port;

Pritiskom na "sijalica 1" menja se izgled aplikacije kao na slici 13.



Slika 13. Izgled prozora nakon aktiviranja "sijalica 1";

Pored toga što se promenio izgled aplikacije, ujedno je i softver poslao na COM port karakter A. Pri realizaciji ovog projekta odlučeno je da veliko slovo označava naredbu mikrokontroleru da odgovarajuće stanje izlaza bude aktivno, a malo slovo označava naredbu mikrokontroleru da odgovarajuće stanje izlaza bude neaktivno.

Izgled gotove pločice sa raspoređenim elementima može da se vidi na slici 14.



Slika 14. Izgled gotove pločice

4. ZAKLJUČAK

Prilikom izrade hardvera za upravljački softver korišćene su sve dostupne i standardne komponente. Prednost ovoga je dostupna nabavka istih tih komponenti koje su korišćene u radu. Kobinovanjem tih komponenti došlo se do jednostavnog upravljačkog sistema. Upravljački sistem trenutno je projektovan da se vrši upravljanje sa samo četiri izlazna releja, ali na štampanoj pločici je dato rešenje za istovremeno upravljanje sa osam izlaznih releja gde se na vrlo lak i jednostavan način može prepraviti kod mikrokontrolera. Mana ovog projekta je što nema neko bežično upravljanje, ali daljim razvojem se može usavršiti ovaj upravljački sistem.

5. LITERATURA

- [1] *Electronic Industries Association. Engineering Dept, EIA standard RS-232-C: Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange, Washington 1969;*
- [2] *PIC16F887, Datasheet, MicroChip, <http://ww1.microchip.com/downloads/en/devicedoc/41291f.pdf>;*
- [3] *MAX220/MAX249, Datasheet, www.maximintegrated.com;*
- [4] *Pinout of PC serial port (RS – 232 DB9) pinout, Tomi Engdahl, 2015;*
- [5] *WH1602B, Datasheet, <http://pdf1.alldatasheet.com/datasheetpdf/view/619137/DBLECTRO/WH1602B.html>*
- [6] *ULN2803, Datasheet, Texas instruments, <http://www.ti.com/product/uln2803a?qqpn=uln2803a>;*

Kratka biografija:



Slobodan Mijailović rođen je u Somboru 1987. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mehatronika, robotika i automatizacija odbranio je 2015.god.

УЛТРАЗВУЧНО МЕРЕЊЕ НИВОА ТЕЧНОСТИ**ULTRASONIC FLUID LEVEL MEASUREMENT**Bojan Lazarevski, *Fakultet tehničkih nauka, Novi Sad***Oblast – MEHATRONIKA**

Kratak sadržaj – U ovom master radu izrađen je uređaj za merenje nivoa tečnosti u rezervoaru uz pomoć ultrazvuka. Merenje se zasniva na merenju vremena prostiranja zvuka. Impulsi odašiljani uz pomoć primopredajnika prolaze kroz tečnost, tipično od dna do površine tečnosti. Površina tečnosti generiše odbojni talas, koji se vraća do primopredajnika. Uređaj za merenje nivoa tečnosti se sastoji od elementa TDC1000 i mikrokontrolera Atmega8.

Abstract – This master thesis is about development of a device for ultrasonic fluid level measurement in a tank. Measurement is based on measuring total time-of-flight of the sound wave in fluid. The pulses transmitted by transducer travel through the fluid, typically from the bottom, to the surface of the fluid. The discontinuity between the fluid and air generates a reflected wave which returns back to transducer.

Ključne reči: Ultrazvučno merenje, merenje nivoa tečnosti.

1. UVOD

Potrebe čoveka za merenjem nivoa tečnosti datiraju još iz perioda pre industrijskog doba. Svaka civilizacija u istoriji ima zapise o načinima merenja određenih veličina. Egipćani su pre više od 2000 godina zahvaljujući praćenju nivoa reke Nil, mogli da upravljaju složenim sistemom za navodnjavanje, koje je podrazumevalo blagovremeno obaveštenje o porastu nivoa vode [1]. Dalji razvoj načina merenja nivoa su vezani za industrijsku revoluciju. Zbog potrebe unapređivanja parne mašine, bilo je neophodno omogućiti merenje temperature, pritiska i protoka.

Nivo predstavlja visinu tečnosti ili usitnjenog materijala u posudi. Merenje nivoa medijuma je neophodno za upravljanje procesima, kao i za kontrolu rada nekih tehnoloških aparata [2]. U ranim industrijskim postrojenjima, postalo je neophodno pratiti više promenljivih, u to vreme, preciznost merenja i način je mnogo zavisio od ljudi koju su bili zaduženi za čitanje rezultata merenja. Merni instrumenti su postavljani na samim mestima merenja. Zatim se kasnije javila potreba za prenošenjem mernih signala na udaljene merače [3].

U početku se to vršilo slanjem hidrauličnih impulsa na udaljene lokacije, da bi kasnije bilo zamenjeno električnim impulsima. U novijem dobu rezultati merenja su počela da se koriste za sisteme povratne sprege, kako bi se omogućilo automatsko upravljanje sistemima.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Stevan Stankovski, red. prof.

Zadatak rada je bilo omogućiti merenje nivoa tečnosti u plastičnom rezervoaru visine 19 cm, korišćenjem neinvazivne metode. Kao odgovarajući način merenja izabrano je ultrazvučno merenje, upotrebom AVR mikrokontrolera, i uređaja za ultrazvučno merenje TDC1000, uz izradu pratećeg računarskog softvera.

2. PREGLED NAČINA MERENJA NIVOVA TEČNOSTI**2.1. Merno staklo**

Jedno od najjednostavniji i najstarijih načina merenja nivoa tečnosti je merno staklo, ili merna cev, koja je bila povezana sa dnom i vrhom rezervoara. Nivo tečnosti u cevi je bio isti kao i nivo tečnosti u rezervoaru. Ovaj način merenja se koristio na parnim mašinama. Prednost ovakvog načina merenja je lak način montaže. Mane su jer tačnost merenja zavisi od čoveka, mogućnost česte pojave curenja tečnosti zbog dotrajalih zaptivnih komponenta.

2.2. Detektori nivoa na principu plovka

Detektor nivoa se sastoji od plovka loptastog oblika koji pliva na površini tečnosti čiji se nivo meri. Mehaničkom vezom položaj plovka se prenosi na senzor položaja. Postoje dva načina izvedbe, jedan način je merenje ugaonog pomeraja, drugi je merenje linearnog pomeraja, u zavisnosti od izvedbe mehaničkog dela. Prednost je relativno jeftina ugradnja. Mana je mali opseg, postojanje pokretnih delova.

2.3. Hidrostatički senzori nivoa

Princip hidrostatičkih senzora zasniva se na primeni Paskalovog zakona, pomoću koga se izračunava vrednost pritiska u odnosu na gustinu tečnosti i visinu. Merenje obuhvata najčešće merenje razlike između hidrostatičnog pritiska na dnu rezervoara, i atmosferskog pritiska ili pritiska u gornjem delu zatvorenog rezervoara. Računa se razlika između ta dva pritiska, na osnovu čega se utvrđuje visina tečnosti. Najčešće se koristi za merenje nivoa u cevima i rezervoarima malog prečnika. Prednost je što nema pokretnih delova i jednostavan način rada. Moguća je primena u otvorenim i zatvorenim rezervoarima, sa zapaljivim ili hemijski agresivnim tečnostima.

2.4. Ultrazvučni senzor nivoa

Ultrazvučni senzor nivoa je zasnovan na odbijanju zvučnih, ultrazvučnih ili mikrotalasnih zračenja od površine fluida. Nivo medijuma je proporcionalan vremenu za koje talas pređe od izvora do prijemnika zračenja. Takođe je moguće meriti i udaljenost razdelne površine između dva fluida koji se ne mešaju. Montaža ultrazvučnog senzora se vrši na vrhu rezervoara,

neophodno je da se postavi tako da senzor registruje nivo najmirnijeg dela razdelne tečnosti. Najčešće se koristi za merenje nivoa tečnosti i sitnozrnastih materijala.

3. OPIS KONFIGURACIJE

Rezervoar korišćen u ovom radu, prikazan na slici 1, izrađen je od plastike, visine 19 cm, širine 16 cm, dužine 13 cm.



Slika 1. Plastični rezervoar

Kao medijum za testiranje je korišćena voda iz česme. Na dno rezervoara je postavljena odvodna cev, kako bi bilo moguće lakše menjanje nivoa tečnosti. S obzirom da rezervoar nema ostavljeno mesto za postavljanje bilo kakvog merača tečnosti, jedina mogućnost je vizuelno utvrđivanje nivoa, ili naknadna ugradnja merača na mesto čepa.

4. ODABIR REŠENJA

S obzirom da je bilo neophodno naći rešenje za merenje nivoa tečnosti za već postojeći rezervoar, neinvazivnom metodom, a uzimajući u obzir veličinu rezervoara, od ponuđenih rešenja, izabrano je ultrazvučno merenje kao najbolje rešenje. Međutim ultrazvučno merenje se zasniva na merenju udaljenosti površine tečnosti od vrha rezervoara, što bi u ovom slučaju bilo moguće samo postavljanjem merača umesto čepa posude. Kao najbolje rešenje odlučena je upotreba uređaja za ultrazvučno merenje proizvođača „Texas Instruments“, TDC1000, koje je namenjeno za ultrazvučno merenje nivoa tečnosti, gde se primopredajnik postavlja na dno rezervoara sa spoljne strane.

5. OPIS NAČINA RADA

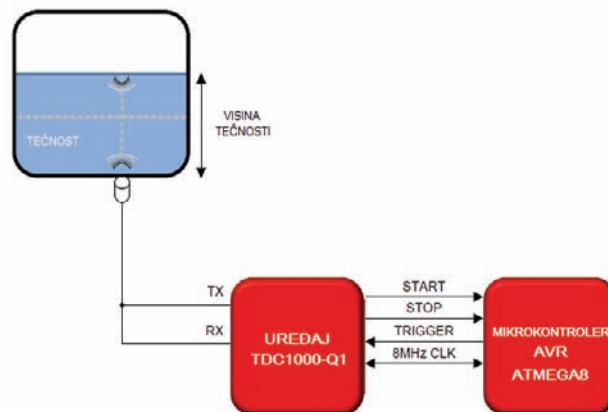
Merenje uz pomoć uređaja TDC1000 se zasniva na merenju vremena prostiranja zvuka. Prednost ovog načina merenja je što je moguće postaviti primopredajnik sa spoljne strane rezervoara, nema pokretnih delova, greška merenja je manja od 1 mm, dok je ultrazvučni metod za merenje otporan na spoljašnje uticaje.

Ultrazvučno merenje nivoa tečnosti koristi jedan piezoelektrični primopredajnik kako bi generisao impuls sa dna rezervoara. Taj impuls putuje kroz zid rezervoara, zatim kroz tečnost u rezervoaru dok ne dođe do površine tečnosti.

Na površini tečnosti jedan deo impulsa se odbija, i na taj način nastaje eho. Vreme koje je potrebno ehu da se vrati

do prijemnika predstavlja visinu nivoa tečnosti. Uređaj se sastoji iz tri dela, prikazanoj na slici 2, piezoelektričnog primopredajnika, uređaja TDC1000 koji služi za posredovanje između primopredajnika i mikrokontrolera.

Uređaj TDC1000 konvertuje analogne signale u digitalne, koji predstavljaju početak merenja (START) i kraj merenja (STOP) vremena prostiranja ultrazvuka [4].



Slika 2. Blok dijagram ultrazvučnog merenja

5. IMPLEMENTACIJA SISTEMA

5.1. Odabir elemenata

Na osnovu prateće dokumentacije za korišćenje uređaja TDC1000, prilikom merenja tečnosti, preporuka proizvođača je da se koristi primopredajnik rezonantne učestanosti 1 MHz.

Na osnovu toga, izabran je piezoelektrični element prečnika 10 mm, i debljine 2 mm proizvođača „STEMNIC“, oznake SMD10T2R111. Za upravljanje merenjem potreban je bio mikrokontroler sa mogućnošću korišćenja prekida programa usled spoljašnjih događaja, merač proteklog vremena, takođe bilo je neophodno da poseduje UART komunikaciju za komuniciranje preko bluetooth-a, kao i SPI komunikaciju za podešavanje parametara uređaja za ultrazvučno merenje.

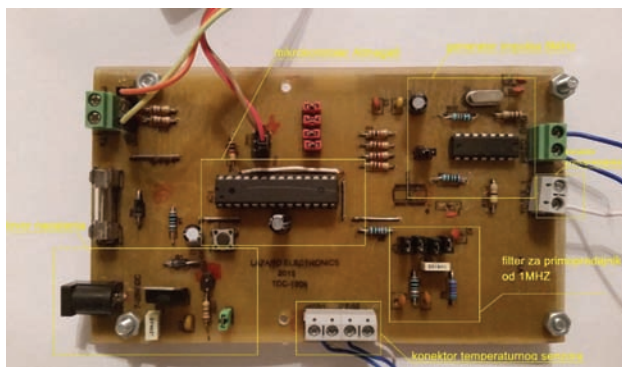
Kao pogodan za ovu primenu, izabran je mikrokontroler Atmega8 proizvođača „Atmel“, koji poseduje 16-bitni vremenski merač, programibilnu UART komunikaciju, SPI serijski interfejs i mogućnost izvršavanja dela programa prouzrokovano promenom stanja na određenom ulazu [5].

Bilo je neophodno napraviti štampanu ploču, koja poseduje izvor napajanja jednosmernog napona od 5 V, generator impulsa od 8 MHz, ulaz za dva temperaturna senzora PT1000, konektor za primopredajnik, i deo za UART komunikaciju preko bluetooth modula, takođe, ploča poseduje i filterski blok za uređaj TDC1000 za primopredajnik od 1 MHz prema preporuci proizvođača.

Izgled izrađene štampane ploče sa označenim blokovima prikazan je na slici 3.

5.2. Instalacija hardvera

Piezoelektrični element je neophodno zalepiti za dno rezervoara sa spoljašnje strane.



S obzirom da je neophodno imati dovoljno elastičnu vezu između rezervoara i primopredajnika, a sa druge strane dovoljno jaku, kao lepilo korišćen je superlepak prema preporuci proizvođača [7]. Nakon sušenja lepka, piezoelement je zaliven topljivom plastikom. Zalepljen primopredajnik je prikazan na slici 4.

Promena temperature medijuma može bitno promeniti brzinu prolaska ultrazvuka kroz mereni medijum, iz tog razloga neophodno je temperaturni senzor PT1000 uroniti u tečnost.

vremenu, neophodno je znati i brzinu prostiranja zvuka kroz mereni medijum, što se može utvrditi eksperimentalno uz dodatne mogućnosti uređaja TDC1000.

Kod ovog programa za brzinu prostiranja zvuka kroz vodu iz česme je primenjeno je 1483.13 m/s. Nivo se izračunava na osnovu izraza (1), gde d predstavlja nivo tečnosti u metrima, T proteklo vreme od slanja do primanja zvučnog impulsa izraženo u sekundama, i C brzinu prostiranja zvuka kroz medijum izraženo u m/s.

$$d = \frac{F_{\text{res}}}{g}. \quad (1)$$

Nakon izračunavanja, mikrokontroler putem bluetooth modula šalje izračunatu visinu izraženu u metrima. Ukoliko se dogodi greška prilikom merenja, TDC1000 šalje impuls greške na ulaz mikrokontrolera. Tada mikrokontroler putem bluetooth modula šalje poruku „ERROR“ i vreme koje je proteklo od početka merenja do događanja greške.

6. OPIS I PRIKAZ RADA KORISNIČKOG INTERFEJSA

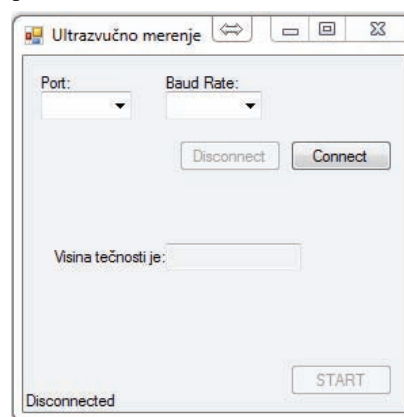
6.1. Opis programa

Program je pisan u programu „Microsoft Visual Studio“ u programskom jeziku „Visual C#“ [8]. Zadatak programa je da uspostavi komunikaciju sa mikrokontrolerom preko COM porta na računaru, zatim nakon uspostavljene komunikacije da korisniku omogući početak merenja i ispis rezultata na ekranu, kako bi na taj način korisniku bilo omogućeno jednostavno korišćenje uređaja za merenje.

6.2. Način upotrebe programa

Prilikom pokretanja programa, iz padajućeg menija neophodno je izabrati odgovarajući COM port, zatim brzinu komunikacije od 9600 bps, nakon toga neophodno je kliknuti na Connect, kako bi se uspostavila veza. Ukoliko je uspostavljanje veze uspješno, u donjem levom uglu prozora ispisuje se „Connected“, u slučaju da povezivanje nije uspjelo, u donjem levom uglu prozora ispisuje se „ERROR“.

Kada je veza uspostavljena merenje se vrši klikom na „START“, nakon završenog merenja u prozoru se ispisuje visina tačnosti izražena u metrima. Prikaz pokrenutog programa je prikazan na slici 5.



Slika 5. *Prikaz prozora programa*

7. REZULTATI RADA I DALJA UNAPREĐENJA

Prilikom testiranja uređaja, bilo je neophodno prvo utvrditi odgovarajuća podešavanja registara uređaja TDC1000. S obzirom da se piezoelektrični element ponaša kao specifičan RLC rezonantni filter, nakon prestanka slanja impulsa oscilacije se nastavljaju. Iz tog razloga neophodno je odrediti vreme potrebno da oscilovanje prestane, kao i broj poslatih impulsa, kako bi neželjeno oscilovanje bilo zamaskirano.

Međutim povećanjem vremena maskiranja smanjuje se minimalna visina očitavanja tečnosti. Nakon eksperimentisanja korišćenjem različitih vremenskih intervala, utvrđeno je optimalno rešenje da vreme trajanja maskiranja traje 8 μ s, što u ovom slučaju omogućava praćenje nivoa tečnosti od 1.1 cm. Kod početnih testiranja javljao se problem previše slabog prijemnog signala, to je rešeno podešavanjem registra za pojačavanje signala. Na ovaj način prijemni signal se pojačava 15 dB.

Eksperimentisanje je prema preporuci proizvođača vršeno dok je nivo u rezervoaru bio 3 cm. Nakon pronalaženja odgovarajućih parametara, pristupilo se testiranju merenja na različitim nivoima. Rezultati merenja su bili odgovarajući, greška merenja je bila približno 3 mm. Na kraju testiranja uspešno je izmeren minimalni nivo od 1.2 cm i maksimalni nivo od 19 cm.

U daljem razvoju ovog uređaja moglo bi se pristupiti razvijanju odgovarajuće andorid aplikacije preko koje bi bilo moguće prikazivanje nivoa tečnosti. Takođe u zavisnosti od primene, bluetooth modul bi mogao biti zamenjen WiFi modulom, što bi uz pomoć interneta omogućilo očitavanje rezultata merenja sa udaljene lokacije. S obzirom da je potrošnja energije uređaja mala, uređaj bi se mogao prilagoditi za merenje nivoa tečnosti u motornim vozilima, ili mobilnim mašinama, gde bi trebalo voditi računa o greški merenja prilikom pomeranja tečnosti u rezervoaru.

8. ZAKLJUČAK

U ovom radu je uspešno realizovano ultrazvučno merenje nivoa tečnosti u rezervoaru bez upotrebe invazivne metode. S obzirom da je za instaliranje uređaja za merenje nivoa je neophodno zalepiti piezoelektrični element na dno rezervoara, ovaj uređaj je moguće koristiti na bilo kojem rezervoaru koji nema mogućnost merenja nivoa.

Takođe prednost ovog uređaja je što nema pokretnih delova, što znači da ima dugačak vek trajanja, fleksibilan je i ima malu cenu izrade. Pogodan je za merenje nivoa agresivnih i zapaljivih tečnosti jer se svi delovi uređaja nalaze van rezervoara.

9. LITERATURA

- [1] Felton, Bob. "Level Measurement: Ancient Chore, Modern Tools." ISA, August 2001.
- [2] http://www.termoenergetika.biz/doc/projektanti/05_Detektori_i_meraci_nivoa.pdf
- [3] Donald R. Gillum "Industrial Pressure, Level, and Density Measurement" ISA, 2009.
- [4] Using Ultrasonic Sensing to Monitor Level in Tanks, Matthew Minasi, Texas Instruments, Jul 2015
- [6] http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontrolleratmega8_1_datasheet.pdf
- [6] <http://www.ti.com/lit/ds/symlink/tdc1000-q1.pdf>
- [7] How to Select and Mount Transducers in Ultrasonic Sensing for Level Sensing and Fluid ID, Matthew Minasi, Texas Instruments, April 2015
- [8] Microsoft Visual C# 2013 Step by Step, John Sharp, Microsoft Press November 15, 2013

Kratka biografija:



Bojan Lazarevski rođen je u Zrenjaninu 1991. god. Master akademske studije na Fakultetu tehničkih nauka upisuje 2014. godine, na smeru Mehatronika, robotika i automatizacija. Master rad odbranio je 2015 god.

BIBLIOTEKA ZA RS485 FULL DUPLEX KOMUNIKACIJU IZMEĐU KONTROLERA KOJI SU DEO UPRAVLJAČKOG SISTEMA ROBOTA MARKO**LIBRARY FOR RS485 FULL DUPLEX COMMUNICATION BETWEEN MICROCONTROLLERS USED IN THE ROBOT MARKO CONTROL SYSTEM**

Tobias Kovačić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – Ovaj rad predstavlja opis biblioteke za ARM mikrokontrolere u kojoj je implementiran protokol za master slave komunikaciju između više mikrokontrolera. Komunikacija se ostvaruje primenom full duplex RS485 standarda. Provera funkcija koje su deo biblioteke je izvršena na kontrolerima STM32F407. U radu su prezentovani rezultati testiranja prenosa podataka i ispitana je robusnost na elektromagnetni šum koji proizilazi od rada motora jednosmerne struje sa četkicama.

Abstract – This work presents description of library for ARM microcontrollers, which are implemented protocol for master slave communication between more microcontrollers. Communication is achieved using of full duplex RS485 standard. Checking the function of library is used STM32F407 microcontrollers. In the work are presented the results of the testing transfer data and for robustness to electromagnetic noise arising from the operation of brushed DC motors.

Ključne reči: Serijska komunikacija, RS485, full duplex, Biblioteka

1. UVOD

U okviru projekta u kojem se razvija robot MARKO [1] se javila potreba za brzu razmenu informacija između više mikrokontrolera. Jedno od mogućih rešenja koja mogu da se primene za ovu namenu je full duplex RS485 standard [2]. Za ovaj komunikacioni standard ne postoji gotova biblioteka prilagođena potrebama projekta koji bi se mogla iskoristiti. Zbog toga je bilo potrebno razviti biblioteku u sklopu koje će se implementirati komunikacioni protokol za razmenu podataka između više kontrolera koji se nalaze u robotu.

2. KOMUNIKACIONI PROTOKOL

Za komunikaciju između mikrokontrolera koji se nalaze u robotu MARKO, razvijen je komunikacioni protokol tako da se omoguće sledeće funkcionalnosti iz ugla master kontrolera: slanje podataka do slave kontrolera, čitanje podataka od slave kontrolera, slanje i čitanje podataka u istom komunikacionom ciklusu. Ako se oba izvršava zajedno, tada mora da prvo čita podatke a posle može slati podatke, da bi vreme bio bolje iskorišćeno.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Mirko Raković.

Tok komunikacije je prikazan na slici 1. Master započinje komunikaciju sa START komandom. Posle šalje adresu izabranog slave uređaja.

Kad slave detektuje START, čeka sledeći podatak, da uporedi sa svojom adresom. Ako je izabran odgovara sa WAIT komandom, ako nije, onda odbaci podatke i čeka sledeći START. Master posle odgovora slave-a nastavlja slanje podataka, funkciju (FUNCTION) upita, 0 za dužinu daljih podataka i CHECK vrednost za proveru ispravnosti glave poruke, u ovom slučaju to je ceo upit.

Za proveru vrednosti se koristi dvodimenzionalni paritet, iz koje jedna dimenzija je hardverski rešen u UART modulu, drugi se softverski računa pomoću bit orijentisanom *eksluziv ili* funkcijom.

Ako slave primi celu glavu i ne nađe razliku kod provere podataka, onda otvara kanal i u daljem se dešava stalna razmena podataka. Pored toga pošalje i da je u ovom slučaju poruka bila u redu, MES_OK.

Posle obrade funkcije upita, kad je spreman odgovor za slanje počinje sa slanjem toga. Ako slave ne šalje odgovor, a nema ni funkcijski kod da šalje onda šalje WAIT funkciju.

Master posle toga kad je poslao upit, započne drugi deo komunikacije, slanje vrednosti.

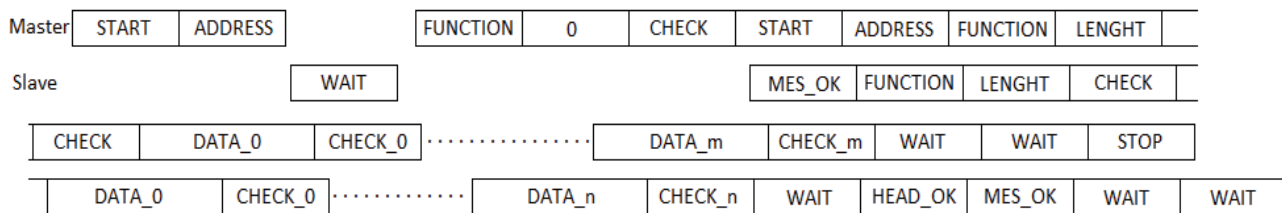
Ponovo se šalje START, adresa slave uređaja (ADDRESS), funkcija (FUNCTION), dužina (LENGTH) podataka koje slede i vrednost za proveru glave poruke. Posle glave šalju se podaci, posle 8 poslanih korisnih podataka (DATA_x) šalje jednu vrednost za proveru ispravnosti poslanih podataka (CHECK_x).

Na kraju po potrebi se dodaje još jedna vrednost za proveru, za podatke od zadnje provere. Ako je master sve poslao šalje WAIT, dok ne primi potvrde od slave-a da je sve primio u redu.

Ako je slave sve potvrdio onda master šalje STOP i zaustavi komunikaciju. Čeka se određeno vreme, da svaki uređaj primeti da je zaustavljena komunikacija i onda počinje komunikacija sa drugim slave uređajem.

Slave striktno se drži protokola, ako primeti grešku i završio sa slanjem svih prethodno primljenih vrednosti, onda pošalje da šta ne valja, glava ili poruka (HEAD_ER ili MES_ER).

Master ako primeti grešku zaustavlja slanje, kad slave primeti prekid, zaustavlja i on slanje, posle izvesnog vremena master pokušava celu komunikaciju iz početka (deo koji je do kraja prošla bez greške ne ponavlja se).



Slika 1. Prikaz razmene podataka u implementiranom komunikacionom protokolu

3. OTPORNOST NA ELEKTROMAGNETNI ŠUM

U radu unutar robota očekivano je slanje struktura koji sadrži redni broj zgloba (int), vrednosti za ugao, brzinu i ubrzanje zgloba (float) i vrednost vremena izvršenje pokreta (float). Napravljen je niz struktura od dva elementa i to je slao master STM32F407 slave uređaju STM32F051.

Slave je pogledao primljenu strukturu, izvršio proveru podataka i u narednom koraku je slao nazad. Sa datim slave uređajem postignuta je brzina komunikacije od 500.000 bps. Podaci su poslani 120.000 puta, za svakih 10ms sledeći. Master je tokom komunikacije slao 55×11 bit, dok slave slao 51×11 bit.

Korisna informacija u oba slučaja je 36×8 bit. Iskorišćenost na bitnom nivou je oko 50%. Za duže podatke iskorišćenost se uvećava. Treba težiti ka što boljim iskorišćenosti ali protokol za opštu namenu ne može da bude efikasan i za kratke i za duže poruke. Neophodni su start i stop bit pri komunikaciji i neophodno je izvršiti adresiranje slave uređaja, kao i korektnu proveru podataka.

Tokom testiranja nije primećena ni jedna greška. Snimak električnih signala na početku prenosa je prikazan na slici 2, najveći šum je oko 0,8V. Sa zelenom linijom je označeno trajanje 11 bita, odnosno jedinica za komunikaciju. Sa narandžastom je označen najveći šum.

Šum koji se javi na prelazu ivica pri bržem snimanju nestaje.

Šum koji se javi, je u trenutku promene na drugom kanalu. Signali su snimljeni na ulaznim pinovima mikrokontrolera, odnosno posle prenosa na 10m sa SF/UTP kablom, sa RS485 konvertorima, koji su napajani sa 3V.

Gore je signal slave-a, dole je signal od mastera.

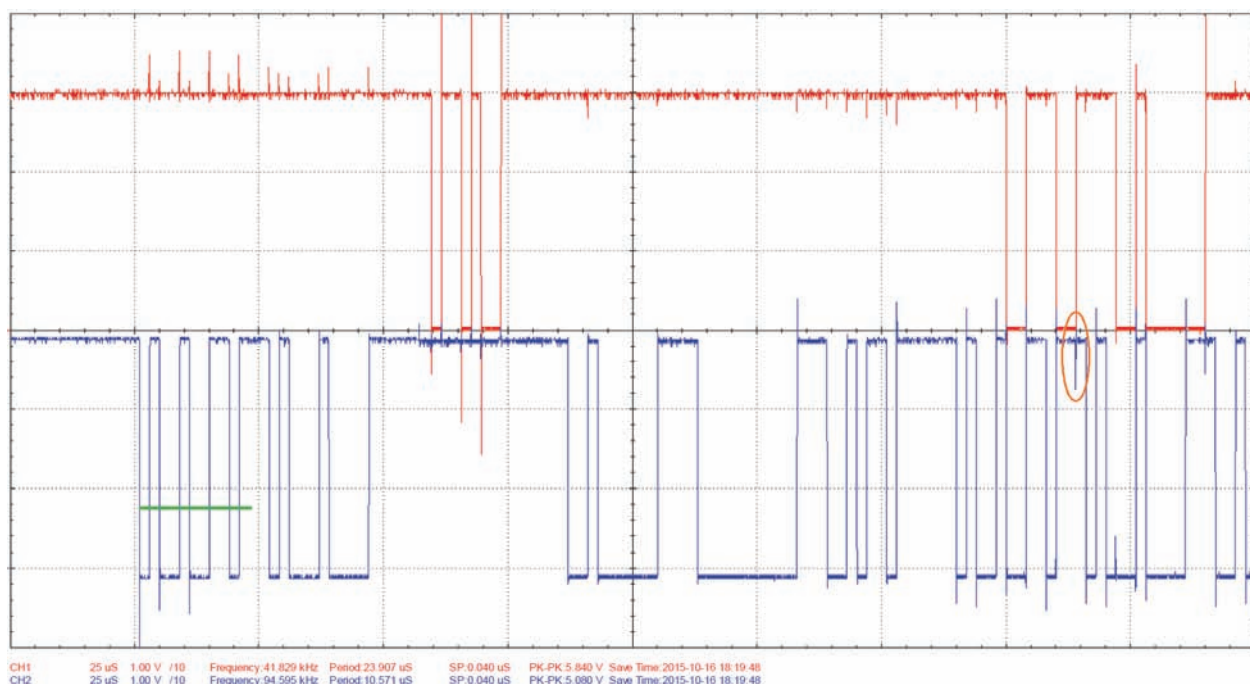
4. ZAKLJUČAK

Napravljena je biblioteka koji zadovoljava postavljene kriterijume za komunikaciju. Testiran je i pokazao se da je moguće koristiti. Ako se eliminiše deo za brojanje i praćenje tipova grešaka, što je bio korišćen za testiranje, brže će kôd izvršavati.

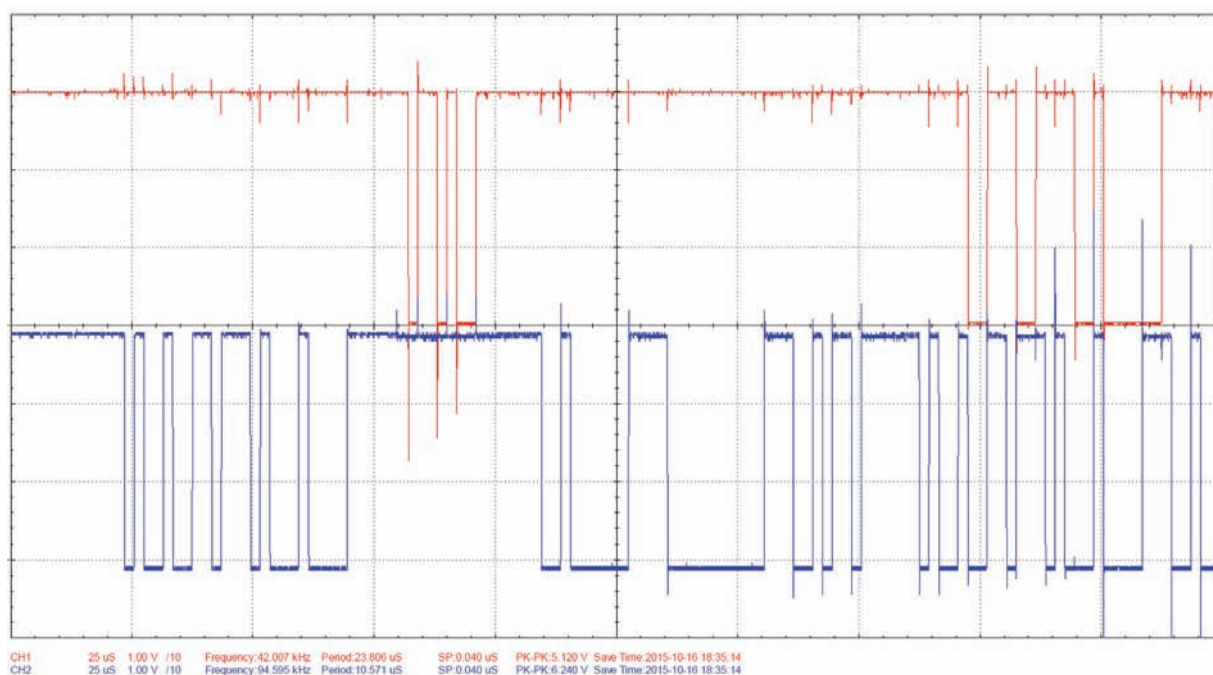
Ako se uvode još neke promene, moguće je dodatno povećavanje brzine komunikacije. Potrebno je promeniti redosled izvršenje prekida sa UARTa, posmatrane ARMovce imaju jedan prekid samo za serijsku komunikaciju.

Potrebno je promeniti, da prvo izvršava provera da li se prijem desio, obrada toga i tek onda provera da li je slanje završen i obrada toga.

Na slici 3 je prikazan isti prenos pored elektromotora, najveća vrednost šuma iznosi 0,6V.



Slika 2. Prenos bez šuma



Slika 3. Prenos pored elektromagnetnog šuma

Trenutno je suprotno. Povećanje iskorišćenosti je moguće sa smanjenjem broja vrednosti za proveru podataka. Testiranje je pokazao da se retko dešava greška u prenosu. Za slane podatke nije potrebno slati proveru posle svakih 8 bajt podataka, već moglo bi za proveru slati na kraju zbira svih poslatih podataka, ograničen do 8 podataka na 1 bajt, dok za više, ograničen na 2 bajta. Dobro je i STM32F051 ali ne treba koristiti pri direktnoj komunikaciji sa STM32F407 mikrokontrolerom ako je cilj brzina.

5. LITERATURA

- [1] Robot MARKO, detaljan opis na sajtu, <http://marko.robot.ftn.uns.ac.rs/>, posmatrano 2015 oktobra
- [2] Electronic Industries Association (1983). Electrical Characteristics of Generators and Receivers for Use in Balanced Multipoint Systems. EIA Standard RS-485

Kratka biografija:



Tobias Kovačič rođen je u Novom Sadu 1991 godine. Bachelor rad na Fakultetu tehničkih nauka iz oblasti Mehatronika – Mehatronika, robotika i automatizacija odbranio je 2014 godine.

RAZVOJ UDALJENE LABORATORIJE ZA UPRAVLJANJE PNEUMATSKIH CILINDARA**DEVELOPMENT OF REMOTE LABORATORIES FOR CONTROL OF PNEUMATIC CYLINDERS**

Danijela Birmančević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – Cilj ovog rada je bio razvoj sistema koji omogućava učenje pneumatskog upravljanja na daljinu. Naročiti naglasak je stavljen na ispitivanje energetske efikasnosti upravljačkih sistema pri primeni različitih načina upravljanja brzinom izvlačenja klipnjače pneumatskog cilindra.

Abstract – The aim of this study was to develop a system that enables learning of pneumatic control remotely. Particular emphasis is placed on testing the energy efficiency of the control systems in the application of different ways of velocity control in driving of the pneumatic cylinder piston rod.

Ključne reči: pneumatski sistem, pneumatsko upravljanje

1. UVOD

Vazduh pod pritiskom je jedan od najstarijih izvora energije koje čovek poznaje i koristi u cilju zamene fizičkog rada u proizvodnji i drugim oblastima savremenog društva.

Vazduh pod pritiskom se primenjuje za opšte namene u fabrikama, radionicama i skladištima, za sisteme pneumatskog transporta, pogon pneumatskih alata, obradu hrane, hemikalija i farmaceutskih proizvoda, bojenje sprejom, oblaganja površina praškovima, za pogon i upravljanje mašina za pakovanje itd.

Pneumatski uređaji se odlikuju jednostavnom konstrukcijom, što omogućava da se pneumatski uređaji koriste gde su uslovi rada teški i gde se od konstrukcije uređaja traži da bude siguran u radu.

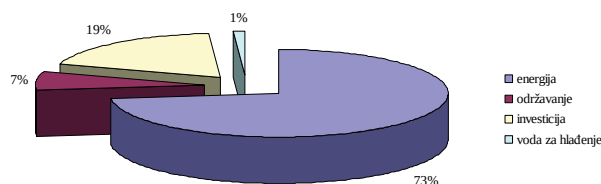
2. ENERGETSKA EFIKASNOST PNEUMATSKOG SISTEMA

Vazduh pod pritiskom je, i pored svih navedenih prednosti, skup energetske resurs za obavljanje rada u industrijskim postrojenjima.

NAPOMENA:

Ovaj rad je proistekao iz master rada, čiji je mentor dr Dragan Šešlija.

Prosečna struktura troškova proizvodnje vazduha pod pritiskom u toku životnog ciklusa prikazana je na slici 1. Uočava se da najveći deo troškova proizvodnje vazduha pod pritiskom otpada na troškove električne energije, investicije i održavanje.



Slika 1. Dijagram isticanja vazduha

Ako se ima u vidu da se u značajnom broju proizvodnih sistema efikasno iskoristi samo jedan deo ukupne proizvodnje vazduha pod pritiskom, onda se jasno nameće potreba za optimizacijom ovih sistema i identifikovanjem i razvojem mera za povećanje njihove energetske efikasnosti.

3. UPRAVLJANJE BRZINOM PNEUMATSKOG CILINDRA

U velikom broju sistema automatizacije postoji potreba regulacije brzine, pa i u pneumatskim upravljačkim sistemima. U njima se upravljanje brzinom izvršnih organa može vršiti na dva načina: smanjenjem brzine i povećanjem brzine.

Mogućnosti za povećanje brzine, nakon definisanih i realizovanih odnosa prečnika cilindra i opterećenja koje savladava, se uglavnom svode na primenu brzoispušnog ventila. Sa druge strane, mnogo su veće mogućnosti smanjivanja brzine.

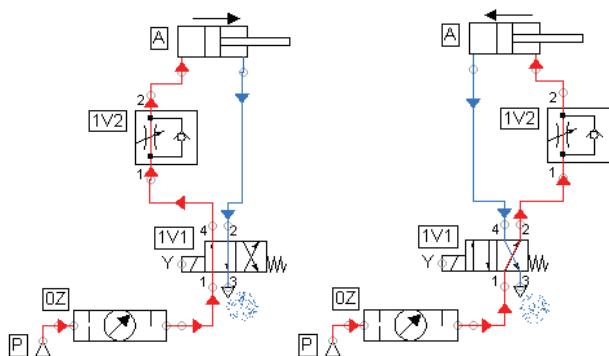
U ovom radu, kao primer, korišćeni su prigušno nepovratni ventili.

U zavisnosti od mesta postavljanja prigušno nepovratnih ventila dolazi do uštede ili potrošnje vazduha pod pritiskom.

Prigušnica se može postaviti na dva načina, i, na taj način, postoje dve metode:

- Meter-IN
- Meter-OUT

za upravljanje brzinom klipnjače cilindra.



Slika 2. Meter-IN i Meter-OUT metode upravljanja brzinom pneumatskog cilindra

Meter-IN metoda se koristi najčešće za upravljanje brzinom kretanja klipnjače cilindra jednosmernog dejstva. Upravljanje se postiže kada se prigušnica promenljivog protoka postavi na dovodni priključak cilindra. Smanjenjem protoka vazduha preko prigušnice se dovodi manji protok u cilindar, a samim tim cilindar ima manju brzinu.

Meter-OUT metoda se često koristi za upravljanje brzinom izvršnih elemenata. Prigušnica se postavlja na izlaznu stranu pneumatskog cilindra.

Ograničenje protoka vazduha pod pritiskom kroz prigušnicu vodi ka većem povratnom pritisku u cilindru, a samim tim i manjoj brzini kretanja cilindra.

4. UPRAVLJANJE SISTEMIMA SA UDALJENIH LOKACIJA

Automatizacija rada koja omogućava upravljanje sa udaljenog mesta ima čitav niz prednosti kao što je, na primer, otklanjanje potrebe za ručnim podešavanjem rada uređaja.

Pri tome, takav sistem mora biti jednostavan, ekonomičan, pouzdan i lak za upravljanje. U slučaju potrebe mora se omogućiti brz i lak prelazak na ručno upravljanje radi održavanja, popravki, nadzora i pregleda.

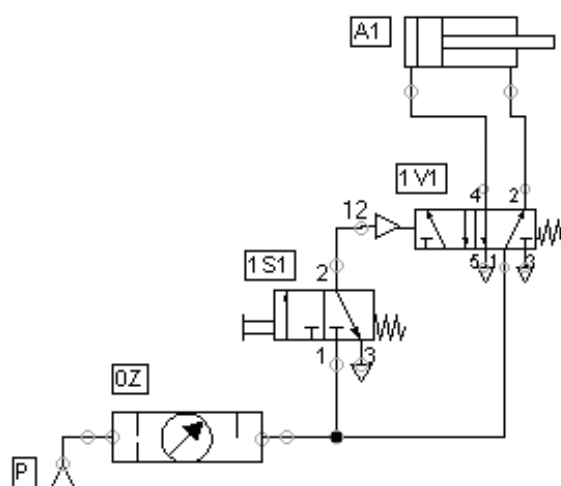
U realnim proizvodnim sistemima ovakva realizacija udaljenog upravljanja može pružiti sledeće prednosti:

- smanjenje troškova zaposlenih,
- smanjenje broja potrebnih radnika i naročito kvalifikovanog osoblja,
- olakšano preventivno održavanje i
- online sprovođenje mera sa udaljene lokacije.

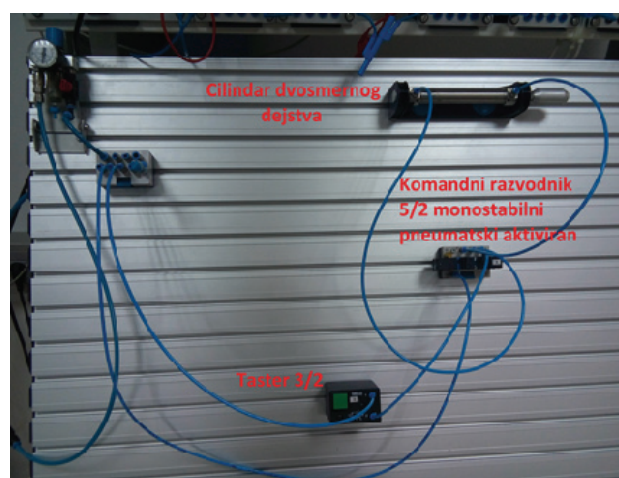
5. PRAKTIČNI DEO

Realizovano je laboratorijsko rešenje uređaja koji omogućava učenje pneumatskog upravljanja brzinom na daljinu.

U tom cilju su na probnom stolu postavljeni jedan cilindar dvosmernog dejstva i konvencionalno postavljeni pripadajući upravljački elementi kako je prikazano na pneumatskoj upravljačkoj šemi (slika 3) i na fotografiji 4.



Slika 3. Šematski prikaz upravljanja cilindrom dvosmernog dejstva



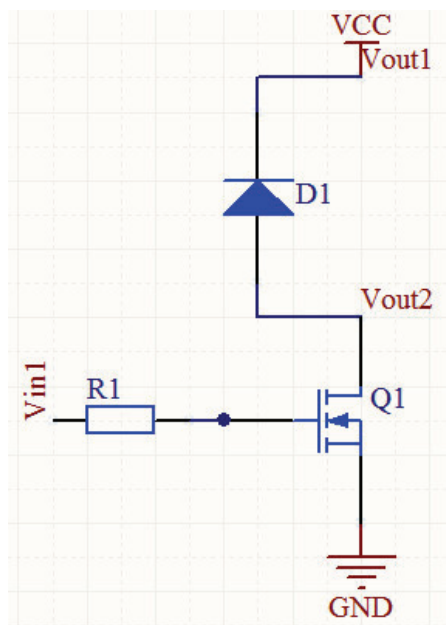
Slika 4. Povezivanje elemenata u laboratoriji na konvencionalan način

Pritiskom na taster 1S1 aktivira se 5/2 komandni razvodnik i na taj način se izvlači klipnjača cilindra dvosmernog dejstva. Kada klipnjača dostigne svoj krajnji izvučeni položaj, vraća se u krajnji uvučeni položaj, pod uslovom da je 3/2 taster otpušten. Ako se iz nekog razloga zadrži 3/2 taster aktivnim, klipnjača cilindra će ostati u izvučenom položaju.

Mikrokontroler pomoću razvodnih ventila vrši upravljanje izvršnim organima uređaja. Informaciju o trenutnim položajima kontroler dobija od senzora, u ovom slučaju to su granični prekidači.

Ukoliko se želi da korisnik pristupa uređaju putem interneta potrebno je da na neki način prati aktivnosti u laboratoriji. U tu svrhu iskorišćena je kamera koja omogućava praćenje eksperimenta uživo.

Radi smanjenja cene koštanja uređaja kao i realizacije upravljanja uređajem sa udaljene lokacije, odabran je jedan relativno jeftin kontroler proizvođača Arduino, model Yun [5]. Pomenuti kontroler na izlaznim nožicama daje napon u vrednosti od 5 VDC, dok je za rad sa razvodnim ventilima korišćenim u ovom radu potreban napon od 24 VDC. U tu svrhu izrađena je štampana elektronska ploča kola prikazana na slici 5.

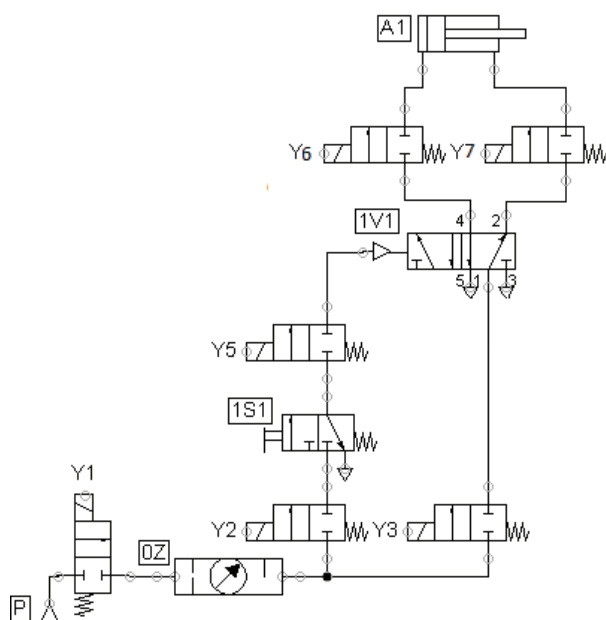


Slika 5. Kolo izlaznog stepena

Vrednosti i oznake komponenti za izradu pomenutog kola su sledeće: $R_1=220\ \Omega$, dioda D_1 ima oznaku 1N4004, MOSFET Q_1 ima oznaku IRFZ44N.

U praktičnom delu predstavljen je najjednostavniji primer indirektnog upravljanja pneumatskim sistemom realizovan sa mogućnošću daljinskog upravljanja kako je prikazano na slici 6 i fotografiji 7

U daljem izlaganju je predstavljena razvijena šema pneumatskog sistema korišćenjem dodatnih 2/2 monostabilnih električno aktiviranih ventila Y1, Y2, Y3, Y5, Y6 i Y7.

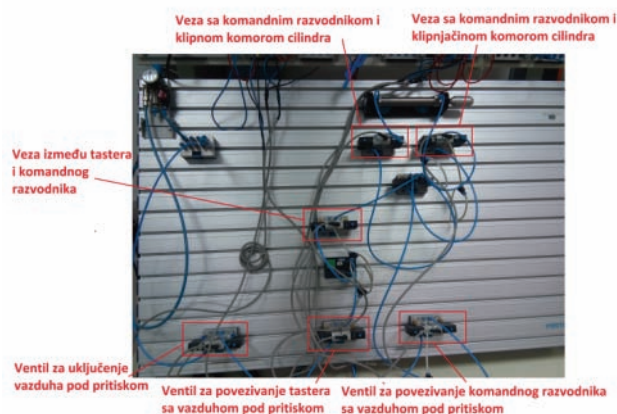


Slika 6. Razvijena šema

Da bi vazduh pod pritiskom dospeo u klipnu komoru cilindra, potrebno je da svi uslovi budu zadovoljeni. Aktiviranjem ventila Y1 vazduh pod pritiskom ulazi u

sistem i na taj način je spreman da opslužuje radne elemente. Aktiviranjem ventila Y2 i Y5 stvara se uslov za premeštanje 5/2 komandnog razvodnika u radni položaj. Aktiviranjem ventila Y3 vazduh pod pritiskom prolazi kroz 5/2 komandni razvodnik, koji je prethodno postavljen u radni položaj. Aktiviranjem ventila Y6 vazduh pod pritiskom ulazi u klipnu komoru, i klipnjača se izvlači. Otpuštanjem 3/2 tastera, klipnjača cilindra se vraća nazad.

Označeni elementi su 5/2 bistabilni električno upravljani ventili, povezani tako da imaju funkciju 2/2 monostabilnih električno aktiviranih ventila.



Slika 7. Povezivanje elemenata u laboratoriji sa omogućavanjem daljinskog upravljanja

6. ZAKLJUČAK

Prilikom realizacije ovog rada, cilj je bio razvoj sistema koji omogućava učenje pneumatskog upravljanja na daljinu.

Naročiti naglasak je stavljen na ispitivanje energetske efikasnost upravljačkih sistema pri primeni različitih načina upravljanja brzinom izvlačenja klipnjače pneumatskog cilindra i prikazana su dva najčešća načina regulacije brzine.

Da bi se omogućilo učenje pneumatskog upravljanja na daljinu realizovan je sistem koji omogućava udaljeni pristup probnom stolu preko interneta a dešavanja na probnom stolu se mogu posmatrati pomoću veb kamere.

Daljinskim upravljanjem se omogućava izbor različitog načina povezivanja komponenti i time različitog upravljanja.

Dalji rad u ovoj oblasti treba usmeriti na omogućavanje upravljanja sa većim brojem komponenti kako bi se udaljenim korisnicima mogli postavljati i složeniji zadaci od primera koji je ovde prikazan.

7. LITERATURA

- [1] Radoslav Korbar, Pneumatika i hidraulika - (http://www.vuka.hr/fileadmin/temp/PiH_skripta.pdf); pristupljeno: 07.10.2012.
- [2] Neven Maleš, Pneumatika i hidraulika, (http://vtsbj.hr/images/uploads/PNEUMATIKA_I_HI_DRAULIKA_3_4_PREDAVAJE.pdf); pristupljeno: 18.09.2012

- [3] Dragan Šešlija, Proizvodnja, priprema i distribucija vazduha pod pritiskom, IKOS, Novi Sad, 2002.
- [4] Meixner, H., Kobler, R., Uvod u pneumatiku, FESTO Didaktik, Fakultet tehničkih nauka, Novi Sad, 1985.
- [5] Bajči, B., Šulc, J., Šešlija, D., Modeliranje PWM upravljanja dvoosnim pneumatskim manipulatorom, Zbornik radova FTN, Septembar 2015.

Kratka biografija:

Danijela Birmančević rođena je u Šapcu 1987. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mehatronike odbranila je 2015 god.

ТЕЛЕМЕТРИЈСКИ СИСТЕМ ЗА ОБРАДУ И АКВИЗИЦИЈУ ПОДАТАКА О МЕРЕЊУ ПОТРОШЊЕ ПРИРОДНОГ ГАСА ПРИМЕНОМ „M-BUS“ ПРОТОКОЛА**TELEMETRY SYSTEM FOR PROCESSING AND ACQUISITION OF NATURAL GAS METERING DATA USING “M-BUS” PROTOCOL**

Лазар Живковић, Мирко Раковић, *Факултет техничких наука, Нови Сад*

Област – МЕХАТРОНИКА

Кратак садржај – У оквиру рада представљен је телеметријски систем за обраду и аквизицију података о мерењу потрошње природног гаса. Мерења се помоћу концентратора прикупљају у редовним временским интервалима са мерила „Landis&Gyr G350“ применом „WM-Bus“ протокола. Након тога се путем „GPRS“-а шаљу на сервер за аквизицију података, где се распакују и уписују у базу података. Корисник може да приступи тим подацима кроз интернет апликацију, у којој се врши провера приступа, слање захтева за подацима серверу за обраду података и приказ добијених резултата. У оквиру рада су приказани елементи реализованог система, као и испитивање рада система.

Abstract – The paper presents a telemetry system for processing and acquisition of natural gas metering data. The measurements are collected on a regular basis by a concentrator from a “Landis&Gyr G350” meter using the “WM-Bus” protocol. After that, they are sent to the data acquisition server via “GPRS”, where they get unpacked and saved in a database. The user can access that data through a web application, which handles user authentication, sending of data requests to the data processing server and the visualisation of the received results. The paper presents elements of the implemented system, as well as system testing.

Кључне речи: телеметрија, природни гас, аквизиција и обрада података, мерило, концентратор, сервер, база података, кориснички интерфејс

1. УВОД

Телеметрија је по дефиницији [1] Међународне електротехничке комисије (енг. IEC – International Electrotechnical Commission): „Процес у коме се мерења врше на удаљеној локацији, а резултати тих мерења се преносе путем телекомуникације.“

Са развојем интернета је дошло до револуције у начину прикупљања података о мерењу потрошње енергената. Створена је могућност честог прикупљања података на удаљеној локацији и тренутног приказивања тих података путем корисничког интерфејса, чиме је знатно унапређено праћење потрошње енергената.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Мирко Раковић.

Тиме је омогућено боље управљање системом за снабдевање и прецизнија наплата потрошених ресурса. Поред тога, једна од главних предности коришћења ове технологије за дистрибутере енергента је могућност даљинског управљања и велика финансијска уштеда, с обзиром да нема потребе за периодичним изласцима на терен ради прикупљања стања мерила.

Проблем који је настао са повећањем броја корисника је истовремено повећање броја мерила потребних за управљање системом и наплату потрошње. С обзиром на често велику међусобну удаљеност мерила, велика количина ресурса је потрошена на редовно прикупљање стања потрошње. Због тога, често су коришћена група мерила, што је довело до немогућности утврђивања појединачне потрошње корисника.

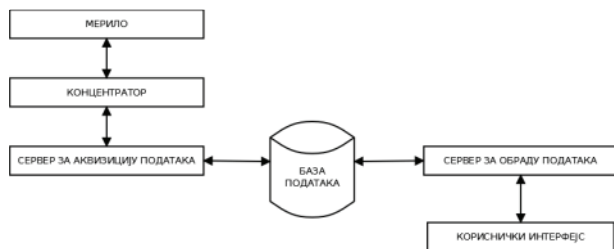
Додатни проблем је била и наплата прекорачења одобрене количине потрошње. С обзиром на то да се нису прикупљале дневне вредности потрошње, није се могло утврдити да ли је, у којој мери и колико пута корисник прекорачивао ту границу. Поред тога, уколико би дошло до квара на неком делу система, потребна је била приметна количина времена до пријаве квара, с обзиром да би се пријава кварова вршила позивом корисничког центра. Свако искључење или поновно прикључење корисника на дистрибутивну мрежу би подразумевало одлазак сервисера на лице места, што би додатно компликовало и продужавало процес.

Због претходно поменутих чињеница, дошло је до развоја концепта напредне мерне инфраструктуре (енг. AMI – Advanced Metering Infrastructure) која представља скуп технологија са циљем да се аутоматизује процес читавања потрошње енергената са мерила, преношења тих информација до централног система где се оне складиште за анализу и утврђивања наплате [2].

На основу комерцијално доступних решења [3] [4] [5] усвојена је општа структура савремених телеметријских система као на слици 1.

Већа учестаност прикупљања података о мерењу потрошње пружа могућност за стварање профила потрошње, наплату у зависности од времена коришћења, прогнозу потребе енергента, праћење протока и квалитета енергента. Поред тога, могућност даљинског управљања дистрибутивним системом омогућава управљање потрошњом у критичним временским периодима, ограничавање укупне потрошње енергије

од стране корисника, као и даљинско искључивање из дистрибуционе мреже. Овакви системи пружају и могућност праћења аларма, као што су: покушаји крађе, цурење, квар или нерегуларност у протоку енергента.



Слика 1. Структура савремених телеметријских система.

С друге стране, проблеми који се јављају коришћењем оваквих система су могућност ограничавања потрошње корисника од стране дистрибутера, могућност даљинског искључивања корисника из дистрибуционе мреже, угрожавање приватности корисника кроз увид у профил потрошње и злоупотреба те информације, нарушавање сигурности система. Поред тога, корисници имају страх од повећања рачуна као последице тачнијег мерења потрошње, нарочито имајући у виду зависност цене енергента од доба дана.

2. ОПИС РЕАЛИЗОВАНОГ РЕШЕЊА

У циљу реализације телеметријског система за обраду и аквизицију података о мерењу потрошње природног гаса, приступило се развоју елемената система у складу са општом структуром савремених телеметријских система (слика 1). Било је потребно развити концентратор који би у редовним временским интервалима од по сат времена прихватао и складиштио "WM-Bus" телеграме са измереним вредностима потрошње природног гаса послате од стране мерила „G350“ британског произвођача „Landis & Gyr“, формиране у складу са стандардом EN 13757 [6]. Комуникација између њих би се одвијала по „DSMR“ (енг. *Dutch Smart Meter Requirements*) протоколу [7] верзије 4. Те податке би даље требало спровести до „gateway“ уређаја унутар система, чије би задужење било да једном у току дана све прикупљене податке пошаље на сервер за аквизицију података. На серверу би требало распаковати те упаковане податке и добијена мерења уписати у базу података. Након тога би та мерења била доступна серверу за обраду података, који би на кориснички захтев, задат кроз кориснички интерфејс, правило параметризване упите у бази података и прослеђивао тражене податке корисничком интерфејсу, где би се ти подаци приказивали кориснику.

2.1 „WM-Bus“ концентратор

Као основа за концентратор узет је уређај „INIT-50“ фирме TO-HET. Овај уређај поседује прикладан хардвер који га чини погодним за решавање оваквог типа задатка, због поседовања два „mikroBUS™“ порта, који ће се користити за прикључење комуникационих модула. Један комуникациони модул је задужен за „WM-Bus“ комуникацију, а други за слање

података на сервер за аквизицију података путем „GPRS“-а. На слици 2 се може видети коришћени уређај са прикљученим комуникационим модулима. Уређај омогућава редовно прихватање „WM-Bus“ телеграма послатих од стране мерила, складиштење тих података у трајној меморији и касније прослеђивање серверу за даљу обраду.

Посебна погодност овог решења састоји се у томе што би било могуће да концентратор уједно има и улогу „gateway“ уређаја, што би смањило потребу за додатним хардвером, као и трошкове имплементације и одржавања. Концентратори који не би имали могућност слања података путем „GPRS“-а би имали улогу рутера, тј. примљене податке би у редовним временским интервалима прослеђивали коришћењем „WM-Bus“ протокола.

Фирмвер концентратора је писан у програмском језику „C“ коришћењем „ATMEL Studio 6“ графичког софтверског развојног окружења.



Слика 2. Концентратор са прикљученим комуникационим модулима.

2.2 Сервер за аквизицију података

Улога сервера за аквизицију података је да омогући концентраторима повезивање на одређени мрежни сокет преко ког ће они слати прикупљена мерења. Након тога би требало да се на серверу распакују примљени пакети са подацима по стандарду [6] и да се упишу у базу података. Поред тога, требало је омогућити примање више пакета у току једне везе са концентратором. За извршавање серверске апликације коришћен је серверски рачунар унутар фирме TO-HET, који ради под „Microsoft Windows Server 2008“ оперативним системом.

У циљу веће компатибилности коришћених технологија, изабрана је комбинација „Java“ програмског језика за имплементацију серверске апликације и „MySQL“ система за управљање базом података. За рад са базом података коришћено је „JDBC“ окружење.

Сервер за аквизицију података прихвата захтеве за повезивањем и проверава да ли су примљени подаци. Уколико јесу, проверава да ли је примљено исправно заглавље. Уколико јесте, чита се дужина послатих корисничких података. Уколико је прочитана исправна дужина, подаци се смештају у бафер одакле се врши разврставање по логовима. Након тога се формирани логови распакују по стандарду [6] и након тога уписују у базу података. Сервер шаље потврде клијенту након примања података и након успешног уписа података у базу.

2.3 База података

Реализована је база података у коју се уписују прикупљена мерења. Поред тога, база података води рачуна о структури система, тј. о мрежи мерила, рутера и „gateway“ уређаја. Подаци унутар базе би требало да буду доступни серверу за обраду података, како би се приказивали на корисничком интерфејсу. Такође, омогућена је контрола приступа корисничком интерфејсу, тј. чување листе корисника са одговарајућим лозинкама и личним подацима. Поред тога, имплементирани су степени права корисника, како би само одређена група корисника имала права да мења, брише и додаје податке кроз кориснички интерфејс.

За реализацију базе података коришћен је серверски рачунар у оквиру фирме TO-NET са инсталираним оперативним системом „Microsoft Windows Server 2008“. Одабран је релациони тип базе података, и систем за управљање релационим базама података „MySQL“.

2.4 Сервер за обраду података

Улога сервера за обраду података је остваривање комуникације између корисничког интерфејса и базе података. Сервер коришћењем „HTTP“ (енг. *Hypertext Transfer Protocol*) прихвата захтеве генерисане кроз кориснички интерфејс. Након тога извршава упите у бази података на основу параметара из захтева. На крају прослеђује тражене податке корисничком интерфејсу у виду „JSON“ (енг. *JavaScript Object Notation*) формата. У оквиру овог рада је реализована претрага и прикупљање података из базе података, без могућности измене, додавања и брисања података.

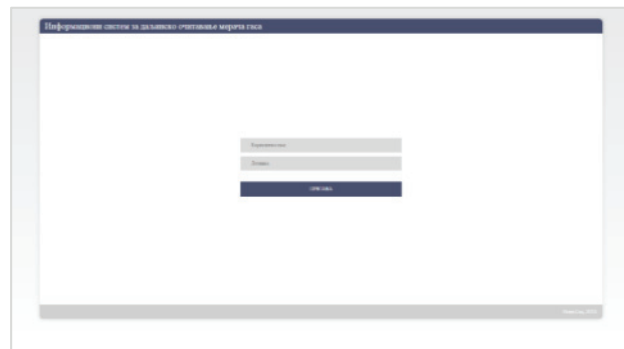
Сервер за обраду података је имплементиран у PHP-у (енг. *PHP- Hypertext Preprocessor*). За реализацију сервера коришћен је серверски рачунар у оквиру фирме TO-NET са инсталираним оперативним системом „Microsoft Windows Server 2008“ и серверском апликацијом „Microsoft Internet Information Services 7“.

2.5 Кориснички интерфејс

Кориснички интерфејс је реализован у виду интернет апликације која омогућава табеларан приказ прикупљених мерења за одређено мерило и временски интервал. Поред тога, апликација има функционалност провере приступа (логовања), због осетљивости података. Апликација остварује комуникацију са сервером за обраду података, којем шаље захтеве коришћењем „HTTP“ и од којег у одговору добија тражене податке у „JSON“ формату.

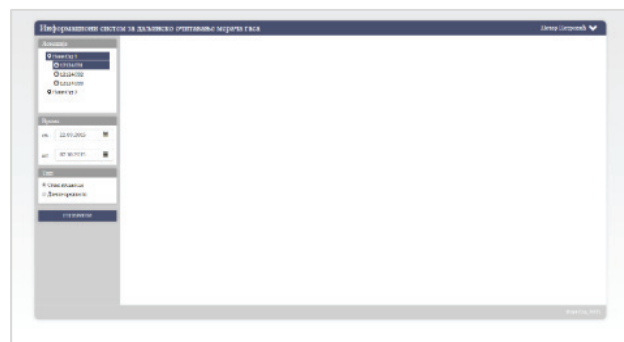
За развој графичког интерфејса коришћене су првенствено технологије „HTML“ (енг. *HyperText Markup Language*) и „CSS“ (енг. *Cascading Style Sheets*) као основни алат за развој интернет страница. Специфичност овог решења је коришћење „Angular JS“ „Javascript“ окружења. За формирање табеле коришћен је додатак „DataTables“ у оквиру „jQuery“ „Javascript“ библиотеке. Приступање интернет апликацији омогућава серверски рачунар у оквиру фирме TO-NET са инсталираним серверском апликацијом „Microsoft Internet Information Services 7“.

На слици 3 се може видети изглед прозора за проверу приступа.



Слика 3. Изглед прозора за проверу приступа.

Након успешне аутентификације, корисник се прослеђује на главни прозор апликације, који се може видети на слици 4. У главном прозору врши се одабир параметара претраге у левом делу екрана и табеларан приказ мерења у десном делу екрана. Захтев се генерише на основу одабране локације, мерила, временског интервала и типа вредности.



Слика 4. Изглед главног прозора апликације.

3. ИСПИТИВАЊЕ РАДА СИСТЕМА

За потребе испитивања рада система коришћено је развојно мерило „Landis&Gyg G350“, које се може видети на слици 5. За прикупљање података је коришћен концентратор са слике 2. На серверским рачунарима су покренуте серверске апликације.

Испитан је рад реализованог система и утврђено је:

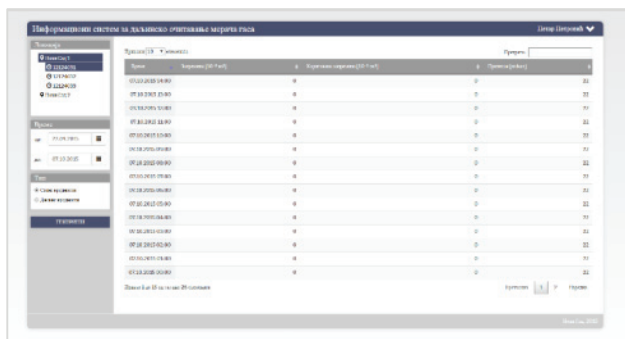
1. да се мерења редовно прикупљају са мерила потрошње природног гаса од стране концентратора, складиште у трајној меморији и након тога прослеђују серверу за аквизицију података путем „GPRS“-а,



Слика 5. Мерило „Landis & Gyr G350“ коришћено за испитивање рада система.

- да се та мерења распакују на серверу за аквизицију података и након тога уписују у базу података,
- да корисник може да приступи интернет апликацији, одабере параметре претраге и генерише захтев који се прослеђује прослеђује серверу за обраду података,
- да сервер за обраду података врши упит у бази података на основу примљених захтева и након тога прослеђује податке који одговарају параметрима претраге,
- да се ти подаци приказују на корисничком интерфејсу у виду табеле.

На слици 6 се може видети изглед главног прозора интернет апликације након успешног прикупљања података из базе. Омогућена је претрага података, приказивање по страницама и сортирање.



Слика 6. Приказ података на интернет апликацији.

4. ЗАКЉУЧАК

У оквиру овог рада је развијено:

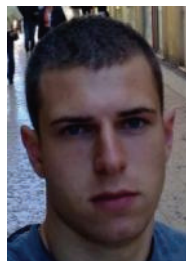
- фирмвер за „WM-Bus“ концентратор, при чему је омогућено: инсталација мерила коришћењем „WM-Bus“ протокола, прикупљање сатних вредности са мерила потрошње природног гаса, складиштење мерења у трајној меморији и слање мерења на сервер за аквизицију података путем „GPRS“-а;

- сервер за аквизицију података, при чему је омогућено: примање података са концентратора у више пакета, распакивање мерења са мерила „Landis&Gyr G350“ и уписивање мерења у базу података;
- база података, при чему је омогућено вођење рачуна о: мерењима прикупљених са мерила потрошње природног гаса, мрежи мерила, рутера и „gateway“ уређаја, корисницима интернет апликације и њиховим правима;
- сервер за обраду података, при чему је омогућено: примање захтева од стране корисничког интерфејса, извршавање упита у бази података на основу параметара из захтева и прослеђивање одговарајућих података у виду „JSON“ формата;
- кориснички интерфејс у виду интернет апликације, при чему је омогућено: провера приступа апликацији, одабир параметара претраге (локација, време и тип мерења) и табеларни приказ мерења која одговарају претрази.

5. ЛИТЕРАТУРА

- [1] International Electrotechnical Commission, IEC ref 701-01-25
- [2] Jennifer Hiscock, Dong-Joo Kang : "AMI Case Book Version 2.0, Spotlight on Advanced Metering Infrastructure", 2014 CEM in Korea
- [3] <http://www.techem.rs/>, датум приступа 20.10.2015.
- [4] <https://www.kamstrup.com/>, датум приступа 20.10.2015.
- [5] <http://www.insa.rs/>, датум приступа 20.10.2015.
- [6] EN 13757
- [7] Dutch Smart Meter Requirements, P2 Companion standard, version 4.0

Кратка биографија:



Лазар Живковић рођен је 28.9.1991. у Лесковцу. Диплому инжењера мехатронике је стекао на Факултету техничких наука на новосадском универзитету 2014. године, одбраном рада са темом „Моделовање и симулација погона хибридних електричних возила“. Рад је урађен под менторством доц. др Мирка Раковића.

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2014. godine učestvovali su sledeći recenzenti:

Aco Antić	Đorđe Ćosić	Milan Rapajić	Slavica Mitrović
Aleksandar Erdeljan	Đorđe Lađinović	Milan Simeunović	Slavko Đurić
Aleksandar Ristić	Đorđe Obradović	Milan Trifković	Slobodan Dudić
Bato Kamberović	Đorđe Vukelić	Milan Trivunić	Slobodan Krnjetin
Biljana Njegovan	Đura Oros	Milan Vidaković	Slobodan Morača
Bogdan Kuzmanović	Đurđica Stojanović	Milena Krklješ	Sonja Ristić
Bojan Batinić	Emil Šećerov	Milica Kostreš	Srđan Kolaković
Bojan Lalić	Filip Kulić	Milica Miličić	Srđan Popov
Bojan Tepavčević	Goran Sladić	Milinko Vasić	Srđan Vukmirović
Bojana Beronja	Goran Švenda	Miloš Slankamenac	Staniša Dautović
Branislav Atlagić	Gordana	Miloš Živanov	Stevan Milisavljević
Branislav Nerandžić	Milosavljević	Milovan Lazarević	Stevan Stankovski
Branislav Veselinov	Gordana Ostojić	Miodrag Hadžistević	Strahil Gušavac
Branislava Kostić	Igor Budak	Miodrag Zuković	Svetlana Nikoličić
Branislava	Igor Dejanović	Mirjana Damjanović	Tanja Kočetov
Novaković	Igor Karlović	Mirjana Malešev	Tatjana Lončar
Branka Nakomčić	Ilija Kovačević	Mirjana Radeka	Turukalo
Branko Milosavljević	Ivan Beker	Mirjana Vojnović	Todor Bačkalić
Branko Škorić	Ivan Tričković	Miloradov	Toša Ninkov
Cvijan Krsmanović	Ivan Župunski	Mirko Borisov	Uroš Nedeljković
Damir Đaković	Ivana Katić	Miro Govedarica	Valentina Basarić
Danijela Lalić	Ivana Kovačić	Miroslav Hajduković	Velimir Čongradec
Darko Čapko	Jasmina Dražić	Miroslav Nimrihter	Velimir Todić
Darko Marčetić	Jelena Atanacković	Miroslav Plančak	Veljko Malbaša
Darko Reba	Jeličić	Miroslav Popović	Veran Vasić
Dejan Ubavin	Jelena Borocki	Mitar Jovanović	Veselin Avdalović
Dragan Ivanović	Jelena Kiurski	Mladen Kovačević	Veselin Perović
Dragan Ivetić	Jelena kovačević	Mladen Radišić	Vladan Radlovački
Dragan Jovanović	Jureša	Momčilo Kujačić	Vladimir Katić
Dragan Kukolj	Jelena Radonić	Nađa Kurtović	Vladimir Radenković
Dragan Mrkšić	Jovan Petrović	Nebojša Pjevalica	Vladimir Strezoski
Dragan Pejić	Jovan Tepić	Neda Pekarić Nađ	Vladimir Škiljajica
Dragan Šešlija	Jovan Vladić	Nemanja	Vlado Delić
Dragana Bajić	Jovanka Pantović	Stanisavljević	Vlastimir
Dragana	Karl Mičkei	Nenad Katić	Radonjanin
Konstantinović	Katarina Gerić	Nikola Brkljač	Vuk Bogdanović
Dragana Šarac	Ksenija Hiel	Nikola Đurić	Zdravko Tešić
Dragana Štrbac	Laslo Nađ	Nikola Jorgovanović	Zora Konjović
Dragi Radomirović	Leposava Grubić	Nikola Radaković	Zoran Anišić
Dragiša Vilotić	Nešić	Ninoslav Zuber	Zoran Brujic
Dragoljub Novaković	Livija Cvetičanin	Ognjen Lužanin	Zoran Jeličić
Dragoljub Šević	Ljiljana Vukajlov	Pavel Kovač	Zoran Mijatović
Dubravka Bojanić	Ljiljana Cvetković	Peđa Atanasković	Zoran Milojević
Dušan Dobromirov	Ljubica Duđak	Petar Malešev	Zoran Mitrović
Dušan Gvozdenac	Maja Turk Sekulić	Predrag Šiđanin	Zoran Papić
Dušan Kovačević	Maša Bukurov	Radivoje Rinulović	Željen Trpovski
Dušan Sakulski	Matija Stipić	Rado Maksimović	Željko Jakšić
Dušan Uzelac	Milan Kovačević	Radovan Štulić	
Duško Bekut	Milan Rackov	Rastislav Šostakov	