



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXXIV

Број: 6/2019

Нови Сад

Едиција: „Техничке науке – Зборници“

Година: XXXIV

Свеска: 6

Издавач: Факултет техничких наука Нови Сад

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

Уредништво:

Проф. др Раде Дорословачки

Проф. др Драгиша Вилотић

Проф. др Срђан Колаковић

Проф. др Владимир Катић

Проф. др Драган Шешилија

Проф. др Миодраг Хаџистевић

Проф. др Растислав Шостаков

Доц. др Мирослав Кљајић

Доц. др Бојан Лалић

Доц. др Дејан Убавин

Проф. др Никола Јорговановић

Доц. др Борис Думнић

Проф. др Дарко Реба

Проф. др Ђорђе Лађиновић

Проф. др Драган Јовановић

Проф. др Мила Стојаковић

Проф. др Драган Спасић

Проф. др Драгољуб Новаковић

Редакција:

Проф. др Владимир Катић, главни
уредник

Проф. др Жељен Трповски, технички
уредник

Проф. др Драган Шешилија

Проф. др Драгољуб Новаковић

Др Иван Пинђер

Бисерка Милетић

Језичка редакција:

Бисерка Милетић, лектор

Софија Рацков, коректор

Марина Катић, преводилац

Издавачки савет:

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

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

CIP-Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

378.9(497.113)(082)
62

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

Месечно

ISSN 0350-428X

COBISS.SR-ID 58627591

ПРЕДГОВОР

Поштовани читаоци,

Пред вама је шеста 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 мастер и 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 мастер радова студената ФТН-а у облику кратког рада у „Зборнику радова Факултета техничких наука“.

Поред студената 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 сваки месец, у оквиру промоције дипломираних мастера.

У овом броју штампани су радови студената мастер студија, сада већ мастера, који су радове бранили у периоду од 29.10.2018. до 27.12.2018. год., а који се промовишу 18.05.2019. год. То су оригинални прилози студената са главним резултатима њихових мастер радова.

Известан број кандидата објавили су радове на некој од домаћих научних конференција или у неком од часописа. Њихови радови нису штампани у Зборнику радова.

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 6., објављени су радови из области:

- машинства и
- електротехнике и рачунарства.

У свесци са редним бројем 7. објављени су радови из области:

- грађевинарства,
- саобраћаја и
- инжењерског менаџмента.

У свесци са редним бројем 8. објављени су радови из области:

- графичког инжењерства и дизајна,
- архитектуре,
- инжењерства заштите на раду и заштите животне средине,
- мехатронике,
- геодезије и геоматике,
- регионалне политике и развоја,
- управљања ризиком од катастрофалних догађаја и пожара,
- инжењерства информационих система и
- анимације у инжењерству.

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. Mirsad Tiganj, UPOREDNA TEHNOEKONOMSKA ANALIZA RAZLIČITIH PROCESA SEČENJA ČELIČNOG LIMA	985-988
2. Nemanja Bandić, Borislav Savković, PELETIRANJE DRVNE MASE I TIPOVI PROIZVODNJE	989-992
3. Uroš Rodić, Marin Gostimirović, REDIZAJN BRUSILICE ZA SPOLJAŠNJE KRUŽNO BRUŠENJE	993-996
4. Ivana Čulafić, UNAPREĐENJE PRIPREME PROIZVODNJE DELOVA ELEKTROMOTORA U KOMPANIJI ATB SEVER	997-1000
5. Branka Sremački, ISPITIVANJE ALTERNATIVNOG TURBINSKOG MERILA PROTOKA GORIVA PRI TOČENJU GORIVA NA BENZINSKIM PUMPAMA	1001-1004
6. Nemanja Dimitrijević, SISTEM ZA PRAĆENJE I PLANIRANJE ALATA U TEHNOLOŠKOM PROCESU MONTAŽE AUTOMOBILSKIH KOMPONENTI	1005-1008
7. Luka Vujičić, METODE OPTIMALNOG UPRAVLJANJA ZA OPTIMIZACIJU PERFORMANSI TRKAČKIH VOZILA	1009-1012
8. Marko Gomola, Dunja Sokolović, PRIMENA VLAKANA NERĐAJUĆEG ČELIKA ZA SEPARACIJU AEROSOLA NASTALOG U METALOPRERAĐIVAČKOJ INDUSTRIJI	1013-1016
9. Nemanja Klještan, NADZOR HABANJA ALATA PRI OBRADI MATERIJALA POVEĆANE TVRDOĆE	1017-1020
10. Josip Horvat, PROJEKAT REKONSTRUKCIJE KOTLARNICE I TEHNIČKO REŠENJE GREJANJA I KLIMATIZACIJE MANIPULATIVNE ZGRADE NA NAFTNOJ SABIRNOJ STANICI	1021-1024

Radovi iz oblasti: Elektrotehnika i računarstvo

STRANA

1. Dejan Radić, PODVODNE AKUSTIČNE SENZORSKE MREŽE	1025-1028
2. Nikola Diklić, Evgenije Adžić, Marko Vekić, REALIZACIJA VEKTORSKOG UPRAVLJANJA ASINHRONOG MOTORA, UPOTREBOM ALATA ZA AUTOMATSKO GENERISANJE KODA	1029-1031
3. Stefan Šoić, ANALIZA UPRAVLJANJA JEDNOSMERNIM MOTOROM BEZ ČETKICA	1032-1035
4. Milica Janković, EVALUACIJA PERFORMANSI KONSENZUS KLASTEROVANJA NAD HISTOPATOLOŠKIM SLIKAMA TUMORA DOJKE	1036-1039
5. Aleksandra Kantar, ANALIZA SEKVENCI MIKROBIOMA KROZ MOTHUR TOK OBRADA	1040-1043
6. Bojan Buzadžija, Zoltan Čorba, PRIKLJUČENJE BIOGASNOG POSTROJENJA NA DISTRIBUTIVNI SISTEM SREDNJEG NAPONSKOG NIVOA	1044-1047
7. Nemanja Mihajlov, PROVERA FUNKCIONALNOSTI RADA RELEJA U SREDNJENAPONSKIM NEBALANSIRANIM DISTRIBUTIVNIM MREŽAMA	1048-1051
8. Jelena Šešlija, PROJEKTOVANJE ZAŠTITE U NISKONAPONSKIM INSTALACIJAMA	1052-1055
9. Filip Frank, JEZIK ZA PODRŠKU EKSTRAKCIJI PODATAKA IZ NESTRUKTURIRANIH IZVORA SA VEBA	1056-1059
10. Jana Vojnović, RAZVOJNO OKRUŽENJE ZA GENERISANJE PROGRAMSKOG KODA SPRING API REST APLIKACIJA	1060-1063
11. Sergej Kešelj, INTERPRETER REST API SERVISA BAZIRAN NA OPENAPI SPECIFIKACIJI	1064-1067
12. Marko Paspalj, MODELOVANJE I IZRADA KONTROLERA ZA ŽALUZINE ZASNOVANOG NA ALGORITMU FUZZY KONTROLERA ZA OPTIMIZACIJU UGLA ZAKRIVLJENOSTI	1068-1071
13. Виолета Новаковић, СИСТЕМ ЗА ПЛАЋАЊЕ УТРОШЕНОГ ВРЕМЕНА	1072-1075
14. Goran Tanasić, Vladimir Rajs, INDUSTRIJSKI MODUL ZA MIKROKONTROLERE I SBC RAČUNARE	1076-1079
15. Žana Bilbija, IMPLEMENTACIJA SPOTIFY VEB KLIJENTA U REACT I DJANGO RADNOM OKRUŽENJU	1080-1083
16. Milica Kapetina, ANALIZA SOLID PRINCIPA U PROGRAMSKOM JEZIKU JAVASCRIPT	1084-1087
17. Filip Surla, UTICAJ SOLARNIH PANELA NA PODEŠENJE ZAŠTITE NA IZVODU	1088-1091
18. Miloš Gagović, UPOTREBA SAVREMENIH METODA MODERNIZACIJE POSTOJEĆIH ASP.NET APLIKACIJA SA CILJEM RADA NA CLOUD PLATFORMI	1092-1095
19. Branko Adžić, POTENCIJAL KORIŠTENJA GEOTERMALNIH PUMPA NA PODRUČJU GRADA BIJELEJINE	1096-1099
20. Bojan Lungulov, Vladimir Rajs, Vladimir Milosavljević, PROFESIONALNI DETEKTOR BEŽIČNIH SIGNALA (PSD-100) - PREVENCIJA VISOKOTEHNOLOŠKIH PREVARA	1100-1103
21. Bojan Kuzmić, Vladan Krsman, KORELACIJA FIZIČKOG I MATEMATIČKOG MODELA PRENOSNE MREŽE	1104-1107

	STRANA
22. Nikolina Modraković, ODREĐIVANJE POLOŽAJA PREKIDAČA ZA 24h NAKON REKONFIGURACIJE MREŽE	1108-1111
23. Igor Tot, INDUSTRIJSKA IMPLEMENTACIJA MODBUS PROTOKOLA	1112-1115
24. Tamara Letić, DETEKCIJA I ANALIZA DUPLICIRANOG JAVASCRIPT KODA UPOTREBOM JSINSPECT ALATA	1116-1119
25. Aleksandar Duduković, METODE OPTIMIZACIJE SERVISA ZA UPRAVLJANJE ALARMIMA U PROCESNOJ INDUSTRIJI	1120-1123
26. Jelena Savković, Vladimir Katić, MOGUĆNOSTI PRIMJENE MODELA PIKO HIDROELEKTRANE	1124-1127
27. Dražen Đanić, PROVERA IDENTITETA I AUTORIZACIJA KORISNIKA U MICROSOFT AZURE PLATFORMI	1128-1131
28. Дејан Марин, ПРОЈЕКТОВАЊЕ LED РАСВЈЕТЕ СА ПРИМЈЕРОМ ЈАВНЕ РАСВЈЕТЕ У КОЗАРСКОЈ ДУБИЦИ ..	1132-1135
29. Bogdan Stajić, Zoran Ivanović, STATIČKA POBUDA SINHRONOG GENERATORA	1136-1139
30. Sonja Živković, PRIMENA INTERNET STVARI U RAZLIČITIM ŽIVOTNIM OKRUŽENJIMA	1140-1143
31. Jelena Spasojević , AUTOMATSKA REINSTALACIJA DMS SCADA SISTEMA POMOĆU DSC-A	1144-1147
32. Vladimir Papuga, REALIZACIJA MAŠINE PRAVILA U RUKOVANJU ALARMIMA U NADZORNO- UPRAVLJAČKOM SISTEMU	1148-1151
33. Vanja Gojnić, Vladimir Katić, PRIMENA SOFTVERA INSEL ZA REŠENJA FN SISTEMA U SRBIJI	1152-1155

UPOREDNA TEHNOEKONOMSKA ANALIZA RAZLIČITIH PROCESA SEČENJA ČELIČNOG LIMA**COMPARATIVE TECHNOECONOMIC ANALYSIS OF DIFFERENT STEEL PLATE CUTTING PROCESSES**Mirsad Tigani, *Fakultet tehničkih nauka, Novi Sad***Oblast – MAŠINSTVO**

Kratak sadržaj – Osnovni cilj istraživanja orijentisan je na optimizaciju procesa sečenja čeličnog lima sa različitih tehnoloških aspekata. Kao ulazne promenljive definišane su vrsta procesa sečenja, debljina čeličnog lima i brzina sečenja, a kriterijumi tehnoekonomske analize su vreme obrade, dimenziona tačnost, geometrijska tačnost, stepen iskorišćenosti materijala, kvalitet obrađene površine i troškovi obrade, odnosno cena dobijanja gotovih delova sečenjem.

Ključne reči: Sečenje čeličnog lima, tehnoekonomska analiza, gasno sečenje, lasersko sečenje, plazma sečenje

Abstract – The aim of the research is oriented to the optimization of the process of steel plate cutting from different technological aspects. The input variables into the process are the type of a cutting process, the thickness of the steel plates and the cutting speed, whereas the criteria of technoeconomic analysis are processing time, dimensional accuracy, geometric accuracy, material utilization, surface quality and processing cost, that is, the cost obtained from the final part cutting.

Key words: steel plate processing, technoeconomic analysis, gas cutting, laser cutting, plasma cutting

1. UVOD

Uporedo sa povećanjem zahteva za obradu materijala u metaloprerađivačkoj industriji, razvijaju se nove metode i tehnologije koje su tačnije, produktivnije i ekonomičnije. Danas se za obradu pločastih materijala sečenjem pored konvencionalnih procesa obrade rezanjem sve više upotrebljavaju nekonvencionalni procesi obrade.

Povećanjem preciznosti i tačnosti navedenih obradnih sistema, smanjuje se potreba za višestrukim naknadnim obradama, uz istovremeno postizanje zadovoljavajuće proizvodnosti i ekonomičnosti.

Poznavanjem karakteristika pojedinog procesa obrade, odnosno njihovih mogućnosti i razlika lakše je doneti odluku koji proces u kom slučaju izabrati, odnosno koji je za koje uslove optimalan. Tema ovog rada je orijentisana na upoređivanje tehnoekonomskih karakteristika tri obradna sistema za sečenje materijala koja koriste tri različite tehnologije sečenja laserom, plazmom i gasom u posmatranom proizvodnom sistemu "Metalopromet d.o.o." iz Kule.

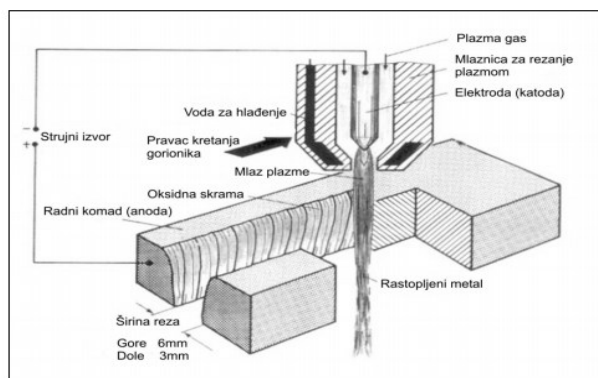
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dejan Lukić, vanredni profesor.

2. OSNOVE PROCESA SEČENJA MATERIJALA

Gasno rezanje/sečenje je postupak razdvajanja metala njegovim sagorevanjem u struji kiseonika, uz istovremeno oduvavanje produkata sagorevanja (troske). Da bi metal sagorevao treba ga zagrejati do temperature paljenja (početka sagorevanja). Gasno rezanje metala spada u termičke postupke, pri čemu se radna temperatura postiže sagorevanjem gorivog gasa (acetilen, propan/butan, zemni gas, ...). Pored toga, sagorevanjem gorivog gasa dobija se deo toplote kojom se radna temperatura održava, a značajan deo toplote dobija se sagorevanjem metala koji se reže. Pravilan proces rezanja se odvija konstantnom brzinom sa neprekidnim mlazom produkata sagorevanja (oksidi u obliku rastopljene troske). Pri tome, plamen koji okružuje mlaz kiseonika, ima zadatak da anulira efekat odvođenja toplote kroz metal van zone rezanja, a mlaz kiseonika mora da sagorevanjem metala obezbedi dovoljno toplote da sagorevanje (rezanje) ostane neprekinuto [1].

Plazma predstavlja četvrto agregatno stanje i suštinski je električno provodljiv, disociran i visokojonizovani gas. U zavarivanju i rezanju se koristi kao relativno tanak mlaz, energije velike gustine (preko 9000A/cm²). Pri sečenju plazmom visoka temperatura mlaza plazme topi i delimično sublimiše radni predmet, a rastop se oduvava iz reza kinetičkom energijom gasa koji izlazi iz mlaznice, slika 2 [1].

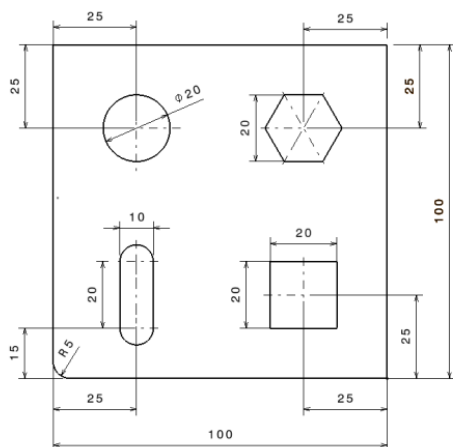


Slika 1. Šematski prikaz gorionika za plazma rezanje [1]

Rezanje plazmom je proces koji se koristi za rezanje čelika i ostalih metala koristeći plazma gorionik. Inertni gas se potiskuje velikom brzinom kroz mlaznicu, dok se u isto vreme uspostavlja električni luk kroz gas do radnog komada (obratka) pretvarajući jedan deo gasa u plazmu. Plazma je dovoljno topla da proreže materijal i da se otopljeni metal oduva sa reza.

Šira primena lasera u industriji počinje u ranim 70-im godinama kada su se pojavili CO₂ i neodijum itrijum aluminijum granat (Nd:YAG) laseri. CO₂ laseri se koriste za sečenje velikom brzinom sa srednjim kvalitetom reza, dok se Nd:YAG laseri koriste za postizanje vrhunskog kvaliteta. Nd:YAG su pogodniji za izradu otvora, uzanih proreza i preciznu obradu, ali njihova cena i troškovi održavanja su mnogo viši u odnosu na CO₂ lasere. Tačnost obrade laserom zavisi od intenziteta i stabilnosti energetskih parametara laserskog zraka i geometrijskih karakteristika optičkog sistema. Zbog konusnog oblika zraka i zbog velikog zagrevanja materijala tokom obrade dolazi do odstupanja dimenzija i oblika. Međutim i pored tih odstupanja, tačnost obrade laserom je vrlo visoka, naročito kod završne obrade $\pm(0,01-0,001)$ mm. Kako bi tačnost obrade bila što je moguće veća, bitno je ostvariti maksimalno fokusiranje laserkog zraka na površini koja se obrađuje. Takođe, tačnost se može povećati ugradnjom dodatnih uređaja, rotacijom laserskog zraka i dr.

Planom eksperimenta je predvideno da se iz čeličnih ploča različite debljine izvrši isecanje određenih delova raspoloživim procesima sečenja, različitim brzinama obrade, kako bi se mogla uraditi uporedna tehnoe ekonomska analiza [3].



Na bazi ovog plana iz dve ploče različitih debljina – 8 i 15 (mm) izrađenih od ugljeničnog čelika Č.1530 (C45), izvršeno je isecanje delova prema crtežu na slici 3. na mašinama za sečenje laserom, plazmom i gasnim sečenjem za uslove posmatranog proizvodnog sistema.

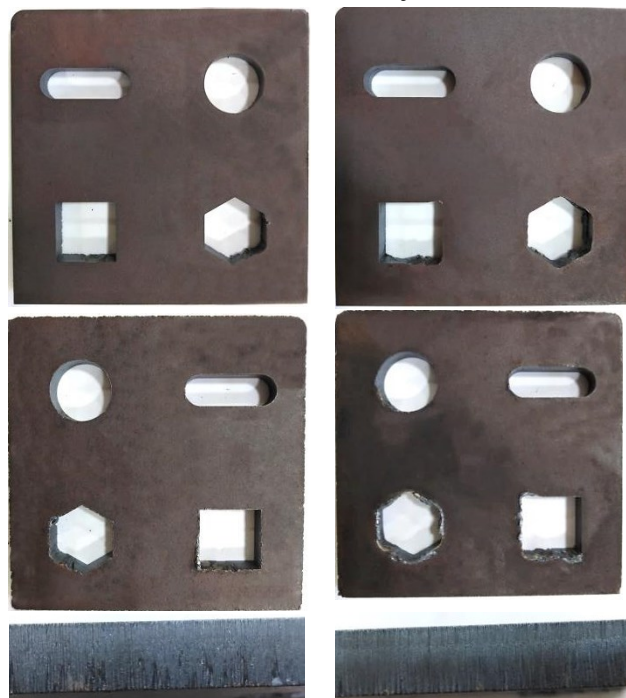
U tabeli 1, 2 i 3 prikazane su maksimalne i minimalne brzine sečenja v [mm/min] kao i proračunska vremena obrade za debljine materijala 8 mm i 15 mm.

Debljina materijala a(mm)	Lasersko rezanje			
	Max brzina v [mm/min]	Min.brzina v [mm/min]	tr [s] pri v _{max}	tr [s] pri v _{min}
8	700	560	59	73
15	380	200	108	205

Debljina materijala a(mm)	Plazma rezanje			
	Max brzina v [mm/min]	Min.brzina v [mm/min]	tr [s] pri v _{max}	tr [s] pri v _{min}
8	4350	2200	9	18
15	1650	940	25	44

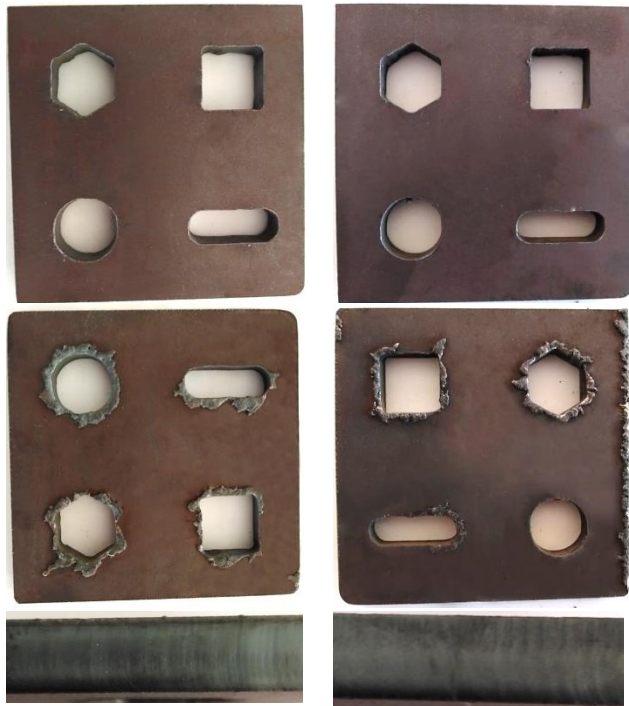
Debljina materijala a(mm)	Gasno rezanje			
	Max brzina v [mm/min]	Min.brzina v [mm/min]	tr [s] pri v _{max}	tr [s] pri v _{min}
8	480	380	85	108
15	450	350	91	117

Na narednim slikama su prikazani isečeni delovi, na svakom od obradnih sistema, za debljinu od 8mm

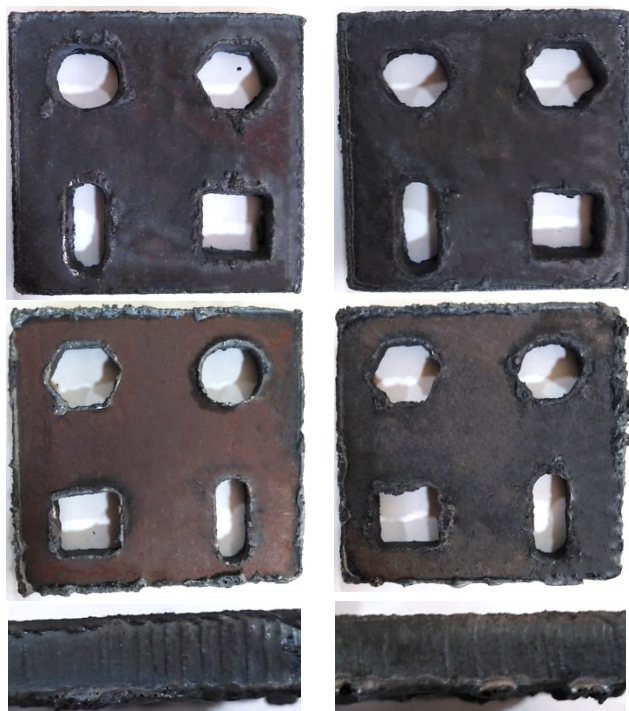


986

Na slici 4 se može videti da je opšte stanje obradaka prihvatljivo i da se ovakvi obradci sečeni laserom mogu koristiti uz dodatnu obradu kod manje zahtevnih proizvoda. Na nekoliko mesta se vide veća brazdanja materijala koja su nastala zbog parametara rezanja (najviše brzine) koji nisu optimalni za ove uslove, već su izabrani na bazi iskustva. Duž ivice reza se može uočiti vrlo mala zona uticaja toplote koja dostiže širinu od 1 do 2 mm. Na bočnim stranama su vidljive brazde i tragovi reza koji su zakošeni u pravcu brzine kretanja laserskog zraka, posebno kod veće debljine materijala i veće brzine.



Slika 5. Prednja, zadnja i bočna strana uzorka debljine 8mm sečenog plazmom pri V_{max} (levo) i V_{min} (desno)



Slika 6. Prednja, zadnja i bočna strana uzorka debljine 8mm sečenog gasom pri V_{max} (levo) i V_{min} (desno)

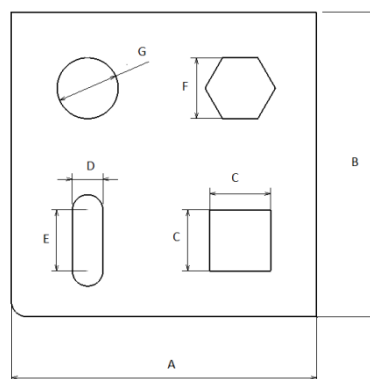
Na bazi vizuelnog posmatranja i analize analize obrađenih delova plazmom (slika 5) može se uočiti da je sa prednje strane stanje obradaka prihvatljivo, dok se na zadnjoj strani javlja taloženje rastopljenog materijala, koji se prilično lako odvaja. Vidljiva je zona uticaja toplote koja na nekim mestima dostiže i do 4 mm, a osim toga nema značajnih promena na površini. Bočna strana je sjajna, prihvatljivog kvaliteta sa vidljivim tragovima rezanja u vidu brazdanja materijala. Rez je zakošen, u obliku konusa. Kako bi ovi obradci mogli da se koriste za potrebe montaže ili neke slične namene potrebno je čišćenje zadnje strane i naknadna obrada konusnih površina ukoliko to zahteva konstrukcija proizvoda.

Na osnovu pregleda obrađenih delova gasnim rezanjem (slika 6) donešeni su sledeći zaključci. Stanje obradaka je prilično lose sa tim da je na prednjoj strani malo bolje nego na zadnjoj strani, gde se javlja taloženje značajne količine materijala u vidu troske. Izrazito je vidljiva zona uticaja toplote koja na nekim mestima dostiže i do 7 mm. Na bočnoj strani se nalazi rastopljena troska, neprihvatljivog kvaliteta sa vidljivim tragovima rezanja u vidu brazdanja materijala. Rez je zakošen, u obliku konusa. Obradci sa ovakvim rezom ne bi mogli da se koriste za potrebe montaže ili za ugradnju u neke druge konstrukcije jer ne zadovoljavaju vizuelnu kontrolu tj. potrebno je čišćenje troske i ponovna obrada dimenzija koje značajno odstupaju, što znači da bi dodaci za obradu bili veliki.

Zbog obima rada prikazaće se samo neki od dobijenih rezultata merenja iz [3].

Tačnost dimenzionih mera

Za merenje dimenzione tačnosti obrade korišćeno je pomično merilo klase tačnosti 0,05 mm. Na slici 7 su prikazane dimenzije koje su merene. Za svaku dimenziju izvršeno je merenje u nekoliko tačaka i usvojeno je maksimalno odstupanje.



Slika 7. Prikaz dimenzija koje su merene

Na osnovu 84 rezultata dimenzionih merenja za različite procese obrade, različite debljine i brzine rezanja dobijene su sledeće srednje vrednosti odstupanja od zadate mere:

- Obrada laserom: 0,47mm
- Obrada plazmom: 0,6mm
- Obrada gasom: 2,55mm

Dobijeni podaci ispitivanja tačnosti merenih dimenzija pokazuju da je najpreciznija mašina za sečenje laserom, dok iza nje slede plazma i na kraju gasno sečenje kod koga je odstupanje od dimenzija značajno.

Rezultati merenja ugla nagiba reza

Ugao nagiba reza, odnosno konus na obradenim delovima je meren mašinskim na obradcima debljine 8mm i 15mm za svaki proces sečenja.

Na delovima sečenim laserom uočen je najmanji konus na obradcima koji iznosi svega $1.5^\circ \div 3^\circ$, gde je veća mera na gornjoj strani obradka.

Na obradcima sečenim plazmom izmeren je konus koji iznosi $2^\circ \div 4^\circ$, na ovim obradcima manja mera je na gornjoj strani obradka, dok je na obradcima sečenim gasom izmeren konus koji iznosi $2^\circ \div 4^\circ$ kod koga je manja mera na gornjoj strani obradka.

Rezultati merjenja ravnosti

Prilikom sečenja materijala dolazi do znatnog uticaja toplote na radni komad. Uticaj toplote na radni komad uzrokuje deformacije na obradku, odnosno njegovo krivljenje.

Na osnovu izmerenih vrednosti maksimalnih odstupanja ravnosti može se zaključiti da je odstupanje od ravnosti, odnosno veličina iskrivljenosti delova usled toplote najveća kod gasnog rezanja a potom kod rezanja plazmom, dok je najmanje kod lasera, što se moglo i pretpostaviti s obzirom na prirodu procesa sečenja.

Rezultati merjenja hrapavosti obrađene površine

Hrapavost obrađene površine je merena pomoću uređaja MarSurf PS1 na spoljašnjoj površini reza na dužini od 17,5 mm. Na slici 8 prikazani su rezultati merenja hrapavosti obrade plazmom za debljinu 8mm pri $v_{\max}$.

Na osnovu rezultat merenja napravljena je tabela 4 dobijenih stepena hrapavosti N za posmatrane procese, mašine i sečene debljine materijala



Slika 8. Rezultati merenja hrapavosti za obradu plazmom za debljinu 8mm kada je V_{max}

Tabela 4. Rezultati merjenja hrapavosti

Djbljina materijala (mm)	Lasersko sečenje		Plazma sečenje	
	$V_{\min}$	$V_{\max}$	$V_{\min}$	$V_{\max}$
8	N9 - N10	N10	N7 - N8	N8 - N9
15	N10 - N11	N10 - N11	N8 - N9	N8 - N9

Rezultati vremena i troškova obrade

Vreme i troškovi proizvodnje predstavljaju osnovne kriterijume za optimizaciju tehnološkog procesa obrade u okviru proizvodnih procesa [4].

Merenje vremena obrade je izvršeno pomoću štoperice, čiji rezultati su predstavljeni u tabeli 5.

Tabela 5. Uporedni prikaz vremena sečenja

a (mm)	Laser		Plazma		Gasno sečenje	
	V _{min}	V _{max}	V _{min}	V _{min}	V _{max}	V _{min}
8mm	106 s	80 s	97 s	74 s	210 s	191 s
15mm	232 s	154 s	80 s	66 s	223 s	209 s

Na osnovu ovih rezultata vidi se da ovo vreme malo odstupa od proračunatih vrednosti, te da je najkraće vreme sečenja plazmom, potom laserom, dok je gasno sečenje očekivano najduže.

Na osnovu analize troškova obrade, odnosno cene gotovih delova na bazi izvršenog proračuna u radu [3], donesen je zaključak da je za razmatrane delove najekonomičnije rešenje obrade - sečenja plazma postupkom, a u slučaju da se traži veća tačnost obrade onda bi se kao rešenje izabralo sečenje laserom.

5. ZAKLJUČAK

U zavisnosti od zahteva koji se postavljaju potrebno je preporučiti odgovarajući postupak rezanja, što je i bio jedan od ciljeva rada. Shodno tome u nastavku su date kratke preporuke na bazi sprovedenih istraživanja. Ako je potrebna velika proizvodnost, zadovoljavajućeg kvaliteta, malo više cene, a pri tome debljina obradaka ne iznosi više od 10-15 mm obrada će se vršiti na laseru. U slučaju da se zahteva dobar kvalitet po prihvatljivoj ceni i da su debljine obradaka veće mora se koristiti plazma, pri čemu treba voditi računa da plazma reže samo metalne delove. Ako se moraju seći jednostavni komadi, debljine preko 50 mm, lošijeg kvaliteta sa niskom cenom, gasno sečenje je najbolji izbor.

U cilju dobijanja detaljnijih podataka o karakteristikama posmatranih obradnih sistema, kao nastavak istraživanja predviđa se optimizacija parametara obrade, ispitivanje procesa obrade za druge vrste materijala, kao i definisanje troškova obrade ovih procesa u zavisnosti od vrste materijala, debljine, parametara obrade i dr.

6. LITERATURA

- [1] Martinovski V., Antić M.: *Noviteti u termičkim postupcima rezanja*, Mašinski Fakultet, Beograd, 2011.
- [2] Gostimirović, M.: *Nekonvencionalni postupci obrade*, FTN izdavaštvo, Novi Sad, 2016.
- [3] Tiganj, M.: *Uporedna tehnoeekonomska analiza različitih procesa sečenja čeličnog lima*, Master rad, Fakultet tehničkih nauka, Novi Sad, 2018.
- [4] Todić, V., Penezić, N., Lukić, D., Milošević, M.: *Tehnološka logistika i preduzetništvo*, FTN izdavaštvo, Novi Sad, 2011.

Kratka biografija:



Mirsad Tiganj je rođen u Vrbasu 1991. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mašinstva – Proizvodno mašinstvo / Mehanizacija i konstrukciono mašinstvo, odbranio je 2018. god.

kontakt: mirsad.tiganj991@gmail.com

PELETIRANJE DRVNE MASE I TIPOVI PROIZVODNJE**PELLETING OF WOODEN MASS AND PRODUCTION TYPES**Nemanja Bandić, Borislav Savković, *Fakultet tehničkih nauka, Novi Sad***Oblast – PROIZVODNO MAŠINSTVO**

Kratak sadržaj – U ovom radu prikazan je proces proizvodnje peleta od drvene mase kroz proizvodne korake, faktori koji utiču na proizvodnju peleta, kao i tipovi i varijante proizvodnih pogona za peletiranje drvene mase.

Ključne reči: Pelet, Biomasa, Obnovljivi izvori energije

Abstract – This paper presents the process of wood pelleting through main steps, factors that influence the production of pellets, as well as the types and variants of the production lines for wood pelleting.

Keywords: Pellet, Biomass, Renewable energy sources

1. UVOD

Obnovljiva energija koristi izvore energije koji se prirodno kontinualno dopunjavaju Suncem, vetrom, vodom, toplotom Zemlje i biljkama. Tehnologije obnovljive energije pretvaraju ova goriva u korisne oblike energije - najčešće električnu energiju, ali i toplotnu, hemijsku ili mehaničku energiju [1].

Danas se fosilna goriva pretežno koriste za zagrevanje i snabdevanje kuća, kao i za goriva automobila. Pogodno je koristiti uglja, naftu i prirodni gas kako bi se zadovoljile energetske potrebe, ali je snabdevanje ovim gorivima ograničeno, zato što se koriste brže nego što se stvaraju. Zbog sigurnosnih razloga i problema sa odlaganjem otpada, Sjedinjene Američke Države će „penzionirati“ većinu nuklearnih kapaciteta do 2020. godine. U međuvremenu, očekuje se da će energetske potrebe zemlje porasti za 33 procenta u narednih 20 godina.

Čak i da je neograničena količina fosilnih goriva, korišćenje obnovljive energije je bolje za životnu sredinu. Često se tehnologije obnovljive energije nazivaju "čiste" ili "zelene", jer su mali zagađivači. Međutim, sagorevanje fosilnih goriva emituje stakleničke gasove u atmosferu, zarobljavajući sunčevu toplotu i doprinoseći globalnom zagrevanju. Naučnici tvrde da je prosečna temperatura Zemlje porasla u prošlom veku. Ako se taj trend nastavi, nivo mora će se povećati, a naučnici predviđaju da će se poplave, talasi vrućina, suše i drugi ekstremni vremenski uslovi sve češće javljati.

2. PELET

Proizvodnja peleta je počela pre više od jednog veka. Koristeći toplotu i pritisak, pelet se može proizvoditi iz različitih materijala za različite namene.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Borislav Savković, docent.

Neke kompanije koje su koristile pelet za proizvodnju stočne hrane počele su 1970-ih da se bave proizvodnjom drvenog peleta kao izvora goriva. Međutim, zbog jeftinih fosilnih goriva koja su bila na raspolaganju, proizvodnja peleta od drveta nastavila je da raste. U 21. veku drveni pelet i uopšte pelet imaju ekspanziju, iako postoje problemi pri snabdevanju. Na slici 1. prikazan je izgled drvenog peleta.



Slika 1. Izgled drvenog peleta

Zbog visokih cena nafte i gasa i zabrinutosti zbog klimatskih promena, pelet postaje čisto i jeftino gorivo za grejanje koje može pomoći smanjenju globalnog zagrevanja. Tokom poslednje decenije postojala su dva glavna faktora koja su izazvala rast peleta na tržištu. Prvi je konstantan rast troškova fosilnih goriva i nestabilnost cena, a drugi je povećana pažnja na korišćenje fosilnih goriva, kao što su nafta i gas i njihov uticaj na životnu sredinu. Ostali faktori koji doprinose razvoju peleta su da se pelet može proizvesti na lokalnom nivou, od lokalnog drveta i biomase. Proizvodnja i distribucija peleta može dati pristupačno gorivo, kao vrstu grejanja i stvarati tako lokalne poslove.

Primarna svrha peletiranja materijala je olakšati rukovanje i smanjiti troškove transporta. Zapremina je često ograničavajući faktor kada se radi o prevozu. Povećanjem gustine materijala, potrebna zapremina za transport određene količine materijala je snižena.

Prednosti peleta u odnosu na briket i drvo kao ogrevni materijal su:

- Cena – pelet od drveta i biomase, kao alternativa fosilnim gorivima, u mnogim zemljama ima konkurentniju i stabilniju cenu, u odnosu na prirodni gas i naftu. Kao takav, drveni pelet se pokazao kao efektna kontramera u borbi protiv globalnog zagrevanja, energetske sigurnosti i povećanja cene nafte.
- Širok asortiman materijala - Što se tiče peleta na bazi drveta i biomase, izvor sirovina je raznovrstan, kao što je otpad od drveta (piljevina, sečka, itd.), dvorišni otpad (trava, lišće, grančice, grmlje itd.), farmški

otpad (kukuruz, kukuruzna stabla, slama itd.) i ostali ostaci otpada iz biomase. Kao tradicionalno grejanje, fosilna goriva su skuplja i lako izazivaju zagađenje životne sredine. Drveće (ogrev) raste sporo i teško je kontrolisati vlagu, pa tako da obe navedene vrste goriva, nisu dobre za peći i kotlove.

- **Zaštita životne sredine** - prednost peleta od drveta i biomase je što ne emituje CO₂. Pri sagorevanju peleta oslobađa se samo CO₂ koji se čuva tokom života biljke i bezopasan je za životnu sredinu, dok sagorevanje fosilnih goriva oslobađa dodatni CO₂ u atmosferu koji se skladišti milionima godina i tako ubrzava globalno zagrevanje.
- **Pogodnost** - Pelet se proizvodi sa jedinstvenim sadržajem vlage, oblikom, veličinom i gustinom koja odgovara potrebama automatizovanih sistema sagorevanja peći i kotlova, a takođe zauzima manje prostora u skladištenju od drugih biomasa.

Ispod su navedeni neki od termina u proizvodnji peleta od strane Pellet Fuels Institute Standard Specification for Residential/Commercial Densified Fuel, October 25, 2010:

- **Bulk density** - zapreminska gustina u uzorkovanom peletu
- **Diameter** - prosečan prečnik uzorkovanih peleta
- **Pellet Durability Index (PDI)** - standardizovani parametar za specifikaciju sposobnosti peleta da se odupre degradaciji prilikom transporta i rukovanja
- **Fines** - procenat materijala peleta u uzorku koji prolazi kroz ekran od 3,7 mm
- **Inorganic ash** - procenat neorganskog materijala u uzorkovanom peletu
- **Length** – težinski procenat peleta dužine 38,1 mm u uzorkovanom peletu
- **Moisture** – sadržaj vlage u uzorkovanom peletu
- **Heating value** – Toplota sagorevanja uzorkovanog peleta

3. PELETIRKE I PROCES PELETIRANJA

Peletirka, tj. mašina za proizvodnju peleta, predstavlja glavni deo procesa peletiranja. Postoje dva tipa peletirki od kojih je jedna sa ravnom matricom (engl. flat die), a druga je sa prstenastom matricom (engl. round die). Ravna matrica za pelet se pojavila prva, a prstenasta je kasnije projektovana.

Uobičajeno je da se ravne koriste za male i srednje proizvodnje peleta, a prstenaste za srednje i velike proizvodnje peleta [2].

3.1. Peletirke sa ravnom matricom

Na slici 2. prikazana je peletirka sa ravnom matricom koja radi na principu pada materijala odozgo na valjke koji se rotiraju preko matrice. Materijal se onda kompresuje između valjaka i površine matrice kroz otvore. Kada pelet izađe kroz otvor matrice, nož odseca pelet na određenu dužinu.

Pogon preko puža i pužnog točka se koristi kod nekih peletirki sa ravnim matricama, dok se kod drugih pogon vrši preko zupčanika.

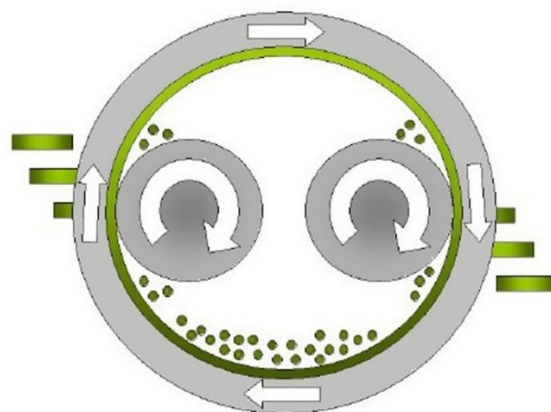


Slika 2. Princip rada peletirke sa ravnom matricom

3.2. Peletirke sa prstenastom matricom

Na slici 3. prikazan je princip rada peletirke sa prstenastom matricom. Peletirke sa prstenastom matricom se sastoje od vertikalne prstenaste matrice sa valjcima na unutrašnjoj strani, koji stvaraju pritisak na matricu.

Materijal se napaja iz spremnika preko kondicionera sa varijabilnom brzinom iznad peletirke; odgovarajući materijal se zatim unosi kroz vrata peletirke i odlazi u centar komore peletirke. U unutrašnjosti komore valjci su stacionarni i matrica je gonjena, slično mašini za pranje veša. U komori materijal se pokreće rotirajućom matricom i zatim se kompresuje valjcima.

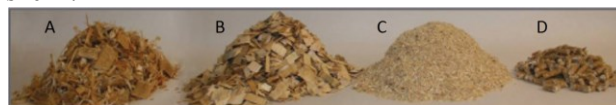


Slika 3. Princip rada peletirke sa prstenastom matricom

3.3. Proces peletiranja

Drveni pelet se često proizvodi iz otpadnog materijala drugih proizvoda, mada nekoliko velikih kompanija koje se bave proizvodnjom peleta, pripremaju gotovo drvo samo za peletiranje [3].

Sirovi materijal koji se koristi za proces peletiranja može se pripremiti na različite načine, primeri su prikazani na slici 4.



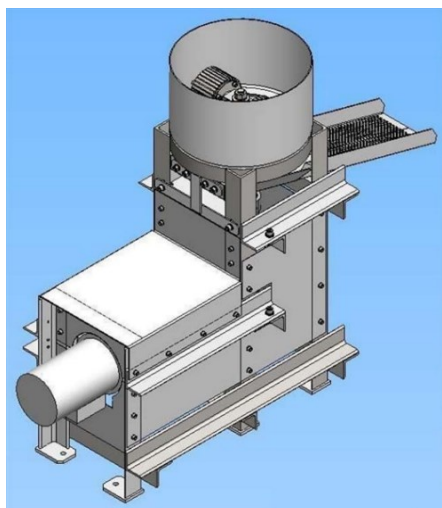
Slika 4. Materijali za peletiranje

Proces peletiranja prolazi kroz nekoliko osnovnih koraka, koji su standardni za sve vrste proizvodnje peleta, bilo da se radi o mini serijama ili masovnoj proizvodnji, a to su [4]:

- ### 3.4. Faktori koji utiču na proces peletiranja

- **Tip drveta (vrsta)**
- **Temperatura**
- **Vlažnost**
- **Veličina čestica i orijentacija vlakana**

4.1. Proizvodnja 250kg/h peleta

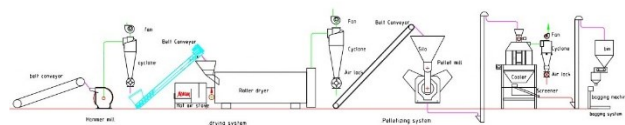


Slika 5. Peletirke za proizvodnju 250kg/h peleta

Dovoljan je jedan radnik koji će ručno vršiti dosipanje materijala na valjke. Materijal mora biti prethodno samleven kako bi mogao proći kroz matricu. Konstrukcija ovakve peletirke omogućava laku i brzu zamenu potrošnih delova u bilo kom trenutku proizvodnje. Kako bi se napravila kompletna linija za proizvodnju peleta, osim

Glavne karakteristike proizvodnih linija ovakvog tipa su jednostavnost upravljanja, ne zauzimaju puno prostora i predstavljaju najjeftiniju varijantu. Idealno je rešenje za one koji se bave preradom drveta, a imaju problem sa otpadom, tj. piljevinom.

4.2. Proizvodnja 500kg/h peleta



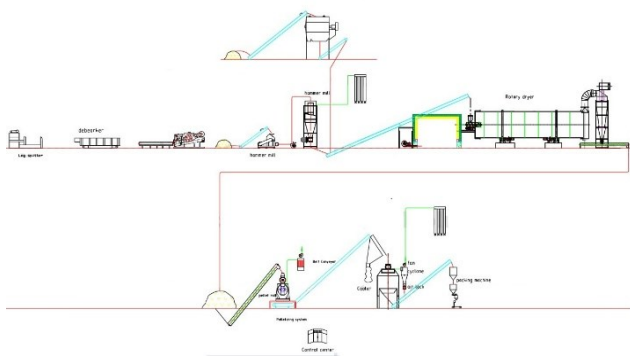
Slika 6. Tip proizvodnje od 500kg/h peleta

1. Sistem za usitnjavanje drvenog otpada
2. Sistem za sušenje
3. Sistem za peletiranje
4. Sistem za pakovanje

Proces peletiranja kreće od ubacivanja drvenog otpada u mlin čekićar. To se može vršiti transportnom trakom kako bi se ravnomerno ubacivao radni materijal. Usitnjeni materijal iz mlina se prenosi pužnim transporterom do sušare. Nakon što materijal prođe kroz sušaru, transportnom trakom se odvodi u silos, odnos spremnik peletirke, gde se materijal pod pritiskom kompresuje u pelet. Gotov pelet se šalje u hladnjak, gde će se ohladiti i pripremiti za pakovanje. Na izlasku is hladnjaka postavlja se sito, kako bi se odstranila sva prašina i pelet koji se pretvorio u sitne delove. Nakon toga pelet se šalje u korpu iznad pakerice, gde se vrši pakovanje u vreće. Između podsistema se instaliraju cikloni koji uvlače prašinu i tako omogućuju čistu sredinu za rad.

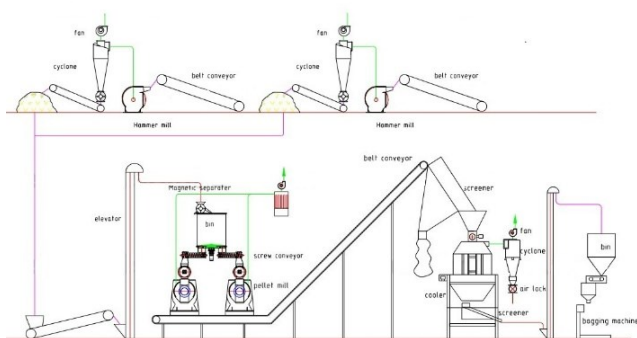
4.3. Proizvodnja 2t/h peleta

Princip rada peletirke isti je kao i kod ostalih peletirki sa prstenastim matricama. Dobijeni pelet se šalje u hladnjak na hlađenje, a posle na pakovanje. Kompletним sistemom se upravlja iz kontrolnog centra.



Slika 7. Tip proizvodnje od 2t/h peleta

4.4. Proizvodnja 4t/h peleta



Slika 8. Tip proizvodnje od 4t/h peleta

Na slici 8. je prikazan tip proizvodnje peleta od 4t/h. Ovakav tip predstavlja masovnu proizvodnju peleta. U proizvodnoj liniji se nalaze dve peletirke sa prstenastim matricama od kojih svaka proizvodi po 2t/h.

Radni materijal može biti sa više izvora, ali se zajedno dovodi u bunker iznad obe peletirke, gde se dalje pužnim prenosnikom ravnomerno dozira u peletirke. Nakon toga pelet ide na hlađenje i pakovanje. Kompletan proces je automatizovan.

5. ZAKLJUČAK

Zbog različitosti upotrebe, procesi proizvodnje drvenog peleta se mogu razlikovati od postrojenja do postrojenja. Na primer, ako se drveni pelet pravi za ličnu upotrebu, nisu potrebni svi proizvodni koraci, tj. proces pakovanja nije potreban zato što se pelet može koristiti direktno iz spremnika peleta. Međutim, ako je pelet napravljen u komercijalne svrhe, takvom peletu je potrebno odgovarajuće pakovanje, odnosno potrebna je mašina za pakovanje peleta u vreće.

Peletiranje biomase ima neke važne prednosti:

- upotreba peleta kao alternativne održive energije je efikasan alat u borbi protiv klimatskih promena;
- manji troškovi transporta, manja količina prašine, veće grejne vrednosti i manje zagađenje [6]
- smanjeno sečenje šuma, pružajući zamenu drvetu koji se koristi kao ogrev

Glavni nedostaci peletiranja biomase su:

- emisije nekvalitetnog peleta dovode do korozije
- sitne čestice peleta mogu poremetiti kontrolu automatskog zagrevanja sistema ili uzrokuju prekid automatskog snabdevanja peletom
- sitne čestice brže gore i dovode do topljenja prašine
- prašina nastala raspadom nekvalitetnog peleta predstavlja zdravstveni rizik, a takođe može dovesti do paljenja i eksplozije tokom rukovanja, skladištenja i transporta.

6. LITERATURA

- [1] Johansson, T. B. , K. McCormick , L. Neij , and W.C. Turkenburg , „*The potentials of Renewable Energy*“, Bonn, 2004.
- [2] Kaltscmitt, M, „*Energie aus Biomasse: Grundlagen, Techniken und Verfahren*“, Springer, 2009.
- [3] J. K. Holm, U. B. Henriksen, J. E. Hustad, L. H. Sorensen, „Toward an understanding of controlling parameters in softwood and hardwood pellets, production“ *Energy & Fuels*, pp. 2686-2694, 2006.
- [4] <http://www.pelletmillsolution.com/blogs/7-Steps-of-Fuel-Pellets-Production-Process.html> (pristupljeno u oktobru 2018.)
- [5] Wolfgang Stelte, Jens K. Holm, Anand R. Sanadi, Søren Barsberg, Jesper Ahrenfeldt, and Ulrik B. Henriksen, „Fuel pellets from biomass: The importance of the pelletizing pressure and its dependency on the processing conditions“, *Fuel*, 90 (11) pp. 3285 – 3290, 2011.
- [6] M.V. Gil, P. Oulego, M.D. Casal, C. Pevida, J.J. Pis, F. Rubiera, „Mechanical durability and combustion characteristics of pellets from biomass blends“, *Bioresource Technology*, vol. 101, pp. 8859–8867, 2010.

Kratka biografija:



Nemanja Bandić rođen je u Doboju 1993. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Proizvodnog mašinstva odbranio je 2016.god.

kontakt: bandic10@hotmail.com



Borislav Savković rođen je u Novom Sadu 1982. god. Doktorirao je na Fakultetu tehničkih nauka 2015. god. gde je i zasnovao radni odnos u zvanju docenta. Autor je jednog univerzitetskog udžbenika i preko 100 naučnih radova. Oblast interesovanja su procesi obrade skidanjem materijala, simulacije kao i ekološko tehnološki sistemi.

REDIZAJN BRUSILICE ZA SPOLJAŠNJE KRUŽNO BRUŠENJE

REDESIGN OF THE EXTERNAL CYLINDRICAL GRINDING MACHINE

Uroš Rodić, Marin Gostimirović, *Fakultet tehničkih nauka, Novi Sad*

Oblast - MAŠINSTVO

Kratak sadržaj – U radu je prikazan redizajn vreteništa tocila sa kliznim uležištenjem, usvojeno rešenje i verifikacija izvedenog rešenja zajedno sa rezultatima ispitivanja. Predstavljen je problem određene brusilice u eksploataciji i predlog rešavanja tog problema izmenom dizajna.

Ključne reči: Industrijski dizajn, Brusilica, Vretenište tocila, Klizni ležaj

Abstract – This paper presents redesign of wheel head with plain bearings, adopted solution, and it's verification together with the results of the test. The problem of a certain grinding machine in the exploitation is presented and the solution for this problem by design change.

Key words: Industrial design, Grinding machine, Wheel head, Plain bearing

1. UVOD

U savremenom društvu se razvoj i unapređenje proizvodnih struktura postavlja kao imperativ, posebno industrijske proizvodnje koja ima najveći potencijal da stvori uslove za stabilan razvoj društva i poboljšanje životnog standarda građana. U tom kontekstu, do posebnog izražaja dolaze sposobnosti inženjera da svakodnevno unapređuju postojeća tehnička i tehnološka rešenja proizvoda, sredstava rada i postupaka obrade, tj. da omoguće njihovu efikasnu primenu u različitim oblastima društvenog života.

U ovom radu će biti reči o dizajnu u proizvodnji alatnih mašina i implementaciji novih rešenja u već postojeće konstrukcije. Konkretno se radi o industrijskom redizajnu postojeće brusilice za spoljašnje kružno brušenje koja mora da ispuni posebno postavljene zahteve određenog korisnika.

U modernim i moćnim mašinama alatkama sve više se povećava značaj uležištenja vratila. Rešenja za ove zadatke nude valjkasti i kuglični, kao i klizni ležajevi. U poslednjih nekoliko godina, sve su veći zahtevi za dug životni vek mašine alatke, kao i za dobre performanse u izradi površina, tačnost oblika i tolerancije izrade delova. Potražnja je naročito za implementacijom hidrodinamičkih segmentalizovanih kliznih elemenata. Ovakvi ležajevi obezbeđuju visoke performanse što se krutosti tiče, obezbeđuju visoke brzine obrtanja, centričnost komada, otporni su na udare, a pokazuju visoku izdržljivost i dugotrajnost.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Marin Gostimirović, red. prof.

Rad prikazuje više idejnih rešenja vreteništa tocila, odabir, realizacija i testiranje najkompetentnije izvedbe, korišćenje uočenih nedostataka, kao i formiranje tehnički korektnog rešenja sa verifikacijom istog.

U analizi sprovedene tehnologije izrade redizajniranog vreteništa tocila sa kliznim uležištenjem, akcenat je bio na praktičnom pristupu projektovanja, izrade i montaže navedenog proizvoda u postojećim proizvodnim uslovima.

Proces verifikacije i kontrole realizovane mašine alatke, sproveden je na mernoj mašini „Taylorond 31c“ sa prikazanim konkretnim rezultatima za različite režime obrade.

Dakle, akcenat celog rada je na pronalaženju adekvatnog rešenja vreteništa tocila brusilice za spoljašnje kružno brušenje po nalogu konkretnog kupca, a da finalni proizvod sadrži zahtevani kvalitet u funkcionalnost, uz neophodnu ekonomsku isplativost.

2. INDUSTRIJSKI DIZAJN

Dizajn se definiše kao intelektualni, tehnički i kreativni proces koncipiranja i oblikovanja predmeta, prostora, usluga i vizuelnih komunikacija [1]. Može se shvatiti kao proces koji zahvata sve faze kroz koje proizvod prolazi u procesu razvoja – od zamisli do konačnog plasmana na tržište. On obuhvata dosta različitih disciplina, gde svaka od njih ima vlastitu specijalizovanu svrhu, pripremu i praksu. Dizajn proizvoda je multidisciplinarni proces koji uglavnom uključuje tržište i tehnološka istraživanja, koncept proizvoda, prototip, konačni razvoj proizvoda i testiranje, kao i post proizvodno prilagođavanje.

Dobar dizajn proizvoda je u funkciji uspešnog poslovanja preduzeća. Postoje tri bitna područja u kojima odlučivanje o dizajnu ima izuzetno značenje:

- identitet preduzeća,
- dizajn proizvoda i
- identitet marke.

Identitet preduzeća predstavlja sliku preduzeća kroz arhitekturu, unutrašnje uređenje, proizvode, logotipe, slogane i ekonomsku propagandu.

Dizajn proizvoda odnosi se na izbor kombinacije funkcionalnih, strukturnih i estetskih obeležja. Funkcionalna obeležja su povezana sa koristi koja se očekuje od proizvoda. Na primer, kišobran treba da pruži zaštitu od kiše. Strukturna obeležja se odnose na mogućnost ispunjavanja funkcionalnih obeležja proizvoda i uključuju odluke o veličini, obliku, materijalu. Na primer, kišobran može biti veličine koja omogućava njegovo nošenje u torbi i može biti izrađen od prozirnog materijala kroz koji

se može gledati. Estetska se obeležja odabiraju i kombinuju tako da daju osećajnu privlačnost proizvodu.

Identitet marke uključuje imena, reči, znakove, simbole i/ili oblike koji se koriste u cilju razlikovanja proizvoda jednog proizvođača ili prodavca od konkurencije. Identitet marke u funkciji je stvaranja imidža proizvoda, utiče na odluke o ceni i stvara percepciju o kvalitetu proizvoda.

Osnovni uslovi koje mora ispuniti dobar dizajn vezani su za:

- upotrebljivost proizvoda,
- ergonomsu prilagodljivost,
- tehničku/ekonomsku pouzdanost,
- estetsku senzibilnost i
- doslednost imidža.

3. OBRADA BRUŠENJEM

3.1. Uopšteno o brušenju

Brušenje je najčešći postupak završne obrade rezanjem, a koristi se i kao postupak grube obrade.

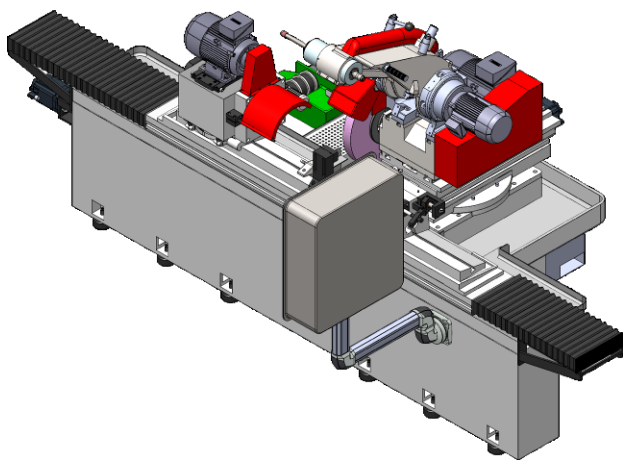
Kao gruba obrada, brušenje se koristi kod čišćenja odlivaka i otkovaka (skidanje oksidnog sloja), a kao završna obrada javlja se u svim slučajevima kada je potrebno poboljšati kvalitet obrađene površine, posle grube obrade ili ako su delovi termički obrađeni [2,3].

Osnovna podela klasičnih mašina za obradu brušenjem podrazumeva:

- brusilice za kružno brušenje,
- brusilice za ravno brušenje,
- brusilice za glačanje,
- brusilice za specijalne namene i
- numerički upravljane brusilice.

4. UNIVERZALNA BRUSILICA FB BSC 237

Predmet ovog rada jeste redizajn univerzalne brusilice za spoljašnje kružno brušenje FB BSC 237 iz proizvodnog programa firme GRINDEX. Firma koja je proizvođač brusilica za brušenje spoljnih i unutrašnjih cilindričnih površina.



Slika 1. Model brusilice FB BSC 237 izrađen u SOLIDWORKS-u

5. VRETENIŠTE TOCILA SA KLIZNIM ULEŽIŠTENJEM

Jedan od upita koji je pristigao od kupaca, bio je da vretenište tocila brusilice za spoljno kružno brušenje bude sa hidrodinamički podmazivanim kliznim ležajevima.

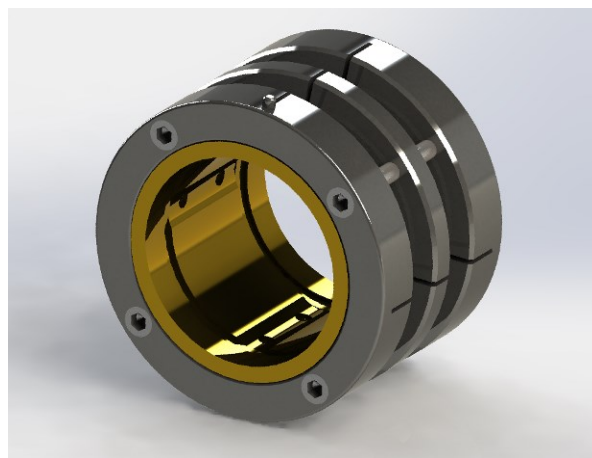
Kao referenca za dalju problematiku, korišćeno je rešenje firme „Livnica Kikinda“ koja je svojevremeno proizvodila ovakva vreteništa, ali po tehnologiji i licenci Nemačkog giganta u proizvodnji alatnih mašina „Fortuna“. Sudeći da Grindex nema licencu za korišćenje ovog patenta, moralo se pristupiti pronalaženju rešenja koje bi se dalje implementiralo utekući proizvodni program. Kasnije ako se ova solucija pokaže kao dobra, dugotrajna i tehnički korektna, patentiraće se i nastaviti ugradnju u iduće generacije mašina alatki.

U tom kontekstu, u daljem radu pristupilo se redizajniranju vreteništa tocila sa novim ležajevima i novim sistemom za podmazivanje.

5.1. Idejno rešenje

Sudeći da postoji zaliha odlivaka od starih mašina, praksa je pokazala da korišćenjem ovih delova, ne samo da se pojeftinjuje proizvodnju, već i u velikoj meri ostvaruje visoka tačnost mašine u daljoj eksploataciji. Takođe, istraživanjem i sagledavanjem tržišta, došlo se na ideju da je dobra solucija iskoristiti klizne ležajeve Nemačke proizvodnje, marke SPIETH (Sl. 2) [4].

Dakle, ovi ležajevi su iskorišćeni za novo radijalno uležištenje redizajniranog vreteništa tocila.



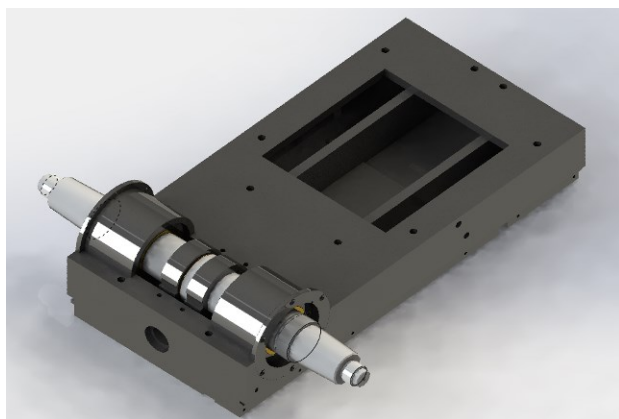
Slika 2. SPIETH radijalni klizni ležaj MSR 85.100

5.2. Realizovano rešenje

Početno idejno rešenje obuhvatilo je dva ista ležaja, jedan pozicioniran sa strane tocila, drugi sa strane remenice i konstrukcijski bi našli mesto unutar dve zasebne čaure (prirubnice), slika 3. Ovakvo rešenje nije baš najbolje u smislu greške koja će se neminovno pojaviti usled netačnosti obrade. Za aksijalni ležaj, isprva, namera je bila da se iskoristi postojeća (Fortunina) tehnologija.

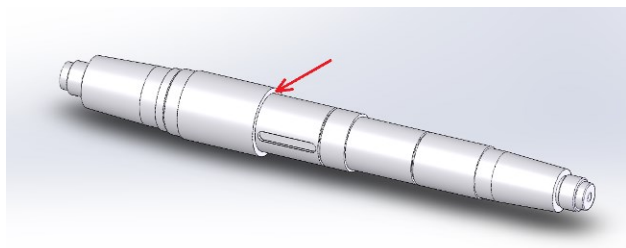
U daljem razmatranju odustalo se od ovog rešenja. Zamisao je da se po potrebi vreteno može izvući na jedan kraj. Ovakva izvedba to ne dozvoljava. U slučaju nekog servisiranja, zamena vretena ili sl., moralo bi se sve demontirati. U staroj (Fortuna) konstrukciji, ovo nije bio problem, štaviše, ovakav je bio redovan postupak (sudeći

da se tamošnja konstrukcija radialnih ležajeva sastojala iz dva dela povezana zavrtnjevima.)



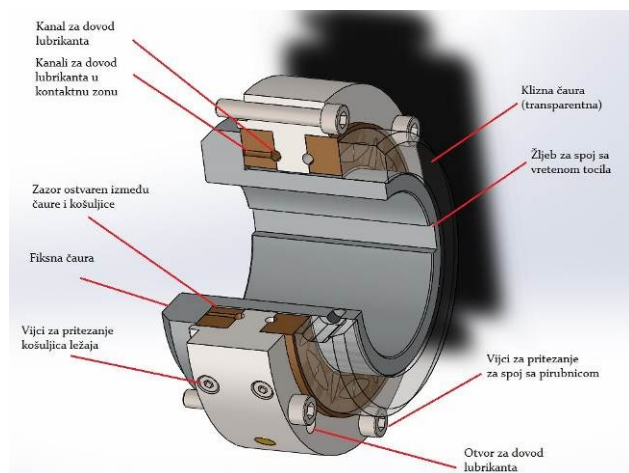
Slika 3. Idejno rešenje vreteništa tocila

U usvojenom rešenju, došlo se na ideju da se izradi stepenasto vratilo (sl. 4). Konus na delu do prirubnice tocila bi bio standardan, kako bi se mogla koristiti stara izvedba prirubnice. Remenica bi morala biti korigovana, iako se radi o istom konusu (1:10), a otvor bi bio manjeg promera. Naime, jeftinije je izraditi novu remenicu koja se pravi od SL nego prirubnicu, protuprirubnicu i sl. Isto tako, bitnije je više ukrutiti deo vretena na koji dolazi točilo, nego suprotno. Pri tome za ovo konstrukciono rešenje, bilo je potrebno uzeti u razmatranje dva ležaja različitog prečnika.



Slika 4. Stepenasto vratilo

Na slici. 5, strelicom je pokazan stepen na koji bi se oslonilo vreteno, a time obezbedilo osiguranje od aksijalnog pomeranja. Izgled prve varijante aksijalnog ležaja prikazan je na slici 5.

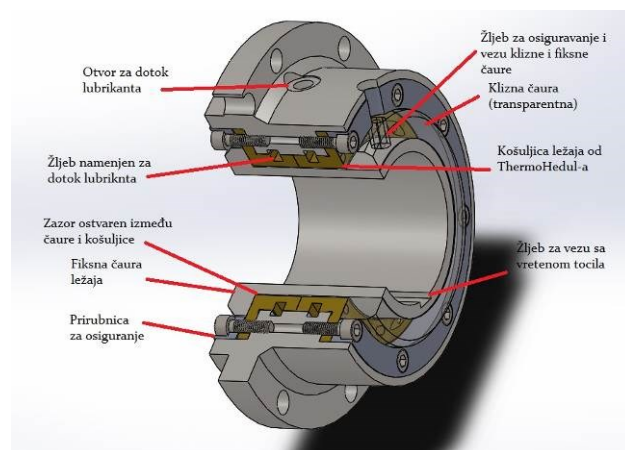


Slika 5. Sklop varijante aksijalnog ležaja

Glavni nedostatak, koji je srećom na vreme uočen je baš u aksijalnom ležaju. Naime, kada se dovede pritisak ulja za podmazivanje, pojavilo se enormno curenje i pritisak u samom vreteništu nije bio dovoljan. Uz ovakav problem, nije bilo moguće pokrenuti motor. Stoga se moralo pribeći ugradnji novog tipa uležištenja

6. VREteniŠTE TOCILA BSC

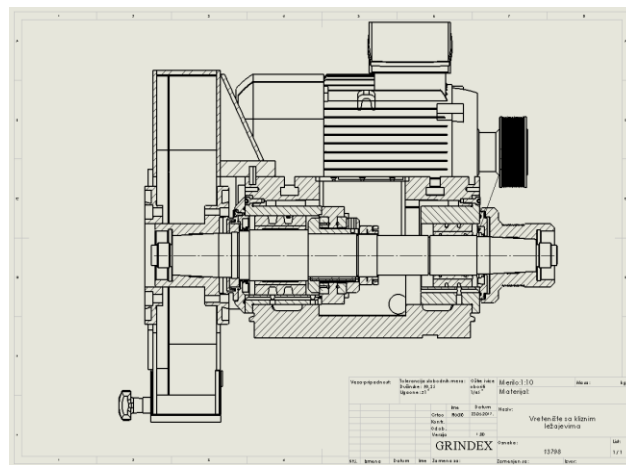
U ovoj konstrukcionoj verziji odabrani su radialni ležajevi po ugledu na prečnik vratila, opterećenje i uslove rada. Pa se pažnja posvetila redizajnu aksijalnog ležaja. Kao prvo, stezanje košuljice u ležaju mora biti sa čela, tj. sa strane na koje se naslanja na kliznu površinu čaure. Mesto naleganja košuljice mora biti obrušeno po mogućnosti bez udara na osu čaure. Lepovanjem se obezbeđuje naleganje na čaure po kojima košuljica ležaja klizi (efekat hidrodinamičkog filma). Ovom verzijom ležaja, obezbeđeno je dobro zaptivanje, postignuto dobro plivanje vretena i temperatura u radu je bila u propisanim granicama.



Slika 6. Presek modela ležaja za BSC vretenište

Pri montaži vreteništa tocila, isprva se montira aksijalni ležaj na prednju prirubnicu. Stegnu se svi zavrtnjevi po obodu. Montiraju dve čaure unutar kućišta. Potom se ubacuju aksijalni ležajevi, vodeći računa o žljebovima kroz koje se radialno fiksira oplata ležaja uz pomoć čivija.

Ostatak konstrukcije se nije mnogo razlikovao od pređašnje verzije.



Slika 7. Sklopni crtež vreteništa tocila sa kliznim uležištenjem

7. VERFIKACIJA REALIZOVANOG REŠENJA

7.1. Merni uređaj Taylorond 31c

Ovaj uređaj potiče iz familije sličnih, čiji su patenti vlasništvo firme Taylor/Hobson. Ovaj tip instrumenta radi na komprimovan vazduh i jednostavan je za rukovanje. Prevashodna namena mu je za merenje kružnosti, ovalnosti, koncentričnosti, ravnosti, upravnosti, kvadratičnosti, cilindričnosti i saosnosti udara. Robusne je konstrukcije i pogodan za upotrebu u pogonskim uslovima.



Slika 8. Taylorond 31c

Instrument poseduje 2 osnovna modula:

1. Bazni modul - Ovaj modul nosi motorizovan radni sto (montiran na vretenu sa vazдушnim jastukom/ležajem) i stub sa motorizovanim nosačima u kojima je montirana motorizovana radijalna ruka (merni krak).
2. PC sa softverskim paketom Taylorond 31c. - Isti obezbeđuje korisničko radno okruženje i omogućava analizu podataka i izračunavanje izlaznih rezultata. Jedan opcion nosač pomoćnog merača – sonde (za rad sa dva merača) takođe je na raspolaganju.

Obradak (predmet merenja) postavlja se na radni sto, a osa predmeta merenja je u ravni sa osom obrtanja vretena. Merni pipak dodiruje površinu obratka koja se okreće oko sopstvene vertikalne ose. Radijalna kretanja se pomoću merača pretvaraju u električne signale, a signali se potom prenose do računara radi dalje obrade.

7.2. Rezultati merenja

U nastavku, biće prikazani rezultati merenja rađeni za tri različita režima rezanja. Po probnom merenju, vrši se potrebna korekcija. Rezultati se upisuju u dijagram tzv. „Slezinger“, ili ti normativ tačnosti mašine.

U prvom merenju dobili su se rezultati: $0.95\ \mu\text{m}$ za kružnost, $0.1\ \mu\text{m}$ za ekscentričnost i udar od $0.95\ \mu\text{m}$. Ovi rezultati su prilično dobri i ispunjavaju tehnički zadatak. Ono što nije dobro jeste merenje cilindričnosti. Greška je $1.281\ \text{mm}$, a ovo se koriguje nivelisanjem, jer je bitno da komad ima konstantnu kupastu krivu, tako da se ne smatra velikom greškom.

U drugom merenju dobijaju se gotovo isti rezultati za kružnost, ekscentričnost i udar, ali cilindričnost je još lošija nego prethodni put.

U trećem merenju je postignut lošiji rezultat za kružnost i udar - $1.55\ \mu\text{m}$, ali ekscentričnosti nema. Usput, korigovana je geometrija da bi cilindričnost bila izmerena na $1.25\ \mu\text{m}$. Ovo je sasvim prihvatljivo rešenje.

8. ZAKLJUČAK

U prikazanom radu predstavljen je jedan realan inženjerski problem, kao i predlog rešenja tog problema. Dat je prikaz jednog modela bruslice za spoljašnje kružno brušenje koja mora da ispuni posebno postavljene zahteve određenog kupca. Posebno je prikazan konstruktivni opis projektovanja, proizvodnje, montaže i funkcionisanja redizajniranog vreteništa tocila bruslice.

U radu se detaljno prikazuje pronalaženje rešenja vreteništa tocila bruslice za spoljašnje kružno brušenje po upitu kupca, a čijem se rešavanju pristupilo sekvencijalno kroz sledeće etape:

- analiziranje postavljenih zahteva,
- planiranje procedure rešavanja postavljenog problema,
- postavka idejnog rešenja,
- odabir tehnički pogodnog rešenja,
- realizacija probnog rešenja,
- testiranje,
- utvrđivanje grešaka i nedostataka,
- korigovanje uočenih nesavršenosti,
- implementacija i realizacija tehnički korektnog rešenja i
- verifikacija izvedenog rešenja.

U postavci idejnog rešenja, u konkretnom slučaju, iskoristilo se postojeće rešenje vretena tocila, kao i stari koncept kućišta. Odmah zatim je realizovano rešenje koje u potpunosti ispunjava postavljene zahteve. Usledilo je detaljno testiranje vreteništa tocila u pogonu firme GRINDEX. Testiranje se obavljalo sve dok se nije utvrdila izdašnost rešenja. Kasnije, utvrđeni su nedostaci koji su se u najvećoj meri odnosili na aksijalni ležaj. Korigovana je konstrukcija istog i utvrđena tehnička korektnost novog rešenja. Nova konstrukcija aksijalnog ležaja je implementirana i pokazala sasvim zavidne rezultate prilikom verifikacije ovog izvedenog rešenja.

9. LITERATURA

- [1]. Sekulić M.: *Proizvodni dizajn - skripta*, Fakultet tehničkih nauka, Novi Sad, 2014.
- [2]. Gostimirović M., Milikić D., Sekulić M.: *Osnove tehnologije obrade skidanjem materijala*, Univerzitet u Novom Sadu, FTN izdavaštvo, Novi Sad, 2015.
- [3]. Jovičić M.: *Obrada brušenjem – identifikacija karakteristika stanja i optimizacija procesa*, Mašinski fakultet, Beograd, 1990.
- [4]. Zvanični katalog za klizne ležajeve firme SPIETH.

Kratka biografija:

Uroš Rodić rođen je u Kikindi 1982. godine. Master rad na Fakultetu tehničkih nauka, smer Proizvodno mašinstvo, odbranio je 2018. godine iz oblasti Proizvodni dizajn.

Marin Gostimirović rođen je 1957. godine. Doktorirao je na Fakultetu tehničkih nauka, Univerziteta u Novom Sadu 1997. godine. Od 2011. godine je u zvanju redovnog profesora. Oblast interesovanja su Procesi obrade skidanjem materijala.

**UNAPREĐENJE PRIPREME PROIZVODNJE DELOVA ELEKTROMOTORA U
KOMPANIJI ATB SEVER****IMPROVING THE PREPARATION OF PRODUCTION OF ELECTRIC MOTOR PARTS
IN THE COMPANY ATB SEVER**

Ivana Čulafić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – U cilju pravovremene proizvodnje i isporuke proizvoda kupcima, proizvodni sistemi koji svoje poslovanje baziraju na pojedinačnoj i maloserijskoj proizvodnji značajnu pažnju moraju usmeriti na pripremu proizvodnje delova i gotovih proizvoda uz sistemsko praćenje narudžbina. Osnovni predmet i cilj istraživanja u ovom radu se odnosi na sistematizaciju i unapređenje rada funkcije pripreme proizvodnje elektromotora, pre svega tehnološke pripreme proizvodnje.

Ključne reči: Priprema proizvodnje, Tehnološki procesi, Delovi elektromotora, Praćenje narudžbina

Abstract – With the aim of timely production and delivery of products to customers, production systems, which base their business on one-off and small batch production, must pay considerable attention to the preparation of production as well as systematic monitoring of orders.

The basic subject and goal of the research in this paper focuses on the systematization and improvement of the preparation of production of electro motors with the primary emphasis on the technological preparation of production.

Keywords: Production preparation, Proces planning, Electric motor parts, Tracking orders

1.0. UVOD

ATB Sever je jedan od najstarijih proizvodnih sistema za proizvodnju rotacionih električnih mašina u jugoistočnoj Evropi. Kompanija danas proizvodi i isporučuje elektromotore razvijene prema zahtevu kupaca koji zadovoljavaju najsloženije tehničke zahteve elektromotornih pogona u nuklearnim termoelektranama, površinskim kopovima uglja i bakra (bageri, transporter, odlagači), podzemnim rudnicima, rafinerijama, platformama za eksploataciju nafte i gasa i dr. [1].

U cilju pravovremene izrade i isporuke proizvoda kupcima, ATB Sever koji svoje poslovanje zasniva na strategiji inženjeringa i proizvodnje prema narudžbini, značajnu pažnju mora da usmeri na sistemsko praćenje narudžbina i pripremu proizvodnje delova i gotovih proizvoda.

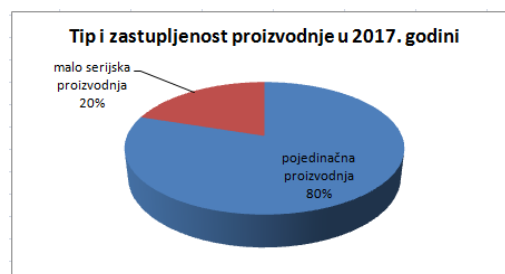
Osnovni predmet i cilj istraživanja u ovom radu se odnosi na sistematizaciju i unapređenje rada funkcije pripreme proizvodnje elektromotora, pre svega tehnološke pripreme proizvodnje odgovarajućih delova i komponenti.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dejan Lukić, vanr. prof.

**2.0. OSNOVNE KARAKTERISTIKE
PROIZVODNJE U KOMPANIJI ATB SEVER**

U posmatranoj kompaniji pretežno je zastupljena malo-serijska i pojedinačna proizvodnja (slika 1). Osnovna odlika ovih tipova proizvodnje je da se proizvodi veliki broj različitih proizvoda u malim, vrlo često jediničnim količinama, pri čemu su proizvodi prilagođeni zahtevima kupca a proizvodnja se odvija prema narudžbini.



Slika 1. Tipovi proizvodnje u kompaniji ATB Sever

Osnovne strategije proizvodnje koje su zastupljene u kompaniji su: [2]:

- ETO (Engineering to order)–Inženjering po narudžbini
- Kod ove strategije kupac definiše specifikaciju proizvoda, a proizvođač projektuje i proizvodi odgovarajući proizvod prema istoj.
- MTO (Make-to-order)–Proizvodnja prema narudžbini
- Kod ove strategije proizvođači čekaju narudžbu od strane kupca a proizvod koji poručuju je već jednom izrađen ETO proizvod ili unapređen proizvod.

Kompanija ATB sever se osim proizvodnje elektromotora u velikom procentu bavi izradom komponenti za sestrinske kompanije iz sastava ATB korporacije čije je sedište u Kini, što iznosi oko 20% od ukupne proizvodnje.

**3.0. FUNKCIJA PRIPREME PROIZVODNJE U
KOMPANIJI ATB SEVER**

Priprema proizvodnje u mašinskoj industriji se deli na tehničku i operativnu pripremu. Tehnička priprema proizvodnje obuhvata dve najvažnije tehničke funkcije proizvodnih sistema. Prva se odnosi na projektovanje proizvoda, poznata pod nazivom konstrukciona priprema, a druga na projektovanje tehnoloških procesa, poznata pod nazivom tehnološka priprema proizvodnje [3].

Osnovni zadaci koji se rešavaju u okviru tehnološke pripreme proizvodnje su:

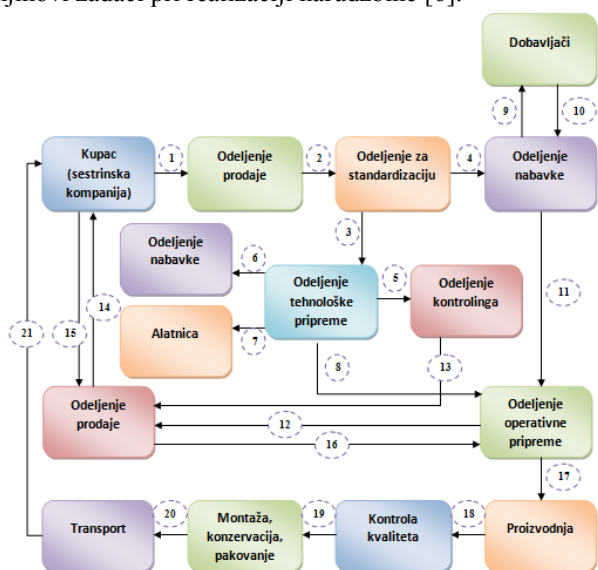
- Analiza tehnološkičnosti konstrukcije proizvoda
- Projektovanje tehnoloških procesa izrade i montaže

- Generisanje upravljačkih informacija
- Konstruisanje specijalnih alata, pribora i drugih uređaja
- Određivanje vremena i troškova proizvodnje
- Tehnoekonomska optimizacija aktivnosti tehnološke pripreme
- Simulacija tehnoloških procesa i drugih aktivnosti, itd.

Operativna priprema obuhvata druge dve funkcije koje se odnose na planiranje i upravljanje proizvodnjom. Planiranje proizvodnje ne označava samo izradu planova već obuhvata niz aktivnosti kao što su definisanje ciljeva, politike, strategije i programa proizvodnje [4].

Informacije koje su od izuzetnog značaja za planiranje proizvodnje su: stanje narudžbina i raspoloživi proizvodni kapaciteti. Osnovni zadaci prilikom planiranja su [5]: proračun potrebnog materijala, proračun kapaciteta i terminiranje proizvodnje.

U nastavku će biti prikazana realizacija pripreme proizvodnje prema narudžbini od strane sestrinskih kompanija, od trenutka poručivanja komponenti do trenutka isporuke. Kako za realizaciju jedne narudžbine nisu uključena samo odeljenja pripreme proizvodnje, biće ukratko predstavljena organizaciona struktura svih odeljenja uključenih u realizaciju narudžbine (slika 2.) i njihovi zadaci pri realizaciji narudžbine [6].



Slika 2. Povezanost odeljenja u kompaniji ATB Sever [6]

Zadaci odeljenja pri realizaciji narudžbine prema slici 2 :

- 1 - Kupac (sestrinska kompanija) kontaktira prodaju (šalje narudžbinu) kako bi poručili delove prema odgovarajućoj konstrukcionoj dokumentaciji.
- 2 - Odeljenje prodaje prosleđuje zahtev odeljenju za standardizaciju gde se vrši šifriranje delova i kreiranje sastavnice za delove.
- 3 - Nakon šifriranja zahtev se dalje prosleđuje u tehnološku pripremu gde se dokumentacija preuzima sa Citrixa, vrši se procena mogućnosti izrade s obzirom na raspoložive tehničke mogućnosti, kreiraju se tehnološki procesi, određuju se radna mesta na kojima delovi mogu da se izrade u zahtevanoj tačnosti kao i potrebna vremena.
- 4 - Odeljenje standardizacije osim prosleđivanja zahteva kupca u tehnologiju, nakon kreiranja sastavnice prosleđuje zahtev odeljenju nabavke.

5 - U odeljenju tehnološke pripreme se nakon kreiranja tehnološkog procesa (radnog plana) formira podloga za izradu kalkulacije na osnovu koje se formira cena u kontroline koja uzima u obzir vremena potrebna za izradu i neophodni materijal.

6 - Prilikom procene tehnološkičnosti izrade delova, kao i mogućnosti izrade, pravi se zahtev za nabavku alata, pribora i merila koja je neophodno kupiti da bi se deo mogao uraditi i prosleđuje se nabavci.

7 - Ukoliko su neophodni specijalni pribori i alati za izradu delova, tehnolog konstruiše alate i pribore i prosleđuje dokumentaciju alatnici na izradu.

8 - Nakon potvrde da deo može da se uradi prema dokumentaciji i kreiranja radnih planova, zahtev se prosleđuje u odeljenje operativne pripreme.

9 - Odeljenje nabavke nakon kreiranja sastavnice, kontaktira kupce kako bi se procenio najkraći rok isporuke za traženi materijal, ukoliko material-pripremak ne postoji na skladištu.

10 - Nakon potvrde od strane dobavljača, nabavka unosi cene materijala za svaku pojedinačnu poziciju u SAP kao i rok isporuke materijala.

11 - Nakon prikupljenih informacija o roku pristizanja materijala, oprećenosti kapaciteta na radnim mestima za koje je predviđena obrada dela, odeljenje operativne pripreme određuje rok isporuke delova. Ukoliko je zahtevani rok isporuke delova manji od 2 nedelje planeri uz konsultaciju sa tehnološkim odeljenjem određuju rok isporuke delova.

12 - Nakon procene roka završetka dela obaveštava se odeljenje prodaje kako bi kontaktiralo kupca radi potvrde za saglasnost sa rokom.

13 - U odeljenju kontroline se na osnovu dobijene podloge za izradu kalkulacije sa vremenima izrade, definisanim radnim mestima i sastavnicom materijala formira cena obrađenog dela koja se šalje odeljenju prodaje kako bi obavestili kupca o ceni obrade poručenog dela.

14 - Odeljenje prodaje nakon dobijanja cene i roka isporuke obaveštava kupca i proverava njegovu saglasnost sa navedenim.

15 - Ukoliko se kupac slaže sa rokom isporuke i cenom, narudžbina se potvrđuje.

16 - Odeljenje prodaje o tome obaveštava odeljenje operativne pripreme.

17 - Operativne priprema nakon potvrde narudžbine treba materijal potreban za izradu dela. Materijal se odnosi na prvo radno mesto koje je predviđeno radnim planom i zajedno sa dokumentacijom dostavlja zaposlenom.

18 - Delovi se obrađuju prema radnom planu i prema predviđenim operacijama. Zaposleni nakon svoje operacije vrše međufaznu kontrolu i proveravaju mere. Nakon kompletne obrade vrši se merenje delova u kontroline kvaliteta gde se utvrđuje da li je deo urađen prema zahtevanoj dokumentaciji.

19 - Ukoliko je deo dobio odobrenje od kontrole kvaliteta, šalje se dalje u montažu (ukoliko je sastavni deo nekog sklopa) ili na pakovanje. Pre pakovanja vrši se konzervacija delova.

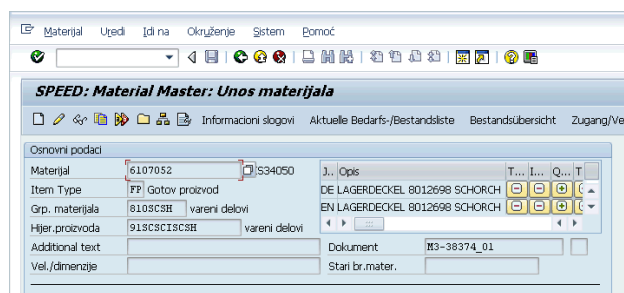
20 - Nakon pakovanja delovi se grupišu u magacinu prema kupcu za koga su poručeni i transportuju se.

21 - Deo stiže do kupca.

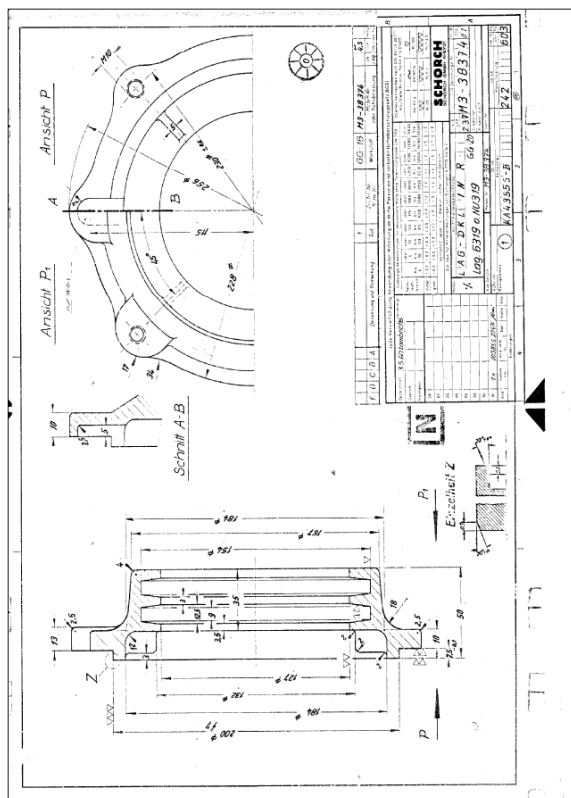
4.0. PRIMERI REŠAVANJA PROBLEMA U TEHNOLOŠKOJ PRIPREMI PROIZVODNJE

4.1. Problem ostvarivanja geometrijske tačnosti

Od strane sestrinske kompanije Schorch poručen je poklopca ležaja oznake 8012698 u seriji od 10 komada. Prvi korak tehnologa nakon dobijanja narudžbine jeste preuzimanje konstruktivne dokumentacije za poručeni deo. Kako bi ustanovili broj crteža po kome se deo obrađuje koristi se maska Z3M1 u SAP-u (slika 3.). Pomoću SAP programskog sistema vidi se da je za deo 8012698, crtež obrade M3-38374_01 (slika 4.). Dokumentacija sestrinskih kompanija se preuzima pomoću softvera Citrix iz baze podataka. Nakon preuzimanja crteža dela donosi se odluka da li je isti moguće obraditi na osnovu zahteva postavljenih dokumentacijom i raspoloživih tehničkih mogućnosti kompanije.



Slika 3. Maska Z3M1 u SAP sistemu

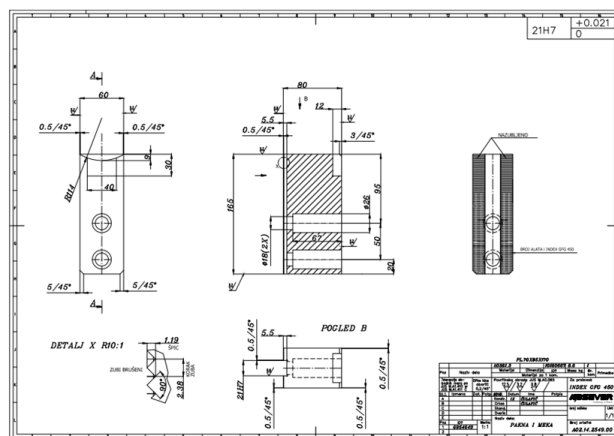


Slika 4. Crtež M3-38374_01

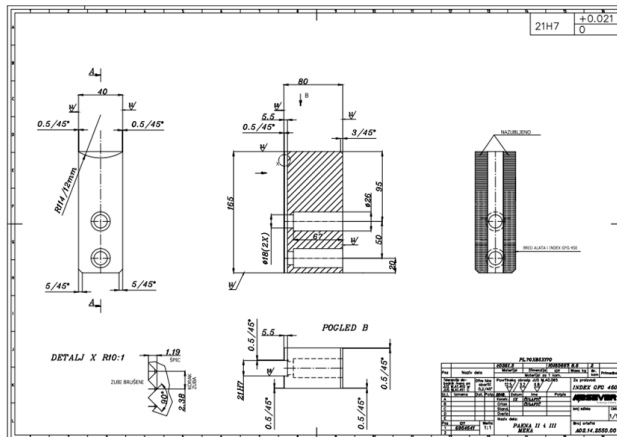
Na osnovu analize konstruktivne dokumentacije, donosi se zaključak da se deo može obraditi i narudžbina se prihvata. Nakon prihvatanja narudžbine, vrši se detaljna analiza tehnološkosti dela i na osnovu toga projektuje se tehnološki proces izrade (radni plan), projektuju se specijalni pribori i kreira NC program.

4.1.1. Prikaz prethodnog rešenja tehnološkog procesa izrade i uočeni problemi

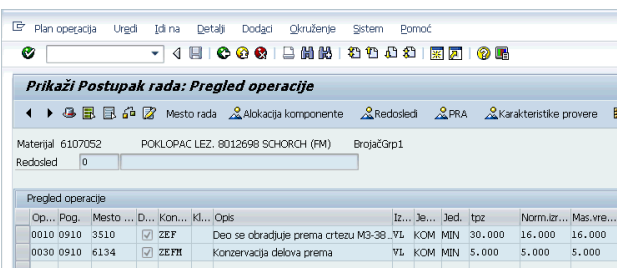
Prvo razmatrano rešenje tehnološkog procesa izrade poklopca odnosi se na obradu iz jednog stezanja, kako bi glavna i pomoćna vremena obrade bila što kraća. Prema tom zahtevu projektovane su specijalne meke pakne za stezanje dela preko spoljašnjeg prečnika Ø228mm koje automatski i pozicioniraju deo kako bi se u okviru operacije struganja moglo izvršiti bušenje i urezivanje navoja. Na slici 5. i 6. prikazane su specijalne meke pakne predviđene za stezanje dela. Na osnovu projektovanog rešenja stezanja dela vrši se konačno projektovanje tehnološkog procesa (slika 7.) i kreira se NC program.



Slika 5. Meka pakna I - A.02.14.2549.00



Slika 6. Meka pakna II i III - A.02.14.2550.00



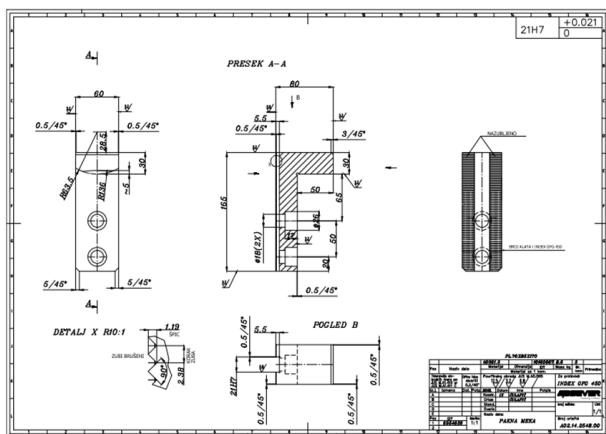
Slika 7. Radni plan poklopca ležaja 8012698

Nakon obrade dela ustanovljeno je da projektovani pribor za stezanje ne steže ravnomerno komad, jer je pakna I oslabljena zbog pozicioniranja a deo se prihvata veoma malom površinom. Konstatovana mera nakon obrade, merena u dve tačke iznosi Ø200 -0.03/-0.11, što je van tolerancije. Nakon toga, izvršena je proba na još četiri komada sa promenom NC programa, promenom pritiska,

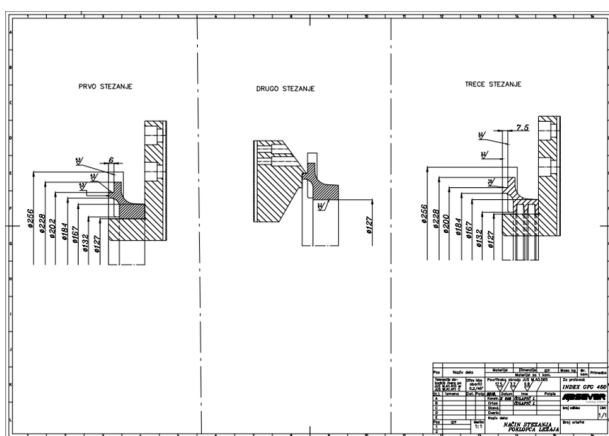
korigovanjem režima ali sa ostavljanjem dodatka na meri Ø200, odnosno deo se obrađuje na meru Ø202f7, kako bi se mogao doraditi na drugim mašinama. Nakon obrade još četiri komada dobijena je jasna slika da je projektovani pribor neadekvatan za obradu datog dela jer na osnovu merenja vidimo da imamo nestabilan proces obrade i nezadovoljavajući kvalitet na meri Ø200f7.

4.1.2. Unapređeno rešenje tehnološkog procesa izrade poklopca ležaja

Zbog neadekvatnosti prvog rešenja pristupilo se kreiranju drugog rešenja, kod koga je projektovan specijalni pribor koji će deo stezati preko veće površine koja je centričnog oblika, na osnovu čega se pretpostavilo da će doći do smanjenja ovalnosti. U ovom slučaju je vreme obrade veće u odnosu na obradu dela iz jednog stezanja, ali se teži postizanju zahtevanog kvaliteta, pa se vreme obrade postavlja u drugi plan. Na slici 8 su prikazane specijalne meke pakne za unutrašnje stezanje. Osim toga za obradu koja se razmatra u drugom slučaju koriste se i standardne tvrde pakne oznake SHUNK 14829. Nakon projektovanja specijalnih pribora, projektovan je i način stezanja dela na svim operacijama (slika 9.), nakon čega je kreiran tehnološki proces - radni plan (slika 10.) i NC program.



Slika 8. Meke pakne za unutrašnje stezanje dela



Slika 9. Način stezanja poklopca ležaja 8012698

Nakon izrade dela izvršeno je njegovo merenje i kreiran je merni izveštaj kontrole kvaliteta. Sa mernog izveštaja se može videti da je promenom tehnologije obrade postignut zahtevani kvalitet i proces obrade je doveden u stabilno stanje, jer je ponovljivost mere konstantna sa manjim odstupanjima (Ø200 -0.06/-0.065).

Op...	Pog.	Mesto...	D...	Kon...	Kl...	Opis	Iz...	Ja...	Jed.	tpz	Norm.iz...	Max.vre...	Kreirano
0010	0910	3510	✓	ZE	F	Deo se obrađuje prema crtežu M3-38_VL	KOM	MIN	30.000	3.000	3.000	3.000	01.06.2016
0011	0910	3510	✓	ZE	F	Stezanje sa tvrdim paknama oznake 1_VL	KOM	MIN	5.000	9.000	9.000	9.000	17.10.2018
0012	0910	3510	✓	ZE	F	Stezanje sa mekim paknama oznake A_VL	KOM	MIN	5.000	6.000	6.000	6.000	17.10.2018
0030	0910	6134	✓	ZE	F	Konzervacija delova prema	VL	KOM	MIN	5.000	5.000	5.000	01.06.2016

Slika 10. Izmenjeni radni plan poklopca ležaja 8012698

Iako je navedeno rešenje obrade poklopca ležaja 8012698 zadovoljavajuće sa tehnološkog aspekta, i ono bi se moglo unaprediti. Ukoliko bi serija bila veća, ekonomski bi bilo isplativo promeniti model odlivka dodavanjem tehnološke baze na meru Ø167mm. Na taj način ne bi ugrozili funkcionalne karakteristike dela, jer spoljašnje kote Ø167mm i Ø184mm nemaju bitnu funkciju. Dodavanjem tehnološke baze tehnološki proces obrade struganja izvodio bi se iz dve operacije, što bi dovelo do smanjenja vremena obrade.

5.0. ZAKLJUČAK

Na osnovu realizovanog istraživanja pokazano je da u primeni strategija proizvodnje koje su zastupljene u kompaniji ETO i MTO, veoma veliki značaj ima sistemski prilaz u praćenju narudžbina i pripremi proizvodnje delova i komponenti elektromotora.

Pored toga, pokazano je koja su to znanja bitna za rad inženjera u tehnološkoj pripremi proizvodnje kompanije prema navedenim strategijama, među kojima su najbitnija: sposobnost analize tehnološkičnosti proizvoda, znanja o zavisnosti tačnosti i kvaliteta površine proizvoda, znanja o različitim procesima proizvodnje, znanja o materijalima, mašinama, alatima, priborima i drugim resursima, znanja za određivanje operacija, zahvata, dodatka za obradu, parametara obrade, znanja za određivanje vremena i troškova proizvodnje, znanja u oblasti primene informacionih tehnologija.

6.0. LITERATURA

- [1] <http://sever.rs/wp02/>
- [2] Lukić, D., Milošević, M., Todić, V.: *Integrirani CAPP sistemi i tehnološka baza podataka, skripta*, Fakultet tehničkih nauka, Novi Sad, 2013.
- [3] Todić, V.: *Projektovanje tehnoloških procesa*, Fakultet tehničkih nauka, Novi Sad, 2004
- [4] Mikac, T., Ljubetić, J.: *Organizacija i upravljanje proizvodnjom*, Tiskano- Hrvatska, 2009.
- [5] Zelenović, D.: *Projektovanje proizvodnih sistema*, Fakultet tehničkih nauka, Novi Sad, 2012.
- [6] Čulafić, I.: *Unapređenje pripreme proizvodnje delova elektromotora u kompaniji ATB Sever*, Master rad, Fakultet tehničkih nauka, Novi Sad, 2018.

Kratka biografija:



Ivana Čulafić rođena je u Vrbasu 1992. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mašinstva – Proizvodno mašinstvo, smer Računarom podržane tehnologije odbranila je 2018. god.

ISPITIVANJE ALTERNATIVNOG TURBINSKOG MERILA PROTOKA GORIVA PRI TOČENJU GORIVA NA BENZINSKIM PUMPAMA**FUEL TURBINE FLOW METER TESTING**Branka Sremački, *Fakultet tehničkih nauka, Novi Sad***Oblast – MAŠINSTVO**

Kratak sadržaj – U ovom radu data su teorijska i laboratorijska rešenja kalibracije turbinskog merila protoka goriva. Objasnjeni su i uticajni faktori procesa kalibracije. Izvršeno je ispitivanje sistema za merenje protoka. Objasnjen je postupak merenja. Date su karakteristike korišćene opreme i prikazani su rezultati ispitivanja.

Ključne reči: Turbinska merila, Kalibracija

Abstract – In this paper, theoretical and laboratory solutions of the calibration of the turbinefuel flow meter are given. The influence factors of the calibration process are also explained. The flow measurement systems was tasted. The measurement procedure is explained. Characteristics of used equipment are given and the test results are shown.

Key words: turbinefuel flow meter, calibration process

1. UVOD

Zapreminski, ili maseni, protok fluida predstavlja najzahvatljiviju karakteristiku fluida u odnosu na mogućnost merenja. Turbinska merila se široko primenjuju zbog sposobnosti da odgovore specijalnim zahtevima u svakoj konkretnoj primeni. Popularnosti ovih merila najviše doprinose njihove metrološke karakteristike koje ih čine superiornim u odnosu na druga merila. Turbinska merila protoka široko se koriste za merenje protoka tečnosti. Ponovljivost je najvažnije svojstvo merila. Merila se moraju kalibrisati i koriste se za različite fluide. Kalibracija je poređenje standarda i očitanih rezultata koje daje uređaj. Pogodniji oblik kalibracionih podataka zove se univerzalna kriva viskoznosti.

2. ZADATAK RADA

Zadatak rada je ispitivanje pištolja za točenje goriva. Potrebno je odrediti tačnosti merenja pri njegovom korišćenju. Tome prethodi teorijski opis kalibracije turbinskih merila i predstavljanje uticajnih faktora. Takođe treba izvršiti ispitivanje uticaja položaja pištolja za točenje goriva u samom uvodniku na tačnost merenja i utvrditi optimalnu visinu uvodnika i ugao konusnih umetaka ispred turbinskog kola za pištolj za točenje goriva.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Maša Bukurov, red. prof.

3. OPŠTE O MERENJU PROTOKA FLUIDA I MERILA

Merenje protoka fluida se ostvaruje direktnom metodom pomoću različitih tipova merila kod kojih se osnovni indikator postavlja neposredno u fluidnu struju. Da bi se moglo vršiti mernje protoka, nezavisno od tipa merila, neophodno je poznavati zavisnost između protoka Q i izlaznog signala merila x , koji je obično dat kao: frekvencija električnih impulsa, električni napon, jačina električne struje, otklon kazaljke na skali, itd. Matematički izraz [1]

$$Q = f(x) \quad (1)$$

naziva se funkcija transformacije merila, a određuje se kalibracijom merila.

U opštem slučaju, za svako merilo su važne sledeće osobine: ponovljivost, pouzdanost i tačnost i to po napisanom redosledu prioriteta. Sama tačnost je od sekundarnog značaja. Temperatura utiče na fizička svojstva radnog fluida i naročito na njegovu viskoznost. Slično kao i temperatura radnog fluida, visok strujni pritisak utiče na promenu dimenzija kućišta merila, pa prema tome i merne komore kada ona sa kućištem predstavlja jedinstvenu celinu.

Turbinsko merilo zahteva linearnu vezu između brzine merila i protoka, u širokom spektru protoka i viskoznosti merenih tečnosti. U dozirnim tečnostima visokog viskoziteta uslovi protoka kroz turbinsko merilo, mogu biti turbulentni, prelazni ili čak i laminarni. Shodno tome, uticaj viskoznosti na preciznost merila postaje veoma važan [2]. Nijedno drugo gledište u problematici turbinskih merila nije izazvalo veće interesovanje i nesuglasice od njihovog ponašanja u viskoznim tečnostima.

4. KALIBRACIJA TURBINSKIH MERILA

Kalibracija nije apsolutna operacija. To je poređenje između mernog instrumenta, uređaja koji se testira ili merila protoka, i standarda. Kroz ovo poređenje, uspostavlja se odnos između količine izmerne uređajem u okviru testa, i merenja iste količine izvedene iz standarda. Na taj način dobijaju se očekivanja kako će uređaj obavljati svoju funkciju, tj. način na koji će se upotrebljavati. Postupak kalibracije izvodi se po određenoj proceduri i uz korišćenje izabranih metoda [3].

Merenje protoka tečnosti zavisi od uspostavljanja opsega merenog protoka i niza uticajnih faktora. Količina tečnosti može se izraziti preko zapremine ili mase. Merenje može biti izraženo količinski ili kao količina po jedinici vremena. Količina merena standardom može se razlikovati od količine koja je prošla kroz uređaj testiranja usled promena u obimu ili čak mase između merila i standarda. Merenje protoka fluida je dinamičan proces i svi merni uređaji su pod uticajem načina i uslova korišćenja. Protočni uređaji su pogođeni od strane temperature, viskoznosti, profila protoka. Oni takođe zavise i od strane spoljašnje sredine, vibracije, stresa i temperature itd. Različiti uređaji su pogođeni na različite načine. Slični standardni će biti podložni istim uticajima.

Da bi se dobilo proverljivo merenje, očekuje se, da bi pri ponovljenom merenju trebalo da se dobije isti rezultat. U praksi, merenje se ponavlja samo u određenom opsegu tokom kratkog vremena i (verovatno) širem opsegu tokom dugog vremenskog perioda ili u različitim okolnostima. Obično se očekuje da kalibracija treba da da neke indikacije o ponovljivosti instrumenta.

Iako može izgledati očigledno, rezolucija uređaja mora biti adekvatna da omogućiti kalibraciji odgovarajuću potrebnu nesigurnost. Da bi se to postiglo, standard mora biti u stanju da izmeri dovoljno tečnosti da odgovara rezoluciji uređaja.

Sva merila protoka intereaguju na neki način sa tečnim fluidom. Na prirodu ove interakcije utiču svojstva tečnosti ili brzine fluida koja prolazi kroz uređaj. Promene u ovoj interakciji menjaju sposobnost uređaja da daje tačnu predstavu o količini. Veličina greške je različita za različite tipove merila i tečnosti. Iz tog razloga je poželjno da se kalibracija vrši koristeći iste tečnosti i cevovode u okviru kojeg će merila normalno funkcionisati.

Greška merenja je razlika između izmerene vrednosti i prave vrednosti izmerene veličine. Merna nesigurnost je parametar povezan sa rezultatom merenja koji karakteriše disperziju vrednosti koje bi se mogle dodeliti merenoj veličini [4]. Da bi se prikazali rezultati kalibracije, potrebno je da priroda podataka koji proizilaze iz merila budu razumljiva. Protočna merila mogu pokazivati protok ili količinu u izvesnom broju na različite načine. Dobijeni podaci o merenju mogu biti prikazani mehaničkim ispisom ili na elektronskom displeju na kome se navodi količina ili protok, a moguć je i elektronski izlaz na bazi impulsa, frekvencije ili struja (mA).

Izlaz ili ekran se zasniva na merenjima (npr. frekvencija, protok, diferencijalni pritisak ili mA). Merenja obično, ponekad, variraju tokom merenja. Normalno je da se uzme prosek, srednja vrednost čitanja, svih dobijenih parametara tokom kalibracije.

5. UTICAJNI PARAMETRI

Dinamička viskoznost je karakteristika tečnosti, koja predstavlja otpor strujanju. Što je veća numerička vrednost dinamičke viskoznosti tečnosti, to je veći otpor tečenju.

Viskoznost tečnosti je jako zavisna od temperature. Povećanje temperature će izazvati smanjenje viskoznosti. Iz tog razloga, temperatura može značajno da utiče na performanse merila protoka.

Dinamička viskoznost njutnovskih fluida podeljena sa gustinom fluida naziva se kinematska viskoznost [5].

$$\nu = \frac{\eta}{\rho} \left[\frac{\text{m}^2}{\text{s}} \right] \quad (2)$$

Viskoznost se najčešće meri viskozimetrom sa rotirajućim cilindrom ili kapilarnim viskozimetrom. Viskozimetrom sa rotirajućim cilindrom mere se tangencijalne sile koje deluju na fluid i na taj način se određuje dinamička viskoznost. Dinamička viskoznost obično se određuje u centipoasima [cP] tj. [mPa·s]. Da bi se dobila kinematska viskoznost mora se prethodno odrediti gustina fluida. Kinematska viskoznost se izražava u centistoksima [cSt] tj. [mm²/s].

Dinamička viskoznost fluida veoma je zavisna od temperature. Sa porastom temperature, viskoznost tečnosti opada, a viskoznosti gasa se povećava. Uticaj pritiska na dinamičku viskoznosti obično se zanemaruje, što je prihvatljivo na niskim pritiscima. Međutim, za pritiske preko 7 MPa, dinamička viskoznost tečnosti može značajno da zavisi od pritiska. Generalno, povećanje pritiska će povećati viskoznost tečnosti [6].

Kinematička viskoznost je odnos dinamičke viskoznosti i gustine. Dakle, ako se gustina menja sa temperaturom ili pritiskom, i kinematska viskoznost će se promeniti. Za gasove, kinematska viskoznost snažno zavisi od pritiska. Kinematska viskoznost je ključni parametar tečnosti koji utiče na performanse turbinskog merila [7].

Strujanje fluida se karakteriše kao laminarni ili turbulentni strujni režim. Kod laminarnog strujanja radi se o kretanju tečnosti u slojevima, glatko klizanje jednog sloja preko drugog. Nema mešanja tečnosti iz sloja u sloj. Dobijeni profil brzina je paraboličnog oblika.

U turbulentnoj struji ne postoje odvojeni slojevi. Količina kretanja fluida prevladava viskozne sile i javlja se intenzivno i kontinualno mešanje fluida duž strujanja.

Kada viskozna tečnost struji preko čvrste površine, sila koja deluje na površinu je u tangencijalnom pravcu. U stvari, tečnost pokušava da povuče čvrstu površinu za sobom. Veličina ove sile zavisi od viskoznosti i brzine fluida.

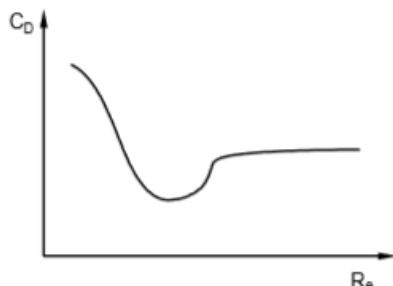
Eksperimentalno izmerena sila otpora trenja F obično se prikazuje kao C_D u funkciji od Re , gde je C_D koeficijent otpora definisan kao:

$$C_D = \frac{F}{\frac{1}{2} \rho v^2 A} \quad (3)$$

gde su:

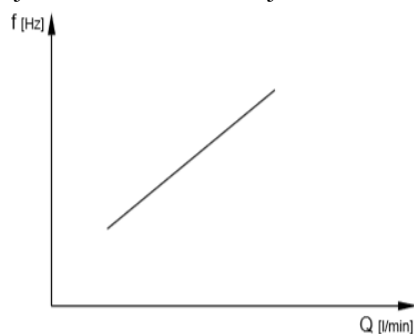
v - brzina fluida
 A - okvašena površina.

Dijagram zavisnosti C_D od Re izgleda kao na slici 1. koeficijent otpora brzo se smanjuje sa povećanjem Reynoldsovog broja pri laminarnom strujanju, raste naglo u prelaznoj oblasti, zatim se ustaljuje i na kraju polako smanjuje u turbulentnoj oblasti.



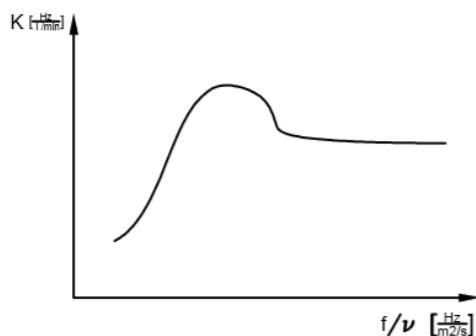
Slika 1. Zavisnost C_D od Re

Kalibracija turbinskog merila protoka sastoji se od registrovanja izlazne frekvencije merila pri određenim veličinama protoka određenog od strane referentnog merila protoka. Rezultat je linearna kriva kao što je prikazano na slici 2. Međutim, ova kriva važi samo za fluide sa približno jednakom kinematskom viskoznošću tečnosti koja se koristi za kalibraciju.



Slika 2. Kalibraciona kriva

Pogodniji oblik kalibracionih podataka zove se univerzalna kriva viskoznosti. Ovo je polu-logaritamaska kriva osetljivosti merila, u funkciji odnosa izlazne frekvencije i kinematske viskoznosti, slika 3. Osetljivost merila poznatija je kao faktor K , i predstavlja broj impulsa merila koje proizvodi svaki litar tečnosti koja protiče kroz njega. Ona je izvedena iz podataka kalibracije na sledeći način: $K=f \cdot 60/Q$.



Slika 3. Osetljivost merila K u funkciji odnosa f/ν

6. METODOLOGIJA ISPITIVANJA

Turbinsko merilo protoka je predmet ispitivanja. Ispitivanje je obavljeno u Laboratoriji za mehaniku fluida Fakulteta tehničkih nauka. Rađena su dva nezavisna niza merenja. Prvi niz merenja radilo se za pet različitih protoka. Za svaki protok izvršeno je pet merenja, pri istim uslovima. Prvi niz merenja služi za određivanje kalibracione krive i matematičkog modela. Drugi niz merenja koristi se za proveru dobijenog matematičkog modela. Pri tome je vršeno merenje za četiri različita protoka koji se nalazi između vrednosti protoka u prvom nizu merenja. Za svaki protok izvršeno je pet merenja pri istim uslovima. Drugim nizom merenja ispitana je tačnost uređaja. Rezultati dobijeni korišćenjem matematičkog modela upoređene su sa vrednostima protoka dobijenim masenim, odnosno zapreminskom metodom (merenje pomoću vage). Postrojenje za ispitivanje prikazano je na slici 4.

Vazduh se iz kompresorske stanice (slika 4) dovodi preko dovodnog voda (1) i finog regulatora pritiska (2) u glavni rezervoar (3) pri čemu je ventil (10) otvoren. Na dovodnom vodu se nalazi grub regulator pritiska i primpremna grupa. Da ne bi toliko oscilovao pritisak u sistemu postavlja se taj fini regulator pritiska. Fini regulator ima mnogo bolju regulacionu grupu i zahvaljujući njemu se održava znatno ujednačeniji pritisak unutar glavnog rezervoara. Kada se postigne pritisak koji se očitava na manometru (15), koji odgovara željenom protoku, kreće sa sa merenjima. To je jedan kontrolni manometar koji ima dobru rezoluciju očitavanja.

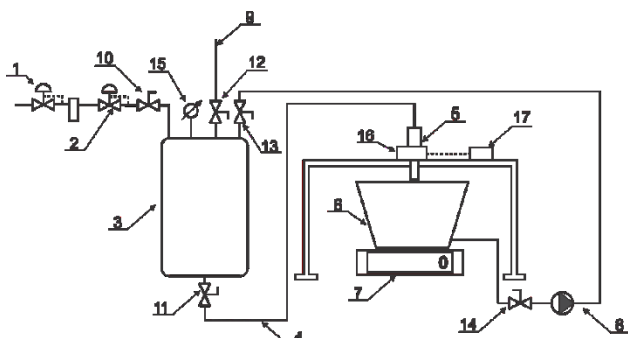
Fluid se pri otvorenom ventilu (11) potiskuje kroz potisni vod i pištolj za točenje goriva (5) na kraju voda (4). Pomoću tog pištolja se startuje i prekida merenje i protok. U cilju simulacije realnih uslova korišćene su izmenljive čaure za usmeravanje goriva koje se montiraju na ulazu u ispitivano turbinsko merilo protoka (16). Fluid nakon merila protoka (16) ulazi u prihvatni rezervoar (6) koji se nalazi na vagi (7). Vagom (7) se određuje masa fluida koji istekne za određen vremenski period. Vaga je apsolutno merilo i potrebno je preračunati maseni protok u zapreminski. Posle svakog procesa merenja, kada se gorivo vrati nazad, potrebno je tarirati vagu na 0. Pomoću akvizicije (17) dobija se promena impulsa u sekundi ispitivanog merila (16). Akvizicija je povezana i sa kompjuterom, da se ne bi pravila velika greška u merenju. Zbog toga je tu postavljen jedan mikro-prekidač, koji, kada se pištolj startuje, on pošalje signal i startuje vreme. Kada se pištolj otpusti on zaustavlja ovaj prekidač i zaustavi se i vreme.

Nakon završetka merenja, zatvara se ventil (10), a zatim se preko odušnog voda (9) i ventila (12) pritisak u rezervoaru izjednačava sa atmosferskim. Odušni vod nam je veza sa atmosferom. Fluid se iz prihvatnog rezervoara (6) pretače preko povratnog voda sa pumpom (8), pri čemu su ventili (13) i (14) otvoreni, u glavni rezervoar (3).

7. KORIŠĆENA OPREMA

Tokom ispitivanja korišćena je sledeća oprema:

- Pripremna grupa MACP401-15A-C-T15-BSP,
- precizni regulator pritiska MAIR100-6A-8K-C-G,
- vaga Shollex tip SHYH,
- pištolj za točenje goriva,
- centrifugalna pumpa „Pedrollo“ model SPm 132A,
- glavni rezervoar,
- pomoćni rezervoar,
- uvodnik,
- umeci,
- ispitivano turbinsko merilo.

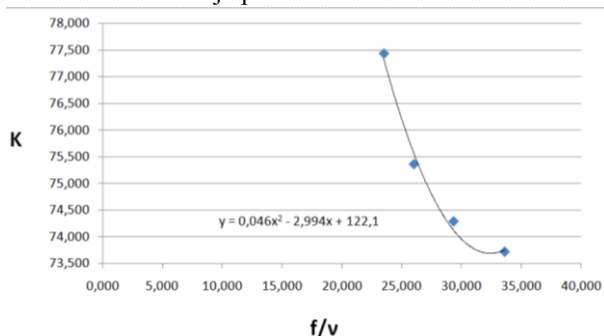


Slika 4. Šematski prikaz postrojenja za ispitivanje

8. REZULTATI ISPITIVANJA

Turbinsko merilo kalibrisano je uz pomoć pištolja za gorivo, u skladu sa uslovima koji su najpribližniji eksplatacionim.

Kalibracioni model je prikazan na slici 5.



Slika 5. Kalibraciona kriva turbinskog merila

Rezultati provere su prikazani u tabeli 1, gde se vidi da je tačnost merenja do $\pm 1,6\%$.

Tabela 1. Rezultati ispitivanja tačnosti turbinskog merila

Imp/s	m [kg]	Q [L/min]	Q_m [L/min]	ε [%]
86,04	56,9	68,55	67,46	1,6
104,40	69,38	83,59	84,68	1,3
115,04	77,92	93,89	93,68	0,22

9. ZAKLJUČAK

Nakon kalibracije i provere pištolja utvrđeno je da je tačnost merenja u granicama $\pm 2\%$.

Utvrđeno je da je neophodno da pištolj za točenje goriva bude što saosniji sa uvodnikom u merilo protoka. Narušavanje saosnosti značajno utiče na grešku merenja. Ako se ispuni uslov saosnosti, dubina uvođenja pištolja u uvodnik nije od značajnog uticaja na tačnost merenja.

Određeno je da visina uvodnika bude 80 mm kako bi se sprečio kontakt pištolja za točenje goriva i radnog kola turbinskog merila. Izmenljivi konusni umeci nisu imali veći uticaj na tačnost merenja kod pištolja za točenje goriva. Bez obzira na tip umetaka tačnost merenja kretala se u granicama od $\pm 2\%$.

10. LITERATURA

- [1] Cvijanović S. Petar, Turbinska merila zapremine tečnosti, monografija.
- [2] W. F. Z. LEE Chief Research Analyst, Central Research and Development Laboratory, Rockwell Manufacturing Company, Pittsburgh, Pa, A Study of Viscosity Effect and Its Compensation on Turbine-Type Flowmeters.
- [3] www.am.unze.ba/mt/2014/Begic.ppt
- [4] <http://www.super-lab.com/fajlovi-upload/brosura-metrolab-2018.pdf>
- [5] Bukurov, M., Mehanika fluida, knjiga prva, FTN izdavaštvo, Novi Sad, 2013.
- [6] Rubin, M., Miller, R.W., and Fox, W.G. Driving Torques in a Theoretical Model of a Turbine Meter, Transactions of the ASME. Journal of Basic Engineering, Paper Number 64 - WA/FM-2, 1965.
- [7] <https://ftimeters.com/wp-content/uploads/2017/.../Universal-Calibrations-FT-Series.pdf> viskoznost.

Kratka biografija:



Branka Sremački rođena je u Subotici 1991. god. Diplomski master rad na Fakultetu tehničkih nauka iz oblasti Mašinstva – Gasna i naftna tehnika odbranila je 2016.god.

SISTEM ZA PRAĆENJE I PLANIRANJE ALATA U TEHNOLOŠKOM PROCESU MONTAŽE AUTOMOBILSKIH KOMPONENTI

SYSTEM FOR TOOL MONITORING AND PLANNING IN ASSEMBLY PROCESS OF AUTOMOTIVE COMPONENTS

Nemanja Dimitrijević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Osnovni cilj ovog rada je predstavljanje sistema za praćenje i planiranje alata u tehnološkom procesu montaže automobilske komponente. Shodno tome, pažnja je usmerena na organizaciju servisne tačke i njegov značaj u tehnološkom procesu u cilju skraćivanja vremena zastoja zbog alata i povećanju ukupne efikasnosti opreme.

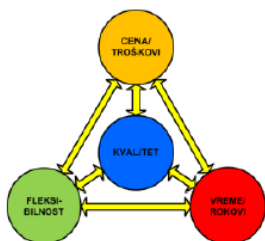
Ključne reči: Praćenje i planiranje održavanja alata, redukcija vremena, preventivno i korektivno održavanje

Abstract – The main objective of this work is to present system for monitoring and planning tools in the technological process of assembly of automotive components. Consequently, attention is focused on the organization of the service point and its significance in the technological process in order to reduce downtime due to tools and increase overall efficiency of the equipment.

Key words: Tool monitoring and planning maintenance, time reduction, preventive and corrective maintenance

1. UVOD

Vreme je danas veoma dragoceno i veoma utiče na proizvodnju i cenu proizvoda. Prisutna je česta promena zahteva tržišta, što znači da industrija treba da se u što kraćem roku prilagodi zahtevima. Potražnja za inovativnim proizvodima koje poseduju neke specifične karakteristike konstantno raste. To znači da kompanija mora da ispunji zahteve kupca za specijalnim karakteristikama proizvoda i traženi kvalitet, da se proizvod izradi u što kraćem roku, a da pri tome cena proizvoda bude minimalna (slika 1). Takav proizvod predstavlja veliki izazov za kompaniju.



Slika 1. Zavisnost uticajnih faktora u proizvodnji

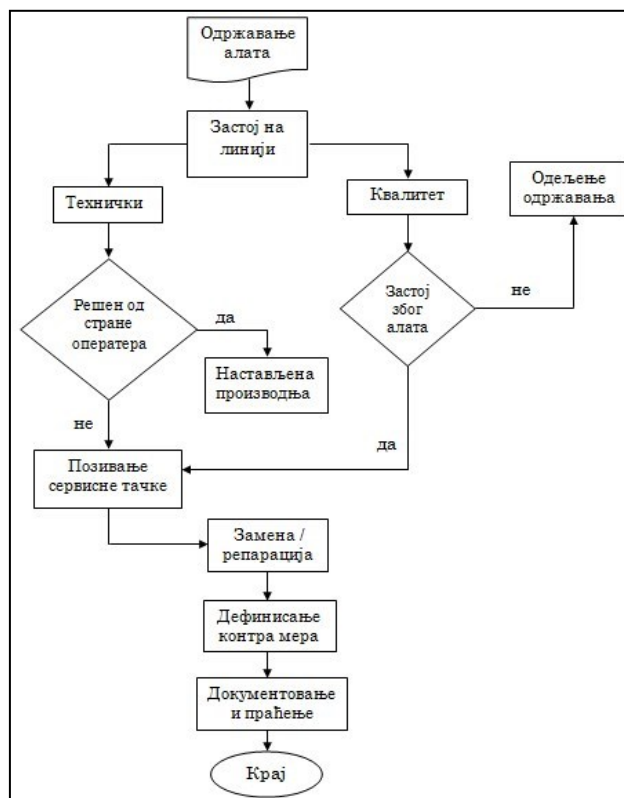
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Mijodrag Milošević, vanr. prof.

Obezbeđivanje traženog kvaliteta proizvoda zahteva ispravne, pouzdane i dugotrajne alate. Samim tim, ispunjenjem tih zahteva proizvodne linije će raditi bez tehničkih zastoja i imaće veliku ukupnu efikasnost opreme. U vezi sa tim, u nastavku će se predstaviti servisna tačka, tok informacija i njen odnos prema proizvodnim linijama. Takođe, prikazaće se i analiza jednog problema i njegovo rešenje [3].

2. PRIKAZ STRUKTURE SERVISNE TAČKE

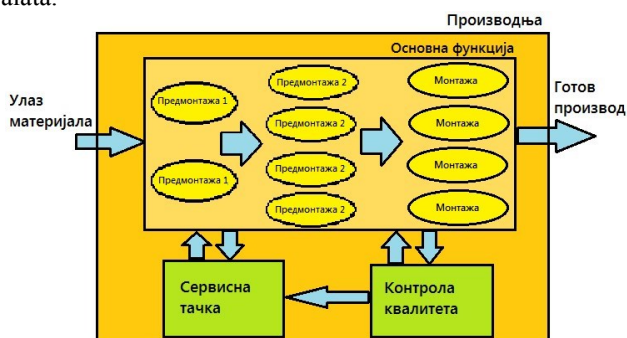
Da bi se moglo izvršiti održavanje tehničkog sistema neophodno je prethodno realizovati sve aktivnosti na nivou razvoja, razrade i istraživanja, kao i analizirati i informacije dobijene preko povratne veze, koje idu preko svih ostalih aktivnosti [2]. Algoritmom datim na slici 2 definisan je tok procesa kada se aktivira servisna tačka u odeljenju proizvodnje. Sve do tada, servisna tačka radi po tačno definisanim zadacima i sopstvenim algoritmom koji će kasnije biti prikazan u radu.



Slika 2. Algoritm toka procesa

Алгоритам се састоји из два дела. Део техничких застоја и застоја због квалитета. Уколико је застој због квалитета везан за алате, оделjenje позива техничаре сервисне тачке, врши се замена или репарација дела, дефинишу се контра мере, документује се и прати статус алата. До оваквих застоја долази најчешће због похабаности компоненти алата који утиче на неправилно постављање и позиционирање делова у производ.

Уколико је застој због кvara алата први корак је да оператор на линији покуша да реши квар (нпр. поновно дотежање компоненти алата). Уколико се проблем реши, производња се наставља под условом контролисања следећег дела од стране вође линије. Уколико оператор не реши проблем позива се техничар из сервисне тачке који даље наставља рад по алгоритму. Врши се репарација или замена, дефинишу се контра мере, документује се и прати статус алата.

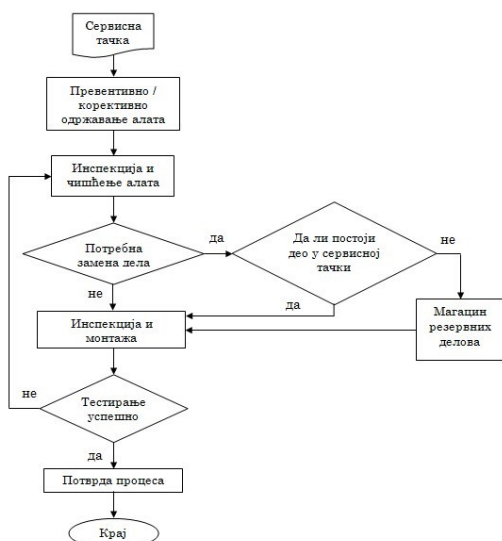


Слика 3. Дијаграм тока информација у производњи

На слици 3. видимо да у оделjenju производње постоје производне линије, који чине основну и најважнију функцију фабрике, сервисну тачку и оделjenje квалитета.

Сервисна тачка је, такође, у повратној веzi са производним линијама. Застоји због алата на линијама се документују, анализирају и доступни су сервисној тачки. Дефинисањем контра мера, модификацијама алата и пружањем потребних обука за коришћење алата, сервисна тачка пружа производним линијама. Такође, извор информација о проблемима је и оделjenje квалитета уколико производ није по спецификацијама, а утврђено је да је проблем због алата.

На слици 4. је приказан алгоритам функционисања сервисне тачке.



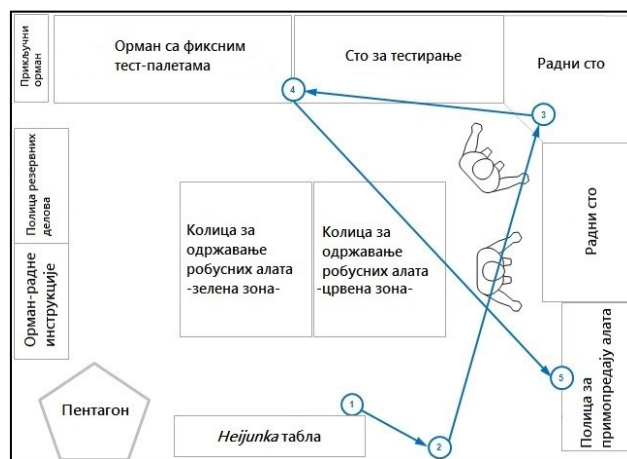
Слика 4. Алгоритам функционисања сервисне тачке

У алгоритму видимо да одржавање алата креће од расклапања, чишћења и инспекције. Уколико је потребно заменити део, а нема га у сервисној тачки, потребно је отићи до магацина резервних делова, узети део, вратити се назад и наставити репарацију. Након монтаже и поновне инспекције, алат се ставља на тест палету и тестира се према предвиђеној процедури за тај алат. Сва додатна podešavanja на алату се врше у овом кораку. Међутим, ако алат не прође тест палету и тестирање буде неуспешно, поново се алат мора расклопити и извршити инспекција. Уколико алат прође корак тестирања, потврђује се процес затварањем картице, односно налога за сервисирање алата, и алат се оставља на полицу са осталим завршеним alatima.

Организација тока алата зависи од врсте одржавања:

- Preventivno održavanje
- Korektivno održavanje

Планирано или preventivno одржавање обухвата низ активности у циљу sprečavanja otkaza i održavanja sistema u radnom stanju [1]. Другим rečima, unapred otklanja uzroke potencijalnih otkaza sistema.



Слика 5. Ток алата приликом preventivnog održavanja

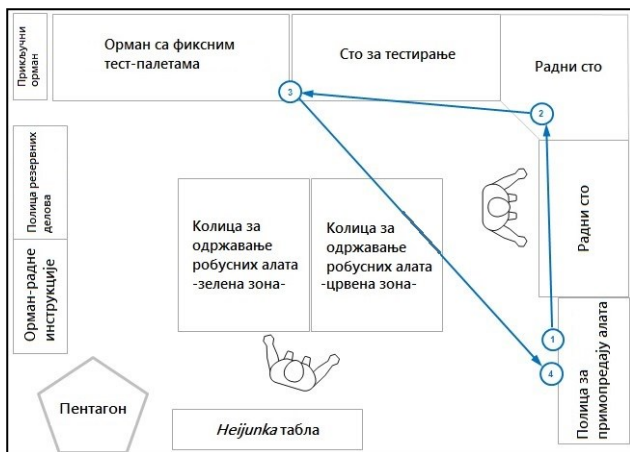
Кораци су следећи (слика 5):

1. Pregled plana održavanja i uzimanje kartica prema planu,
2. Odlazak po alat u vraćanje u servisnu tačku,
3. Rasklapanje, servisiranje i sklapanje alata,
4. Testiranje i podešavanje alata,
5. Ostavljanje alata na polici za primopredaju, popunjavanje kartice i ostavljanje kartice nazad na tablu.

Корективно одржавање представља skup radnji i postupaka kojima се технички систем из stanja otkaza враћа у радно stanje [1]. Другим rečima, технички систем се поправља да би се довео у радно stanje. Овај тип одржавања је још увек најчешћи начин одржавања техничких sistema.

Кораци приликом korektivnog održavanja су следећи (слика 6):

1. Uzimanje alata i radnog naloga iz crvene zone sa police za primopredaju alata,
2. Rasklapanje, servisiranje i sklapanje alata,
3. Testiranje i podešavanje alata,
4. Ostavljanje alata u zelenoj zoni na polici za primopredaju alata, popunjavanje radnog naloga i odlaganje naloga u registrator sa ostalim nalogima za servisiranje alata.



Slika 6. Tok alata prilikom korektivnog održavanja

3. SISTEM ZA PRAĆENJE ODRŽAVANJA ALATA

Praćenje alata se izvodi na dva načina. Prvi način je praćenje alata na vremenskoj bazi, a drugi način je praćenje alata na bazi proizvedenih delova.

Sistem praćenja alata na vremenskoj bazi se sastoji od razlike sadašnjeg datuma i datuma poslednjeg održavanja alata (slika 7). Razlika pokazuje broj dana od poslednjeg održavanja alata i ukoliko taj broj pređe definisanu vrednost, status alata prelazi u planiranje alata za održavanje [3].

Common		Date	9/2/2018	OK	Upozorenje
Tool name	Plate No.	Register button	Tool Status	Number of maintenances	Date of last maintenance
3C10-1	A05000011			7	8/29/2018
3C10-2	A05000012			7	8/29/2018
3C10-3	A05000013			7	8/29/2018
3C10-4	A05000014			6	7/10/2018
3C11-1	A05000015			7	6/29/2018
3C11-2	A05000016			7	8/29/2018
3C11-3	A05000017			8	8/29/2018
3C14-1	A05000018			6	8/30/2018
3C14-2	A05000019			6	8/30/2018
3C15-1	A05000020			6	8/29/2018
3C15-2	A05000021			6	8/29/2018
C16-1	A05000022			4	8/29/2018

Slika 7. Praćenje alata na vremenskoj bazi

U sistem praćenja alata na vremenskoj bazi ulaze alati koji su jednostavni, sastoje se samo od nekoliko komponentata, imaju povećanu postojanost na liniji i obezbeđuju traženi kvalitet tokom velike većine svog životnog veka. Takvi alati se ne repariraju nego zamenjuju sa rezervnim. Takođe, alati koji spadaju u ovaj sistem su slični alati koji se razlikuju samo u nekim malim sitnicama.

Postoje tri statusa alata u proizvodnji (slika 8): zeleni (alat nije dostigao vrednost za planiranje održavanja), žuti (alat je dostigao vrednost za planiranje održavanja i ulazi u sistem za planiranje) i crveni (alat je dostigao kritičnu vrednost i za njega se planira dodatno održavanje).

Common		Date	9/2/2018	OK	Upozorenje
Tool name	Plate No.	Register button	Tool Status	Number of maintenances	Date of last maintenance
3C10-1	A05000011			7	8/29/2018
3C10-2	A05000012			7	8/29/2018
3C10-3	A05000013			7	8/29/2018
3C10-4	A05000014			6	7/10/2018
3C11-1	A05000015			7	6/29/2018
3C11-2	A05000016			7	8/29/2018
3C11-3	A05000017			8	8/29/2018
3C14-1	A05000018			6	8/30/2018
3C14-2	A05000019			6	8/30/2018
3C15-1	A05000020			6	8/29/2018
3C15-2	A05000021			6	8/29/2018
C16-1	A05000022			4	8/29/2018

Slika 8. Status alata u proizvodnji

Potvrda procesa se obavlja preko forme za registrovanje održavanja alata (slika 9).

Slika 9. Forma za registrovanje održavanja alata

Forma sadrži sledeće podatke:

- Jedinstvena oznaka alata,
- Linija,
- Naziv alata,
- Rezultat inspekcije alata koji sadrži pitanja vezana za alat,
- Rezultat održavanja alata ukoliko je nešto zamenjeno,
- Polja za dodatne komentare,
- Spisak tehničara,
- Taster za poništavanje forme,
- Taster za snimanje forme.

U bazi podataka se nalaze rezultati potvrde procesa (slika 10). Ona sadrži:

- Sve podatke iz registracione forme (jedinstveni broj alata, liniju, naziv alata, popunjen upitnik, dodatne komentare),
- Vreme i datum kada je izvršena konfirmacija procesa,
- Ime i prezime tehničara koji je servisirao alat.

Tool ID	Line	Tool name	Check result	Maintenance result	Technician	Date	Time
A05000001	OK	OK	OK	OK	OK	9/17/2018	2:55:05 PM Plaka 2 WSA2
A05000002	OK	OK	OK	OK	OK	9/18/2018	2:54:26 PM Plaka 2 WSA1
A05000003	OK	OK	OK	OK	OK	9/18/2018	2:54:33 PM Plaka 4 WSA1
A05000004	OK	OK	OK	OK	OK	9/18/2018	2:54:38 PM Plaka 5 WSA1
A05000005	OK	OK	OK	OK	OK	9/18/2018	2:54:43 PM Plaka 2 WSA2
A05000006	OK	OK	OK	OK	OK	9/18/2018	2:54:48 PM Plaka 3 WSA2
A05000007	OK	OK	OK	OK	OK	9/18/2018	2:54:53 PM Plaka 4 WSA2
A05000008	OK	OK	OK	OK	OK	9/18/2018	2:54:58 PM Plaka 5 WSA2
A05000009	OK	OK	OK	OK	OK	9/18/2018	2:55:03 PM Plaka 6 WSA2
A05000010	OK	OK	OK	OK	OK	9/18/2018	2:55:08 PM Plaka 7 WSA2
A05000011	OK	OK	OK	OK	OK	9/18/2018	2:55:13 PM Plaka 8 WSA2
A05000012	OK	OK	OK	OK	OK	9/18/2018	2:55:18 PM Plaka 9 WSA2
A05000013	OK	OK	OK	OK	OK	9/18/2018	2:55:23 PM Plaka 10 WSA2
A05000014	OK	OK	OK	OK	OK	9/18/2018	2:55:28 PM Plaka 11 WSA2
A05000015	OK	OK	OK	OK	OK	9/18/2018	2:55:33 PM Plaka 12 WSA2
A05000016	OK	OK	OK	OK	OK	9/18/2018	2:55:38 PM Plaka 13 WSA2
A05000017	OK	OK	OK	OK	OK	9/18/2018	2:55:43 PM Plaka 14 WSA2
A05000018	OK	OK	OK	OK	OK	9/18/2018	2:55:48 PM Plaka 15 WSA2
A05000019	OK	OK	OK	OK	OK	9/18/2018	2:55:53 PM Plaka 16 WSA2
A05000020	OK	OK	OK	OK	OK	9/18/2018	2:55:58 PM Plaka 17 WSA2
A05000021	OK	OK	OK	OK	OK	9/18/2018	2:56:03 PM Plaka 18 WSA2
A05000022	OK	OK	OK	OK	OK	9/18/2018	2:56:08 PM Plaka 19 WSA2
A05000023	OK	OK	OK	OK	OK	9/18/2018	2:56:13 PM Plaka 20 WSA2

Slika 10. Baza podataka sa rezultatima potvrde procesa

Baza podataka nam služi za analizu svakog alata pojedinačno. Ukoliko je došlo do neplaniranog zastoja zbog alata može se baza podataka filtrirati i videti kada je poslednji put servisiran alat, rezultate servisiranja i ko je poslednji put servisirao alat.

Sistem praćenja alata na bazi proizvedenih delova se sastoji od pokazivača koliko je alat uradio ciklusa od poslednjeg održavanja (slika 11). Broj ciklusa pokazuje broj od poslednjeg održavanja alata i ukoliko taj broj pređe definisanu vrednost, status alata prelazi u planiranje alata za održavanje. Status alata je isti kao u sistemu na vremenskoj bazi (zeleni, žuti i crveni status).

Crmping		Date		9/2/2018	Clean	Warning	Dirty
Tool name	Plate No.	Register button	Produced amount	Tool Status	Number of maintenances	Date of last maintenance	Overall produced amount
Jaws 1A	A01010047		3979		43	8/25/2018	323168
Jaws 1B	A01010048		3979		44	8/25/2018	323168
Jaws 1C	A01010049		3979		43	8/25/2018	324980
Jaws 1D	A01010050		3979		43	8/25/2018	324980
Jaws 2A	A01010051		6721		1	8/2/2017	7422
Jaws 2B	A01010052		6721		1	8/2/2017	7422
Jaws 3A	A01010053		6284		25	8/24/2018	180933
Jaws 3B	A01010054		6284		24	8/24/2018	180933
Jaws 3C	A01010055		6132		24	8/24/2018	180933
Jaws 3D	A01010056		6132		24	8/24/2018	180933
Jaws 4A	A01010057		0		11	8/29/2018	110741
Jaws 4B	A01010058		0		12	8/29/2018	110741
Jaws 4C	A01010059		0		11	8/29/2018	110893
Jaws 4D	A01010060		0		12	8/29/2018	110893
Jaws 5A	A01010061		7460		80	8/23/2018	806316
Jaws 5B	A01010062		7460		80	8/23/2018	806316
Jaws 5C	A01010063		0		84	8/31/2018	898851
Jaws 5D	A01010064		0		84	8/31/2018	898851

Slika 11. Tabela na bazi proizvedenih delova

U sistem praćenja alata na bazi proizvedenih delova ulaze alati koji su komplikovaniji, sastoje se od većeg broja standardnih komponentata (linearni i radijalni ležajevi, pneumatski cilindri...), imaju povećanu postojanost na liniji i ključni su za obezbeđenje kvaliteta tokom velike većine svog životnog veka. Takvi alati se repariraju, a preventivno se menjaju standardni delovi tih alata [3].

4. PREDSTAVLJANJE I REŠENJE PROBLEMA SA KALIBRACIONIM PINOVIMA

Proces provlačenja je visokoproduktivan i vrlo precizan postupak koji se primenjuje za finu obradu otvora i profilnih oblika. Glavno kretanje je pravolinijsko i izvodi ga alat. Alat za provlačenje je oblika kojeg treba da bude finalni deo. Razlikuje se unutrašnje i spoljašnje provlačenje. U ovom slučaju se koristi kalibracioni pin koji postavlja konačnu dimenziju na delu.

Provlačenjem kalibracionog pina kroz čauru koja je upresovana u aluminijumsko kućište dobijamo konačnu meru sa propisanim tolerancijama. Posle izvesnog vremena, kalibracioni pin se pohaba, mera izlazi iz granica tolerancije i pin se mora zameniti sa novim.

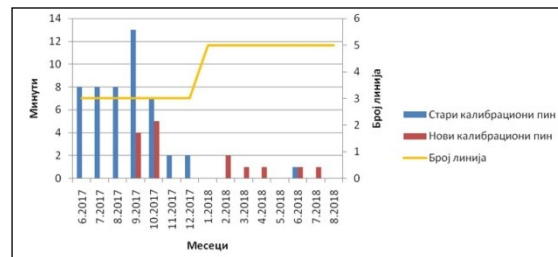
Zbog povećanih tehničkih zastoja i troškova zamene kalibracionih pinova potrebno je pronaći bolje rešenje koje će smanjiti troškove zamene i tehničke zastoje. Pošto se geometrija kalibratora nije mogla promeniti zbog dimenzionih ograničenja, pokušano je sa materijalom koji će imati veću površinsku tvrdoću i glatku poliranu spoljašnju površinu.

Materijal koji je korišćen je 1.2379 (H 55SrVMo12-1), iliti alatni čelik Č.4850, i imao je tvrdoću od 58HRC do 60HRC. Materijal sadrži 1.55% ugljenika, 0.3% silicijuma, 0.3% mangana, 12% hroma, 0.7% molibdena, 0.8% vanadijuma. Karakteristike ovog čelika je vrlo visoka otpornost na abrazive, umerena obradivost, dobra postojanost, žilavost.

Novi kalibracioni pin koje je testiran i koji se pokazao kao bolje rešenje je od materijala G10. Ovaj materijal u sebi sadrži 1% ugljenika, 15% hroma, 1% molibdena, 0.2% vanadijuma i 1.5% kobalta. Tvrdoća materijala nakon termičke obrade je oko 90 HRA. Kada se prebaci u HRC tvrdoća je oko 74-76 HRC. Kako je nakon termičke obrade ovaj materijal dosta tvrdi, a takođe i otporniji na habanje, uzet je za izradu kalibracionih pinova.

Na slici 12 vidimo da se od 11. meseca 2017. godine redukovala potrošnja kalibracionih pinova. Razlog toga je zamena novih pinova sa starim, što se može videti povećanom potrošnjom u devetom i desetom mesecu iste godine. Ukupna potrošnja starih kalibracionih pinova u ovom posmatranom periodu je 49 komada. Potrošnja

novih pinova u posmatranom periodu je 15 komada. Potrošnja je redukovana za 70%. Postignuta je ušteda u troškovima, stabilniji proces i ostvaren sigurniji kvalitet proizvoda i montaža proizvoda prema specifikacijama.



Slika 12. Dijagram zamene kalibracionih pinova

5. ZAKLJUČAK

Prednost servisne tačke u proizvodnji se može videti kroz više stvari: stabilan sistem održavanja alata i planiranja održavanja alata, sistem testiranja i simulacije rada alata kao na liniji i praćenje i analiza tehničkih gubitaka na dnevnom nivou za svaku liniju posebno.

Prostorni plan servisne tačke i tačno definisanje kretanja tehničara prilikom svake situacije (preventivno i korektivno održavanje) standardizuje rad i smanjuje mogućnost greške tehničara u nekim od tih koraka.

Velika prednost je, takođe, mogućnost podešavanja alata na servisnoj tački. Simulacijom rada alata kao na liniji može se podesiti alat i odmah nakon toga biti pušten u proizvodnju bez dodatnih podešavanja. Ovim redukujemo vreme zastoja na liniji i podešavanje alata na samoj liniji. Glavni fokus i efikasnost servisne tačke se postiže sistemom za praćenje i planiranje održavanja alata. Praćenje na vremenskoj bazi sadrži jednodielne alate koji omogućuju dugotrajnu postojanost na liniji. Praćenje alata na bazi proizvedenih delova sadrži alate koji su složeniji, sastoje se iz sklopova i imaju veći uticaj na kvalitet proizvoda. Prednost praćenja alata na bazi proizvedenih delova omogućava fokusiranje na pojedinačne alate, a ne na grupu alata, što povećava kapacitet servisne tačke.

6. LITERATURA

- [1] Milovanović, Z., Papić, L.J., "Održavanje i pouzdanost tehničkih sistema", Mašinski fakultet, Univerzitet u Banja Luci, 2007.
- [2] Todić, V., Penezić, N., Lukić, D., Milošević, M "Tehnološka logistika i preduzetništvo", FTN, Novi Sad, 2011.
- [3] Dimitrijević, N.: „Sistem za praćenje i planiranje alata u tehnološkom procesu montaže automobilskih komponenti“, master rad, FTN, Novi Sad, 2018.

Kratka biografija:



Nemanja Dimitrijević rođen je u Šapcu 1992. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mašinstva – Proizvodno mašinstvo, smer računarom podržane tehnologije, odbranio je 2018. god.

METODE OPTIMALNOG UPRAVLJANJA ZA OPTIMIZACIJU PERFORMANSI TRKAČKIH VOZILA

OPTIMAL CONTROL METHODS FOR THE OPTIMIZATION OF RACE CAR PERFORMANCE

Luka Vujičić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHANIZACIJA I KONSTRUKCIONO MAŠINSTVO

Kratak sadržaj – Razvijen je metod za simulaciju vremena kruga zasnovan na teoriji optimalnog upravljanja. Korišćen je kinematski model vozila sa jednim tragom. Problem je rešen metodom direktne kolokacije. Metodom je izvršena minimizacija vremena kruga sa istovremenom optimizacijom parametara.

Ključne reči: Teorija optimalnog upravljanja, Dinamika vozila, Optimizacija

Abstract – An optimal control lap time simulation method is developed. The vehicle model used is a kinematic single track model. The problem is solved with the direct collocation method. An example lap time minimisation study with simultaneous parameter optimization is performed..

Keywords: Optimal control theory, Vehicle dynamics, Optimization

1. UVOD

Osnovni kriterijum u razvoju u moto-sportu je izvršiti dati manevar za minimalno vreme. Pri tome fizičke granice trkačke staze i tehnička ograničenja definisana pravilima moraju da se poštuju. U granicama ovih pravila konstruktori i trkački timovi razvijaju i podešavaju vozila da minimizuju vreme kruga.

U ovom procesu se koriste simulacije da bi se ispitao uticaj konstrukcionih i parametara podešavanja na performanse vozila.

Oblasti razvoja uključuju generisanje aerodinamičke sile potiska, poboljšanje prijanjanja pneumatika, karakteristika pogonske grupe i oslanjanja i dr. U donošenju odluka inženjeri koriste tzv. alate za simulaciju vremena kruga da ispitaju uticaj ovih parametara na vreme kruga.

Pregled razvijenih metoda je dat u [1, 2].

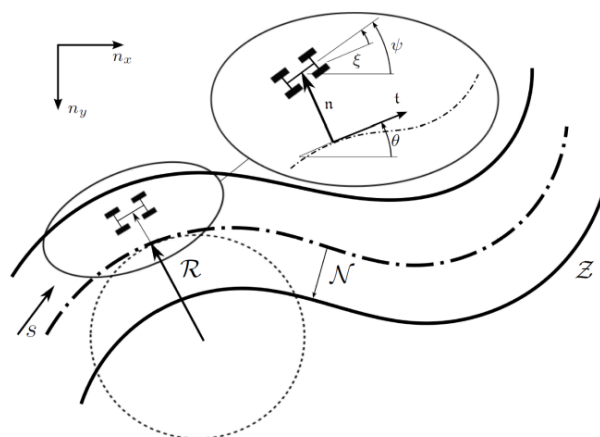
Cilj rada jeste razvoj metoda za optimizaciju performansi zasnovanog na optimalnom upravljanju. Metod treba da bude fleksibilan u smislu lakih izmena modela i daljeg razvoja metoda.

2. MODELOVANJE I FORMULACIJA PROBLEMA

2.1. Modelovanje staze

Da bi se definisao manevar koji vozilo treba da izvrši potreban je matematički opis staze.

Za opis staze korišćen je kurvilinearni koordinatni sistem.



Slika 1. Kurvilinearni opis staze [3]

Kako je prikazano na slici 1., pozicija težišta je definisana kurvilinearnom apscisom $s(t)$ i vektorom $\mathbf{n}(s(t))$. Prva veličina jeste rastojanje pređeno srednjom linijom od startne linije, dok je druga rastojanje težišta vozila od srednje linije u pravcu normalnom na vektor tangente srednje linije $\mathbf{t}(s(t))$.

Staza je u bilo kojoj tački s opisana poluprečnikom krivine R , $C=1/R$ i polu-širinom N . Vektor tangente srednje linije $\mathbf{t}$ definisan je uglom orijentacije staze θ . Kako je ξ ugao između podužne ose vozila i $\mathbf{t}$, apsolutni ugao zakretanja vozila je $\psi = \theta + \xi$.

Podrazumeva se da je pređeni put $s(t)$ funkcija vremena koja isključivo raste i da su, prema tome, rastojanje i vreme alternativne nezavisne promenljive.

Relacije između lokalnog koordinatnog sistema vozila i globalnog kurvilinearnog sistema date su sledećim jednačinama [3]:

$$\frac{ds}{dt} = \frac{v \cos(\beta + \xi)}{1 - nC} \quad (1)$$

$$\frac{dn}{dt} = v \sin(\beta + \xi) \quad (2)$$

$$\frac{d\xi}{dt} = \frac{d\psi}{dt} - C \frac{ds}{dt} \quad (3)$$

NAPOMENA:

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

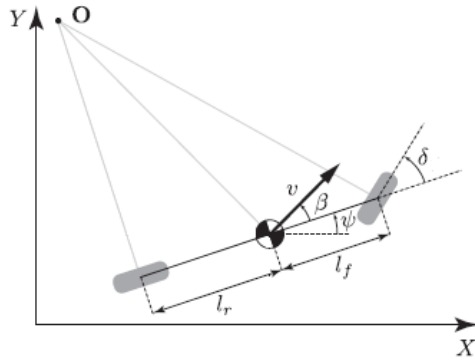
2.2. Modelovanje vozila

Korišćen je kinematski model vozila sa jednim tragom [4]. Promenljive stanja i upravljačke promenljive su definisane redom vektorima stanja i upravljanja:

$$x = (v \ \beta \ \psi) \quad (4)$$

$$u = (\delta \ a) \quad (5)$$

gde je v brzina težišta, β ugao bočnog klizanja, ψ ugao zakretanja vozila, δ ugao zakretanja upravljanog točka i a ubrzanje težišta u istom pravcu kao i brzina. Ove veličine su prikazane na slici 1.



Slika 2. Kinematski model vozila sa jednim tragom [4]

Jednačine koje opisuju model su:

$$\frac{dv}{dt} = a \quad (6)$$

$$\frac{d\beta}{dt} = \tan^{-1}\left(\frac{l_r}{l_r + l_f} \tan \delta\right) \quad (7)$$

$$\frac{d\psi}{dt} = \frac{v}{l_r} \sin \beta \quad (8)$$

Parametri modela su rastojanje težišta od prednje i zadnje osovine, redom l_f i l_r .

Pri simulaciji dinamičkih sistema kao nezavisna promenljiva se obično koristi vreme. Međutim u ovom slučaju, vreme i pređeni put se mogu smatrati alternativnim nezavisnim promenljivim. Za ovu primenu, korišćenje pređenog puta kao nezavisne promenljive ima prednosti. Naime, uspostavlja se eksplicitna veza sa pozicijom vozila na stazi i smanjuje konačna veličina problema. Promena nezavisne promenljive se može izvršiti samo ako postoji tačna korespondencija između s i t . Ovo je tačno samo ako se vozilo kreće samo u pravcu rasta s . Faktor za promenu je recipročna vrednost brzine duž srednje linije staze[3]:

$$S_f = \left(\frac{ds}{dt}\right)^{-1} = \frac{1 - nC}{v(s) \cos(\beta(s) + \xi(s))} \quad (9)$$

Ovom promenom uvodimo i dva nova stanja, normalno rastojanje od srednje linije n i ugao između podužne ose vozila i tangente srednje linije staze ξ .

Izvodi stanja po pređenom putu dobijaju se pomoću jednačina (2,3,6,7,8), tako što se iste pomnože faktorom S_f :

$$\frac{dn}{ds} = \frac{dn}{dt} S_f \quad (10)$$

$$\frac{d\xi}{ds} = \frac{d\xi}{dt} S_f \quad (11)$$

$$\frac{dv}{ds} = \frac{dv}{dt} S_f \quad (12)$$

$$\frac{d\beta}{ds} = \frac{d\beta}{dt} S_f \quad (13)$$

$$\frac{d\psi}{ds} = \frac{d\psi}{dt} S_f \quad (14)$$

2.3. Formulacija problema optimalnog upravljanja

Cilj ovog metoda jeste da se pronađu optimalne putanje stanja i kontrola dinamičkog sistema (trkačko vozilo) za koje je vreme manevra minimalno. Pri tome se moraju poštovati ograničenja dinamike vozila, ograničenja staze i ograničenja vrednosti promenljivih stanja i upravljanja. Ovi zahtevi su formulisani u sledeći problem optimalnog upravljanja:

$$\min J = \int_{s_0}^{s_f} \frac{1 - n(s)C(s)}{v(s) \cos(\beta(s) + \xi(s))} ds \quad (15)$$

$$\text{s.t. } \frac{dx}{ds} = f(x(s), u(s)) \quad (16)$$

$$x(s_0) = x_0 \quad (17)$$

$$x(s_f) = x_f \quad (18)$$

$$x_L \leq x(s) \leq x_U \quad (19)$$

$$u_L \leq u(s) \leq u_U \quad (20)$$

$$N_l \leq n(s) \leq N_r \quad (21)$$

Funkcija cilja J jeste minimizacija vremena manevra. Jednačina (16) definiše ograničenja dinamike vozila. Ograničenja početnog i krajnjeg stanja su definisana jednačinama (17,18). Jednačine (19,20) su ograničenja vrednosti promenljivih stanja i upravljanja, gde indeksi L i U označavaju donje i gornje granične vrednosti koje se mogu odrediti na osnovu poznavanja dinamičkog sistema. Na kraju, vozilo se zadržava u okviru granica staze ograničenjem definisanim jednačinom (21), gde su N_l i N_r leva i desna polu-širina staze.

2.4. Rešavanje problema optimalnog upravljanja

Izabrani numerički metod rešavanja je direktna kolokacija [5], u kojoj se kontinualni problem optimalnog upravljanja pretvara u diskretni nelinearni program.

Interval pređenog puta na kom je problem definisan $[s_0 \dots s_f]$ je diskretizovan u N segmenata:

$$s_0 = s_1 < s_2 \dots < s_N = s_f \quad (22)$$

Diskretni korak h_i je definisan sledećom jednačinom:

$$h_i = s_{i+1} - s_i \quad \text{za } i = 1 \dots N-1 \quad (23)$$

Sada kada je problem diskretizovan formuliše se nelinearni program:

$$\min J = \sum_{i=1}^{N-1} \frac{h_i}{2} (S_{fi} + S_{fi+1}) \quad (24)$$

$$\Phi_i = xi + 1 - xi - \frac{hi}{2} (x'_i + x'_{i+1}) = 0 \quad \text{za } i = 1 \dots N-1 \quad (25)$$

$$Y = x_N - x_f = 0 \quad (26)$$

$$\Psi = x_1 - x_0 = 0 \quad (27)$$

$$\square \quad \square \quad \square \quad t_l \leq n(s) \leq t_r \quad (28)$$

$$\square \quad x_L \leq x(s) \leq x_U \quad (29)$$

$$\square \quad u_L \leq u(s) \leq u_U \quad (30)$$

Funkcija cilja J definisana je primenom trapezoidnog pravila numeričke integracije na integral recipročne vrednosti brzine duž srednje linije staze S_f u granicama u kojim je problem definisan. Isto pravilo numeričke integracije je primenjeno u ograničenju dinamike sistema Φ , gde x' predstavlja jednačine (10-14).

Ograničenja u jednačinama (26-30) predstavljaju diskretizovana ograničenja problema optimalnog upravljanja koja su definisana jednačinama (17-21).

2.5. Implementacija

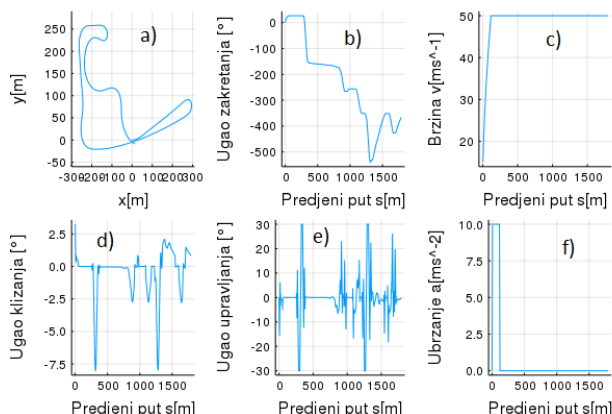
Implementacija metoda je izvršena u programskom jeziku Julia [6], koristeći jezik za algebarsko modelovanje JuMP [7]. Za rešavanje konačnog nelinearnog programa korišćen je solver IPOPT [8]. Prvi i drugi izvodi potrebni za formiranje Jakobijeve i Heseove matrice koje IPOPT zahteva kao ulaz izračunati su metodom algebarske diferencijacije koja je deo paketa JuMP.

Prednost korišćenja ovog okruženja je to što je sve od modelovanja vozila, preko formulacije i transkripcije problema optimalnog upravljanja do rešavanja rezultujućeg nelinearnog programa urađeno u istom programskom jeziku. Ovo olakšava i ubrzava implementaciju metoda. Dodatne prednosti su performanse programskog jezika i činjenica da je sav korišćen softver otvorenog tipa (open source).

3. REZULTATI SIMULACIJA

3.1. Minimalno vreme kruga

Prva studija je minimizacija vremena manevra za ceo krug staze. Korišćena staza je prikazana na slici 3.a. Dužina staze je 1792[m], sa konstantnom širinom od 6 [m] celom dužinom staze i vozi se u smeru kazaljke na satu. Problemu su dodata dodatna ograničenja na količinu promene upravljačkih promenljivih između dva koraka. Parametri modela i informacije o rešenju su prikazani u tabelama 2. i 3. u dodatku.

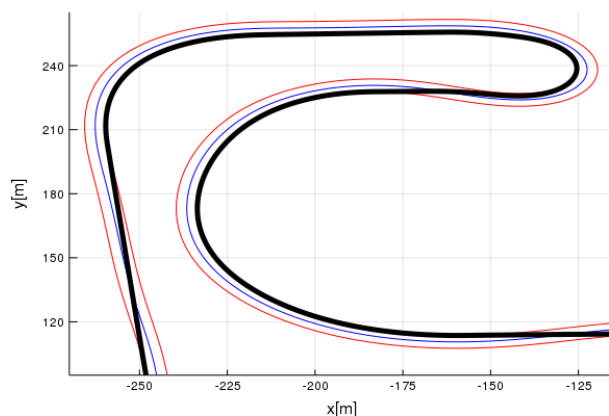


Slika 3. Rezultati simulacije

Na ostalim dijagramima na slici 3. prikazana su stanja i upravljačke promenljive. Može se videti da brzina počinje od zadate početne vrednosti 0 [ms⁻¹] i raste do 50 [ms⁻¹], što je zadata gornja granična vrednost, na kojoj ostaje do kraja manevra. Ubrzanje takođe kreće od zadate početne vrednosti od 0 [ms⁻²] i raste do 10 [ms⁻²], dok brzina ne dostigne gornju graničnu vrednost, kada opet opada na 0 [ms⁻²]. Može se videti da ubrzanje ni u jednom trenutku ne dostiže negativnu vrednost i da kao posledica toga brzina ne opada.

Sistem se ovako ponaša zbog značajno pojednostavljenog modela vozila koji ne predstavlja verno ponašanje stvarnog vozila. Naime, model ne uzima u obzir sile koje deluju na vozilo, među njima i sile koje generišu pneumatici u kontaktu sa podlogom. Upravo ove sile ograničavaju bočno ubrzanje, a time i brzinu koju vozilo može da postigne u krivinama.

Ovo se odražava i na proračunatu putanju vozila koja je prikazana za jedan segment staze crnom linijom na slici 4. Vozilo ulazi u ovaj segment iz šikane desno-levo na unutrašnjoj strani i prelazi na unutrašnju stranu za sledeću desnu lakat krivinu, nakon čega prelazi na unutrašnju liniju za sledeću dugu levu krivinu, koja je poslednji deo ovog segmenta. Celom dužinom staze vozilo teži unutrašnjim linijama u krivinama, koje zadržava i na pravcima. Razlog za ovo jeste što vozilo brzo dostigne maksimalnu brzinu, nakon čega traži najkraću putanju unutar granica staze.



Slika 4. Optimalna putanja vozila za jedan segment staze

3.2. Istovremena optimizacija parametara

Prethodna studija je proširena istovremenom optimizacijom parametara. Za ovu studiju model je blago izmenjen. Osnovni model ima dva parametra l_f i l_r . Za ovu studiju ova dva parametra su zamenjena jednim, raspodelom mase na prednjoj osovini d_f i konstantom $l=l_f+l_r$, koja predstavlja ukupno osovinsko rastojanje. U jednačinama modela vozila l_f i l_r su zamenjeni sledećim izrazima:

$$l_f = (1-d_f)l \quad (31)$$

$$l_r = d_f l \quad (32)$$

Osnovni problem optimalnog upravljanja je proširen jednostavno dodajući d_f u promenljive i dva dodatna ograničenja, jedno za jednakost parametra u svim koracima i drugo za granične vrednosti parametra:

$$d_f(s_i) = d_f(s_{i+1}) \quad \text{za } i = 1 \dots N-1 \quad (33)$$

$$d_{f \min} \leq d_f(s) \leq d_{f \max} \quad (34)$$

Rezultati ove studije u poređenju sa prvom dati su u tabeli 1.

Tabela 1. Rezultati optimizacije parametara

	Osnovno	Optimizovano
Vreme kruga [s]	36.2784	36.2779
d_f [-]	0.48	0.401
l_f [m]	1.716	1.976
l_r [m]	1.584	1.323

4. ZAKLJUČAK

Na osnovu gore prezentovanog rada se mogu izvesti sledeći zaključci:

- Kinematski model vozila za jednim tragom korišćen u radu je jednostavan za implementaciju i parametrizaciju sa samo dva parametra. Takođe, metod sa ovim modelom pokazuje dobru efikasnost. Međutim model je previše pojednostavljen, što je pokazano i naglašeno u rezultatima simulacija koji ne reprodukuju ponašanje stvarnog vozila
- Izbor metoda direktne kolokacije za numeričko rešavanje problema optimalno upravljanja se pokazao kao dobar. Metod se pokazao kao robustan i fleksibilan, što je pokazano jednostavnom izmenom modela i problema za omogućenje istovremene optimizacije parametara
- Okruženje u kom je izvršena implementacija metoda se pokazala dobro. Ceo metod je implementovan u programskom jeziku Julia. Paket za optimizaciju JuMP je moćan sa svojom funkcionalnošću algoritamske diferencijacije, koja nije pre korišćena za ovu primenu

Kao pravac daljeg istraživanja i razvoja metoda prioritet ima model vozila, kako bi metod mogao da uvid u optimalno ponašanje stvarnog vozila. Predlaže se korišćenje dinamičkog modela sa nelinearnim modelom pneumatika. Korisno bi bilo i istražiti uticaj složenosti modela na efikasnost metoda, odnosno vremena proračuna.

5. DODATAK

Tabela 2. Parametri modela i simulacije

Simbol	Opis	Jedinica
l_f [m]	Rastojanje težišta od prednje osovine	1.72
l_r [m]	Rastojanje težišta od zadnje osovine	1.58
v_{max} [ms ⁻¹]	Gornja granična vrednost brzine	50
a_{max} [ms ⁻¹]	Gornja granična vrednost ubrzanja	10
δ_{min} [°]	Donja granična vrednost ugla upravljanja	-30
δ_{max} [°]	Gornja granična vrednost ugla upravljanja	30
r_a [$\frac{ms^{-2}}{korak}$]	Količina promene ubrzanja	1
r_b [$\frac{°}{korak}$]	Količina promene ugla upravljanja	2
v_0 [ms ⁻¹]	Početna brzina	15
ψ_0 [°]	Početni ugao zakretanja	0
a_0 [ms ⁻²]	Početno ubrzanje	0
δ_0 [°]	Početni ugao upravljanja	0

Tabela 3. Informacije o rešenju prve studije

Opis	Vrednost	Jedinica
Diskretni korak h	0.8	m
Broj diskretnih tačaka	2240	-
Broj promenljivih nelinearnog programa	15680	-
Trajanje proračuna	33.6	s

6. LITERATURA

- [1] D. Casanova, "On minimum time manouvering: The theoretical optimal lap", PhD Thesis, Cranfield University, 2000.
- [2] T. Völkl, "Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen", PhD Thesis, Technische Universität Darmstadt, 2013.
- [3] G. Perantoni, D.J.N. Limebeer, "Optimal control for a formula one car with variable parameters", *Vehicle System Dynamics*, Vol. 52, pp. 653-678, September 2014.
- [4] J. Kong, M. Pfeiffer, G. Schildbach, F. Borrelli, "Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design", *IEEE Intelligent Vehicles Symposium*, 2015
- [5] J.T. Betts, "Practical Methods for Optimal Control and Estimation using Nonlinear Programming: Second Edition, *Advances in Design and Control*", Society for Industrial and Applied Mathematics, 2010.
- [6] J. Bezanson, A. Edelman, S. Karpinski, V. Shah, "Julia: A Fresh Approach to Numerical Computing", *SIAM Review*, Vol. 59, pp. 65-98, 2017.
- [7] I. Dunning, J. Huchette, M. Lubin, "JuMP: A modeling language for Mathematical Optimization", *SIAM Review*, Vol. 59, pp. 295-320, 2017.
- [8] A. Wächter, L.T. Biegler, "On the Implementation of a Primal-dual Interior Point Filter Line Search Algorithm for Large-scale Nonlinear Programming", *Mathematical Programming*, Vol. 106, pp. 25-57, 2006.

Kratka biografija:



Luka Vujičić rođen je u Novom Sadu 1992. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mehanizacije i konstrukcionog mašinstva – Automobilsko inženjerstvo odbranio je 2018.god.
kontakt: luka.vujicic@gmail.com

**PRIMENA VLAKANA NERĐAJUĆEG ČELIKA ZA SEPARACIJU AEROSOLA
NASTALOG U METALOPRERAĐIVAČKOJ INDUSTRIJI****APPLICATION OF STEEL FIBERS FOR THE SEPARATION OF AEROSOL
GENERATED IN METALWORKING INDUSTRY**Marko Gomola, Dunja Sokolović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ENERGETIKA I PROCESNA TEHNIKA**

Kratak sadržaj – *Primenom sredstava za hlađenje i podmazivanje (SHP) u metaloprerađivačkoj industriji dolazi do formiranja SHP aerosola. Ovi SHP aerosoli predstavljaju opasnost po zdravlje radnika, te ih je neophodno efikasno ukloniti iz radne sredine.*

Koalescentna filtracija je često primenjivana metoda za uklanjanje SHP aerosola. U ovom radu ispitana je mogućnost primene vlakana od nerđajućeg čelika kao filtracionog sloja za separaciju SHP aerosola. Vlakna nerđajućeg čelika su se pokazala kao relativno dobar materijal u koalescentnoj filtraciji za ovu vrstu separacije.

Ključne reči: *Sredstva za hlađenje i podmazivanje – SHP sredstva, SHP aerosoli, koalescentna filtracija*

Abstract – *With the application metal working fluids (MWF) in the metalworking industry, MWF aerosol is formed. These MWF aerosols represent a health risk for workers and must be effectively removed from the workplace. Coalescence filtration is often used to remove MWF aerosols. In this work, the possibility of using stainless steel fibers as a filtration layer for the separation of MWF aerosol is investigated. Stainless steel fibers have proven to be relatively good in coalescent filtration for this type of separation.*

Keywords: *Metal working fluids – MWF, MWF aerosol, coalescence filtration*

1. UVOD

U toku operacija obrade metalnih delova, kao što su: struganje, glodanje, brušenje, itd., dolazi do trošenja površine alata usled trenja metal o metal izazvanog kontaktom sa metalnim obratkom, kao i do zagrevanja tih površina. Da bi se smanjilo trenje i obezbedilo adekvatno hlađenje u zoni kontakta i time produžio radni vek mašine alatke i obezbedio visok kvalitet obrađenog metalnog elementa, u toku obrade nanose se sredstva za hlađenje i podmazivanje (SHP sredstva). SHP sredstva pored ovoga omogućavaju spiranje metalnih opiljaka koji se formiraju i ometaju obradu. SHP sredstva, iako imaju veoma pozitivne efekte na proces obrade metala, predstavljaju izrazito opasne materije po zdravlje radnika, zbog čega se mora voditi računa kako i pod kojim uslovima se

primenjuju. Pored SHP sredstava koji predstavljaju hazardne supstance, još opasniji su aerosoli koji nastaju njihovom primenom. SHP aerosoli mogu nastati na tri načina: udarom mlaza SHP sredstva o površinu kontakta, centrifugalnom silom, odnosno rotiranjem struga, ili pak isparavanjem tečnosti, odnosno njenim hlađenjem i kondenzacijom. Veličina kapi SHP aerosola kreće se u opsegu od 0,1 do 15 μm. Ovako male kapi imaju mnogo veću mogućnost prodiranja duboko u disajne puteve čoveka, što može da dovede do alergijskih reakcija. Pored toga izloženost ovom SHP aerosolu dovodi do iritacije očiju i kože, respiratornih smetnji, astme, a ako izloženost traje duže vreme može izazvati čak kancer: ždrela, jednjaka, pluća, kože, bešike, pankreasa, i reproduktivnih organa [1-3].

Od strane Američkog ministarstva za rad “Occupational Safety and Health Administration-OSHA” propisana granica izloženosti radnika SHP aerosolu nastalog od mineralnog ulja, iznosi 5 mg/m³ uprosečeno na osmočasovni radni dan. Od strane “Environmental Protection Agency- EPA” su uvedeni standardi kojima se ograničava količina aerosola koji sadrži čestice određene veličine, i to za čestice sitnije od 10 μm PM10 i čestice sitnije od 2,5 μm PM2.5 (PM- Particulate Matter), umesto ograničenja ukupne količine aerosola [1].

Radi zaštite zdravlja radnika, radne i životne okoline, neophodno je ukloniti SHP aerosol iz radne atmosfere, kao i iz otpadnog vazduha pre ispuštanja u atmosferu. U ovu svrhu se često koristi koalescentna filtracija. Koalescentna filtracija je filtracija kod koje se pomoću adekvatno odabranog i dimenzionisanog filtracionog sloja omogućava ukupnjavanje sitnih kapi SHP aerosola, koje nakon prolaska kroz njega imaju mogućnost da se istalože. Taloženje ovih kapi pre filtracionog sloja nije bilo moguće jer su one previše sitne, pa kao takve lebde u vazduhu [3, 4].

Filtracioni sloj može biti izgrađen od granula ili pak vlakana. Mnogo češće se koriste vlaknasti slojevi jer je kod njih lakše menjati geometriju sloja (poroznost i permeabilnost sloja). Materijali od kojih se mogu formirati vlaknasti slojevi mogu biti različiti, od staklenih vlakana, preko različitih polimera do nerđajućeg čelika [4, 5].

Cilj ovog eksperimentalnog rada bio je da se ispita mogućnost primene vlakana nerđajućeg čelika u koalescentnoj filtraciji za separaciju aerosola nastalog u metaloprerađivačkoj industriji. Filtracioni sloj je formiran od čeličnih vlakana debljine 20 μm. Ispitivana je efikasnost separacije SHP aerosola formiranog od 6%

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Dunja Sokolović, vanr. prof.

vodene emulzije. U cilju optimizovanja rada koalescera ispitivana je efikasnost separacije primenom četiri različite brzine ventilacionog vazduha: 1 m/s, 3 m/s, 4 m/s i 5 m/s.

2. EKSPERIMENTALNI PROGRAM

Glavni mehanizam kojim je formiran SHP aerosol u ovom eksperimentu je delovanjem centrifugalne sile. Uticaj čvrstih čestica koje su mikronskih veličina je zanemaren. Takođe uslovi eksperimenta isključuju prisustvo plivajućeg ulja i mikroorganizama. Za potrebe realizacije eksperimentalnog programa, korišćeno je komercijalno SHP sredstvo mineralnog porekla. Za pripremu emulzije, korišćena je voda gradskog vodovoda. Kao simulator mašine alatke, korišćen je laboratorijski aerosol generator uz pomoć kojeg je formiran SHP aerosol. SHP aerosol AE06, nastao je atomiziranjem 6% vodene emulzije E. Emulzija E je formirana upotrebom navedenog SHP sredstva i temperirane vode. Filtracija je vršena u koalesceru u kojem je filtracioni sloj formiran od vlakna nerđajućeg čelika, debljine 20 µm, oznake SS20. Debljina filtracionog sloja iznosila je 5cm, dok je permeabilnosti sloja bila $K_{01}=7,56 \times 10^{-9} \text{ m}^2$, a poroznosti 0,96.

Eksperiment je vršen pri četiri različite brzine vazduha: 1, 3, 4 i 5 m/s.

2.1. Metode rada i eksperimentalne tehnike

Za određivanje karakteristika aerosola, korišćen je optički spektrometar „Particle Size Analyzer PCS 2010“. Karakteristike koje meri ovaj uređaj su: raspodela veličine čestica, masena i brojčana koncentracija aerosola.

Merna mesta na kojim je vršeno uzorkovanje SHP aerosola su:

1. Pre filtera – IN – „inlet“
2. Nakon filtera – AF – „after filter“

Efikasnost separacije aerosola primenom koalescentne filtracije, analizirana je preko dve karakteristike SHP aerosola:

1. masene koncentracije (C_m)
2. brojčane koncentracije (C_n)

Efikasnost separacije računata je prema formulama (1) i (2):

$$Em = \frac{(C_{m_{in}} - C_{m_{af}})}{C_{m_{af}}} \cdot 100 \% \quad (1)$$

$$En = \frac{(C_{n_{in}} - C_{n_{af}})}{C_{n_{af}}} \cdot 100 \% \quad (2)$$

Gde su:

Em- efikasnost praćena preko masene koncentracije

En- efikasnost praćena preko brojčane koncentracije

$C_{m_{in}}$ - masena koncentracija na mernom mestu IN

$C_{m_{af}}$ - masena koncentracija na mernom mestu AF

$C_{n_{in}}$ - brojčana koncentracija na mernom mestu IN

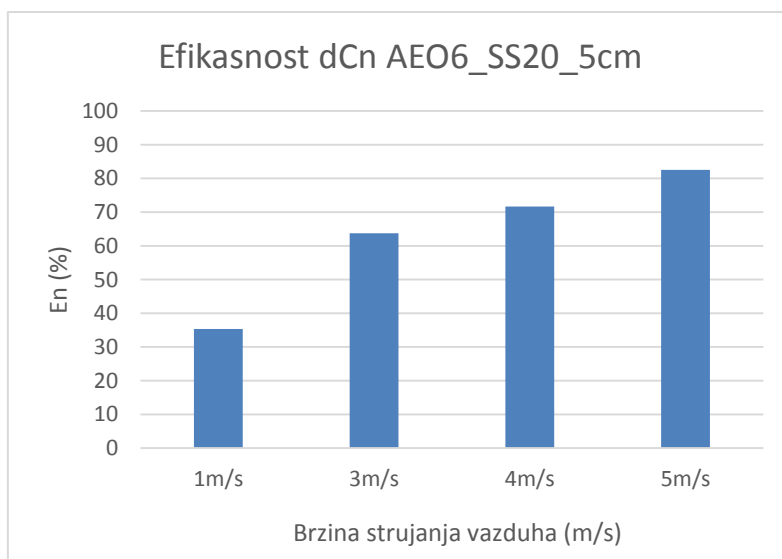
$C_{n_{af}}$ - brojčana koncentracija na mernom mestu AF

Aparatura na kojoj je izvođen ogled sastoji se od: aerosol generatora, ventilacone cevi, filtracionog sloja i ventilatora smeštenog na kraju cevi. Merenja su vršena 60 minuta od pokretanja ogleda.

3. PRIKAZ I ANALIZA REZULTATA

Na slici 1. prikazana je efikasnost separacije praćene preko brojčane koncentracije SHP aerosola. Pri brzini 1 m/s efikasnosti En , iznosi 35,33%, dok pri brzini od 3 m/s ona iznosi 63,73%. Trend rasta se nastavlja, pa tako pri brzini od 4 m/s efikasnost iznosi 71,65%, a pri najvećoj brzini strujanja vazduha od 5 m/s efikasnost En iznosi čak 82,53%. Usled povećanja brzine vazduha, dolazi do porasta efikasnosti En , što ukazuje da je koalescencija kapi bolja pri većim brzinama strujanja vazduha.

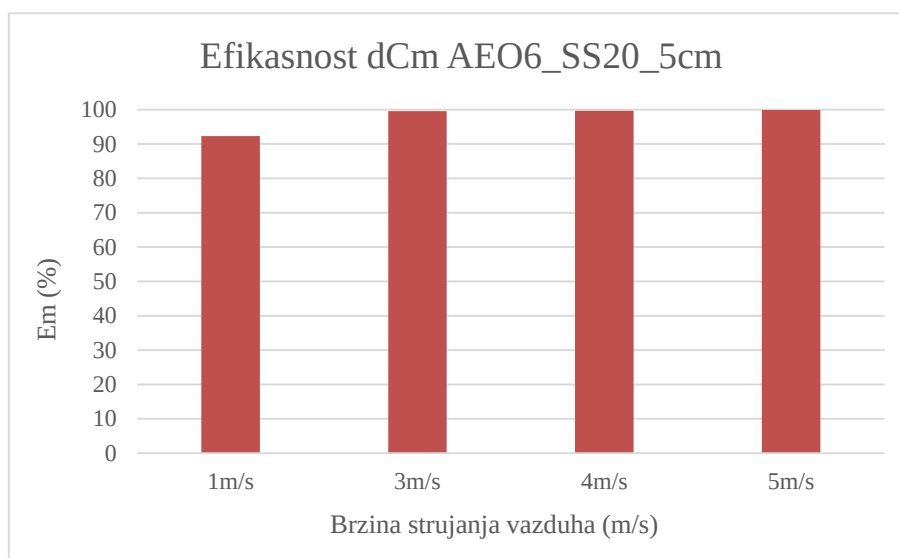
Pri višim brzinama je intenzivnije strujanje vazduha, što izaziva bolje mešanje SHP aerosola, a to dovodi do više slučajnih sudara između kapi, i kao rezultat ima veći broj koalesciranih (ukrupljenih) kapi, a samim tim i višu efikasnost separacije. Stoga, kada je koncentracija sitnih kapi u aerosolu koje treba ukloniti velika, potrebno je kao radnu brzinu odabrati brzinu od 5m/s.



Slika 1. Efikasnost En za sve brzine za materijal SS20

Na slici 2. prikazana je efikasnost separacije praćene preko masene koncentracije za sve ispitivane brzine. Već pri najnižoj brzini od 1 m/s vrednost efikasnosti separacije E_m je izrazito visoka i iznosi 92,31%, što nije bio slučaj kada se efikasnost separacije pratila preko brojčane koncentracije E_n . Za sledeću brzinu od 3 m/s,

separacije praćene preko masene koncentracije, u odnosu na efikasnost separacije praćene preko brojčane koncentracije govori o tome da je ostao određen broj najsitnijih kapi (≤ 1) u SHP aerosolu nakon filtracionog sloja.



Slika 2. Efikasnost E_m za sve brzine za materijal SS20

efikasnost E_m je skoro pa maksimalna, čak 99,59%. Pri daljem povećanju brzine strujanja vazduha, trend porasta se nastavlja, ali sa obrzirom da je već pri brzini od 3m/s postignuta visoka efikasnost separacije, povećanje je zanemarljivo malo.

Kada je propisana granična vrednost SHP aerosola preko ukupne masene koncentracije tada radna brzina strujanja vazduha treba da bude 3 m/s.

Razlika između efikasnosti praćene preko brojčane i masene koncentracije za iste uslove se može objasniti na sledeći način. Sitne kapi, manje od $1\mu\text{m}$, imaju gotovo zanemarljivu masu pa je njihov udeo u masenoj koncentraciji mali, iako brojčana koncentracija sitnih kapi može da bude visoka.

4. ZAKLJUČAK

Porastom brzine strujanja vazduha, raste i efikasnost separacije nezavisno preko koje koncentracije SHP aerosola se prati.

Pri višim brzinama je intenzivnije strujanje vazduha, što izaziva bolje mešanje SHP aerosola, tj. više slučajnih sudara između kapi, što kao rezultat ima veći broj koalesciranih (ukrupljenih) kapi. Efikasnost separacije praćene preko brojčane koncentracije je znatno manja, nego kada se ona prati preko masene koncentracije.

Primeru radi pri brzini od 1 m/s E_n je 35,33%, dok je E_m 92,31%. Ove se dešava, zbog toga jer sitne kapi imaju malu masu, koja ne figuriše značajno u masenoj koncentraciji, dok to nije slučaj kada je brojčana koncentracija upitanju.

Vlakana nerđajućeg čelika pokazala su se kao relativno dobar materijal za separaciju SHP aerosola. Kada je bitno postići propisanu granicu ukupne masene koncentracije SHP aerosola, onda radna brzina vazduha treba da bude

3m/s, jer je pri toj brzini E_m 99,59%. Dok kada je bitno efikasno ukloniti i najsitnije čestice SHP aerosola ($\leq 1\mu\text{m}$) tada radna brzina vazduha treba da bude 5m/s jer se u tom slučaju postiže najveća efikasnost separacije E_n 82,53%.

Visoka radna brzina pri uklanjanju sitnih kapi predstavlja ograničenje primene vlakana nerđajućeg čelika za separaciju SHP aerosola.

5. LITERATURA

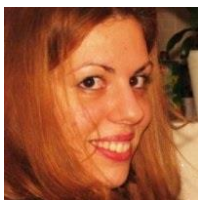
- [1] D. Sokolović, Istraživanje fenomena aerosola formiranog od emulzija mineralnih i drugih ulja u vodi, doktorska disertacija, Univerzitet u Novom Sadu, Tehnološki fakultet, Novi Sad, 2012.
- [2] U.S. department of health and human services, Occupational exposure to metalworking fluids, National Institute for Occupational Safety and Health, Publication No. 98-116, Mart 1998.
- [3] D. Sokolović, W. Höflinger, Z. Zavargo, R. Šećerov Sokolović, Uticaj ventilacije komore mašine alatke na osobine SHP aerosola, Hemijska industrija, 2012, 66(1), 67-77.
- [4] D. Govedarica, D. Sokolović, Separacija emulzija koalescencijom u sloju vlakana, Univerzitet u Novom Sadu, Fakultet Tehničkih nauka, Novi Sad 2014.
- [5] S. U. Patel, Improving performance and drainage of coalescing filters, doktorska disertacija, The University of Akron, Akron, 2010.

Kratka biografija:



Marko Gomola rođen je u Sremskoj Mitrovici 1994. god. Master rad na Fakultetu tehničkih nauka iz oblasti Energetike i procesne tehnike odbranio je 2018.god.

kontakt: markogomola@gmail.com



Dunja Sokolović rođena je u Novom Sadu 1981. Doktorirala je na Tehnološkom fakultetu u Novom Sadu, UNS., a od 2017 je u zvanju vanrednog profesora na Fakultetu tehničkih nauka. Oblast interesovanja su procesna tehnika, održivi razvoj, zaštita životne sredine.

NADZOR HABANJA ALATA PRI OBRADI MATERIJALA POVEĆANE TVRDOĆE TOOL MONITORING WHILE CUTTING HARDENED MATERIALS

Nemanja Klještan, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PROIZVODNO MAŠINSTVO

Kratak sadržaj – U ovom radu je rečeno nešto više o nadzoru alata prilikom obrade rezanjem. Izvršen je eksperiment koji se zasnivao na nadzoru alata prilikom obrade materijala povećane tvrdoće. Nadzor alata podrazumijeva praćenje životnog vijeka alata kao i praćenje loma alata. Alati koji su se koristili su pločice od tvrdog metala. Na kraju je izvršeno poređenje rezultata eksperimenta.

Ključne reči: Nadzor, alati, habanje, sile rezanja

Abstract – This paper is telling us something more about tool monitoring while cutting. There was an experiment that was based on tool monitoring while cutting hardened materials. Tool monitoring implies tool life monitoring and tool fracture monitoring. Tools that were used were inserts from hard metal. In the end experiment results were compared.

Keywords: Monitoring, tools, wear, forces cutting

1. UVOD

Sa razvojem tehnologija obrade rezanjem razvijali su se i sistemi za praćenje i nadzor alata u tehnološkim obradnim sistemima. Primjenom sistema za praćenje i nadzor alata u jednom tehnološkom obradnom sistemu moguće je povećati produktivnost. Naime alati se i danas uglavnom mijenjaju na osnovu neke subjektivne procjene operatera. Tako operateri uglavnom ne rizikuju i mijenjaju alate prije nego habanje uopšte počne. Ukoliko ipak operater čeka da uoči loš kvalitet obrađene površine ili habanje alata moguće je da će se pojaviti škart a samim tim i određeni gubici.

Kada se priča o nekim visokoproduktivnim tehnološkim obradnim sistemima, gubici mogu biti mnogo veći ukoliko dođe do zastoja proizvodnje usljed pojave škarta i kolozije nego kod fleksibilnih tehnoloških sistema. U tom slučaju čitava jedna linija u proizvodnji trpi gubitke. Zato se sistemi praćenja i nadzora alata u većoj mjeri primjenjuju u visokoproduktivnim i visokoautomatizovanim obradnim tehnološkim sistemima.

Pored lako uočljivih prednosti jednog ovakvog sistema za praćenje i nadzor alata postoje i neki nedostaci kao što je cijena samog sistema a zatim i troškovi radne snage koja kontroliše i održava ove sisteme u proizvodnji.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aco Antić, vanr. prof.

2. PREDMET I CILJ ISTRAŽIVANJA

Sistemom nadzora i praćenja alata moguće je pratiti habanje alata sa svrhom da se alat zamjeni pravovremeno nakon što dostigne kritičnu tačku habanja. U cilju eksperimentalnog ispitivanja izvršice se mjerenje postojanosti i procesa habanja pločica, sa istom geometrijom a različitim radijusima zaobljenja vrha, presvlakama i lomačima. Cilj je da se utvrdi kako radijusi, presvlake i lomači utiču na sile tj. otpore rezanja.

Predmet istraživanja u ovom radu će se zasnivati na poređenju, postojanosti i sila rezanja pločica različitih geometrijskih karakteristika i njihovom testiranju u radnim uslovima. Eksperimentalnim ispitivanjem mjeriće se sile tj. otpori rezanja preko senzora koji su ugrađeni u glavno vreteno CNC struga. Uporedo sa mjerenjima otpora rezanja, posmatraće se promena rezne geometrije alata na alatnom mikroskopu, merenjem porasta bočnog habanja kao i kvalitet obrađene površine obratka.

3. SISTEMI ZA PRAĆENJE I NADZOR ALATA

Nadzor alata podrazumijeva nadgledanje odnosno praćenje njegovih dimenzija koje se mijenjaju prilikom habanja definišući životni vijek alata (TLM-Tool life monitoring) ili loma alata (TBR-Tool brucakage recovery) koji mogu uzrokovati razni faktori ili kolizija. Dakle nadzor alata vrši se na osnovu praćenja njegovog stanja a izvodi se sa merno-kontrolnim operacijama. Takve operacije se izvode u okviru obradnog sistema jer nema smisla vaditi alat povremeno iz mašine i vršiti mjerenja [1, 2].

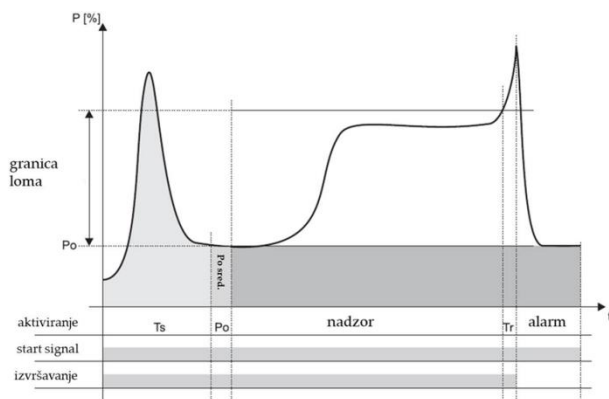
Kada je reč o TLM funkciji pri kojoj se prati veličina komponente otpora rezanja i u momentu dostizanja neke granične vrednosti konstatuje pohabanost alata, te aktivira potreba njegove izmene, bitno pitanje predstavlja baš poznavanje te granične veličine. Ona može biti zadata na osnovu proračuna ili eksperimentima utvrđene vrednosti, ali poteškoće koje sputavaju ovakva izvođenja nastaju zbog rada alata sa raznim elementima režima rezanja.

Za razliku od fiksno postavljene granice otpora rezanja kao kriterijuma identifikacije dopuštenog habanja alata, postoje rješenja kod kojih se ova granica automatski određuje na bazi praćenja porasta komponenti otpora rezanja i registrovane početne vrednosti sa novom oštricom alata.

Još složeniji slučaj nastaje kada je u pitanju TBR funkcija merno-kontrolnog sistema. Primenom ovih sistem omogućava se zadavanje gornje granične vrednosti čije preoračunje prouzrokuje generisanje signala o detektovanju loma alata. Problem koji se pri tome javlja jeste sama veličina granične vrednosti, s obzirom da alat vrši obradu

većeg broja zahvata sa raznim elementima režima rezanja praćenih različitim opterećenjima. Takođe je moguća pojava lokalnih odstupanja osobina materijala što uzrokuje povećanje otpora rezanja koja ne smeju biti identifikovana kao lom.

Otežavajuću okolnost predstavlja i mogućnost pojave loma alata bez povećanja otpora rezanja, već samo sa naglim padom, što se dešava kod alata sa keramičkim pločicama.



Slika 1: Standardni oblik signal nadzora alata [4]

Primenom sistema za nadzor u pogonu dobija se određena pomoć prilikom podešavanja. Sistem za nadzor se fokusira se na detalje i automatski reaguje na nastale greške. Takođe pruža mogućnost brzog reagovanja na nastale promene u proizvodnji kao i praćenje procesa obrade pomoću trenda. Pomoću ovih sistema moguće je i optimizovati proces obrade [5].

4. MJERENJE SILA REZANJA PRILIKOM OBRADJE ČELIKA POVEĆANE TVRDOĆE

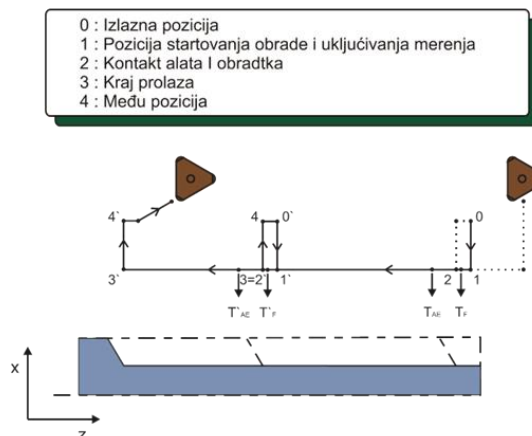
4.1 Postavka eksperimenta

Na osnovu postavljenog cilja rada izvršeno je eksperimentalno ispitivanje sila rezanja koje se javljaju prilikom obrade pločicama sa različitim: lomačima, radijusima, presvlakama i reznom geometrijom. Eksperiment je sproveden sa usvojenim režimima obrade na predmetu obrade kružnog poprečnog presjeka. Prilikom obrade koristile su se mogućnosti koje pruža fleksibilni tehnološki modul INDEX GU600 (CNC strug) odnosno mjerile su se sile rezanja pomoću senzora koji su instalirani na glavno vreteno mašine. Alati koji su korišteni u eksperimentu su pločice od tvrdog metala. Nakon prvog prolaza sa svakom od pločica izmeren je kvalitet obrađene površine tj. hrapavost koju iste daju. Zatim se pristupalo daljoj obradi i mjerenju sila rezanja. Planom eksperimenta je predviđeno da svaka pločica uradi tri prolaza te da se na osnovu ta tri prolaza ustanovi kolike su sile rezanja u svakom od njih, kvalitet obrađene površine i stepen pohabanosti pločice.

Iako se radi o malom broju prolaza, pretpostavlja se da će ipak doći do određenog habanja pločica jer se obrađuje materijal povećane tvrdoće (45HRC). Pločice od tvrdog metala nije preporučljivo koristiti za obradu materijala tvrdoće preko 50 HRC jer prilikom obrade materijala koji su tvrdi od te vrijednosti pločice se brzo habaju te se postavlja pitanje isplativosti. Na kraju predviđenog broja

prolaza pločice su postavljene na mikroskop kako bi se izmerilo habanje koje je nastalo prilikom obrade.

Proces obrade izvršen je na cilindričnom obradku stegnutom u steznu glavu i poduprt sa hidrauličnim konjićem. Obrada koja je izvođena je uzdužno struganje. Prečnik obrade je u opsegu od 80 do 45 mm a dužina jednog prolaza 100 mm.



Slika 2. Zahvati obrade pri izvođenju eksperimenta

Koristiće se CNMG pločice japanskog proizvođača Kyocera sa nekoliko različitih lomača, presvlakama i radijusa, tabela 1 [3].

Tabela 1. Specifikacija pločica koje se koriste u eksperimentu

Oblik	Opis	Dimenzije (mm)			
		Dužina rezne ivice	Debljina	Rupa	Radijus
	CNMG120408KH CA315	12,7	4,76	5,16	0,8
	CNMG120412KH CA315				1,2
	CNMG120408KG CA315	12,7	4,76	5,16	0,8
	CNMG120412KG CA315				1,2
	CNMG120408PG CA025P	12,7	4,76	5,16	0,8

Primjenjivaće se dva radijusa 08 i 12. Takođe primjenjuje se tri različita lomača: KH, KG i PG. Sa slike se lako uočava i razlika u boji presvlakke na pločicama. Iako su sve pločice sa slike presvučene CVD presvlakama, prve dvije pozicije sa KH i KG lomačima su presvučene malo drugačijom presvlakom koja je namjenjena obradi tvrdih materijala. Isto tako i lomači na tim pločicama su namjenjeni obradi tvrdih materijala. Prije svega namjenjeni su obradi livenih gvožđa koja sama po sebi imaju veću tvrdoću od sirovih čelika, samim tim su i pogodniji za obradu čeličnih materijala koju imaju veću tvrdoću tj. koji su termički obrađeni.

Posle odabira pločica za testiranje potrebno je odabrati i odgovarajući držač alata koji se montira na nosač alata. Bira se naravno držač koji odgovara CNMG geometriji i obliku pločice. Držač koji je odabran je kvadratnog poprečnog presjeka dimenzija 25x25mm. Nakon odabira odgovarajućeg držača pločica se steže na držač. Držač sa

pločicom se dalje postavlja i steže u odgovarajući nosač, odnosno nosač koji može da prihvati poprečni presek držača 25x25mm. Prije nego se taj nosač postavi na mašinu potrebno je izmjeriti korekcije alata sa nosačem na ZOLLER uređaju za podešavanje alata i izvršiti unos korekcija na upravljačkoj jedinici mašine na osnovu rezultata mjerenja kako ne bi došlo do kolizije prilikom obrade.

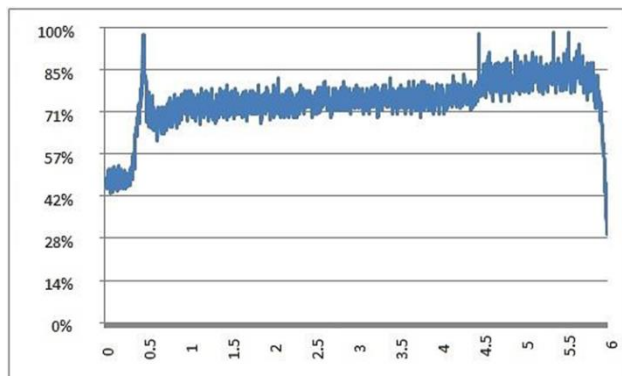
Nakon mjerenja alata u nosaču na Zoller mašini za mjerenje, alat se postavlja i montira na mašinu. Potom se sve neophodne informacije zajedno sa rezultatima mjerenja ubacuju u upravljački program putem Siemens upravljačke jedinice koja se nalazi na mašini. Prethodno je postavljen obradak na mašinu o kojem je već bilo riječi. Radi se o kaljenom obratku velike tvrdoće (45 HRC) kružnog poprečnog presjeka. Materijal obratka je čelik Č1730.

Tokom izvođenja eksperimenta vođeno je računa o izboru režima obrade kojima će se vršiti obrada. Obzirom da je materijal koji se obrađuje kaljeni čelik, te je samim tim jako tvrd, režimi su prilagođeni obradi za to stanje materijala koji se u bitnoj mjeri razlikuju u odnosu na obradu sirovih čelika. Brzina rezanja je obrnuto srazmjerna tvrdoći materijala. Što je materijal koji se obrađuje tvrdi to je brzina rezanja manja. U ovom slučaju se obrađuje materijal tvrdoće oko 45 HRC pa je potrebno rezati manjim brzinama. Međutim, zbog primjene optimizovanog eksperimenta u laboratorijskim uslovima, režimi obrade su u određenom procentu povećani kako bi se rezni alat brže habao. Odabrana je brzina od $V_c=160$ m/min. Kako ne bi dolazilo do trenutnog loma pločica pri ulasku u zahvat ili preopterećenja mašine definisana je dubina rezanja od $a = 2$ mm. Pomak koji je usvojen je sličan kao i za obradu sirovih čelika $f = 0.2$ mm/obrt.

4.2 Rezultati eksperimenta

Svaka od pločica u jednom prolazu prelazi put od 100 mm. Prilikom prvog prolaza sistem se obučava. Zatim se u drugom prolazu određuju granice na osnovu obučenog sistema i u trećem prolazu mašina samostalno radi i mjeri otpore rezanja uz prethodno obučavanje i određivanje granica. Kao što je već rečeno radi se o materijalu povećane tvrdoće od koga je sačinjen obradak pa je hrapavost odnosno srednje aritmetičko odstupanje profila R_a mjerena nakon prvog prolaza jer i nakon prvog prolaza postoji habanje pojedinih pločica.

CNMG120408PG CA025P



Slika 3. Sile rezanja za *CNMG120408PG CA025P*

Na slici 3 prikazuje grafik sile rezanja u toku vremena praćenja. Vertikalna osa grafika prikazuje vrijednosti sile rezanja u procentima. Vrijednost sile rezanja od 100% odgovara vrijednosti od 3,5 volta (analogni signal koji se šalje sa mašine izražen je u voltima i direktno srazmjern silama) jer vrijednosti sile rezanja nisu bile veće od onih koje su odgovarale vrijednosti analognog signala od 3,5 volti. Na osnovu toga će se dalje vršiti poređenja svake od pločica tako što će se sile rezanja prikazivati u procentima.

Konkretno sa grafika na slici 4 se vidi da sile rezanja koje pravi *CNMG120408PG* na početku imaju neke veće oscilacije. Te oscilacije postoje zbog ulaska alata u zahvat zbog čega se trenutno naglo destabilizuje proces rezanja formirajući veće oscilacije u silama rezanja. Nakon stabilizacije proces se ustaljuje i bilježe se neke prosječne vrijednosti sile od oko 71%. Vremenom te sile se povećavaju da bi pred sam izlaz alata iz zahvata imale prosječnu vrijednost od oko 84%.



Slika 4. Pločica *CNMG120408PG CA025P* nakon obrade

Slika 4.3 prikazuje pohabanost pločice u velikoj mjeri i to samo posle tri prolaza. To je razlog zašto su se sile rezanja tako brzo i drastično povećavale. Krzanje se uočava i na leđnoj i na grudnoj površini kao i promjene u boji pločice što nam govori da su termalna opterećenja bila velika.

Desni deo slike prikazuje izražen pojas habanja kao parametar koji je praćen prilikom određivanja pohabanosti pločice. Na samom početku rezanja sa grafika koji nije prikazan vidi se da su prosječne sile rezanja na oko 32% što i jeste logično jer ova pločica ima oštriju reznju ivicu i radijus 08.

CNMG120408KG CA315

Srednje aritmetičko odstupanje profila koje je izmjereno posle prvog prolaza ove pločice je $R_a=5,264\mu\text{m}$. Dakle kvalitet obrađene površine je lošiji u odnosu na kvalitet postignut sa prethodnom pločicom što se i očekivalo obzirom da ova pločica ima tuplju reznju ivicu kao i manji grudni ugao. Prosječna vrijednost sile se kretala oko 47% od postavljene granice opterećenja što je manje u odnosu na prethodni tip pločice.

Veliki uticaj na dobijanje ovakvih rezultata sile rezanja ima geometrija pločice koja je drugačija u odnosu na prethodnu pločicu u smislu da ima manje oštru reznju ivicu i da je vrijednost grudnog ugla manja nego kod prethodne pločice.

Ova pločica se zbog lomača koji ne odgovara ovradi ovog materijala polomila usljed prevelikog termalnog opterećenja.

CNMG120412KG CA315

Prosječna vrijednost sila rezanja koju je imala ova pločica je oko 76%. Kao i prethodna pločica radila je dosta stabilnije za razliku od prve pločice sa kojom se obrađivalo. Međutim baš kao i prethodna pločica lomač strugotine nije bio pogodan za ovu obradu pa je i ovde usljed pregrijavanja pločice i promjene strukture došlo do loma. Sa ovom pločicom je postignut najlošiji kvalitet obrađene površine što se i očekivalo jer je rezna ivica tuplja u odnosu na rezu ivicu na prvoj pločici a ima i radijus veći od obje prethodne pločice. Izmereno srednje aritmetičko odstupanje profila je $Ra=6,129\mu m$.

CNMG120408KH CA315 – Pločica je imala prosječnu vrijednost sila rezanja od oko 36% od maksimalnih vrijednosti. Sile rezanja koje prouzrokuje ova pločica su nešto veće u odnosu na početne vrijednosti sile rezanja koje je imala pločica sa oštrijom reznom ivicom u kvalitetu CA025P ali i manje u odnosu na sile rezanja koje je dala pločica sa KG lomačem u istom kvalitetu (CA315).

CNMG120412KH CA315 - Prosječne vrijednosti sila rezanja sa ovom pločicom su oko 43%. Dakle pločica sa radijusom 12 je kao što je i bilo očekivano uzrokovala pojavu većih sila u odnosu na istu pločicu sa radijusom 08. Pločice sa KH lomačem su imale daleko bolje odvođenje strugotine a samim tim i toplote u zoni rezanja pa se nisu imale već su čak imale i drastično bolju postojanost.

5. ZAKLJUČAK

Savremeni način proizvodnje nameće potrebu korištenja sistema za nadzor i praćenje alata jer su optimizacija procesa obrade i ušteda na vremenu sve bitniji segmenti proizvodnje i pridaje im se sve veći značaj. Ukoliko ovakvi sistemi figurišu u proizvodnji moguće je na adekvatan način pratiti postojanost alata. Takođe, dimenzionalne karakteristike i kvalitet obrađene površine kao bitan segment zahtjevanog kvaliteta proizvoda direktno je povezan za stepenom pohabanosti alata u smislu da se alat treba zamjeniti kada ne daje zahtjevani kvalitet što direktno utiče na smanjenje broja škart komada.

Mjerenje sila odnosno otpora rezanja sprovedeno je sa ciljem primjene rezultata mjerenja u proizvodnji.

Uporedo sa silama rezanja mjerena je i hrapavost. Na osnovu rezultata mjerenja sila rezanja i hrapavosti se lako zaključuje da posotji zavisnos između sila, hrapavosti obrađene površine i pohabanosti alata. Dakle sile rezanja su direktno srazmjerne stanju rezne geometrije alata što se vidi na osnovu kvaliteta obrađene površine odnosno srednjeg aritmetičkog odstupanja.

Pločica u kvalitetu CA025P sa radijusom 08 je imala najoštriju rezu ivicu i proizvodila je najmanje sile rezanja na samom početku eksperimenta (pločica se posle brzo habala pa su sile zbog toga bile veće) kao i najbolji kvalitet obrađene površine odnosno najmanje srednje aritmetičko odstupanje profila.

Međutim, pločica se brzo pohabala, došlo je do drastičnog povećanja sila rezanja a samim tim i do lošijeg kvaliteta obrađene površine.

Pločica sa radijusom 08 i lomačem KG u kvalitetu CA315 je proizvodila veće sile rezanja i nešto lošiji kvalitet obrađene površine.

Pločica sa istim lomačem i kvalitetom ali sa radijusom 12 je proizvodila najveće sile rezanja. Kvalitet obrađene površine je srazmjerno tome bio najlošiji uzimajući u obzir ovu i prethodno spomenute dvije pločice.

6. LITERATURA

- [1] Milikić D., Gostimirović M., Kovač P., : „*Osnove tehnologije obrade rezanjem*“ Fakultet tehničkih nauka, Novi Sad, 2013.
- [2] Kovač P., „*Teorija obradnih procesa*“ Fakultet tehničkih nauka, Novi Sad, 2010
- [3] Kyocera katalog 09.10.18': <https://brochure.kyocera-unimerc.com/Indexablecuttingtools/SingleLanguageBrochures/UK/general-catalog-2016-2017/#/>
- [4] Patrick De Vos, Jan-Eric Stahl: „*Metal cutting theories in practice*“
- [5] Antić, A.: „*Prepoznavanje stanja pohabanosti alata za obradu rezanjem primenom neuro-fazi klasifikatora*“, *Doktorska disertacija*, Fakultet tehničkih nauka, Novi Sad, 2010“

Kratka biografija:



Nemanja Klještan rođen je u Vlasenici 1994 god. Diplomski master rad na Fakultetu tehničkih nauka iz oblasti proizvodnog mašinstva odbranio je 2018.god. kontakt: kljesta@hotmail.com

PROJEKAT REKONSTRUKCIJE KOTLARNICE I TEHNIČKO REŠENJE GREJANJA I KLIMATIZACIJE MANIPULATIVNE ZGRADE NA NAFTNOJ SABIRNOJ STANICI**PROJECT OF RECONSTRUCTION OF THE BOILER ROOM AND TECHNICAL SOLUTION FOR HEATING AND AIR CONDITIONING OF MANIPULATIVE BUILDING ON OIL COLLECTION STATION**

Josip Horvat, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Radom su obuhvaćene teorijske osnove grejanja i klimatizacije kao i proračuni svih elemenata vezanih za ovu oblast kao i praktična provera zamenjenog kotla, provera postojećeg dimnjaka zahtevima novo instalisanog kotla, kao i tehničko rešenje grejanja i klimatizacije manipulativne zgrade na naftnoj sabirnoj stanici uz pomoć toplotne pumpe „vazduh – voda“.

Ključne reči: Rekonstrukcija, kotlarnica, toplotna pumpa, grejanje i klimatizacija, sabirna naftna stanica

Abstract – The paper deals with theoretical basics of heating and air conditioning as well as calculations of all elements related to this area as well as practical check of the replaced boiler, checking the existing chimney with the requirements of the newly installed boiler, as well as the technical solution for the heating and air conditioning of the manipulative building on the oil collection station with the help of a heat pump "air - water".

Ključne reči: reconstruction, heat pump, heating and air conditioning, oil collection station

1. UVOD

Radom je obuhvaćeno tehničko rešenje rekonstrukcije postojeće gasne toplovodne kotlarnice za grejanje građevinskih objekata, procesnih aparata i zamenu dotrajalog kotla, kao i tehničko rešenje grejanja i klimatizacije manipulativne zgrade uz pomoć toplotne pumpe „vazduh-voda“ na naftnoj sabirnoj stanici investitora, čija osnovna delatnost je proizvodnja i prerada nafte. Prvi deo rada sadrži teorijske osnove.

U njima je se nalazi metodologija proračuna kasnije primenjenih toplotnih gubitaka građevinskih objekata i potrošača, dimnjaka i cevne mreže, kao i teorijske osnove toplotne pumpe.

Drugi deo rada sadrži proveru snage novog kotla, kontrolni proračun dimnjaka, proračun toplotnih gubitaka i dobitaka građevinskih objekata, cevne mreže, odabir toplotne pumpe „vazduh-voda“ i razmeštaj grejnih / rashladnih tela po građevinskim objektima.

2. ZADATAK RADA

Za potrebe investitora čija osnovna delatnost je proizvodnja i prerada nafte potrebno je izraditi mašinski projekat koji treba da obradi termo-tehničke instalacije na jednoj naftnoj sabirnoj stanici. Grejni konzum sastoji se od građevinskih objekata, rezervoara za naftu i grejača procesne opreme.

Projekat treba da sadrži:

- Rekonstrukcija postojeće toplovodne gasne kotlarnice dodavanjem jednog ogranka za grejanje procesnih aparata i provera novo-zamenjenog kotla uz proveru postojeće instalacije za odvod produkata sagorevanja.
- Toplovodni razvod do svih elemenata konzuma
- Instalacija klimatizacije manipulativne zgrade uz korišćenje toplotne pumpe „vazduh-voda“

Napomena: Tehnološki potrošači i dovod gasa do kotla nisu predmet ovog projekta.

3. ANALIZA TERMIČKIH POTREBA OBJEKTA MANIPULATIVNE ZGRADE

Za objekat manipulativne zgrade uz poznate arhitektonske podloge sa sastavom pregrada izvršen je proračun građevinske fizike. Proračun gubitaka i dobitaka toplote dao je podatke o termičkim potrebama objekta u zimskom i letnjem periodu. Za zimski režim usvojena je spoljašnja projektna temperatura od -18°C , dok su unutrašnje projektne temperature prostorija usvajane u skladu sa namenom prostorije. Za letnji režim usvojena je spoljašnja projektna temperatura od $+34^{\circ}\text{C}$, dok su unutrašnje projektne temperature svih klimatizovanih prostorija $+26^{\circ}\text{C}$.

4. USVOJEN SISTEM GREJANJA I KLIMATIZACIJE ZA MANIPULATIVNU ZGRADU

Iz proračuna za toplotne gubitke (dobitke) usvajaju se grejna/rashladna tela tako što za svaku prostoriju u katalogu proizvođača pronađemo odgovarajući „FAN COIL“ na osnovu gubitaka/dobitaka, usvajamo veću vrednost grejnog/rashladnog kapaciteta.

Iz proračuna su usvojeni „FAN COIL“ uređaji proizvođača „EDEN“:

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Momčilo Spasojević.

1. WIND 10, toplotnog kapaciteta 1246W/
rashladnog kapaciteta 865W;
2. WIND 30, toplotnog kapaciteta 2586W/
rashladnog kapaciteta 2072W;
3. WIND 40, toplotnog kapaciteta 3279W/
rashladnog kapaciteta 2530W;

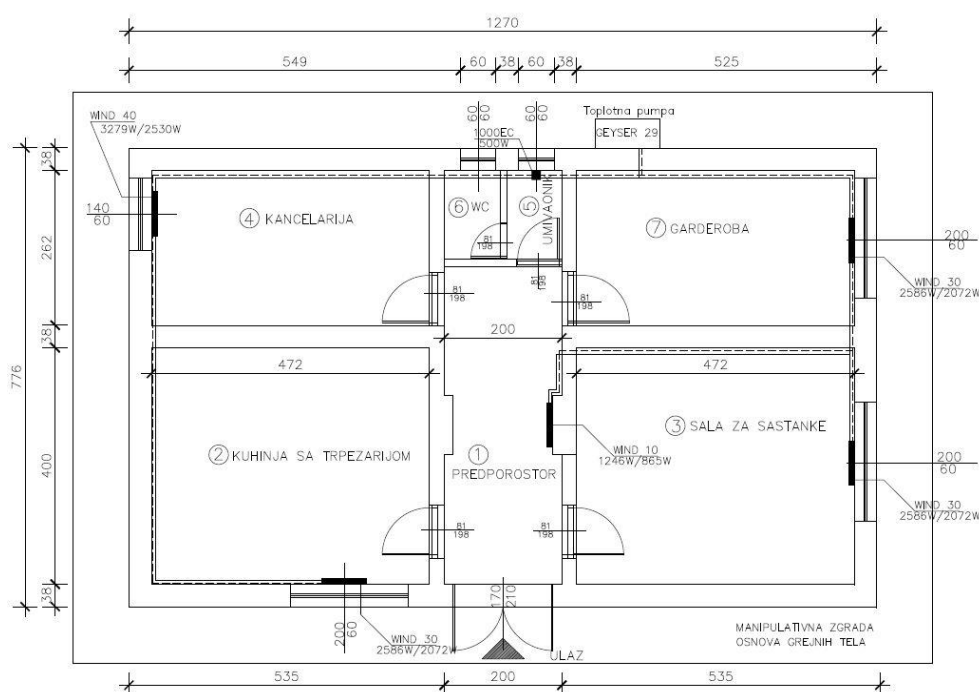
WIND 10 i WIND 30 uređaji su usvojeni prema potrebnom toplotnom grejnom kapacitetu, dok je WIND 40 usvojen prema toplotnom rashladnom kapacitetu zbog ostalih električnih uređaja koji se u prostoriji nalaze. Cevna mreža koja povezuje „FAN COIL“ uređaje sa toplotnom pumpom mora biti zaštićena termičkom izolacijom sa integrisanom parnom branom.

Kondenzat iz „FAN COIL“ jedinica se vodi do najbliže olučne vertikale u posebnom kanalu koji se nalazi u podu prostorije neposredno pored zida.

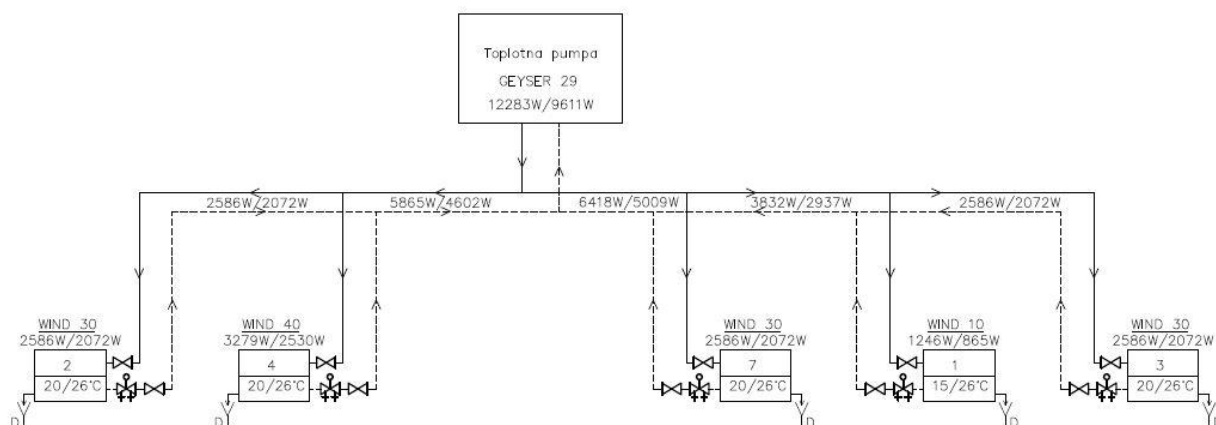
Za potrebe grejanja neklimatizovanih sanitarnih prostorija usvojen je mali električni grejač koji pokriva gubitke u istim prostorijama.

Nakon što smo usvojili grejna/rashladna tela sabiramo njihove toplotne grejne/rashladne kapacitete i dobijamo da je ukupni instalisani kapacitet uređaja koje toplotna pumpa treba da snabde toplotnom/rashladnom energijom:

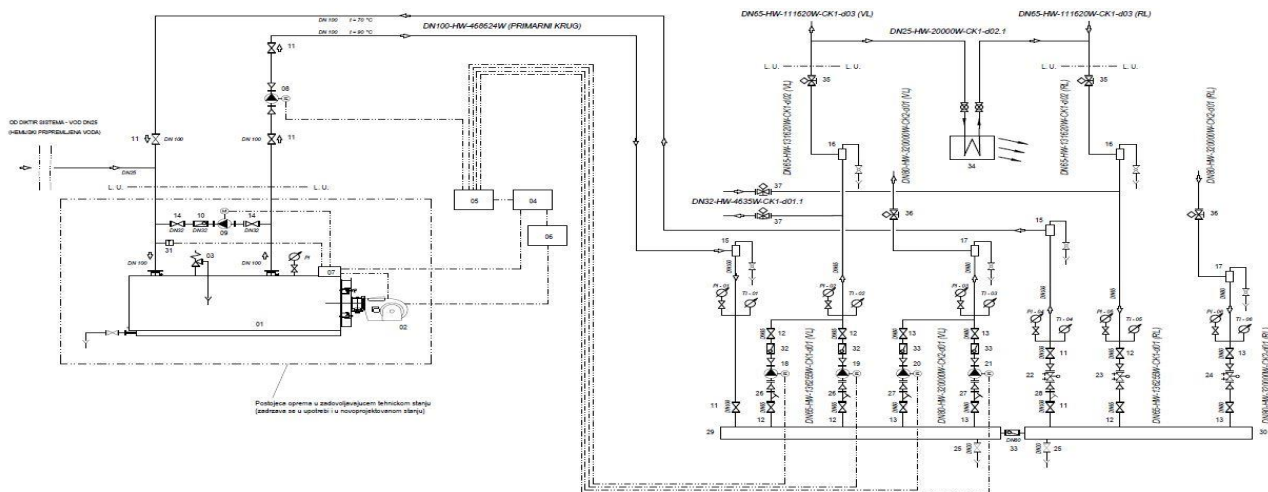
- 12,95 kW toplotnog kapaciteta
- 29,22 kW rashladnog kapaciteta



Slika 1. Osnova manipulativne zgrade



Slika 2. Šema opreme za grejanje i klimatizaciju manipulativne zgrade



Slika 3. Šema povezivanja toplovodne instalacije

Iz prethodnog se usvaja toplotna pumpa proizvođača "BLUE"BOX" tip GEYSER 29. Toplotna pumpa je morala biti u izvedbi "VAZDUH-VODA" usled nemogućnosti bušenja ili kopanja zbog podzemnih cevni instalacija koje su postojeće na naftnoj sabirnoj stanici.

5. TOPLOVODNA INSTALACIJA GREJANJA NA NAFTNOJ SABIRNOJ STANICI I REKONSTRUKCIJA GASNE KOTLARNICE

5.1. Toplotna instalacija u okviru stanice

Za poznati instalisani grejni kapacitet elementa toplotnog konzuma ovim tehničkim rešenjem predviđena je toplovodna cevna mreža. Razvod je dominantno nadzemni vođen sa izolacijom od mineralne vune u plaštu od Al lima. U najvišim tačkama je predviđeno odzračivanje, a u najnižim tačkama drenaža.

5.2. Rekonstrukcija toplovodne instalacije kotlarnice

Postojeća instalacija grejanja na Sabirnoj Stanici kao izvor toplotne energije koristi zidanu gasnu kotlarnicu sa kotlom termičke snage 500 KW. Kotao je postavljen pre nekoliko meseci, da bi zamenio prethodni kotao termičke snage 350 KW.

Rekonstruisani toplovodni razvod je koncipiran u vidu dva grejna ogranka (CK-01 i CK-02), za koje su u prostoru kotlarnice postavljene po dve cirkulacione pumpe (radna i rezervna). Hidroforsko postrojenje, HPV i diktir sistem su postojeća oprema i u zadovoljavajućem tehničkom stanju, pa će biti zadržani i u okviru novoprojektovane instalacije.

Predviđena je zaštita opreme u kotlarnici za teorijski slučaj havarije u konstrukciji procesnih grejača u kojima je pritisak procesnog fluida značajno viši od pritiska grejne vode - što bi moglo da implicira prodor ugljovodonika u toplovodnu instalaciju.

Za sprečavanje neželjenih posledica tog scenarija, na toplovodnim cevima (i potis i povrat) krugova CK-01 i CK-02 predviđeni su blokadni ventili koji će biti locirani u kotlarnici, odnosno izvan prostora kotlarnice (u neposrednoj blizini).

Blokadni ventili treba da zatvore i onemoguće protok tople vode (odnosno mešavine tople vode i ugljovodonika, u havarijskom slučaju) već na pritisku od 3 bar, a podešavanje tog pritiska blokade se može i naknadno menjati u toku eksploatacije.

5.3. Provera postojeće instalacije za odvod dimnih gasova

Odvod dimnih gasova u atmosferu biće ostvaren pomoću termički izolovanog čeličnog samostojećeg dimnjaka prečnika 400 mm i efektivne visine 8 m (postojeći). S obzirom da je predmetni dimnjak bio prethodno povezan na kotao kapaciteta 350 KW, bilo je neophodno proveriti mogućnost da se zadrži postojeći dimnjak.

Za maseni protok dimnih gasova koji odgovara radu kotla pri maksimalnom kapacitetu proračunat je pad pritiska pri strujanju kroz dimnjaču i dimnjak, kao i uzgon dimnjaka. S obzirom da je uzgon dimnjaka veći od otpora strujanja, zaključak je da postojeći dimnjak odgovarajući i za novoprojektovane uslove.

5.4. Hidraulički proračun toplovoda

Ulazni podatak za usvajanje cirkulacionih pumpi je rezultat hidrauličkog proračuna. Za poznati konzum i režim 90/70°C izračunati su protoci tople vode po deonicama i sračunat je ukupan otpor cevovoda, koji obuhvata pad pritiska usled trenja i pad trenja usled lokalnih otpora.

6. ZAKLJUČAK

Iz primernog projekta možemo zaključiti da novozamenjeni kotao odgovara potrebama postojećeg i novoprojektovanog toplotnog konzuma sa rezervom u slučaju proširenja toplotnog konzuma. Takođe postojeći dimnjak je zadovoljavajući i neće biti nikakvih smetnji pri radu novozamenjenog kotla.

Dobrim izborom prečnika toplovoda i izborom odgovarajućih cirkulacionih pumpi stvoreni su svi preduslovi za pravilan rad sistema grejanja.

Problem grejanja i klimatizacije manipulativne zgrade je rešen uz pomoć „FAN COIL“ uređaja i toplotne pumpe koji će zimi grejati, a leti hladiti manipulativnu zgradu. Manipulativna zgrada je jedini objekat na stanici koji ima potrebe za hlađenjem u letnjem periodu, te je zbog toga usvojeno takvo tehničko rešenje.

Toplotna pumpa je u izvedbi „Vazduh-Voda“ što je bilo jedino tehnički moguće zbog mnogobrojnih cevnih instalacija (što bi onemogućavalo bilo kakvu vrstu kopanja ili bušenja tla). S obzirom da je toplotna pumpa jedini generator toplote u manipulativnoj zgradi u zimskom periodu, uređaj je izabran da može da zagreje i pri veoma niskim spoljašnjim temperaturama.

Na ovaj način će se smanjiti troškovi grejanja i klimatizacije za manipulativnu zgradu iz razloga što je povoljan odnos dobijene rashladne/toplotne energije od uređaja u poređenju sa potrošenom električnom energijom.

7. LITERATURA

- [1] Grejanje i klimatizacija sa primenom solarne energije; S. Zrnić, Ž. Čulum; GRO „KULTURA“, Beograd
- [2] Centralno grejanje principi i praksa; J. Danon; NIRO „TEHNICKA KNJIGA“, Beograd
- [3] Projektovanje postrojenja za centralno grejanje; B. Todorović; MAŠINSKI FAKULTET, Beograd
- [4] Princip projektovanja sistema grijanja; E. Kulić; SMEITS, Beograd
- [5] Grejanje i klimatizacija sa primenom potrošne vode i rashladnom tehnikom; H. Reknagel, E. Šprenger; IRO „GRAĐEVINSKA KNJIGA“, Beograd
- [6] *** Technical catalogue geyser; BLUEBOX
- [7] *** Technical catalogue WIND FAN COIL; EDEN;
- [8] *** <https://www.wilo-select.com>; WILO

Kratka biografija:



Josip Horvat rođen je u Novom Sadu 1990. god, gde je završio osnovnu i srednju školu. Diplomirao je na Fakultetu tehničkih nauka, studijski program Mašinstvo, smer Energetika i procesna tehnika. Zaposlen je u kompaniji “Naftna Industrija Srbije”.

PODVOodne AKUSTIČNE SENZORSKE MREŽE UNDERWATER ACOUSTIC SENSOR NETWORKS

Dejan Radić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu, fokusirali smo se na jedinstvene karakteristike akustičnog komunikacionog kanala. Proučavali moguće napade i kontramere komunikacionih protokola u PASM i predloženi SDAM prototip.

Ključne reči: PASM, sigurnosni protokoli, SDAM prototip.

Abstract – In this paper we have focused on the unique characteristics of the acoustic communication channel, studied possible attacks and countermeasures of communication protocols in UASNs, and proposed SDAM prototype.

Keywords: UASNs, secure communication protocols, SDAM prototype.

1. UVOD

Tokom proteklih nekoliko godina, došlo je do brzog rasta istraživanja na podvodnim akustičnim senzorskim mrežama (PASM) zbog njihove široke primene u mnogim podvodnim scenarijima, uključujući okeanografsko prikupljanje podataka, asistent navigacije, praćenje zagađenja, istraživanje u blizini obale, prevencija katastrofa i tako dalje.

U većini od gore navedenih aplikacija, PASM su raspoređene bez nadzora, pa čak i u neprijateljskim okruženjima, tako da se mora uzeti u obzir bezbednost komunikacije među senzorskim čvorovima.

Tako u ovom radu, govorimo i o bezbednosnim problemima za PASM.

Podvodne mreže senzora omogućavaju primene čiji su potencijali još nedovoljno istraženi i da poboljšaju našu sposobnost posmatranja i predviđanja velikih vodenih površina. Ključni izazov u dizajniranju proizilazi iz karakteristika podvodnog akustičnog kanala, koji pokazuje veliko slabljenje, visok nivo šuma, odbijanje talasa, veliko i uz to i promenljivo kašnjenje, kao i Doplerov efekat.

Pored toga, izazovi prostiranja u podvodnom okruženju rezultiraju vremenski i prostorno različitim koeficijentima podvodnih kanala, što iziskuje veoma specifičan dizajn protokola.

NAPOMENA:

Ovaj rad predstavljamo iz master rada čiji mentor je dr Dragana Bajić, red. prof.

2. PODVOodne AKUSTIČNE SENZORSKE MREŽE

Akustične komunikacije su tipična tehnologija fizičkog sloja u podvodnim mrežama. U stvari, radio talasi se prostiru na velikim udaljenostima kroz provodljivu slanu vodu samo na vrlo niskim učestanostima (30-300 Hz), što zahteva velike antene i veliku prenosnu snagu.

Optički talasi nemaju tako veliko slabljenje ali zato imaju veliku disperziju.

Osim toga, prenos optičkih signala zahteva visoku preciznost zbog uskih laserskih zraka. Zato se linkovi u podvodnim mrežama obično zasnivaju na akustičnoj bežičnoj komunikaciji.

Tabela 1. Tipični propusni opsezi podvodnih kanala [1].

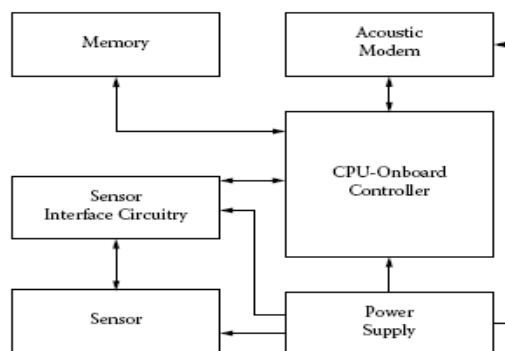
	Domet [km]	Opseg [kHz]
Veoma velik	1000	< 1
Velik	10 do 1000	2 do 5
Srednji	1 do 10	≈ 10
Mali	0.1 do 1	20 do 50
Veoma mali	< 0.1	> 100

Tabela 1 prikazuje propusne opsege podvodnog akustičnog kanala za različite opsege. Akustični linkovi su otprilike klasifikovani kao vertikalni i horizontalni, prema pravcu zvučnog zraka (talasa) u odnosu na morsko dno.

Njihove karakteristike prostiranja se značajno razlikuju, posebno u vezi sa vremenom disperzije, višestrukom putanjom širenja i varijacijom kašnjenja.

2.1. Podvodni senzori

Tipična interna arhitektura podvodnog senzora je prikazana na slici 1.

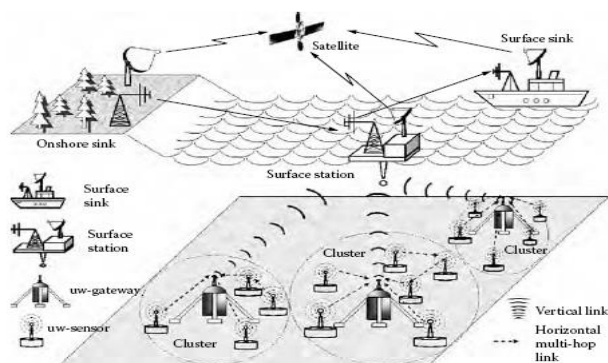


Slika 1. Unutrašnja organizacija podvodnog senzorskog čvora [2]

Sastoji se od glavnog kontrolera/procesora, koji se povezuje sa jednim okeanografskim instrumentom ili senzorom kroz senzorsko kolo. Kontroler prima podatke iz senzora i može da ih sačuva u internu memoriju, procesira i pošalje ih na drugi mrežni uređaj, kontrolišući akustični modem.

2.2. Komunikaciona arhitektura

Referentna arhitektura za 2D podvodne mreže je prikazana na slici 2. Grupa senzorskih čvorova je usidrena na dnu okeana. Podvodni senzorski čvorovi su povezani sa jednim ili više podvodnih mrežnih prolaza putem bežičnih akustičnih linkova. Podvodni mrežni prolazi su mrežni uređaji koji su zaduženi za prenos podatka iz mreže sa morskog dna na površinsku stanicu.



Slika 2. 2D podvodne senzorske mreže [2]

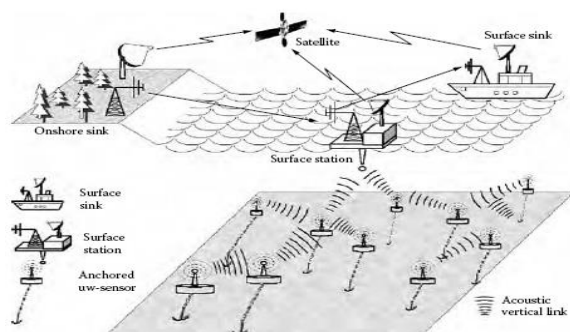
Da bi se to postiglo, opremljeni su sa dva akustična primopredajnika, odnosno vertikalnim i horizontalnim primopredajnikom.

Horizontalni primopredajnik se koristi od strane podvodnih mrežnih prolaza da komunicira sa senzorskim čvorom u cilju:

- slanja komandi i konfiguracijskih podataka do senzora (podvodni mrežni prolaz do senzora) i
- prikuplja podatke monitoringa (senzori do podvodnog mrežnog prolaza).

Vertikalni link se koristi od strane podvodnih mrežnih prolaza da prenesu podatke na površinsku stanicu.

Trodimenzionalne podvodne mreže se koriste za otkrivanje i posmatranje pojava koje se ne mogu adekvatno posmatrati pomoću senzorskih čvorova na morskome dnu, tj. za obavljanje kooperativnog uzorkovanja u 3D okeanskom okruženju.



Slika 3. 3D podvodne senzorske mreže [2]

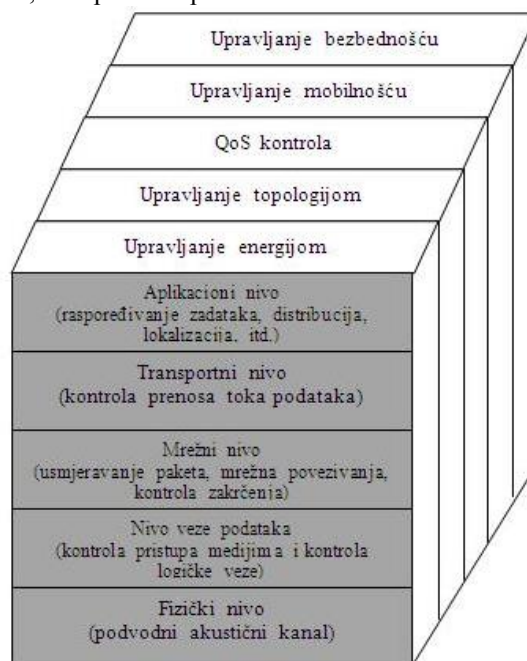
U 3D podvodnim mrežama, senzorski čvorovi plutaju na različitim dubinama. U ovoj arhitekturi, datoj na slici 3, svaki senzor je usidren na morsko dno i opremljen plutajućom bovom koja se može naduvati pumpom. Bova gura senzor na površinu okeana. Dubina senzora reguliše se podešavanjem dužine žice koja povezuje senzor za sidro pomoću elektronski kontrolisanog motora koji se nalazi na senzoru.

2.3. Sigurna komunikacija za podvodne senzorske mreže

Akustična komunikacija ima nekoliko loših osobina, kao što je nizak propusni opseg, veliko kašnjenje i veliku verovatnoću greške po bitu. Pored toga, kvalitet veza u akustičnom kanalu je loš, uglavnom zbog višeputnih prostiranja i vremenske varijabilnosti medijuma. Pored gore navedenih tačaka, u PASM, svi senzorski čvorovi slobodno plutaju sa okeanskom strujom, što 3D mrežnu konfiguraciju čini dinamičnom. Stoga velika koncentracija čvorova nije pogodna mrežna struktura.

Pokušaji zlonamernih napada mogu se svrstati u dve grupe: napade na senzorske čvorove i napade na mrežne protokole. Prva vrsta zlonamernog napada je najefikasniji metod da se ošteti PASM. Međutim, ova metoda nije praktična u realnim aplikacijama, pošto, kao što je rečeno, podvodni senzorski čvorovi nisu gusto raspoređeni: teško je da se istovremeno uništi nekoliko čvorova. Daleko si češći napadi na mrežne protokole, posebno na komunikacione protokole. Kada su komunikacioni protokoli razbijeni, cela mreža je beskorisna.

Kao što je prikazano na slici 4, mrežna arhitektura PASM se sastoji od pet nivoa: fizičkog nivoa, nivoa podataka, mrežni, transportni i aplikacioni nivo.



Slika 4. Mrežna arhitektura PASM

Najčešći zlonamerni napad na fizičkom sloju je *jamming* napad, što je popularan tip napada, odbijanja usluge (Denial of Service-DoS) napada. U *jamming* napadu, zlonamerni čvor koji pokušava da prekine komunikaciju

između senzorskih čvorova se naziva jammer. *Jammer* šalje mnogih beskorisnih signala na istom frekvencijskom opsegu. Pošto je frekvencijski opseg akustične komunikacije uzak (od herca do kiloherca), PASM su veoma ranjive na *jamming* napade.

Stoga je u [3] predložen protokol za detekciju podvodnog Jamming-a (UWJDP UnderWater Jamming Detection Protocol) koji otkriva i ublažavan *jamming* napad. Pretpostavlja se da se jamming napad vrši ubacivanjem velikog broja zlonamernih paketa u mrežu. Postoje tri faze u UWJDP, detekcija suseda, *jamming* detekcija i mapiranje *jammed* oblasti.

U fazi *jamming* detekcije, za detekciju se koriste odnosi poslatih i isporučenih paketa (PSR -packet send ratio i PDR - packet delivery ratio) kao i količina utrošene energije (ECA - energy consumption amount). U [4], efekti napada na PASM se proučavaju na osnovu testova u stvarnom okruženju, gde su zlonamerni čvorovi svrstani u dve kategorije: jednostavni (Dummi signal) *jammer* i pametni (obmanljiv) *jammer*.

Prva vrsta jammera ne zna ništa o mreži i koristi šum da ometa akustični komunikacijski kanal.

Druga vrsta jammera zna protokol mreže i može da se pretvara da je legitimni čvor za pokretanje zlonamernih napada. Mrežne performanse pod *jammer* napadom se simuliraju, a rezultati simulacije pokazuju da *jamming* napadi na PASM mogu lako da se pokrenu, što može drastično da degradira performanse mreže.

Rad u [4] daje nova saznanja za istraživače za dalja izučavanja *jamming* napada. Pored već pomenutih zlonamernih šema *jamming*-a tu je i prijateljski *jamming*, koji se može koristiti za poboljšanje bezbednosti mreže.

Na primer, u [5], autori predlažu osiguranu podvodnu komunikacijsku šemu baziranu na kooperativnom prijateljskom *jamming*-u, koji je jamming posredstvom analognog mrežnog kodovanja (Jamming through Analog Network Coding- J-ANC).

Za razliku od konvencionalnih kooperativnih zaštićenih šema od *jamming*-a, koja unose veštački šum kao izvore *jamming*-a, J-ANC upotrebljava isti kod širenja upotrebom legitimnog linka.

Paket prenesen prijateljskim *jammer*-om poznat je legitimnom čvoru, ali ne i prislušivaču. Zbog toga, nakon dobijanja ometanog paketa, legitimni čvor može ispravno dekodovati primljeni paket, dok prislušivač ne može.

2.4. Softverski definisane podvodne akustične mreže

Nedostatak sporazuma standardizacije za PASM nameće dodatne prepreke u dizajnu rekonfigurabilne mreže. Prostorne i vremenske varijacije podvodnih kanala zahtevaju rekonfiguraciju komunikacijskih parametara podvodnih uređaja kako bi obezbedili stabilan rad u smislu verovatnoće bitske greške (*bit error rate* BER) i verovatnoće paketske greške (*packet error rate* PER). Trenutno, komercijalni modemi za adaptaciju adresa, zbog varijacije adresa kanala, koriste unapred definisane načine. Takva rešenja imaju nedostataka:

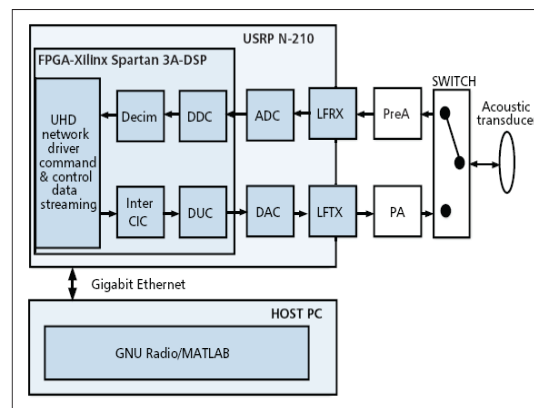
- spособnost da se prebaci u realnom vremenu među konačan broj operativnih modova
- mehanizmi donošenja odluka, da odluči i primeni adaptaciju i

- spособnost da se dinamički prilagođavaju svim mogućim okruženjima zbog konačnog broja operativnih modova.

Shodno tome, postoji potreba za podvodne uređaje koji mogu:

- olakšati raspoređivanje i testiranje novih protokola
- premosti jaz između različitih mrežnih uređaja i protokola za rešavanje problema interoperabilnosti u heterogenim PASM.

U [6] je opisan softverski definisan akustični modem (SDAM) prototip za postojeće podvodne komunikacije, na osnovu komercijalnih dostupnih komponenti.



Slika 5. Hardverska arhitektura predloženog SDAM prototipa [6]

SDAM je kao i njegovo jezgro SDR povezan sa širokopojasnim akustičnim pretvaračem preko pojačala snage /pretpojačavača. SDAM iskorišćava jedinstvene sposobnosti i karakteristike SDR da ispunji potrebu za fleksibilnošću lako rekonfigurabilnih UAN uređaja [6].

Predloženi SDAM zasniva se na USRP N210, koji je komercijalno dostupna FPGA- bazirana, SDR platforma. Za rad su odabrane LFTKS i LFRKS *daughterboards* (DC -30 MHz), koji omogućavaju razvoj polu-dupleks primopredajnika koji rade u frekventnom opsegu odabranog omnidirekcionog akustičnog pretvarača, Teledine Reson TC4013, od 1 Hz do 170 kHz.

Kako bi se poboljšao opseg komunikacije SDAM, korišteno je linearno širokopojasno pojačalo snage (PA), Benthovave BII-5002 i predpojačalo napona (PreA), TELEDINE Reson VP2000.

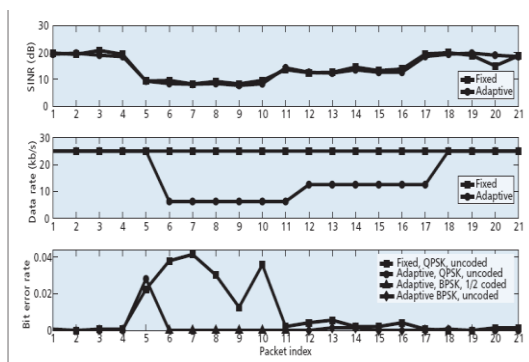
Ugrađen je elektronski prekidač, Mini-Circuits ZKS80-DR230 + da omogući rad jednog akustičnog transduktora kao predajnika i prijemnika u vremenskoj podeli dupleks modu (načinu).

Algoritmi za obradu i protokoli, signala u osnovnom opsegu, uglavnom su implementirani u glavnom računaru (Host PC), koji je povezan sa USRP N210 Gigabitnim Ethernetom (Ethernet) (GigE).

2.5. Eksperimenti

Interesantno je pokazati do kakvih eksperimentalnih rezultata se došlo prilikom testiranja prototipa SDAM. U U [6] su testirali adaptacione mogućnosti u realnom vremenu.

Slika 6 prikazuje eksperimentalnu brzinu prenosa podataka i evaluaciju BER rezultata za fiksne i adaptivne šeme.



Slika 6. Poređenje adaptivne sa fiksnim (ne-adaptivnim) šemama u smislu brzine prenosa podataka i BER za različite SINR profile [6].

Modulaciona šema i kodni količnik menjaju se prema algoritmu čiji je cilj maksimalizacija brzine prenosa podataka za zadati BER. Na slici 6 BER prag je empirijski postavljen na 10^{-3} , dok SINR varira između 10 i 20 dB. U obe, adaptivne i fiksne (ne-adaptivne) šeme, SDAM, emitovanje počinje najvećom mogućom brzinom prenosa podataka. U neadaptivnim šemama brzina ostaje konstantna ali BER prelazi dozvoljeni prag. S druge strane, u adaptivnoj šemi SDAM prilagođava modulaciju i kodni količnik u realnom vremenu sa smanjenjem SINR. Kao rezultat, brzina prenosa se podešava na nižu vrednost da se zadovolje unapred definisana BER ograničenja. Naknadno, čim se povećava SINR (od paketa 11 do 17) SDAM menja modulaciju da bi se na kraju vratio na početnu kada SINR dostigne početni nivo od 20 dB.

3. ZAKLJUČAK

Zbog velikog vremena propagacije u akustičnim kanalima i zbog pokretljivosti podvodnih čvorova, PASM se mora posmatrati kao mreža otporna na kašnjenje. Bezbedni

komunikacioni protokoli se dizajniraju na osnovu interakcije između senzorskih čvorova kojom se otkrivaju zlonamerni čvorovi i time osigurava integritet osetljivih podataka, njihov ispravan prijem, a takođe i privatnost i poverljivost.

4. LITERATURA

- [1]. M. Stojanovic. Acoustic (Underwater) Communications. In ed. J. G. Proakis, *Encyclopedia of Telecommunications*. John Wiley and Sons, New York, (2003).
- [2]. D. Codiga, J. Rice, and P. Baxley. Networked Acoustic Modems for Real-Time Data Delivery from Distributed Subsurface Instruments in the Coastal Ocean: Initial System Development and Performance, *Journal of Atmospheric and Oceanic Technology*. 21(2), 331–346, (2004).
- [3]. S. Misra *et al.*, “Jamming in Underwater Sensor Networks: Detection and Mitigation,” *IET Commun.*, vol. 6, no. 14, Sep. 2012, pp. 2178–88.
- [4]. M. Zuba *et al.*, “Vulnerabilities of Underwater Acoustic Networks to Denial-of-Service Jamming Attacks,” *Security Commun. Net.*, Feb. 2012, pp. 1–11.
- [5]. H. Kulhandjian, T. Melodia, and D. Koutsonikolas, “Securing Underwater Acoustic Communications through Analog Network Coding,” *Proc. SECON*, June 2014, pp. 1–9.
- [6]. E. Demirors *et al.*, “Design of a Software-Defined Underwater Acoustic Modem with Real-Time Physical Layer Adaptation Capabilities,” *Proc. Int'l. Conf. Underwater Networks & Systems*, ser. WUWNET '14, 2014, pp. 25:1–25:8.

Kratka biografija:

Dejan Radić rođen je u Ljuboviji 1981. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva odbranio je 2018.god.

kontakt: dejane81@gmail.com

**REALIZACIJA VEKTORSKOG UPRAVLJANJA ASINHRONOG MOTORA,
UPOTREBOM ALATA ZA AUTOMATSKO GENERISANJE KODA**

**VECTOR CONTROL IMPLEMENTATION OF INDUCTION MOTOR, USING
AUTOMATIC CODE GENERATION TOOLS**

Nikola Diklić, Evgenije Adžić, Marko Vekić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu se prati razvoj indirektnog vektorskog upravljanja asinhronog motora, razvijanom u okruženju za automatsko generisanje koda. Korišćeni alat za grafičko razvijanje i generisanje koda je MATLAB/SIMULINK. Cilj rada je da se istaknu prednosti koje nudi automatsko generisanje koda, upotrebjeno pri razvijanju složenog algoritma vektorskog upravljanja.

Ključne reči: mikrokontroler, grafički kod, vektorsko upravljanje, PWM modulacija

Abstract – This paper follows the development of indirect vector control of induction motor, developed in a model based environment for automatic code generation. The tool used for code generation is MATLAB/SIMULINK. Goal of the paper is to point out the benefits that automatic code generation provides, in developing of complex algorithm, such as vector control.

Keywords: microcontroller, graphic code, vector control, PWM modulation

1. UVOD

Zahtevi modernog društva za čistijom energijom i većom efikasnošću, dovode do toga da oblast upravljanja elektromotornim pogonima (*motor control*) postaje sve interesantnija za izučavanje i implementaciju. Najaktuelnije i najoptimalnije metode upravljanja se baziraju na vektorskoj kontroli, odnosno, na primeru asinhronih motora, svode se na procenu položaja rotorskog fluksa. Jedna od najčešće primenjivanih metoda je indirektna procena položaja vektora rotorskog fluksa (*Indirect field oriented control - IFOC*).

Uz navedene zahteve, u modernoj industriji je sve prisutniji pristup automatskog generisanja koda (*automatic code generation*), putem programa koji se baziraju na razvoju modelovanjem sistema (*Model based development*). Ovaj rad ima za cilj da objedini navedene trendove i predloži novi pristup pri razvoju, kako algoritma, tako i kodiranja u oblasti upravljanja elektromotornim pogonima, konkretno asinhronim motorima.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Marko Vekić.

Cilj rada jeste da se čitaocu, postepeno, praktično prikaže proces razvoja i implementacije indirektnog vektorskog upravljanja asinhronim motorom.

2. PRINCIP VEKTORSKOG UPRAVLJANJA ASINHRONIM MOTOROM

Vektorsko upravljanje asinhronim motorom, zasniva se na rasprezanju upravljanja momentom i fluksom. To takođe podrazumeva pojednostavljen princip upravljanja sličan principu upravljanja motorom jednosmerne struje, gde se pobuda i struja armature potpuno nezavisno kontrolišu. Posledica potpuno raspregnute kontrole se ogleda u tome što se vektor struje i vektor fluksa mogu postaviti upravno jedan na drugi, čime se postiže maksimalan momenat za date amplitude vektora. Odnosno, može se reći da asinhroni motor time počinje da se ponaša kao linearni konvertor momenta, gde je za konstantnu amplitudu fluksa, moment direktno srazmeran struji motora. Time se postiže poboljšanje dinamičkih i regulacionih karakteristika asinhronog motora, što mu omogućava primat na tržištu. Za realizaciju vektorskog upravljanja, neophodno je prvo transformisati matematički model asinhronne mašine iz originalnog trofaznog *abc* domena, u dvofazni domen. Elektromagnetni moment koji mašina razvija, predstavljena u dvofaznom rotirajućem *dq* koordinatnom sistemu, data je jednačinom (1). Iz jednačine se eliminacijom *q* komponente fluksa, jasno vidi kako asinhroni motor postaje linearni konvertor momenta.

$$m_e = \frac{3}{2} p \frac{L_m}{L_r} (\psi_{rd} i_{sq} - \psi_{rq} i_{sd}) \quad (1)$$

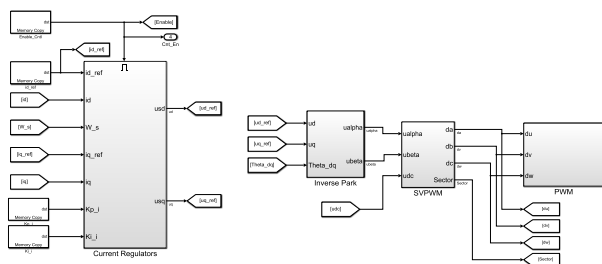
3. HARDVERSKI OPIS SISTEMA

U praktičnoj realizaciji rada korišćen je asinhroni motor snage 1,1 kW, napajan preko industrijskog pretvarača učestanosti *N700*, proizvođača *Hyundai*. Na ovom pretvaraču, originalna upravljačka (matična) ploča zamenjena je *LARA-100 - Power Electronics Launch Ramp Motherboard* matičnom pločom, čime je ostvarena otvorena platforma za proizvoljno kreiranje aplikacije. *LARA-100* matična ploča kontrolisana je od strane mikrokontrolera, proizvođača *Texas Instruments* iz serije *C28335*. Ovaj kontroler koristi *Code Composer Studio* kao alat za spuštanje koda, prethodno izgenerisanog upotrebom MATLAB/SIMULINK-a. Zatvaranje povratne sprege omogućeno je preko već postojećih LEM sondi na energetskom pretvaraču, dok je brzina očitavana

inkrementalnim enkoderom proizvođača SIEMENS. Za potrebe *online* debugovanja, omogućeno je posmatranje bilo kog signala iz grafičkog koda, putem *Perun Power Desk* okruženja u realnom vremenu. Ova mogućnost je značajno doprinela brzini razvoja algoritma i analizi rezultata rada.

4. SOFTVERSKA REALIZACIJA KODA

Grafička realizacija koda zasniva se na modularnom razvoju unutar SIMULINK-a. Ovo podrazumeva činjenicu da se odgovarajući blokovi, izvršavani sa istom periodom odabiranja, grupišu u jedan blok. Takođe je ostvareno razdvajanje grafičkog koda u manje podsisteme, koji realizuju pojedinačnu funkcionalnost. Najvažniji deo grafičkog koda, za upravljanje elektromotornim pogonima, smešten je u *PWM* prekidnu rutinu (20 kHz). U njoj su implementirani proračuni regulatora, odgovarajuće transformacije veličina iz jednog u drugi domen, kao i *SVPWM* modulator. Od važnijih funkcionalnosti sistema upravljanja jedino merenje brzine zahteva nešto manju učestanost izvršavanja, pa je tako izmeštena u glavni podsistem koji se izvršava na 100 Hz. Deo ovako realizovanog grafičkog koda prikazan je na slici (1).



Slika 1 Deo grafičkog koda prekidne (PWM) rutine

Grafički kod formiran je od pojedinačnih modula (merenje struje, merenje brzine, *SVPWM*...). Nakon toga, izvršena je njihova integracija u jedinstveni sistem. Ovaj princip razvoja, omogućio je eksperimentalnu proveru grafičkog koda, nakon svake nadogradnje.

5. EKSPERIMENTALNA PROVERA

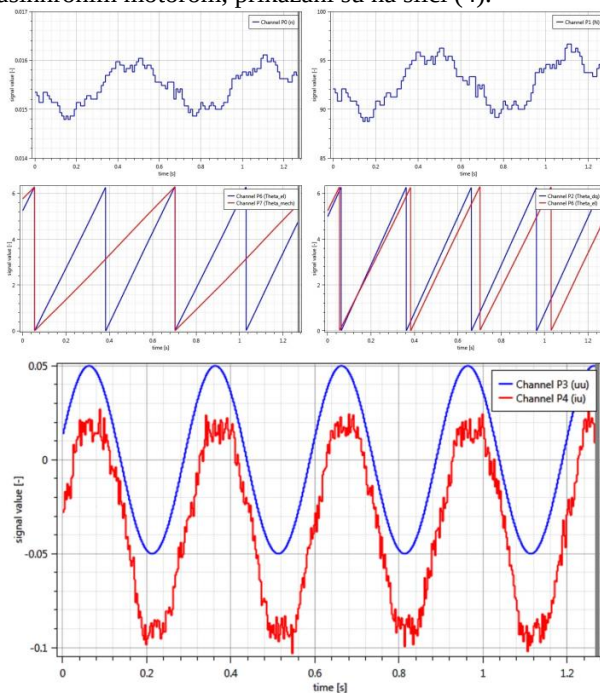
Tri značajna trenutka u razvoju algoritma su:

- otvorena petlja,
- strujna regulaciona petlja i
- brzinska regulaciona petlja.

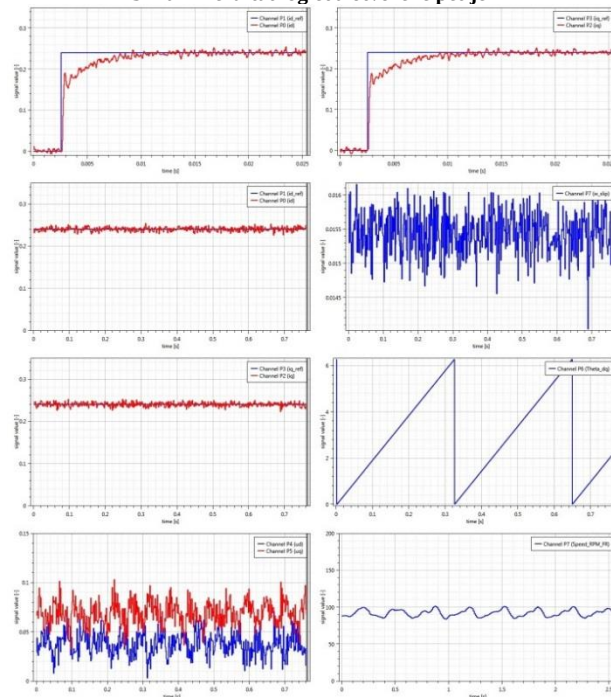
Ogledi izvršeni u tri navedene faze razvoja, dovoljan su pokazatelj valjanosti sistema. Prvo je izvršen ogled u otvorenoj petlji. Zadavane su različite referentne frekvencije napona i posmatrani su brzina motora, ugao *dq* koordinatnog sistema i naponi na motoru. Jedan od rezultata ogleda prikazan je na slici (2).

Kao provera valjanosti koda strujne petlje posmatrani su tranzijentni procesi struja i njihovo ponašanje u stacionarnom stanju. Rezultati ovog ogleda dati su slikom (3). Konačna provera ponašanja sistema na ulaznu referencu brzine, ogleda se u zadavanju tranzicija reference i praćenju odziva brzine. Pored praćenja odziva brzine, veličine od interesa su *q* komponenta struje i napon *DC*

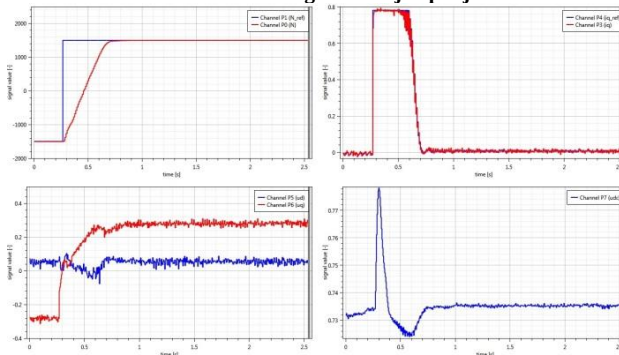
kola. Rezultati ovog ogleda i konačna potvrda ispravne implementacije indirektnog vektorskog upravljanja asinhronim motorom, prikazani su na slici (4).



Slika 2 Rezultati ogleda otvorene petlje



Slika 3 Rezultati ogleda strujne petlje



Slika 4 Rezultati ogleda brzinske petlje

6. ZAKLJUČAK

Korišćenje savremenih metoda generisanja koda za upravljanje naizmeničnog pogona, upotrebom mikrokontrolera, pokazalo se kao veoma dobro iz svih aspekata. Grafički kod je mnogo pregledniji i čitljiviji od koda pisanog u drugom programskom jeziku. Algoritam sistema je vizuelno pristupačan, što olakšava debugovanje sistema. Prednosti koje nude alati, kao što je npr. *PERUN Power Desk*, omogućavaju jasnu prezentaciju veličina i signala, kojima se upravlja ili koje operater zadaje.

Čitaocu rada, prepušteno je da sam pokuša idejnu nadogradnju algoritma upravljanja i da doprinese razvoju ovog novog, modernog načina programiranja, koji nalazi sve veću primenu u industriji. Indirektno vektorsko upravljanje asinhronim motorom je samo jedan od mogućih algoritama za upravljanje naizmeničnim pogonima. U industriji, isti nailazi na izmene u vidu uprošćenja, pojednostavljenja i izmena koje se odnose na samu energetska efikasnost motora. Modularnost grafičkog koda omogućava da se kod prilagodi najnovijim metodama i specifičnim slučajevima aplikacije, kao i da se prenese (uz minimalne izmene) na upravljanje sinhronom mašinom.

7. LITERATURA

- [1] Darko P. Marčetić, Mikroprocesorsko upravljanje energetskim pretvaračima, FTN Izdavaštvo, 2014.
- [2] Veran Vasić, Đura Oros, Energetska elektronika u pogonu i industriji, FTN Izdavaštvo, 2012.
Primer: B. Klaus, P. Horn, Robot Vision, Cambridge, MA, MIT Press, 1986.
- [3] Evgenije Adžić, LARA-100 (Power Electronics Launch Ramp) Motherboard uputstvo
- [4] Slobodan N. Vukosavić, Električne mašine, Univerzitet u Beogradu, Elektrotehnički fakultet, Akademska misao, Beograd 2010.
- [5] Stevan Grabić, Upravljanje energetskim pretvaračima – skripta, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, 2016.
- [6] Embedded Coder Support Package for Texas Instruments C2000 Processors Examples, 2017.
dostupno na:
https://www.mathworks.com/help/supportpkg/texasinstrumentsc2000/examples.html#responsive_offcanvas
- [7] MATLAB Support, 2017. dostupno na:
https://www.mathworks.com/support/index.html?s_c_id=pl_support

Kratka biografija:



Nikola Diklić rođen je u Novom Sadu 1994. god. Diplomski rad pod nazivom „Digitalno U/f upravljanje asinhronim motorom upotrebom automatskog generisanja koda u Matlab/Simulink-u odbranio na Fakultetu tehničkih 2017.god.

Od 2018. godine zaposlen je u preduzeću Continental u Novom Sadu.



Evgenije Adžić je diplomirao 2005, magistrirao 2007, i doktorirao 2014. godine redom na Fakultetu tehničkih nauka. Oblasti interesovanja su mu elektromotorni pogoni, digitalno upravljanje i električna i hibridna vozila. Od oktobra 2018. je zaposlen u preduzeću Continental u Novom Sadu.



Marko Vekić je diplomirao 2005, magistrirao 2007, i doktorirao 2014. godine redom na Fakultetu tehničkih nauka. Oblasti interesovanja su mu energetska elektronika u distributivnim i prenosnim mrežama, kvalitet električne energije i mikromreže. Od 2014. je zaposlen na Fakultetu tehničkih nauka u zvanju docenta

ANALIZA UPRAVLJANJA JEDNOSMERNIM MOTOROM BEZ ČETKICA**ANALYSIS OF THE BRUSHLESS DC MOTOR CONTROL**Stefan Šoić, *Fakultet tehničkih nauka, Novi Sad***Oblast – Energetika, elektronika i telekomunikacije**

Kratak sadržaj – Jednosmerni motori bez četkica (BLDC) su izuzetno atraktivni zbog svojih dobrih dinamičkih karakteristika u odnosu na klasične jednosmerne motore. Takođe, za razliku od sinhronih mašina sa stalnim magnetima (PMSM), razmatrano upravljanje BLDC motorom ne zahteva kontinualnu informaciju o poziciji rotora, već je dovoljno odrediti 6 diskretnih pozicija tokom električnog ciklusa. To se ostvaruje sa 3 holova senzora pravilno razmaknuta na statoru. To čini upravljanje BLDC motorom dosta jednostavnijim, a pogon jeftinijim u odnosu na sinhroni motor. Rad takođe razmatra i mogućnost besenzorskog upravljanja koje bi dodatno snizilo cenu pogona, uz minimalan gubitak u kvalitetu dinamičkog odziva motora.

Ključne reči: jednosmerni motor bez četkica, kontrola bez senzora pozicije

Abstract – Brushless DC (BLDC) motors are attractive due to their better dynamic characteristics compared to brushed DC motors. Also, in contrast to the permanent magnet synchronous machine (PMSM), analysed BLDC motor control doesn't require continuous information about rotor position. It requires to detect only six discrete rotor positions in electrical cycle, which can be obtained by 3 hall sensors. This makes BLDC motor control much simpler and the whole drive significantly cheaper compared to PMSM drive. This paper also considers sensorless BLDC control implementation, which would additionally reduce the drive cost with minimal losses in performance.

Ključne reči: brushless DC motor, sensorless BLDC control

1. UVOD

Iako nosi naziv jednosmerni motor bez četkica, BLDC ustvari spada u sinhronu motore sa stalnim magnetima. Sa njima deli konstrukciju; stalni magneti su locirani na rotoru, dok je radni trofazni namot na statoru. Razlika je jedino u načinu namotavanja statorskih namota, koji su kod BLDC motora tako namotani da obezbede trapezoidalnu raspodelu magnetskog polja po obodu statora, umesto sinusne.

To implicira identičan oblik indukovane elektromotorne sile u vremenu; u svakoj od tri faze u toku električnog perioda ona ima interval od 120 stepeni u kom je pozitivna, interval od 120 stepeni u kom je negativna, kao i dva prelazna intervala od 60 stepeni u kom menja znak.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vlado Porobić, vanr.prof.

Ovakav oblik elektromotorne sile omogućio je znatno jednostavnije upravljanje nego što je to slučaj kod sinhronih mašina sa sinusnom raspodelom fluksa.

Da bi jedan fazni namot doprinosio elektromagnetnom momentu, potrebno je da elektromotorna sila (EMS) i struja koja teče kroz njega budu istog znaka. Tako se dolazi do ideje da, na osnovu tri fazne elektromotorne sile međusobno pomerene za 120 stepeni, postoji 6 radnih sektora; u svakom od njih pozitivna struja treba da teče kroz namot sa pozitivnom indukovanom EMS, a negativna struja kroz namot sa negativnom indukovanom EMS. Treći namot, kod kojeg je EMS u prelaznom intervalu, ostavlja se u neaktivnom stanju. Zahvaljujući ovome, sve što treba znati o rotoru je u kom od šest električnih sektora se on trenutno nalazi, na osnovu čega se mogu generisati odgovarajuće referentne fazne struje.

Očigledno je da skup senzor pozicije poput enkodera, za detektovanje u kom od 6 radnih sektora se motor nalazi nije potreban. Na osnovu toga da li se koristi senzor pozicije ili ne, postoje 2 pristupa upravljanja:

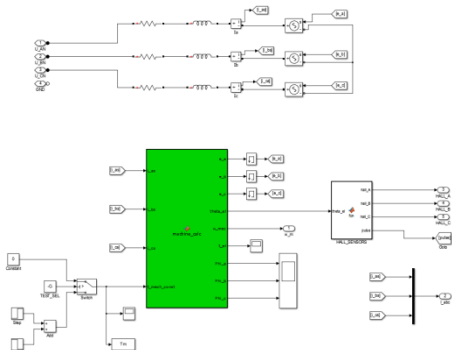
- Senzorsko upravljanje – za senzore pozicije se koriste 3 holova senzora, razmaknuta međusobno za 120 stepeni (električnih). Oni kao izlaz daju digitalni signal, zavisno od toga koji magnetni pol rotora detektuju u datom trenutku. Kombinacijom ova 3 signala može se utvrditi koji u kom radnom sektoru se motor nalazi u svakom trenutku.
- Besenzorsko upravljanje – vrlo atraktivan vid upravljanja, iz razloga što dodatno snižava cenu pogona. Ovde se ne koriste nikakvi senzori pozicije, već se radni sektor utvrđuje obradom merene elektromotorne sile motora. U ovom radu se kao rešenje za detektovanje momenata za komutaciju koristi integracija elektromotorne sile.

2. MODEL BLDC MOTORA

Model motora koji je ovde korišćen realizovan je kao kombinacija RL kola sa kontrolisanim naponskim izvorima koji predstavljaju elektromotornu silu sa jedne strane, i Matlab funkcije koja pokriva mehanički deo rada motora sa druge strane.

Ulazi u funkciju koja predstavlja mehanički deo motora su fazne struje motora, kao i moment opterećenja. Izlazi su mehaničke veličine – ugao, brzina i elektromagnetni moment motora, kao i fluksni obuhvati i elektromotorna sila motora. EMS ustvari i predstavlja vezu između električnog i mehaničkog dela motora, pošto se taj izlaz iz

Na osnovu ugla motora koji je izlazna veličina mehaničkog dela motora, kreiran je i blok koji generiše signale koji predstavljaju izlaz holovih senzora. Slika koja predstavlja implementaciju opisanog modela je data na slici 1.

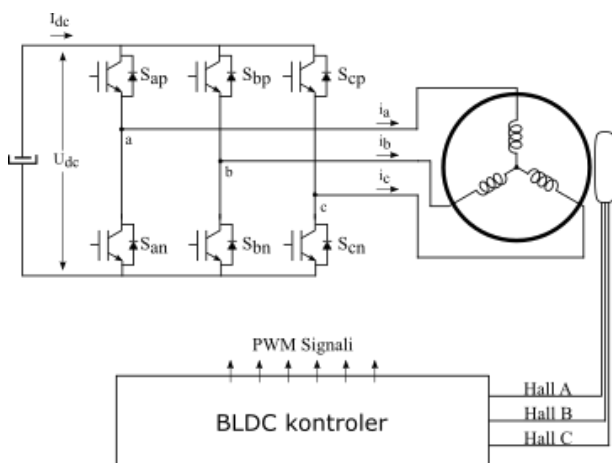


Slika 1. Model BLDC motora

Prednost ovog modela u odnosu na ugrađenu komponentu koja predstavlja ovaj motor u samom *Simulink*-u je u tome što realizovani model obezbeđuje više izlaznih signala, kao i mogućnost snimanja svih signala unutar samog motora, što znatno olakšava razvoj upravljačkog algoritma i analizu postojećih problema na njemu. Takođe, uz modifikaciju nad električnom delu, ovaj model ostavlja prostor za rad sa motorima čiji trofazni namot je vezan u trougao, što nije moguće sa ugrađenom komponentom, koja nudi samo namotaje vezane u zvezdu.

3. SENZORSKO UPRAVLJANJE

Slika 2 prikazuje uobičajeni izgled ovakvog pogona. BLDC motor se napaja preko trofaznog invertora. Na osnovu signala sa 3 holova senzora, kao i zadate reference brzine, kontroler generiše 6 upravljačkih impulsa za gejtove tranzistora u invertoru.

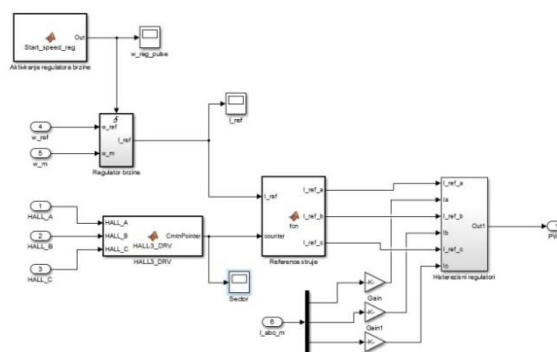


Slika 2. Šema pogona sa BLDC motorom i holovim senzorima

Kao što je pomenuto, holovi senzori daju digitalni izlaz; on je na logičkoj jedinici ako detektuje jedan pol rotora, a na logičkoj nuli ako detektuje drugi. To znači da u električnom periodu senzor ima 2 intervala od po 180

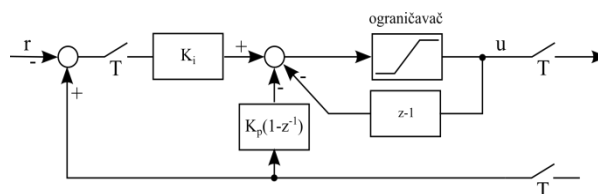
stepeni, sa izlazom koji je naizmenično jednak jedinici i nuli. Uz međusobnu razmaknutost senzora od 120 električnih stepeni, omogućeno je da je za svaki radni sektor motora(koji traje 60 električnih stepeni) različita izlazna kombinacija na holovim sensorima.

Realizacija upravljačkog algoritma u programu *Simulink* prikazana je na slici 3. On je smešten u prekidnu rutinu koja se poziva na svakih 20μs, period na kom se vrši i prekidanje tranzistorskih prekidača. Ulazni podaci su referentna i merena brzina, signali sa holovih senzora, kao i merene fazne struje.



Slika 3. Senzorsko upravljanje – upravljački algoritam

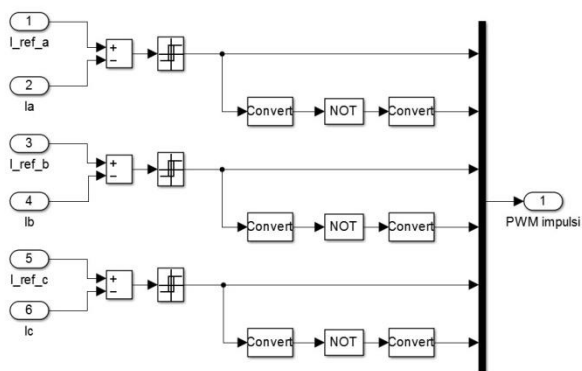
Regulator brzine se izvršava na svakih 500 μ s, tako da se on aktivira na svaki 25. ulaz u prekidnu rutinu. Na osnovu razlike referentne i merene brzine, regulator proračunava referencu struje potrebnu da se ta greška eliminiše. Regulator je realizovan u inkrementalnoj formi, a namotavanje integralnog dejstva je sprečeno. Blok dijagram koji opisuje realizaciju regulatora dat je na slici 4.



Slika 4. Blok dijagram regulatora brzine

Na osnovu signala sa holovih senzora utvrđuje se u kom radnom sektoru je motor trenutno. Na osnovu tog signala, koji uzima celobrojni vrednost između nula i pet, kao i zadate reference struje iz regulatora brzine, generišu se referentne struje za svaku fazu. U zavisnosti od sektora, fazna referentna struja može biti jednaka referenci regulatora (ako je odgovarajuća EMS pozitivna), jednaka referenci pomnoženoj sa -1 (ako je EMS negativna), ili jednaka nuli ukoliko ta faza treba da bude neaktivna (ako je EMS u prelaznom intervalu).

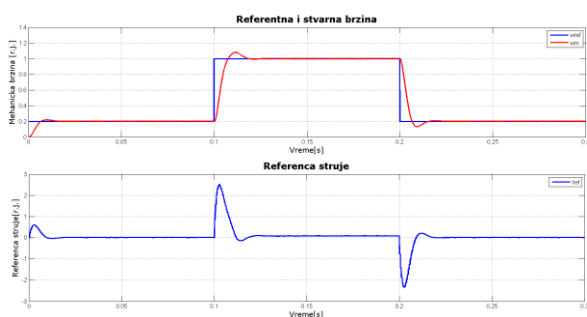
Generisane reference faznih struja se uz odgovarajuće merene fazne struje vode na 3 strujna regulatora, koji omogućavaju da se te reference ostvare. Strujni regulatori su realizovani kao jednostavni histerezisni regulatori. Svaki od njih upravlja odgovarajućom granom trofaznog invertora; ukoliko je merena struja manja od odgovarajuće reference, aktivira se gornji tranzistor u grani i obrnuto. Implementacija strujnih regulatora u simulaciji je data na slici 5.



Slika 5. Histeresizni regulatori faznih struja

Rezultati simulacije pri promeni referentne brzine su dati na slikama 6 i 7.

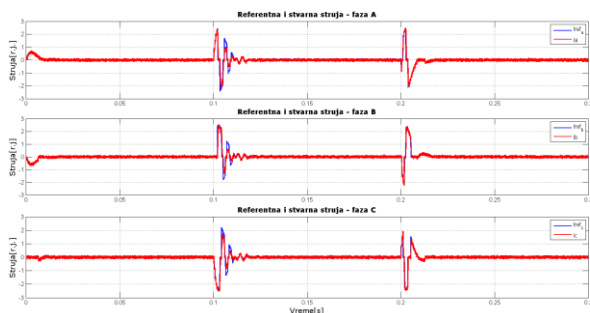
Na slici 6 se može videti da prilikom promene reference brzine regulator brzine zadaje strujnu referencu određenog znaka, zahvaljujući čemu se stvarna brzina brzo smiruje na željenoj vrednosti, uz blagi preskok.



Slika 6. Brzina i referencna struja na promenu reference brzine

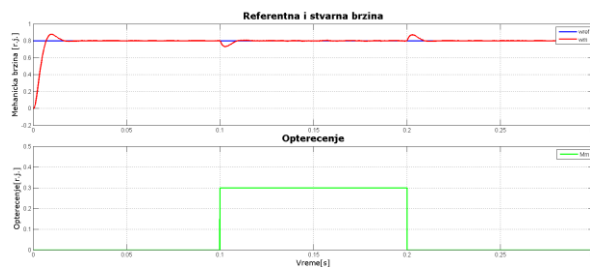
Slika 7 demonstrira odziv strujnih regulatora. Reference se, u skladu sa izlazom regulatora brzine, aktiviraju samo u trenucima promene referentne brzine.

Fazne struje te promene približno isprate, što se ogleda i u kvalitetnom odzivu brzine prikazanom na slici 5.



Slika 7. Referentne i stvarne fazne struje pri promeni reference brzine

Pogon je takođe testiran na promenu opterećenja. Na slici 8 se može videti pravougaoni signal koji predstavlja mehaničko opterećenje. Može se primetiti da je pogon uspeo da vrati brzinu na zadatu referencu u momentima naglog opterećenja, odnosno rasterećenja.



Slika 8. Referentna i stvarna brzina pri nagloj promeni opterećenja

4. BESENZORSKO UPRAVLJANJE

Besenzorsko upravljanje je izuzetno atraktivno za primene u kojim je glavni cilj što niža cena pogona. U tom slučaju se izbacuju holovi senzori za detekciju pozicije. Zbog toga se detekcija pravovremene komutacije svodi ne merenje elektromotorne sile – pošto je jedna faza uvek neaktivna, merenje napona na njoj ustvari daje EMS te faze. Najčešći metod se svodi na detektovanje prolazaka faznih EMS kroz nulu – na osnovu prethodna dva prolaska kroz nulu može se proceniti sledeći trenutak za komutaciju.

Međutim, ovaj metod ima dve bitne mane. Motor se ne može startovati na ovaj način, potrebno je da motor dostigne određenu brzinu pre nego što detekcija nultih trenutaka EMS postane korisna. Takođe, sličan problem se javlja i pri naglim promenama brzine, pošto procena za sledeći trenutak komutacije postaje neprecizna.

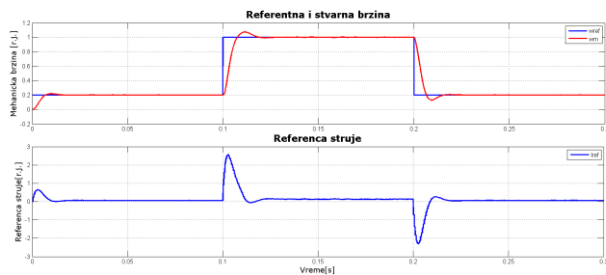
U ovom radu je zato korišćen metod integracije elektromotorne sile. I dalje se meri napon na neaktivnoj fazi, ali se nakon zabeleženog prolaska kroz nulu počinje sa integraljenjem odgovarajuće EMS. Ovaj integral ustvari predstavlja magnetni fluks; pošto je poznato da važi formula:

$$e = \frac{d\psi}{dt} = \frac{d\psi}{d\theta} * \frac{d\theta}{dt} = \frac{d\psi}{d\theta} * \omega, \quad (1)$$

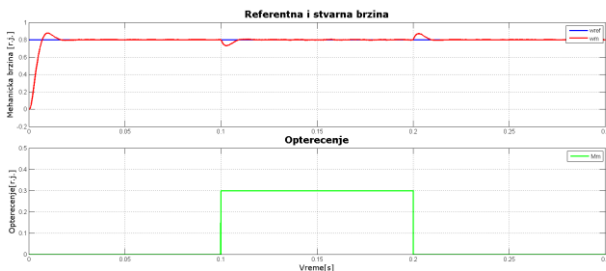
jasno je da EMS zavisi od brzine motora. Time će i integral elektromotorne sile takođe zavisiti od brzine motora; Što se motor sporije obrće, sporije će integral EMS dostizati neki, unapred definisani prag. Za određenu vrednost tog praga fluksa postići će se željeni cilj; trenutak za komutaciju će moći pravovremeno da se definiše bez obzira na brzinu motora.

Simulacija besenzorskog pogona se neznatno razlikuje od iste kod senzorskog; ulazi u prekidnu rutinu su sada elektromotorne sile motora, umesto signala sa holovih senzora. U skladu sa tim je promenjen i blok za određivanje radnog sektora motora, koji beleži trenutak prolaska odgovarajuće EMS kroz nulu i potom je integrirali dok vrednost tog integrala ne dostigne prag fluksa, koji je utvrđen eksperimentalno.

Nad besenzorskim pogonom potom su izvedena ista dva testa kao i nad senzorskim. Odzivi su prikazani na slikama 9 i 10.



Slika 9. Brzina, referenca struje pri promeni reference brzine – besenzorski pogon



Slika 10. Referentna i stvarna brzina pri nagloj promeni opterećenja

Poredeći slike 9 i 10 sa odgovarajućim slikama koje prikazuju odzive pogona sa senzorskim upravljanjem (6 i 8), može se primetiti da su odzivi praktično identični. Uz iste regulatore brzine i struje koji su korišćeni u oba modela, dalje se zaključuje da blok koji na osnovu integracije elektromotorne sile detektuje pravovremeni trenutak komutacije daje ekvivalentne rezultate bloku koji to radi pomoću holovih senzora. Što je još bitnije, besenzorski pogon je potpuno samostalno izvršio zalet, bez prethodnog „open loop“ starta. Takođe, pri naglim promenama reference brzine, odzivi su pokazali stabilan rad pogona.

5. ZAKLJUČAK

Jednosmerni motori bez četkica (eng. BLDC) beleže veliki rast u komercijalnoj primeni, zbog jednostavnije konstrukcije i bolje dinamike od jednosmernih motora. Realizovano upravljanje koristeći holove senzore za povratnu spregu po poziciji rotora pokazuje jednostavnost upravljanja ovim motorom u odnosu na sinhroni motor sa permanentnim magnetima sa sinusnom raspodelom fluksa. Pogon u kome je implementirano besenzorsko upravljanje pokazuje na to da je moguće ukloniti bilo kakve senzore pozicije, bez gubitaka na performansama.

Primenjeni metod za detektovanje pravovremenih trenutaka komutacije ukazuje na to kako neke opšte poznate mane besenzorskih pogona sa BLDC motorom mogu biti eliminisane. U papiru nije razmatran rad pogona u oblasti slabljenja polja, kao ni implementacija četvorokvadrantnog rada, što bi mogli biti neki od ciljeva budućeg rada.

6. LITERATURA

- [1] Krishnan, Ramu. *Electric motor drives: modeling, analysis, and control*. Vol. 626. New Jersey: Prentice Hall, 2001.
- [2] Krause, Paul, et al. *Analysis of electric machinery and drive systems*. Vol. 75. John Wiley & Sons, 2013.

Kratka biografija:

Stefan Šoić rođen je u Vrbasu 1993. god. Osnovne akademske studije na Fakultetu Tehničkih nauka upisao je 2012. godine, smer Energetika, elektronika i telekomunikacije. Godine 2016. upisao je master akademske studije na Fakultetu Tehničkih Nauka.

PRILOG

Parametri jednosmerne mašine bez četkica korišćeni u simulaciji:

- linijska otpornost: $R_{L-L}=0.28 \Omega$;
- linijska induktivnost: $L_{L-L}=0.46 \text{ mH}$;
- broj pari polova: $p=4$;
- moment inercije: $J_m=1.398 \cdot 10^{-5} \text{ kg} \cdot \text{m}^2$;
- nominalna struja: $I_n=8.4 \text{ A}$;
- napon dc kola: $U_{dc}=24 \text{ V}$;
- nominalni moment motora: $M_n=0.42 \text{ Nm}$;
- nominalna mehanička brzina: $\omega_n=419 \text{ rad/s}$;
- namotaji motora su spregnuti u zvezdu, čemu je i prilagođen model motora. Za analizu motora u sprezi trougao, bila bi potrebna modifikacija električnog dela motora.

EVALUACIJA PERFORMANSI KONSENZUS KLASTEROVANJA NAD HISTOPATOLOŠKIM SLIKAMA TUMORA DOJKE**EVALUATION OF CONSENSUS CLUSTERING PERFORMANCE ON HISTOPATHOLOGICAL BREAST CANCER IMAGES**

Milica Janković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – Energetika, elektronika i računarstvo

Kratak sadržaj – U ovom radu prikazana je analiza i izdvajanje obeležja sa histopatoloških slika tumora dojke kako bi se postiglo njihovo klasterovanje na benigne i maligne uzorke. Korišćeno je šest metoda za izdvajanje obeležja, PCA metoda redukcije dimenzionalnosti, konsenzus i polunadgledano klasterovanje i adjusted rand indeks kao mera validacije klasterovanja.

Ključne reči: *Konsenzus klasterovanje, polunadgledano učenje, histopatološke slike, tumor dojke, pca*

Abstract – In this paper we present histopathological breast cancer image analysis, feature extraction, and their clustering into benign and malignant. We used six methods for feature extraction, PCA for dimensionality reduction, ensemble clustering and semi-supervised learning were evaluated and adjusted rand index was used as an external validation measure.

Keywords: *Consensus clustering, semi-supervised learning, histopathological images, breast cancer, PCA*

1. UVOD

Prema istraživanjima Svetske zdravstvene organizacije (World Health Organization), u 2012. godini je umrlo 8.2 miliona ljudi od kancera i očekuje se da će se ta brojka povećati na 27 miliona do 2030. godine. Kancer dojke se najčešće javlja kod žena, a od raka dojke mogu oboleti i muškarci. Rak dojke je stotinu puta češći kod žena, nego kod muškaraca. Zloćudni ili maligni tumori dojke su drugi po redu uzrok smrti, odmah posle karcinoma pluća. Od svih smrtnih ishoda malignih tumora, smrtnost od karcinoma dojke uzima oko 20% [1].

Biopsija je jedini način da se sa sigurnošću utvrdi priroda uočenih promena. Nakon biopsije vrši se histopatološka analiza uzoraka tkiva i klasifikacija benignih i malignih promena od strane patologa. Manuelna analiza i klasifikacija histopatoloških slika donekle je subjektivan proces, dodatno podložan greškama usled umora i obima posla. Računarska podrška i automatska klasifikacija histopatoloških slika bile bi značajan doprinos u svakodnevnom radu patologa, kao podrška proceni patologa donešenoj na osnovu znanja i iskustva.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila prof. dr Tatjana Lončar Turukalo.

Kako bi se poboljšala histopatološka analiza uloženo je dosta truda, posebno kako bi se poboljšala automatska klasifikacija malignih i benignih slika za kompjutersku dijagnostiku. Na osnovu svih dosadašnjih radova se može zaključiti da je glavna prepreka, u razvoju novih metoda za histopatološku analizu nedostatak dovoljno velike anotirane baze podataka koja je dostupna široj javnosti za evaluaciju algoritama koji bi omogućili finu klasifikaciju uzoraka, ne samo na benigne i maligne, već i utvrđivanje podvrsta tumora.

Stoga je u radu [2] korišćena i predstavljena baza podataka *BreaKHist*, sakupljena upravo da omogući olakšan razvoj algoritama za klasifikaciju ovih slika. U tom radu je prvo vršeno izdvajanje obeležja, a zatim klasifikacija gde je postignuta tačnost od 80% do 85% ovisno o faktoru uvećanja slike.

U ovom radu je korišćena ista baza podataka i isti skup obeležja, ali je umesto klasifikacije vršeno klasterovanje. Evaluiraju se performanse jedne metode za nenadgledano učenje i jedne metode za polunadgledano učenje pri grupisanju histopatoloških slika na osnovu teksturalnih obeležja.

2. MATERIJALI I METODE**2.1. Baza podataka**

U ovom radu korišćena je pomenuta *BreaKHist* baza podataka, koja sadrži slike koje prikazuju mikroskopski prikaz biopsije benignih i malignih tumora dojke.

Digitalne slike su dobijene tako što je mikroskop povezan sa digitalnim fotoaparatom. One su 24-bitne slike u boji (8 bita za svaki RGB kanal) sa uvećanjima 40×, 100×, 200× i 400×. Veličina slike je 700 × 460 piksela, bez kompresije i normalizacije. Baza slika sadrži 7909 slika podeljenih na benigne i maligne tumore.

Uvećanje	Benigni	Maligni	Ukupno
40x	625	1370	1995
100x	644	1437	2081
200x	623	1390	2013
400x	588	1232	1820
Ukupno	2480	5429	7909
Pacijenti	24	58	82

Tabela 1. Raspodela slika po klasama i uvećanjima

2.2. Metode

2.2.1. Izdvajanje obeležja

Pre izdvajanja obeležja vršena je predobrada slika normalizacijom. Algoritam koji se koristi automatski pronalazi ispravan vektor bojenja i vrši korekciju boje [3]. Nakon korekcije vektora bojenja, vrši se korekcija inteziteta.

Za izdvajanje obeležja korišćeni su uglavnom teksturalni deskriptori koji spadaju u najčešće korišćene teksturalne deskriptore. Izbor ove vrste obeležja ima smisla na manjim uvećanjima, dok na uvećanju od 400x pored ovih atributa mogla je biti vršena i segmentacija i karakterizacija ćelija, ali je zbog obima izostavljeno iz ovog istraživanja.

LBP (*Local Binary Patterns*) je operator koji izračunava raspodelu binarnih oblika u kružnom susedstvu svakog piksela [4]. Susedstvo karakteriše poluprečnik R i broj suseda P . Intenzitet susednih piksela se poredi sa intenzitetom centralnog piksela: svakom od P suseda dodeljuje se vrednost 1 ako je intenzitet posmatranog piksela veći ili jednaki intenzitetu centralnog piksela, u suprotnom dodeljuje se vrednost 0. Za svaki piksel se na ovaj način formira binarni obrazac. LBP se za svaki piksel u kompaktnom zapisu može prikazati kao binarni kod:

$$LBP(p) = \sum_{i=0}^{P-1} 2^i \delta(f(q_i) - f(p)) \quad (1)$$

gde su $f(q_i)$ i $f(p)$ intenziteti piksela q_i i p , a δ je Kronecker funkcija. Ovde se koristi rotaciono-invarijantni uniformni LBP, a histogram LBP koda se koristi kao teksturalni deskriptor.

CLBP (*Completed Local Binary Pattern*) je jedna od najnovijih varijanti LBP-a [5]. Ima tri komponente izvučene iz lokalnog regiona: centralni piksel, znak i intenzitet. Centralnom pikselu je dodeljen binarni kod nakon globalne primene praga. Za komponente znaka i intenziteta, razmatrano je susedstvo centralnog piksela radijusa R veličine P suseda, slično LBP-u. Tri komponente se spajaju kako bi se formirao završni CLBP histogram. Ovde se koristi rotaciono-invarijantni uniformni CLBP sa parametrima $P = 24$ i $R = 5$.

LPQ (*Local Phase Quantization*) predstavlja novu metodu koja je otporna na zamućenje u slikama. Ona se bazira na kvantizaciji DFT faze u lokalnim prozorima slike [6]. LPQ operator se računa lokalno za svaki piksel, a rezultati se predstavljaju pomoću histograma. U ovom radu je korišćena varijacija LPQ-a koja se zove LPQ-TOP, koja podrazumeva korišćenje različitih vrednosti parametara.

GLCM (*Gray-Level Co-Occurrence Matrices*) se koristi za karakterizaciju tekture slike i bazira se na matricama koje ukazuju na prostornu zavisnost intenziteta. U ovom radu, za izračunavanje GLCM-a se koriste četiri pravca sa uglovima od 0° , 45° , 90° , 135° i osam nivoa intenziteta. Zatim se nad datim GLCM matricama izračunavaju Haralick-ovi atributi [7]. Završni vektor se formira izračunavanjem i konkatencijom prosečnih vrednosti 13-dimenzionalnog vektora za sva četiri smera.

PFTAS (*Parameter-Free Threshold Adjacency Statistics*) je verzija TAS bez parametara [8]. Prvi korak je binarizacija slike upotrebom tri opsega: $[\mu + \sigma, \mu - \sigma]$, $[\mu - \sigma, 255]$ i $[\mu, 255]$, gde je σ standardna devijacija piksela. Prag μ se izračunava na osnovu Otsu metode. Nakon toga se prebrojavaju beli pikseli koji imaju i (0-8) belih suseda i formira odgovarajući histogram. Završni vektor obeležja se dobija tako što se 81-dimenzionalni vektor obeležja poveže sa svojom bitski negativnom verzijom.

ORB (*Oriented FAST and Rotated BRIEF*) se zasniva na FAST detektoru ključnih tačaka i BRIEF deskriptoru ključnih tačaka [9]. U ovom radu korišćena je OpenCV implementacija algoritma, bez promene parametra. Najbolji rezultati se postižu sa 500 ključnih tačaka.

Nakon izdvajanja obeležja vrši se redukcija dimenzionalnosti primenom analize osnovnih komponenti (PCA). Cilj PCA je da predstavi skup uzoraka što tačnije u prostoru sa manjim brojem dimenzija. Očuvanje informacija postiže se očuvanjem varijanse podataka u što većoj meri, za svaki skup je određen prostor sa manjim brojem dimenzija, tako da se u tom prostoru sačuva više od 99% varijanse iz prvobitnog prostora obeležja.

2.2.2. Klasterovanje

Klasterovanje je tehnika istraživanja podataka koja objekte (koji se opisuju atributima) sličnih osobina deli u grupe (klustere), čineći ih preglednijim i korisnijim.

Klasterovanje na osnovu akumulacije dokaza (EAC - *Evidence Accumulation Clustering*) je pristup koji koristi veći broj particija dobijenih nekim jednostavnijim algoritmom klasterovanja da bi formirao novu matricu sličnosti koja reflektuje sličnost uzoraka na osnovu verovatnoće njihovog pojavljivanja u istom klasteru procenjene nad ansamblom particija [10]. Prvo se počinje višestrukom podelom podataka korišćenjem brzog k -means (KM) u veliki broj kompaktnih i malih klastera. Slučajnom inicijalizacijom algoritma KM se dobijaju različite particije, a i broj klastera pri svakom pokretanju algoritma bira se na slučaj iz opsega $\left[\frac{\sqrt{N}}{2}, \sqrt{N}\right]$, gde je N ukupan broj raspoloživih slika. Nakon toga se podaci iz particija mapiraju u matricu koasocijacije koja predstavlja novu meru sličnosti između slika:

$$co_assoc(i, j) = votes_{ij} / M \quad (2)$$

gde je M broj particija u ansamblu, a $votes_{ij}$ je broj koliko puta je par uzoraka (i, j) dodeljen istom klasteru u M klasterovanja. Zatim je nad matricom koasocijacije vršeno hijerarhijsko klasterovanje *average linkage* metodom, a kao parametar korišćen je samo broj klastera.

Polunadgledano konsenzus klasterovanje (SSCC - *Semi-Supervised Consensus Clustering*) se bavi problemom pronalaženja konsenzusne particije podataka koristeći prilikom pronalaženja rešenja znanje o labelama određenog procenta raspoloživih slika [11]. Ako sa $\mathcal{L} \subset \mathcal{I}$ označimo indekse podataka koji su labelirani i sa $l_i \in \{1, \dots, k\}$ labelu koja odgovara i -toj slici, $i \in \mathcal{L}$.

Problem koji rešava SSCC može da se formuliše kao problem minimizacije sa ograničenjem:

$$\phi^* \in \arg \min_{\hat{\phi} \in \mathcal{I} \rightarrow \{1, \dots, k\}} \sum_{u=1}^m d(\hat{\phi}, \phi_u) \quad (3)$$

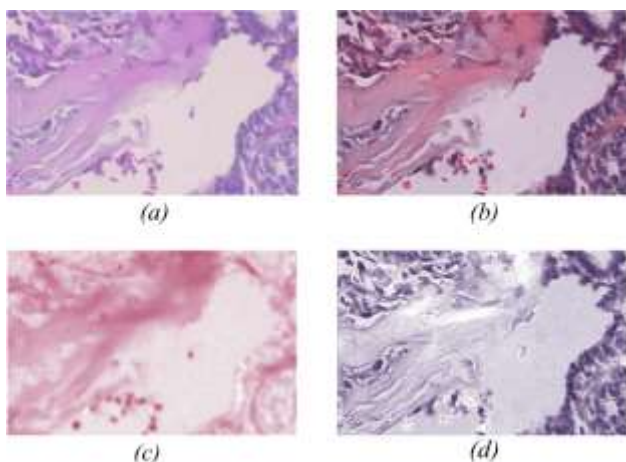
gde je $\hat{\phi}(i) = l_i$ za svaki $i \in \mathcal{L}$, a $\phi_u \in \mathcal{I}_u \rightarrow \{1, \dots, k_u\}$ je funkcija koja koduje podelu podskupa tačaka podataka indeksovanih sa $\mathcal{I}_u \subseteq \mathcal{I} = \{1, \dots, N\}$ u k_u klastera. U ovom radu ansambl je konstruisan korišćenjem ansambla od 200 particija KM algoritma i 20% poznatih labela.

Mere validacije klasterovanja je vršena primenom *adjusted rand* indeks (ARI) kao eksterne mere za validaciju klasterovanja.

3. REZULTATI

3.1. Rezultati normalizacije

Normalizacija boje je vršena nad slikama iz baze podataka za sva tri RGB kanala. Jedan primer efekata ove normalizacije prikazan je na Slici 1. Na ovaj način izbegnute su varijacije u obeležjima usled neujednačene obojenosti uzoraka koja zavisi i od kvaliteta boja i od starosti uzorka.



Slika 1. (a) Prikaz malignog tumora; (b) Prikaz slike nakon izdvajanja boje (hematoksilin); (c) Prikaz slike nakon izdvajanja boje (eozina) i (d) Prikaz slike nakon normalizacije

3.2. Rezultati izdvajanja obeležja

Nakon normalizacije vršeno je izdvajanje obeležja. Za svako uvećanje i za svaku grupu obeležja izdvajanje je vršeno nezavisno, a dobijene su matrice dimezije $N \times d$ gde je N broj vrsta koji odgovara broju raspoloživih slika za dato uvećanje (Tabela 1), a d broj kolona koji odgovara broju obeležja za sva tri RGB kanala prikazan u Tabeli 2.

Naziv	LBP	CLBP	LPQ	GLCM	PFTAS	ORB
Broj obeležja	30	4056	768	39	162	96

Tabela 2. Ukupan broj obeležja iz svake grupe

3.3. Rezultati redukcije dimenzionalnosti

Nad matricama obeležja vrši se redukcija dimenzionalnosti PCA metodom tako da se očuva 99% varijanse. U Tabeli 3 prikazan je novi broj obeležja nakon redukcije dimenzionalnosti. Potrebno je uočiti da je za različita uvećanja PCA analiza rezultovala različitim brojem obeležja.

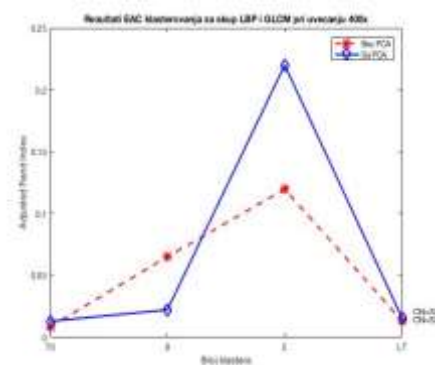
Uvećanje	LBP	CLBP	LPQ	GLCM	PFTAS	ORB
40x	11	1304	140	8	31	75
100x	11	1237	144	9	32	74
200x	12	1191	147	9	30	66
400x	12	1205	166	9	29	65

Tabela 3. Vrednosti kolona matrice obeležja nakon PCA

3.4. Rezultati klasterovanja

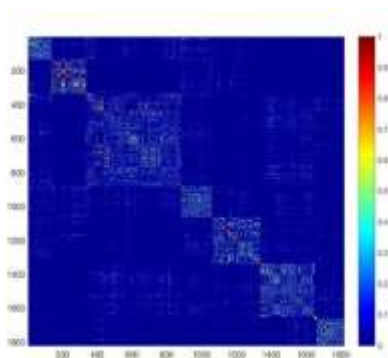
Kod EAC klasterovanja za broj klastera se uzimaju vrednosti 2, 8 (ukupan broj podvrsta tumora), 70 (radi procene načina grupisanja uzoraka) i broj dobijen *life-time* (LT) kriterijumom, odnosno broj klastera koji se dobija u najdužem opsegu presecanja dendrograma. Dok se kod SSCC klasterovanja za broj klastera zadaju vrednosti 2, 8, 50 i 70. Klasterovanje je vršeno za svaku grupu obeležja zasebno za različit broj klastera sa i bez PCA. Nakon toga za skup svih obeležja zajedno za različit broj klastera sa i bez PCA. I na kraju za skup obeležja sa i bez PCA, koji je formiran spajanjem onih obeležja koja su postigla najveću tačnost u diplomskom radu [12] u kojem je vršena klasifikacija nad istim skupom, takođe za različit broj klastera. Ovde će, zbog obimnosti, biti prikazani samo neki od rezultata.

Rezultati EAC klasterovanja sa i bez PCA za skup koji je dobijen spajanjem LBP i GLCM obeležja za različit broj klastera (2, 8, 70, *LT* kriterijum) za uvećanje 400x su prikazani na Slici 2. Zapaža se da se sa smanjenjem broja klastera poboljšavaju rezultati dok primena PCA ovde dobro utiče posebno kada je broj klastera dva gde je ARI 0.22.



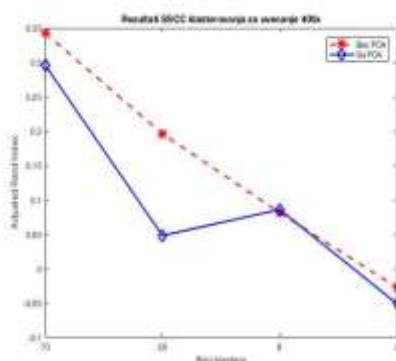
Slika 2. Rezultati EAC klasterovanja sa i bez PCA za skup LBP i GLCM obeležja za različite brojeve klastera

Na Slici 3. prikazana je sortirana matrica koasocijacije za slučaj osam klastera gde se uočava slaba sličnost među uzorcima iz istog klastera, koja se ogleda i u malom ARI indeksu (Slika 2).



Slika 3. Sortirana matrica koasocijacije nad LBP i GLCM skupom obeležja bez PCA ($k=8$)

Na Slici 4. prikazani su rezultati SSCC klasterovanja za skup svih obeležja združeno pre i nakon primene PCA za različit broj klastera (2, 8, 50, 70). Na osnovu slike uočava se malo poboljšanje kada je broj klastera 70 gde je ARI 0.3426, a to je najbolji rezultat postignut u odnosu na druga uvećanja. Takođe se uočava da se rezultati pogoršavaju sa smanjivanjem broja klastera i primenom PCA, uglavnom jer se manje grupe uzoraka izoluju u manje klastere, dok je većina uzoraka u jednom klasteru.



Slika 4. Rezultati SSCC klasterovanja sa i bez PCA za sva obeležja i različit broj klastera (2, 8, 50, 70)

Za uvećanje 400x u Tabeli 4. prikazana je raspodela uzoraka po klasterima, za broj klastera 8. U vrstama su naznačene originalne labele. Uočava se veliko mešanje među uzorcima što ukazuje na veliku sličnost između nekih benignih i malignih tumora i potrebu za dodatnim obeležjima ili njihovom nelinearnom transformacijom u cilju bolje separabilnosti uzoraka.

	Broj klastera							
	1	2	3	4	5	6	7	8
Benigni	92	118	0	87	121	40	86	44
Maligni	187	0	246	200	52	222	20	305

Tabela 4. Rezultati SSCC bez PCA za $k=8$

4. ZAKLJUČAK

Klasterovanjem, čak i primenom polunadgledanog učenja nije dobijeno zadovoljavajuće grupisanje uzoraka po klasama. Uočeno je veliko izdvajanje određenih grupa uzoraka, što u postupku hijerarhijskog klasterovanja uslovljava da se ti specifični uzorci prvi grupišu u manje klastere, dok veliki broj benignih i malignih uzoraka

ostaje u istom klasteru čak i pri povećanju broja klastera na 50 ili 70. Kako bi se poboljšali rezultati klasterovanja jedan od mogućih pristupa bi bio izbor različitih parametara prilikom izdvajanja obeležja i testiranje njihovog uticaja kako na EAC tako i na SSCC. Pored toga potrebno je razmotriti i druga obeležja, posebno za veća uvećanja. Dodatni izazov bi bio pokušaj klasterovanja uzoraka u podtipove malignih i benignih tumora, kao i pokušaj klasterovanja po pacijentima.

5. LITERATURA

- [1] World Health Organization: <http://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/>
- [2] F. A. Spanhol *et al.*, "A Dataset for Breast Cancer Histopathological Image Classification", IEEE Transactions on Biomedical Engineering, str. 1455-1463, 2016.
- [3] M. Macenko *et al.*, "A Methode For Normalizing Histology Slides For Quantitive Analysis", IEEE International Symposium on Biomedical Imaging, str. 209., 2013.
- [4] T. Ojala *et al.*, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", IEEE Trans. Pattern Anal. Mach. Intell., str. 971-987, 2002.
- [5] Z. Guo *et al.*, "A completed modeling of local binary pattern operator for texture classification", IEEE Trans. Image Process, str. 1657-1663, 2010.
- [6] V. Heikkilä *et al.*, "Blur insensitive texture classification using local phase quantization", Proc. 3rd Int. Conf. Image Signal Process., str. 236-243, 2008.
- [7] R. Haralick *et al.*, "Textural features for image classification", IEEE Trans. Syst. Man Cybern., str. 610-621, 1973.
- [8] L. P. Coelho *et al.*, "Structured literature image finder: extracting information from text and images in biomedical literature", New York : Springer, str.121, 2010.
- [9] E. Rublee *et al.*, "ORB: An efficient alternative to SIFT or SURF", Proc. IEEE Int. Conf. Comput. Vision, str. 2564-2571, 2011.
- [10] L.N. Fred, "Data clustering using evidence accumulation", IEEE Object recognition supported by user interaction for service robots., 2002.
- [11] H. Aidos, "Semi-Supervised Consensus Clustering for ECG Pathology Classification", Proc European Conf. Machine Learning and Principles and Practice of Knowledge Discovery in Databases, str. 150-164, 2015.
- [12] A. Antić, "Klasifikacija histopatoloških slika tumora dojke", Diplomski rad, Novi Sad : FTN, 2018.

Kratka biografija:



Milica Janković rođen je u Doboju 1991. god. Master rad na Fakultetu tehničkih nauka iz oblasti Energetike, elektronike i telekomunikacija – Obrada signala, odbranila je 2018.god. Kontakt: milica24111991@live.com

ANALIZA SEKVENCI MIKROBIOMA KROZ MOTHUR TOK OBRADA**ANALYSIS OF MICROBIOME SEQUENCES USING MOTHUR***Aleksandra Kantar, Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu analizirane su 16S rRNA sekvence mikrobioma kroz Mothur tok obrada radi naknadne klasifikacije uzoraka na osnovu utvrđenog taksonomskog sastava. Klasifikacija je vršena random forest algoritmom i postignuta je tačnost od 95,46%. Nakon prikaza i diskusije rezultata, rezultati su upoređeni sa prethodnim istraživanjima nad istim podacima.

Ključne reči: 16S rRNA, klasifikacija, mikrobiom, Mothur

Abstract – In this paper 16S rRNA microbiome sequences were analyzed using Mothur pipeline for their further classification based on the obtained taxonomic description. Classification is performed using random forest algorithm achieving accuracy of 95,46%. The results were discussed and compared with the previous findings from the literature.

Keywords: 16S rRNA, classification, microbiome, Mothur,

1. UVOD

Neki od mikroba su veoma važni za zdravlje čoveka, dok su neki od njih izuzetno patogeni i predstavljaju uzrok raznih bolesti. Mikrobiom predstavlja ukupni genetski materijal mikroba prisutnih u ljudskom organizmu. Na svaki ljudski gen dolazi oko 100 gena mikroba, čime se objašnjava i značaj analize sprežanja i uticaja na genetskom nivou [1]. Trenutno najzastupljenija tehnika za proučavanje mikrobiološkog diverziteta jeste sekvenciranje marker gena 16S rRNA kod prokariota ili 18S rRNA kod eukariota, s obzirom na to da su ovi geni visoko-očuvani u okviru vrste. Grupe mikrobioloških vrsta koje pokazuju određen nivo sličnosti predstavljaju OTU (Operational Taxonomic Unit) jedinice. OTU struktura pojedinačnih uzoraka grupiše se u tkz. OTU tabele, gde je za svaki uzorak naznačen kvantitet vrsta iz određene OTU jedinice. U slučaju kada postoji veliki broj uzoraka, OTU tabele su, po pravilu, retke.

U ovom radu se primenjuje Mothur [2] tok obrade podataka za analizu 16S rRNA sekvenci i utvrđivanje taksonomskog sadržaja uzoraka mikrobioma. U ovom postupku je kao referentna baza korišćena SILVA baza podataka, verzija 132 [3]. Na raspolaganju nije bio čitav genom, već samo 16S rRNA marker gen. Iako postoje ograničenja u korišćenju ovih sekvenci, kao što su nizak

stepen evolucije i manjak korelacije sa funkcijama organizma, nije se pojavio nijedan drugi marker koji se može naći u svim organizmima, koji ima nizak stepen rekombinacije gena i ima dovoljno genetskih informacija za razlikovanje vrlo srodnih organizama.

U ovom istraživanju korišćeni su podaci prikupljeni i analizirani u okviru studije „Moving pictures of the human microbiome“ [4] u kojoj su analizirani uzorci mikrobioma uzimani iz usta, creva i kože jednog muškarca i jedne žene u 396 vremenskih trenutaka, pri čemu je ukupno uzeto 1967 uzoraka. Prethodna istraživanja nad ovim skupom podataka, sem navedenog [4] koji analizira promene u taksonomskom sastavu obuhvatila su obradu podataka i formiranje OTU tabela u paketu QIIME [5], koji je funkcionalno sličan Mothur-u, ali sa izvesnim razlikama u predobradama i parametrima korišćenih algoritama, što može da proizvede i neke razlike u rezultatima [6]. Jedno istraživanje obuhvatilo je obradu podataka uz korišćenje SILVA, ali i Greengenes [7] referentne baze podataka, a zatim klasifikaciju Random Forest [8] klasifikatorom, dok je drugo istraživanje obuhvatilo klasterizaciju podataka nakon formiranja OTU tabele.

Cilj ovog istraživanja jeste izvršavanje sekvenciranja u Mothur toku obrada i upoređivanje rezultata klasifikacije podataka dobijenih na taj način i u QIIME paketu, kako bi se uočile eventualne razlike u performansama klasifikacije nad dobijenim OTU tabelama nakon ova dva postupka obrade, o čemu se vode debate u akademskoj zajednici [6].

2. OPIS PODATAKA

U bazi podataka korišćenoj u ovom radu nalazi se oko 69 miliona sekvenci 16S rRNA gena, formiranih na osnovu 1967 uzoraka. Od žene je uzeto 135 uzoraka iz usta, 268 sa kože i 131 uzorak iz creva, a od muškarca 373 uzorka iz usta, 724 sa kože i 336 uzoraka iz creva. Podaci su preuzeti preko MG-RAST API servera nakon koraka za filtriranje i ukupna veličina skupa podataka iznosi oko 12 GB.

3. METODOLOGIJA RADA**3.1. Mothur**

Za potrebe ovog istraživanja korišćen je operativni sistem Bio-Linux i Mothur softver, verzija 1.40.5. Mothur je trenutno najviše citiran alat za analizu 16S rRNA genetskih sekvenci, koji može da se primenjuje i na vrlo velikim setovima podataka. Ograničenje u vidu brzine izvršavanja komandi može biti raspoloživa količina RAM

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Tatjana Lončar Turukalo, vanr. prof.

memorije na računaru, a s obzirom na to da većina računara ima procesore sa više jezgara, brzina izvršavanja može se povećati paralelizacijom procesa.

Da bi se formirale OTU tabele, prethodno je vršena obrada sekvenci gena u *Mothur*-u. Originalni podaci dobijeni sa MG-RAST servera su *fna* tipa, što je jedan od tipova *fasta* formata koji sadrži sekvence nukleotida. Pre početka obrade, takođe je potrebno opredeliti se za referentnu bazu podataka za poravnanje sekvenci i klasifikaciju. U ovom radu izabrana je SILVA baza podataka [3].

Nakon preuzimanja podataka, potrebno je prebrojati sekvence, odrediti njihovu prosečnu dužinu, broj nedefinisanih nukleotida, itd. Posle toga treba očistiti skup uklanjanjem sekvenci koje ne zadovoljavaju određene kriterijume: dužina im je mnogo veća od proseka, sadrže nedefinisane nukleotide, imaju niz homopolimera većih od dužine koju korisnik zahteva i slično. Smanjivanjem broja sekvenci povećava se brzina dalje obrade podataka i smanjuje mogućnost greške pri obradi. Pošto su poravnanje i obrada zahtevni procesi u pogledu vremena i računarskih resursa, potrebno je utvrditi sve jedinstvene sekvence kako se njihova obrada ne bi višestruko ponavljala. Ovim se još više smanjuje broj sekvenci koje se u nastavku obrađuju. Posle navedenog sređivanja sekvenci, neophodno je uraditi njihovo poravnanje.

U bioinformatici, poravnanje je vrlo važno i svrha poravnanja je višestruka. Istraživači mogu otkriti funkciju novih sekvenci gena vršenjem poravnanja sa sekvencama koje već postoje u bazama podataka, mogu izvući zaključke o sličnosti sekvenci gena kod različitih bliskih vrsta tokom evolucije, ili naći strukturno ili funkcionalno slične regione gena u proteinima. Poravnanje sekvenci je procedura poređenja para sekvenci traženjem niza karaktera ili obrazaca koji se pojavljuju istim redosledom u sekvencama. Mogu se porediti i više od dve sekvence istovremeno, i ovaj proces može biti globalni ili lokalni.

Posle poravnanja, potrebno je proveriti da li se sve sekvence nalaze u određenom regionu i da li su sve one iste dužine. To se vrši pomoću filtriranja, a, pošto *Mothur* prilikom poravnanja ubacuje prazna mesta ili tačke radi označavanja umetnutih ili obrisanih nukleotida, ti karakteri se u ovom koraku odstranjuju, radi prevencije grešaka. Ovim uklanjanjem se znatno smanjuje dužina sekvenci. Dalje se klasteruju sekvence po principu sličnosti, recimo radi se grupisanje onih sekvenci koje se međusobno razlikuju za najviše dva nukleotida. Himerne sekvence sastoje se od dela jedne sekvence i dela druge, tako da izgledaju kao jedinstvene sekvence, ali nisu dobre za dalju analizu, tako da, ako je sekvenca identifikovana kao himerna u nekom uzorku, uklanja se iz svih uzoraka u kojima se nalazi, što je primenjeno i u ovom istraživanju. Ovim korakom se ponovo vrši redukcija broja sekvenci u skupu podataka, ali je potrebno još proveriti da li i dalje postoje nepoželjne sekvence nakon svih prethodno navedenih koraka. U zavisnosti od cilja i teme istraživanja, istraživači mogu želeći da dobiju samo određene vrste za dalju analizu, tako da se izvršava klasifikacija sekvenci korišćenjem referentne baze, i nakon toga mogu se ukloniti one vrste koje nisu od

značaja za istraživanje. Posle ovoga sekvence bi trebalo da se sekvence dodele OTU jedinicama na osnovu taksonomije, pri čemu je potrebno definisati i procenat sličnosti. U ovom istraživanju procenat sličnosti je 97%, što odgovara vrsti, dok, na primer, 90% sličnosti odgovara familiji. Kada se sekvence dodele OTU jedinicama, formira se OTU tabela koja za svaki uzorak ukazuje na broj sekvenci iz svake OTU.

3.2. Klasifikacija

Klasifikacija nekog uzorka zasniva se na pronalaženju sličnosti sa unapred određenim uzorcima koji su pripadnici različitih klasa, pri čemu se sličnost dva uzorka određuje analizom njihovih karakteristika. Pri klasifikaciji se svaki uzorak svrstava u neku od klasa sa određenom tačnošću, a algoritam uči na osnovu već poznatih klasifikacija i na taj način se gradi skup pravila na osnovu kojih će se kasnije vršiti klasifikacija novih uzoraka.

Random Forest (RF) klasifikator je ansambalski klasifikator zasnovan na ansamblu stabala odlučivanja. Osnovna ideja ansambalske klasifikacije je da se, umesto jednog klasifikatora, kombinuje više pojedinačnih klasifikatora slabijih performansi. Svaki od klasifikatora izvodi klasifikaciju nezavisno od ostalih, tako da se, na kraju, uzorku dodeljuje ona klasa koju je izabralo najviše klasifikatora. U slučaju *RF* algoritma, ansambl se formira od stabala odlučivanja, pri čemu se broj stabala povećava dok se tačnost klasifikacije ne ustali. Svako stablo odlučivanja generiše se na isti način po određenom algoritmu, ali se svako stablo trenira korišćenjem drugačijeg butstrep skupa uzoraka. Označimo sa N broj uzoraka, a ukupan broj varijabli, odnosno atributa označimo sa M . Butstrep trening skup ima istu veličinu kao originalni skup, a formira se tako što se iz originalnog skupa podataka na slučajan način odabere N uzoraka sa vraćanjem. Na ovaj način oko 1/3 uzoraka ne bude uključena u treniranje svakog od stabala [9]. Da bi se stabla dekorelisala uvodi se dodatno ograničenje – u svakom čvoru odlučivanja na slučaj se bira m od M varijabli koje utiču na odlučivanje u tom čvoru stabla, pri čemu je $m \ll M$. Prilikom generisanja stabala, broj m je konstantan.

Istraživanja su pokazala da stepen greške *RF* klasifikatora zavisi od korelacije između stabala u ansamblu tako što povećanje korelacije utiče na povećanje greške klasifikatora. Ovaj problem rešava se donekle slučajnim odabirom malog podskupa obeležja prilikom odlučivanja u svakom čvoru. Na grešku utiče i tačnost klasifikacije svakog pojedinačnog stabla. Stablo odluke sa vrlo malom greškom predstavlja dobar (jak) klasifikator, što doprinosi smanjenju ukupne greške klasifikacije na osnovu ansambla. Redukcijom parametra m i slučajnim izborom obeležja redukuje se korelacija, ali i tačnost klasifikatora, dok povećanje ovog parametra utiče na povećanje korelacije i tačnosti, pa stoga treba izabrati optimalnu vrednost m koja je kompromis između dekorelacije stabala u ansamblu i njihove pojedinačne tačnosti. Optimalna vrednost m može se naći uz pomoć *OOB (out-of-bag)* greške. Kada se, radi generisanja stabala, iz trening skupa uzimaju uzorci sa vraćanjem, otprilike jedna trećina uzoraka se ne koristi, odnosno nije uzeta za

generisanje stabala [9]. Ovi *OOB* podaci koriste se za izračunavanje procene greške klasifikacije svakog stabla odluke. Nakon formiranja stabla, kroz njega se puštaju svi *OOB* podaci kao test. Potom se za svaki uzorak bira klasa koja je dobila najviše glasova od stabala za koje je uzorak bio *OOB*. Odnos broja puta kada izabrana klasa *j* nije jednaka pravoj klasi za sve uzorke i ukupnog broja klasifikacija (tj. broja uzoraka) daje *OOB* procenu greške.

Prednosti *RF* klasifikatora su velika tačnost čak i kod velikih skupova podataka i velikog broja klasa, zadržava određenu tačnost i kada nedostaje deo podataka, može davati procene koje varijable su važne za klasifikaciju, ne dolazi do prilagođenja podacima (*overfitting*), itd. Jedna od njegovih mana jeste vremenska složenost u odnosu na jednostavnije klasifikatore.

Tačnost klasifikacije predstavljena je pomoću matrica konzistencije. Matrica konzistencije jeste matrica čije vrste označavaju originalne klase uzoraka, kojih ima šest, a kolone predstavljaju klase koje je algoritam prepoznao i u koje je smestio uzorke. Na osnovu ove matrice može se videti kako rezultat klasifikacije odgovara stvarnim klasama.

U ovom istraživanju korišćena je implementacija *Random Forest* klasifikatora iz programskog jezika *Python*. Za procenu tačnosti klasifikacije korišćena je funkcija *accuracy_score* iz *scikit* biblioteke u *Python*-u.

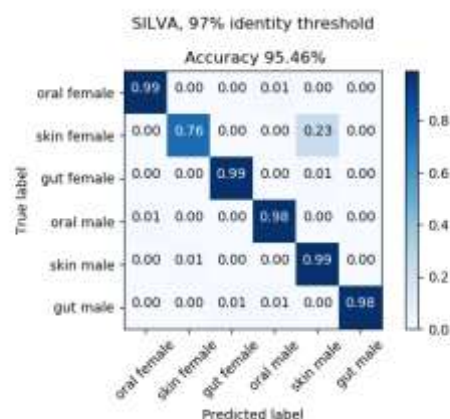
4. REZULTATI

Radi dobijanja OTU tabele, izvršena je obrada podataka u *Mothur* toku. Taj proces je bio vremenski i računarski veoma zahtevan. S obzirom na to da *Mothur* na osnovu ulaznih podataka generiše fajlove koji mogu biti veoma veliki, osnovni preduslov za implementaciju je dovoljno RAM memorije, a poželjno je imati i više procesora radi paralelizacije procesa. Za potrebe ovog istraživanja računarski resursi obuhvatali su 32 GB RAM memorije, 16 procesora i oko 1 TB mesta na hard disku. Neki algoritmi su zahtevali više vremena i računarskih resursa, a neki manje, u zavisnosti od kompleksnosti. Računarski i memorijski najzahtevnija bila je procedura računa udaljenosti između sekvenci i formiranja matrice udaljenosti, u kojoj se sekvence dodeljuju OTU jedinicama. Drugo „usko grlo“ ove obrade bio je lošiji kvalitet samih podataka, pa su, u toku obrade, neki uzorci morali biti izostavljeni iz skupa za dalju obradu. Izostavljeno je ukupno 74 uzorka od postojećih 1967, svodeći broj uzoraka na 1893.

Dobijena OTU tabela sa zahtevanih 97% sličnosti radi svrstavanja sekvenci u OTU jedinice ima dimenzije 1893 x 1149. U vrstama su smešteni uzorci, dok su u kolonama OTU jedinice, a u samoj tabeli broj sekvenci dodeljen svakoj od OTU jedinica. OTU tabela je retka, odnosno u uzorcima nisu prisutne sve OTU jedinice, već obično neki manji podskup. Najmanji broj sekvenci imaju uzorci uzeti sa kože žene, a najveći uzorci uzeti iz creva muškarca.

Random Forest klasifikacijom dobijene su matrice konzistencije i određena je tačnost dodele uzoraka odgovarajućim klasama. Parametar koji je variran jeste broj stabala u klasifikatoru, odnosno veličina ansambla. Tačnost dodele uzoraka klasama dobijena prilikom

promene ovog parametra varira od 94,98% za 80 stabala do 95,46% za 100 stabala. Veći broj stabala u klasifikatoru (150, 200) daje rezultate u navedenom intervalu, tako da povećanje broja stabala iznad 100 nema uticaja na poboljšanje tačnosti klasifikatora, stoga je izabrana veličina ansambla od 100 stabala. Na slici 1. prikazana je normalizovana matrica konzistencije dobijena nakon klasifikacije *RF* klasifikatorom sa 100 stabala.



Slika 1. Normalizovana matrica konzistencije, 100 stabala

Ukupna tačnost dodele uzoraka odgovarajućim klasama iznosi 95,46%. Uzorci iz svake klase su uglavnom u potpunosti raspodeljeni u odgovarajuću klasu. Veća odstupanja mogu se приметiti u slučaju klase čiji su uzorci sa kože žene. Sa gotovo potpunom tačnošću, gde je svega po jedan uzorak netačno raspoređen, raspoređeni su uzorci uzeti iz creva i usta žene. Iako je iz klase sa uzorcima iz usta muškarca izbačeno čak 60 uzoraka, rezultati za ovu klasu su veoma dobri, tako da manjak uzoraka nije uticao na tačnost klasifikacije.

Variranjem broja stabala utvrđeno je da su svi uzorci manje ili više dobro klasifikovani, osim u slučaju klase uzoraka sa kože žene, gde je uvek bio najveći broj pogrešno klasifikovanih uzoraka u klasu uzoraka sa kože muškarca. Uzrok pogrešne klasifikacije uzoraka sa kože žene u klasu uzoraka sa kože muškarca (a značajno manje obrnuto) je verovatno usled višestruko većeg broja uzoraka u klasi kože muškarca, čime ove nejednake apriorne verovatnoće utiču na pomeranje granice odlučivanja. Dodatno, koža je najviše izložena mešanju mikroba sa mikrobima iz sredine, što može biti dodatni uzrok pogrešne klasifikacije. Najbolji rezultati u sva četiri slučaja dobijeni su za klase uzoraka iz usta i creva žene, gde je samo po jedan uzorak pogrešno klasifikovan kao uzorak klase usta muškarca, odnosno kože muškarca, respektivno.

Formirana OTU tabela sa 97% sličnosti korišćena je za konsenzus klasterovanje uzoraka u šest klastera. Ansambl klasterovanja formiran je korišćenjem particija spektralnog klasterovanja nad matricama sličnosti dobijenim sa 24 različite metrike. Konsenzus particija je na kraju dobijena hijerarhijskim klasterovanjem sa *ward* merom povezivanja nad novom matricom sličnosti dobijenom na osnovu ansambla [10]. Rezultati konsenzus klasterovanja ukazuju na veoma dobro grupisanje ovih

podataka – dobijena je vrednost 0.61 za *Adjusted Rand Index*, odnosno indeks za validaciju klasterovanja koji predstavlja meru sličnosti između dve particije. Sličnost sa rezultatima dobijenim u ovom istraživanju je u lošijem grupisanju uzoraka sa kože, s tim što su klasterovanjem i uzorci sa kože muškarca u velikom broju raspoređeni pogrešno. Potpuna tačnost dobijena je za uzorke iz klase usta žene i creva žene. Može se zaključiti da je primenjenim klasterovanjem dobijena lošija raspodela uzoraka i samim tim manja ukupna tačnost nego prilikom klasifikacije.

Drugo istraživanje [11] zasnovano je, takođe, na analizi tabele formirane u QIIME paketu i klasifikaciju pomoću *Random Forest* klasifikatora. U ovom slučaju, uzorci koji pripadaju klasi kože žene raspoređeni su u odgovarajuću klasu sa manjom tačnošću od ostalih. Za razliku od rezultata dobijenih obradom podataka u *Mothur*-u, za klasu usta žene klasifikacija svih uzoraka je dala tačan rezultat, odnosno nije bilo uzoraka iz ove klase koji su netačno klasifikovani. Tačnost dodeljivanja uzoraka iz određene klase u odgovarajuću klasu, kao i ukupna tačnost u ovim podacima malo je veća nego tačnost dobijena obradom podataka u *Mothur*-u i iznosi 96,29%. OTU tabela dobijena obradom u QIIME paketu bila je dimenzija 1967 x 50963. Može se zaključiti da je obrada podataka u QIIME paketu uz klasifikaciju uzoraka dala za nijansu bolje rezultate od obrade u *Mothur*-u.

Očigledno je da nelinearnost koja se unosi prilikom klasifikacije podataka doprinosi većoj tačnosti. Sa druge strane, algoritmi nenadgledanog učenja su od važnosti s obzirom na sve manji broj labeliranih podataka naspram dostupnih podataka generisanih naprednim tehnologijama. Unapređenje tačnosti grupisanja uzoraka u nenadgledanom scenariju je stoga od interesa ne samo u oblasti računarske biologije, već i uopšte.

5. ZAKLJUČAK

Rezultati istraživanja ukazali su na uska grla u resursima raspoloživim za ovu analizu. Iako se *Mothur* paket oslanja na C++ i ostvaruje značajne uštede u vremenu, složenost realizovanih algoritama iziskuje i vremenske i računarske resurse. Rezultati klasifikacije uzoraka pokazali su da je *Random Forest* dobar izbor za ovu vrstu problema i da daje bolje rezultate od konsenzus klasterovanja korišćenog u drugom istraživanju. Takođe, na osnovu dobijenih rezultata može se zaključiti da obrada ovih podataka u različitim paketima – *Mothur* i QIIME, kao i veličina OTU tabele, odnosno broj OTU jedinica, nije uticao na postignutu tačnost klasifikacije, ali da izvesna razlika postoji jer se greške klasifikacije nisu javljale na uzorcima istog staništa. Utvrđeno je da optimalan broj stabala *RF* klasifikatora iznosi 100, a najlošiji rezultati prilikom klasifikacije dobijeni su za uzorke iz klase koža kod žena, za koju je pokazano i da ima najmanji ukupan

broj sekvenci koje joj pripadaju. S druge strane, uzorci iz klase usta i creva žene imaju najvišu tačnost prilikom klasifikacije. Buduća istraživanja mogla bi obuhvatiti formiranje OTU tabela sa različitim stepenom sličnosti među sekvencama, kao i korišćenje drugih klasifikatora i algoritama za klasterovanje.

6. LITERATURA

- [1] J. Qin, et al., "A human gut microbial gene catalogue established by metagenomic sequencing," *nature*, 464.7285: 59, 2010.
- [2] P. D. Schloss, et al. "Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities", *Applied and environmental microbiology* 75.23, p.7537-7541, 2009.
- [3] <https://www.arb-silva.de/documentation/release-132/> (pristupljeno u septembru 2018.)
- [4] J. G. Caporaso, C. L. Lauber, E. K. Costello, D. Berg-Lyons, A. Gonzalez, J. Stombaugh, D. Knights, P. Gajer, J. Ravel, N. Fierer et al., "Moving pictures of the human microbiome," *Genome Biol*, vol. 12, no. 5, 2011.
- [5] <http://qiime.org/> (pristupljeno u septembru 2018.)
- [6] E. Plummer, et al., "A comparison of three bioinformatics pipelines for the analysis of preterm gut microbiota using 16S rRNA gene sequencing data," *Journal of Proteomics & Bioinformatics*, 8.12: 283, 2015.
- [7] <http://greengenes.secondgenome.com/> (pristupljeno u oktobru 2018.)
- [8] T. K. Ho, "Random decision forests," Document analysis and recognition, proceedings of the third international conference on. vol. 1. IEEE, 1995.
- [9] G. James, et al., "An introduction to statistical learning," New York: springer, 2013.
- [10] D. Pavlović, "Stabilna procena klastera konsenzus klasterovanjem uzoraka mikrobioma", Diplomski rad, Fakultet tehničkih nauka, Novi Sad, 2017.
- [11] S. Brdar, T. Lončar-Turukalo, V. Crnojević, B. Stres, "Clustering and classification of human microbiome data: evaluating impact of different settings in bioinformatics workflows," Book of Abstracts of the second Belgrade BioInformatics Conference – BelBi 2018, in *Biologia Serbica*, vol. 40, No1, 2018, pp. 67 Belgrade, June, 18-22, 2018.

Kratka biografija:



Aleksandra Kantar rođena je u Kikindi 1994. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Obrada signala odbranila je 2018.god.
kontakt: kantar.alex@gmail.com

**PRIKLJUČENJE BIOGASNOG POSTROJENJA NA DISTRIBUTIVNI SISTEM
SREDNJEG NAPONSKOG NIVO A****CONNECTION OF THE BIOGAS PLANT TO THE MEDIUM VOLTAGE
DISTRIBUTION SYSTEM**

Bojan Buzadžija, Zoltan Čorba, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj–U ovom radu je teorijski, praktično i eksperimentalno predstavljeno biogasno postrojenje priključenona distributivni sistem srednjeg naponskog nivoa. Na početku rada je dat teorijski uvod o obnovljivim izvorima električne energije (biogas) i Pravilima o radu distributivnog sistema. Nakon ovoga je analiziran primer priključenja biogasne elektrane na distributivni sistem električne energije i obrađeni su rezultati merenja parametara električne energije da bi se utvrdilo da li elektrana zadovoljava uslove za priključenje na distributivni sistem električne energije (DSEE).

Ključne reči: Pravila o radu DS, Parametri električne energije, Sredjenaponski nivo, Biogasna elektrana

Abstract –In this paper, the biological plant connected to the distribution system of the medium voltage level is theoretically, practically and experimentally presented. At the beginning of the paper a theoretical introduction on renewable electricity sources was given (biogas) and the operating rules of the distribution system. After this, an example of the connection of the biogas plant to the distribution system of electricity is analyzed and the results of the measurement of the parameters of electrical energy were processed to determine whether the plant meets the requirements for connection to the distribution system of electricity (DSE).

Key words: Rules on the operation of the distribution system, Parameters of electrical energy, Medium voltage level, Biogas power plant

1. UVOD

Postoji više načina da se iz obnovljivih izvora dobije električna energija. To su energija vodotokova, vetra, sunčeva energija, biomasa, geotermalna energija i druge. Od svih navedenih, u ovom radu je akcenat na procesu dobijanja električne energije iz energije biogasa [1].

Biogasna elektrana za priključenje na distributivni sistem srednjeg naponskog nivoa, mora da zadovolji uslove iz Pravilnika o radu distributivnog sistema i Tehničke preporuke Elektroprivrede Srbije. Gas koji se oslobađa iz fermentora, pokreće turbine gasnog generatora koji proizvodi električnu energiju. Proizvedena energija se preko niskonaponskog (NN) i visokonaponskog (VN) postrojenja u transformatorskoj stanici (TS) elektrane i

prenosnog odnosa transformatora, prenosi u mrežu. Mesto priključenja elektrane na DSEE je opremljen 20 kV rasklopnim postrojenjem, daljinskom stanicom i obračunskim merenjem. Naveden je konkretan primer jedne biogasne elektrane, izvršeno je merenje parametara električne energije u niskonaponskom bloku koji se nalazi u TS elektrane i iz dobijenih rezultata su izvučeni zaključci, da li dobijene vrednosti odgovaraju parametrima koji su navedeni kao uslov za priključenje na DSEE.

**2. PRAVILA I PREPORUKE PRIKLJUČENJA
MALE ELEKTRANE NA SREDNJI NAPONSKI
NIVO**

Pravila i preporuke priključenja male elektrane na DS propisuje nadležni operator DS. Operator DS (ODS) propisuje uslove za priključenje kako bi obezbedio siguran paralelan rad DS i male elektrane (ME).

Za priključenje i bezbedan paralelan rad ME sa DS, elektrana mora da zadovolji kriterijum maksimalno dozvoljene snage generatora u elektrani, kriterijum dozvoljenih vrednosti napona u stacionarnom režimu, kriterijum dozvoljenog strujnog opterećenja elemenata distributivne mreže, kriterijum snage kratkog spoja (KS), kriterijum flikera i kriterijum dozvoljenih struja viših harmonika i interharmonika.

Kriterijumom maksimalno dozvoljene snage generatora u elektrani sa energetske pretvaračima ili asinhronim generatorima proverava se da pri uključenju generatora promena napona na mestu priključenja na DS ne prekorači vrednost od 2% na SN, odnosno 3% na NN. Maksimalno dozvoljena prividna snaga generatora u elektrani (S_{ngm}) u [MVA] izračunava se prema sledećim formulama: za priključenje na SN,

$$S_{ngm} = \frac{S_{KS}}{50 * k} \quad (1)$$

za priključenje na NN,

$$S_{ngm} = \frac{S_{KS}}{3,33 * k} \quad (2)$$

gde je S_{KS} snaga trofaznog kratkog spoja, u [MVA].

Kriterijumom dozvoljenih vrednosti napona u stacionarnom režimu proverava se tako da u okviru normalnog pogona DS, vrednost napona u bilo kojoj tački DS ostane u dopuštenim granicama.

Kriterijumom dozvoljenog strujnog opterećenja elemenata distributivne mreže se proverava, da za vreme rada elektrane, struje u elementima DS ne pređu propisane vrednosti.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Zoltan Čorba, docent.

Kriterijumom snage KS se proverava ukupna vrednost struje trofaznog KS (snaga) na mestu priključenja elektrane u pogonu. Vrednost struje ne sme preći maksimalne dozvoljene vrednosti struja KS na koje je dimenzionisana oprema u DS.

Kad je **kriterijum flikera** u pitanju elektrana sa n generatora ukupne snage S_E može da se priključi na DS ako je ispunjen uslov:

$$A_{It} = (c_{fE} * \frac{S_E}{S_{KS}})^3 = (\frac{c_{fg}}{\sqrt{n}} * \frac{S_E}{S_{KS}})^3 \leq 0,1 \quad (3)$$

gde je: A_{It} dugotrajni faktor smetnji elektrane, c_{fE} koeficijent flikera elektrane i c_{fg} koeficijent flikera generatora. Koeficijent flikera označava osobinu elektrane da proizvodi flikere.

Kriterijum dozvoljenih struja viših harmonikase proverava pomoću izraza:

$$I_{vhg} \leq I_{vhdoz} = I_{vhs} \nu \mu * S_{KS} \quad (4)$$

gde je: I_{vhg} stvarna i I_{vhdoz} dozvoljena vrednost struje višeg harmonika u [A], a $I_{vhs} \nu \mu$ je dozvoljena vrednost struje višeg harmonika svedena na jediničnu snagu KS na mestu priključenja na DS u [A/MVA].

Dozvoljene vrednosti struja viših harmonika svedenih na snagu KS su date u tabeli 1. Ugradnjom odgovarajućih zaštitnih i drugih tehničkih uređaja u elektrani, treba obezbediti da uključanje elektrane na DS bude izvršeno samo ako je na svim faznim provodnicima prisutan napon mreže sa strane DS.

Elektrana se povezuje na DS samo preko jednog priključka izuzimajući priključak opšte potrošnje elektrane i ostale potrošnje na lokaciji elektrane.

Kada je u pitanju zaštita generatora i elektrane postoje sistemski zaštita, zaštita od ostrvskog rada i zaštita voda elektrane.

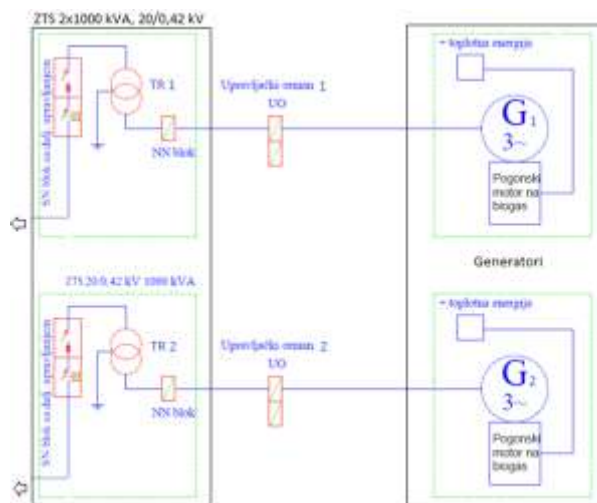
Faktor snage elektrane u odnosu na DS treba da je veći od 0,95 u kapacitivnom i induktivnom režimu rada.

Elektrana mora biti opremljena odgovarajućom opremom, softverom i komunikacionim servisom radi dostavljanja neophodnih podataka, mora da se obezbedi oprema koja će omogućiti prenos informacija do nadležnog centra upravljanja ODS.

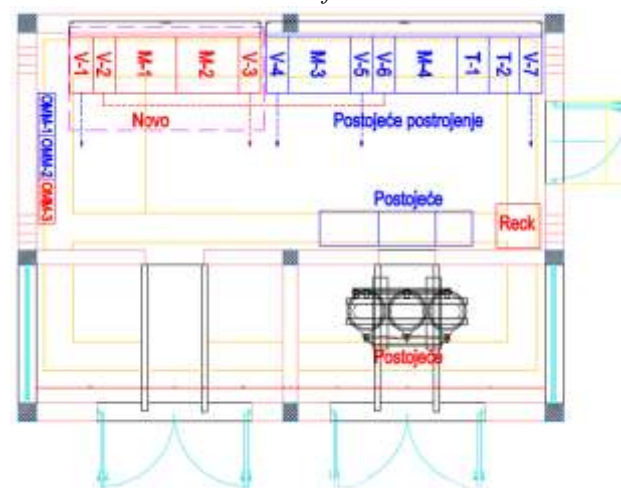
3. PRIMER PRIKLJUČENJA BIOGASNOG POSTROJENJA NA DISTRIBUTIVNI SISTEM SREDNJEG NAPONSKOG NIVOA

Primer priključenja biogasnog postrojenja na DS je prikazan na elektrani koja je izgrađena u Čurugu, koristeći podatke, informacije i iskustva na izgradnji tog postrojenja. Na slici 1. prikazana je šema generatora elektrane i ZTS „Velvet farm ME“. Prikazani su 20 kV vodovi za priključenje elektrane na DSEE, novo prefabrikovano 20 kV razvodno postrojenje u SF6 tehnici proizvođača „SIEMENS“, NN razvod za napajanje sistema daljinskog upravljanja i nadzora (SDUN). Izgled 20 kV razvodnog postrojenja je dat na slici 2. U okviru postrojenja se nalaze: V-1 vodna ćelija prema ČRS-u br. 35 „Izvod Čurug“, V-2 spojna ćelija, M-1 napajanje RTU daljinske stanice, M-2 obračunsko merenje OMM3, V-3 TS „Velvet farm ME“ Generator 1, V-4 TS „Velvet farm ME“ Generator 2, M-3 obračunsko merenje OMM2, V-5

MBTS „Budućnost“, V-6 spojna ćelija, M-4 obračunsko merenje OMM1, T-1 trafo polje 1, T-2 trafo polje 2 i V-7 sopstvena potrošnja TS „Velvet farm ME“.



Slika 1. Šematski prikaz generatora biogasne elektrane i ZTS „Velvet farm ME“



Slika 2. TS „Velvet farm“ 20/0,4 kV/kV (I i II faza izgradnje)

Nazivni naponi U_n distributivne mreže, na koju se priključuje ME je $U_n = 20$ kV, dok je vrednosti maksimalne dozvoljene struje (snage) trofaznog KS i struje zemljospoja od 14,5 kA (500 MVA). Stvarna struja trofaznog KS sa strane DSEE na mestu priključenja elektrane na DSEE, u subtranzijentnom periodu je $I_{KS} = 1,316$ kA, odnos $R/X = 0,803$. Elektroenergetska oprema u DSEE na 20 kV naponu je dimenzionisana na dozvoljenu strujutrofaznog kratkog spoja 14,5 kA.

Distributivna mreža 20 kV ima radijalno napajanje iz prenosnog sistema i uvođenjem ME ona više nije radijalno napajana. Uzemljenje neutralne tačke distributivne mreže 20 kV, izvode se tako što je neutralna tačka mreže 20 kV izolovana ili uzemljena preko niskoomske impedance.

Sinhroni generator pokreće se preko elastičnog gasnog motora. Generator se izrađuje kao samoregulišući sinhroni generator i sa potrebnom opremom i pomoćnim komponentama. Podaci o generatoru: kogeneraciono postrojenje J 312 GS-D25 Stamford CG634, pogonska snaga 788 kVA odnosno 1134 A pri $\cos \varphi = 0,8$,

nominalan aktivan učinak 635 kW, napon 400 V, frekvencija 50 Hz, broj obrtaja 1500 ⁰/min, vrsta zaštite IP 23 i klasa izolacije H.

Rasklopna oprema u ćelijama novog 20 kV postrojenja u OMP je u skladu sa koncepcijom ODS-a, rasklopni aparati su daljinski upravljani.

U objektu mesta priključenja (OMP) je ugrađena daljinska stanica i u neposrednoj blizini je postavljen novi antenski stub, na koji je montirana antena radi komunikacije sa nadređenim dispečerskim centrom, a koja je povezana sa daljinskom stanicom. Komunikacija sa daljinskom stanicom se realizuje se komunikacionim IEC 61850 putem fiberoptičkog kabla, a može se realizovati i po principu Scada-Scada.

U sklopu priključka je ugrađen merni uređaj za obačunsko merenje primopredaje električne energije između ME i DSEE, koji se smešta u orman mernog mesta tipa MOMM-PI/2 i povezuje se sa mernim transformatorima u mernoj ćeliji. Navedeni orman mernog mesta se montira na zid unutar OMP [3, 4].

4. ANALIZA MERENJA PARAMETARA ELEKTRIČNE ENERGIJE BIOGASNOG POSTROJENJA

Sa uređajem za merenje kvaliteta električne energije izvršeno je merenje u bloku niskog napona (BNN) biogasne elektrane.

Po priključenju uređaja za merenje memorisane su trenutne vrednosti napona i struje, sva tri napona istovremeno, sve tri struje istovremeno i struje i naponi po fazama.

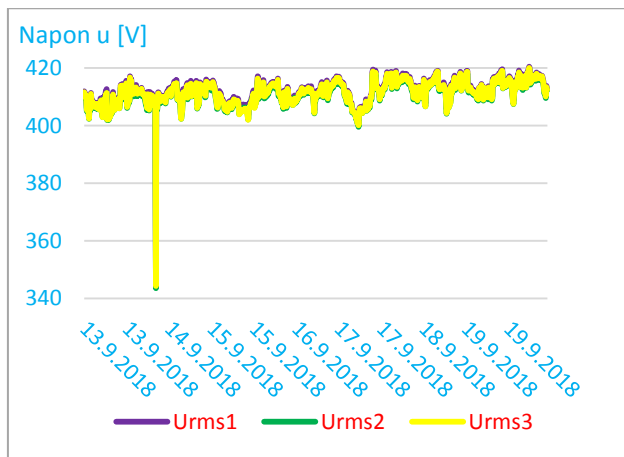
Na slici 3. prikazano je priključenje mernog uređaja u NN bloku TS male elektrane (ME).



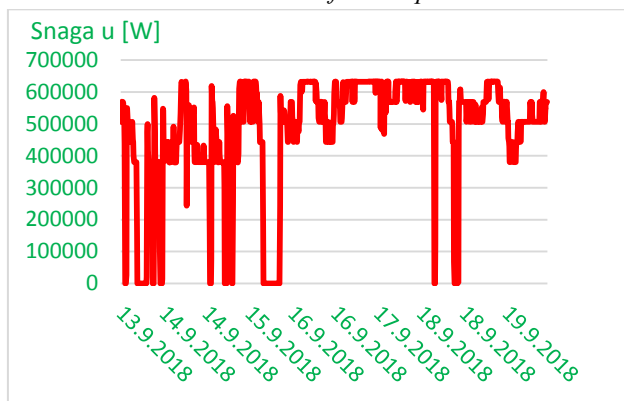
Slika 3. Priključenje uređaja za merenje

Zabeleženi su sledeći parametri električne energije: frekvencija, međufazni napon sve tri faze, struja sve tri faze, ukupno harmonijsko izobličenje napona THDU, ukupno harmonijsko izobličenje struje THDI, aktivna snaga (W_{total}), faktor snage (PF , PF_{mean}) i harmonike RMS_2 (od 2-25 harmonika).

Na slici 4. prikazan je međufazni napon, dok je na slici 5. prikazana aktivna snaga.

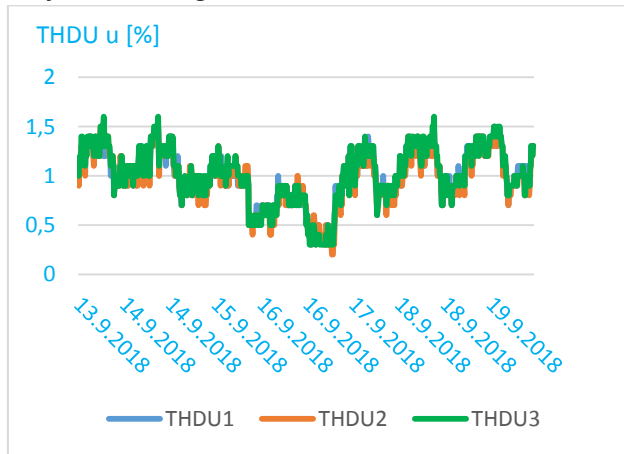


Slika 4. Međufazni napon

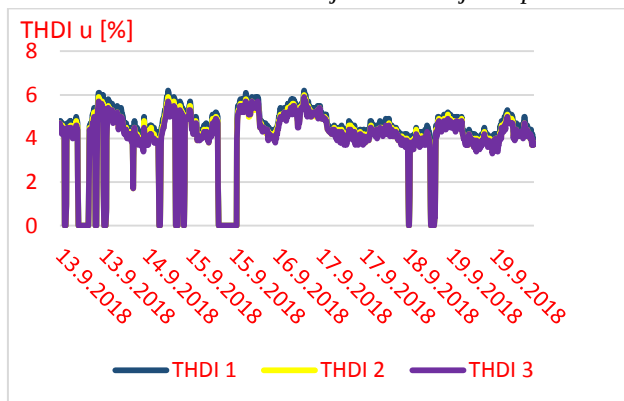


Slika 5. Aktivna snaga

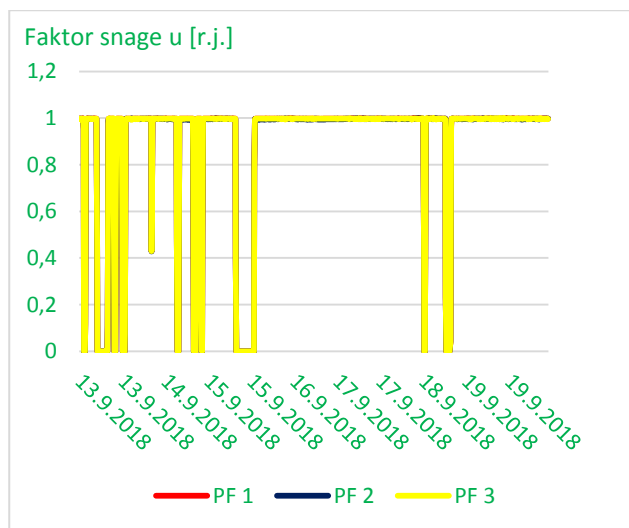
Na slici 6, 7 i 8 su redom prikazane THD napona, THD struje i faktor snage.



Slika 6. Totalna harmonijska distorzija napona



Slika 7. Totalna harmonijska distorzija struje



Slika 8. Faktor snage za sve tri faze

Prikazani grafici su dobijeni kao rezultat merenja parametara električne energije u ME. Maksimalna vrednost napona na sve tri faze iznosi 420,5V, a minimalna 343,6 V. Minimalna vrednost napona je zabeležena 14.09.2018. u 08:50h, što nam govori da je u tom trenutku došlo do propada napona ili kratkotrajnog prekida pošto je usrednjavanje na 10 minuta. Izuzimajući taj propad napona, minimalna vrednost napona iznosi 399,6 V, što je prihvatljivih $\pm 10\%$ U_n . Maksimalna vrednost aktivne snage iznosi 633,314 kW. Maksimalna vrednost THDU iznosi 1,6 %, dok je minimalna vrednost 0,2 %. Vrednosti THDU su manje od 8%, što ispunjava propisane standarde SRPS EN 50160. Maksimalna vrednost totalne harmonijske distorzije struje iznosi 6,2%.

4.1. Provera kriterijuma harmonijskih izobličenja

Upoređivanjem rezultata merenja stvarne vrednosti struje generatora I_{vhg} sa dozvoljenom vrednošću struje na mestu priključenja I_{vhdoz} , koja se dobija proizvodom dozvoljenih vrednosti struja za naznačeni naponski nivo $U_n=0,4$ kV i snage KS generatora $S_{KS}=\sqrt{3} \cdot U_n \cdot I_{KS}$ za $I_{KS}=1,316$ kA.

Iz dobijenih vrednosti prikazanih u tabeli 1. dolazimo do zaključka da izmerene vrednosti struja viših harmonika su manje i/ili jednake sa maksimalnim dozvoljenim strujama viših harmonika:

$$I_{vhg} \leq I_{vhdoz} \quad (5)$$

Znak + u tabela 1. predstavlja pozitivan rezultat, tj. ispunjenje uslova kriterijuma dozvoljenih vrednosti harmonika struje. Kako 5. harmonik ne ispunjava uslov, neophodan je filter za 5. harmonik ili da se izvrši ponovno merenje parametara električne energije radi ponovne provere ispunjenja kriterijuma.

Tabela 1. Dozvoljene i stvarne vrednosti struja viših harmonika

Harmonik	$I_{vhs \nu \mu}$ [A]	$I_{vhdozmax}$ [A]	I_{vhg} [A]	Komentar
2	0,029	1,321	0,005	+
3	/	/	0,318	+
4	0,009	0,410	0,000	+
5	0,029	1,321	1,740	-
6	0,012	0,546	0,000	+
7	0,041	1,867	0,249	+

8	0,004	0,182	0,000	+
9	/	/	0,030	+
10	0,007	0,319	0,000	+
11	0,026	1,834	0,094	+
12	0,005	0,228	0,000	+
13	0,019	0,865	0,039	+
14	0,003	0,137	0,000	+
15	/	/	0,000	/
16	0,003	0,137	0,000	+
17	0,011	0,501	0,025	+
18	0,002	0,091	0,000	+
19	0,009	0,410	0,000	+
20	/	/	0,000	/
21	/	/	0,000	/
22	/	/	0,000	/
23	0,006	0,273	0,000	+
24	/	/	0,000	/
25	0,005	0,228	0,000	+

5. ZAKLJUČAK

Izgradnjom ME se poboljšava pouzdanost i kvalitet EES. Za pravilno funkcionisanje novoizgrađenih ME i TS za priključenje na DSEE, obavezno je poštovanje svih obrađenih uslova iz pravila o radu DS i propisa. Izgrađene TS i oprema su u skladu sa propisanim standardima, ispunjavaju sve uslove zaštite, kvaliteta, sigurnosti i pouzdanosti. Analizom parametara i poređenjem sa pravilima, došlo se do zaključka da ME ispunjava uslove za bezbedan rad i za bezbedno priključenje na DSEE srednjeg naponskog nivoa.

7. LITERATURA

- [1] Oficijalni sajt kompanije "Global Seed" DOO, Čurug (poslovni sistem) <http://www.globalseed.info/obnovljivi-izvori-energije.php>
- [2] Pravila o radu distributivnog sistema - "EPS Distribucija" DOO Beograd
- [3] Uslovi za projektovanje i priključenje, "EPS Distribucija" DOO Beograd, Ogranak "Elektrodistribucija Novi Sad", Pogon Žabalj
- [4] Idejni projekat IDP G-520 od Februara 2018. Izgradnja II faze, rekonstrukcija

Kratka biografija:



Bojan Buzadžijaj je rođen u Novom Sadu 1990. godine. Diplomski rad na Fakultetu Tehničkih Nauka u Novom Sadu iz oblasti Elektrotehnika i Računarstvo - energetska elektronika i električne mašine odbranio je 2016. godine.



Zoltan Čorba je docent na Fakultetu tehničkih nauka u Novom Sadu, na katedri za Energetsku elektroniku i pretvarače. Oblast interesovanja su obnovljivi izvori i kvalitet električne energije.

PROVERA FUNKCIONALNOSTI RADA RELEJA U SREDNJENAPONSKIM NEBALANSIRANIM DISTRIBUTIVNIM MREŽAMA**CHECKING FUNCTIONALITY OF THE RELAY OPERATION IN MEDIUM-VOLTAGE UNBALANCED DISTRIBUTED NETWORKS**

Nemanja Mihajlov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je dat pregled relejnih zaštita koje se koriste u srednjenaponskim nebalansiranim distributivnim mrežama. Opisane su distantna i prekostrujna zaštita sa svojim karakteristikama. Kvarovi koji se dešavaju na srednjem naponu, kao i primeri podešenja distantnih i prekostrujnih releja.

Ključne reči: Rad releja, relejna zaštita, kvarovi, nebalansirane mreže

Abstract – In this paper is given an overview of relay protection used in medium-voltage unbalanced distribution networks. Distant and overcurrent protection with its characteristics are described. Middle-voltage faults as well as examples of distant and overcurrent relay settings.

Keywords: Relay operation, relay protection, faults, unbalanced networks

1. UVOD

Funkcionalnost rada releja smatra se jednom od najbitnijih karakteristika savremenih elektroenergetskih sistema (EES). Esencijalnost ispravnog rada releja ogleda se u tome da svi planirani i neplanirani kvarovi kao i opasna pogonska stanja, koja remete sigurnost i pouzdanost EES, budu na vreme otklonjeni. Time se omogućava nesmetan rad mreže u kojoj su instalisani releji. Srednjenaponske distributivne mreže odlikuju se radijalnim ili slaboupetljanim pogonom, gde se različite vrste relejnih zaštita primenjuju.

Kada je reč o nebalansiranim mrežama, tipični predstavnici ovih mreža su severnoameričke distributivne mreže. Transformatori sa specifičnim spregama (sprežnim odnosom) i potrošači sa svojim zahtevima za isporukom električne energije, čine nestandardnim ove EES u odnosu na evropske mreže, te se s toga poseban tretman nad ovim mrežama i primenjuje.

Relejna zaštita i provere u ovim mrežama zahtevaju: brzinu, koordinaciju, preciznost, tačnost i usmerenje. Iza svega navedenog, krije se atraktivnost u podešenju rada relejne zaštite u ovim mrežama (prim. aut. severnoameričkim).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Duško Bekut, red. prof.

Nakon uvoda, u drugoj glavi date su teorijske osnove funkcionalnosti relejne zaštite, ispitane na prekostrujnoj i distantnoj zaštiti.

U trećoj glavi rada opisane su srednjenaponske nebalansirane distributivne mreže, sa vrstama kvarova koji se pojavljuju u njima. U prvom delu je predstavljena struktura distributivnih sistema, prednosti i mane evropskih i severnoameričkih sistema. U drugom delu predstavljeni su transformatori sa specifičnim sprežnim odnosom kao i regulatori napona sa svojim funkcionalnostima.

U četvrtoj glavi dat je praktičan primer transformatorske stanice sa prekostrujnim relejima. Ispitivanja funkcionalnosti prekostrujne zaštite odrađena su preko softverskih rešenja.

Nakon zaključka iznetog u petoj glavi, navedena je literatura, korišćena za realizaciju ovog rada.

2. TEORIJSKE OSNOVE

Distributivni sistemi su izloženi toku struje kroz elemente EES koja može da pređe nominalne vrednosti. Razlog zašto struje izlaze izvan svojih zadatih limita mogu biti zbog preopterećenja određenih elemenata (npr. vodova) i kratkih spojeva ili startovanja motora.

Stoga, upravljanje potrošnjom, (*Demand-Side management*), kao i krive preopterećenja ili soft startovanje motora, neki su od načina za izbegavanje potencijalnog preopterećenja [1].

Dodatno, distributivne mreže opremljene su zaštitnim relejima koji daju komande prekidačkoj opremi za određene akcije u cilju suzbijanja abnormalnih sistemskih uslova.

Optimalna podešenja zaštite potrebna su za radne uslove elektroenergetskog sistema, kako bi se osigurao rad unutar unapred postavljenih zahteva za pouzdanim snabdevanjem električnom energijom i sigurnošću elemenata mreže. Neophodna je zaštita opreme, osoblja, ljudi i mreže generalno govoreći.

Međutim, trenutna podešenja koja se koriste u EES, često nisu na optimalnom nivou. Mreža često nije optimalna za određeno stanje sistema, te je s toga neophodno da se podešenje relejne zaštite prilagodi stvarnim uslovima koji se menjaju. Kada se promeni topologija mreže, podešenja zaštitnih uređaja treba što pre ažurirati kako bi se zaštitio sistem od novih potencijalnih kvarova.

2.1. Prekostrujna relejna zaštita

Svaki prekostrujni relej da bi ispravno funkcionisao mora biti energizovan. Ova vrsta energije može da se obezbedi iz strujnih transformatora (obezbeđuje sistem) ili iz kondenzatorskih uređaja, kada su u pitanju manji sistemi, dok se kod velikih sistema energizacija obezbeđuje u vidu posebne baterije. Radna struja za sve tipove prekostrujnih releja može biti ili podešena ili podesiva.

Kontakti releja se zatvaraju kada se predviđena podešena vrednost struje premaši. Tada kontakti releja iniciraju strujni prekidač da odseče vrednost struje koja se pojavila. U zavisnosti od toga koju brzinu želimo da postignemo sa datim prekostrujnim relejom, razlikujemo tri vrste strujnih karakteristika:

- Trenutna strujna karakteristika.
- Strujno-vremenska karakteristika.
- Inverzna vremenska karakteristika.

2.2. Prekostrujna zaštita za kvarove sa zemljom

Kvarovi sa zemljom su najčešći tip kvara koji se pojavljuje u distributivnim mrežama. S toga, mora postojati koncept šticećenja od kvarova nastalih sa zemljom. Kod takvih kvarova, deo struje putuje kroz zemlju, a vrednost struje zavisi od konfiguracije mreže, tipa kvara (npr. jednopolni sa zemljom, dvopolni sa zemljom), lokacije kvara, i tipa sistema uzemljenja. Ova vrednost struje kvara predstavlja sumu faznih struja koja nije jednaka nuli, kao u normalnim uravnoteženim uslovima sistema. Kod nebalansiranih mreža ta struja je poznata i kao struja ostataka (reziduala) i to iskazana formulom:

$$I_{res} = \sum I_{ph} = I_{ph1} + I_{ph2} + I_{ph3} > 0 \quad (1)$$

gde su:

I_{ph1} – struja u fazi 1,

I_{ph2} – struja u fazi 2,

I_{ph3} – struja u fazi 3,

I_{ph} – ukupna vrednost struje.

Kvarovi sa zemljom – zemljospoji funkcionišu na sličan način kao i prekostrujna zaštita namenjena za fazne kvarove. Kada se operativna zona rada releja, iskazana karakteristikom releja, nalazi u stanju funkcionalnosti – tada će biti omogućen nesmetan rad releja koji su namenjeni za kvarove sa zemljom. Neophodno je da zemljospojna zaštita bude vrlo osetljiva na struje kvara, koje mogu imati niske vrednosti, u slučajevima kada su impedansa kvara ili impedansa uzemljenja velikih vrednosti. S druge strane, prag struje podešenja kod kvarova sa zemljom treba biti iznad vrednosti nulte struje, koja proizilazi iz dozvoljenih radnih uslova neuravnoteženosti u mreži, i na taj način izbeći pogrešno okidanje releja.

2.3. Distantna zaštita

Rad distantnih releja nije regulisan striktno strujom ili snagom, već zavisi od odnosa primenjenog napona i struje. Releji stoga efektivno mere impedansu u zoni koju štite. U uslovima kvara, napon je degradiran, i tada se struja u velikom intenzitetu povećava. Ovo se detektuje distantnim relejom kada impedansa opadne tokom kvara. Iniciran je prekid napajanja, koji se obavlja slanjem

signala strujnom prekidaču da prekine strujno kolo. Ovaj tip releja (*distance relays*) koristi se u nadzemnim i kablovskim mrežama. Prednost im se ogleda u brzini, kada druge zaštite nisu u mogućnosti da ispoštuju zahtevane vremenske uslove. Impedansom može da se odredi dužina nadzemnog voda ili kabla, te se distantna zaštita naziva i zaštita distantom impedansom.

2.4. Karakteristike distantnih releja

Kada se analizira impedantna karakteristika distantnog releja, tada se može definisati i izraziti kao jednostavna karakteristika koja se nalazi u (R–X) ravni. Na apscisi se nalazi otpornost (R) izražena u [Ohm], a na ordinati se preslikavaju vrednosti reaktanse (X) koja se, takođe, izražava u [Ohm]. Distantni relej sa ovom karakteristikom je osetljiv samo na modul merene impedanse. Velika mana je što ne postoji usmerenje u radu releja sa ovom karakteristikom. Usmerenje se mora obezbediti na neki dodatni način, i upravo se ovim favorizuju ugaono admitantni releji koji se najčešće koriste kada je zahtevano usmerenje releja u mreži.

Linija koja predstavlja liniju voda se pomera po MHO dijagramu (ugaono admitantnoj karakteristici) u zavisnosti od situacije, za različite vrste simuliranih kvarova. Tako za trofazne kvarove koji su u sve tri faze zahvaćeni simetrično, relej će pratiti standardnu MHO karakteristiku. Što je veća impedansa izvora (*source impedance*), veći je i krug u R–X ravni, kome je tim postupkom povećan kapacitet za toleranciju otpornosti – uticaj rezistanse, a baš upravo postojanje otpora na mestu kvara predstavlja problem pri radu distantnih releja.

3. ZAŠTITA NEBALANSIRANIH MREŽA

Razmatraće se specifičnosti i razlike između evropskih i severno-američkih distributivnih mreža. Severno-američki i evropski tip mreže su dva sistema distributivnih naponskih nivoa koji se najčešće koriste u celom svetu.

3.1. Struktura distributivnih sistema

Opsezi naponskih nivoa koji se koriste u gore navedenim EES prikazani su u Tabeli 1.

Izbor napona zavisi od tipa opterećenja (kućna, komercijalna ili industrijska potrošnja). Takođe zavisi i od veličine opterećenja, kao i udaljenosti na kojoj je locirano dato opterećenje [1].

Tabela 1. – Distributivni naponi Severne Amerike i Evrope

Naponski tip	Severno-američki sistem	Evropski sistem
Primarni distributivni napon (faza–faza)	Od 4 do 35 kV	Od 6.6 do 33kV
Trofazni sekundarni napon (faza–faza)	208, 480 ili 600V	380,400 ili 416V
Monofazni sekundarni napon (faza sa zemljom)	120/240, 277 ili 347V	220,230 ili 240V

Primetno je da su distributivni naponi u evropskim sistemima viši nego u severno-američkim sistemima. To nosi određene prednosti i mane. Prednosti: Sistem može

da podnese više prenesene snage, imaće manje padove napona i manje gubitke za dati tok snage. Posledično, sistem može da pokrije mnogo šire oblasti. Zbog velikog dometa dalekovoda, sistem zahteva manje transformatorskih stanica u distribuciji.

Mane: Previše prekida napajanja potrošačima sledi usled problema na dalekovodima, odnosno povećan je broj kvarova, što posledično smanjuje pouzdanost. Stoga, glavni fokus je na održavanju željenog opterećenja shodno naponskim profilima.

3.2. Kvarovi u nebalansiranim mrežama

Većina električnih kvarova u distributivnim elektroenergetskim sistemima (EES), obezbeđuje pored značajne predate snage i gubitke koji su značajni, ukoliko kvar nije na vreme otklonjen iz sistema. Kada je kvar detektovan relejnom zaštitom u sistemu, prekidači će odvojiti deo mreže sa kvarom od ispravnog pogona. Zaštita je stoga ekstremno važna za normalan rad u bilo kom EES. Releji moraju da ispunjavaju zadata očekivanja u smislu podešenja i tačnosti reagovanja, u suprotnom kvar će opstati i doći će do dodatnih prekida ili oštećenja. Takođe, relejna zaštita ne sme da reaguje kada to od nje nije zatraženo. U tim situacijama dolazi do potencijalnog ugrožavanja toka struje i pojaviće se eventualni kvarovi na tom delu mreže, umesto ispravnih, tj. „zdravih“ faza struje.

Kvarovi se mogu podeliti na:

- kvarove između faza (međufazne),
- kvarove sa zemljom (zemljospoje).

Vrednost fazne struje kvara najčešće zavisi od impedanse izvora (*source-a*) i impedansi na putu do mesta kvara. Međutim, kada je reč o kvarovima sa zemljom, tu je bitan još jedan važan faktor, a to je način na koji je sistem uzemljen tj. proverava da li je uopšte mreža uzemljena. Kako se impedansa u nultom režimu povećava, tako struja zemljospoja smanjuje svoju vrednost, te je s toga teško detektovati ovu vrstu kvara. Dispečeri uprkos ovom problemu, često se susreću sa veoma visokom vrednošću impedanse uzemljenja nultog režima.

U nesimetričnom sistemu nije moguće, kao u simetričnom, prilike u mreži odrediti posmatranjem stanja samo u jednoj fazi, pa nije moguće ni trofaznu mrežu neposredno zameniti s ekvivalentnom jednofaznom šemom, a uz to, određivanje prilika na osnovu trofazne šeme zahteva duge proračune. Ovakvi režimi se računaju metodom simetričnih komponenti. Simetrične komponente omogućavaju da se nesimetrični trofazni sistem raspregne na tri sistema (direktni, inverzni i nulti), koji se analiziraju metodama za analizu monofaznih sistema [2].

3.3. Specifična Y-Y sprega transformatora

Konfiguracija ovih transformatora je sledeća: Trofazni su, sa tri stuba (trostubni), sa magnetnim jezgrom, Y-Y sprega transformatora ili u posebnim slučajevima autotransformator. Problem koji se razmatra je sledeći:

U transformatoru povezanosti namotaja Yy (zvezda-zvezda), sa jezgrom, sekundarni namotaj se ponaša kao neuzemljena zvezda, odnosno kao Δ (trougao) sa visokom impedansom. Potencijalna oštećenja mogu nastati usled struje cirkulacije u ovom slučaju. Prekostrujni relej je energizovan preko strujnih transformatora povezanih tako

da uvećavaju efektivnu tercijarnu struju. U slučaju dvonamotajnih transformatora, neophodna nulta sekvenca je obezbeđena povezivanjem visoke i niskonaponske strane strujnih transformatora [3, 4].

3.4. Regulatori napona

Funkcija regulatora napona ogleda se u tome da održe vrednosti naponskih prilika u zadatim granicama. Tokom pada napona, regulatori napona povećavaju vrednost napona do željenog nivoa, osetljivog opterećenja i u obratnom slučaju tokom prenapona, imaju funkciju da smanjuju vrednosti napona. Najčešće, pri upotrebi regulatora napona u praksi, nalazi se na situaciju kada treba da se smanji vrednost pada napona. Dva ključna uslova za postizanje zadovoljenja potrošača moraju biti razmatrana. Prvi je kontinualno napajanje električnom energijom od strane distributivnog preduzeća. Drugi je dobar kvalitet električne energije koji je obezbeđen između ostalog regulatorima napona, prekidačkom opremom, kao i uređajima za poboljšanje kvaliteta električne energije (VAR kompenzatori, STATCOM-i itd).

Napon na različitim terminalima u mreži mora biti održavan na konstantnim vrednostima aproksimativno. Dozvoljene varijacije limita zavise od IEC-a (*International Electrotechnical Commission*) ili od IEEE standarda.

Ukoliko se želi postizanje snabdevanja potrošača konstantnom amplitudom napona i bez narušenih naponskih limita, cena električne energije sa proizvođačke strane biće znatno povećana. S druge strane, ukoliko je napon takve prirode da narušava limite, tada se u svrhu izbegavanja plaćanja visoke cene snabdevačima električnom energijom, ulaže u prekidačku opremu. Upravo je između ove dve granične situacije potrebno naći balans i kompromis.

4. PROVERA FUNKCIONALNOSTI RELEJA

Provera efikasnosti zaštite uključuje proveru sledećih kriterijuma:

- osetljivosti,
- opterećenja,
- tajminga (vremena okidanja),
- usmerenja (smjera).

Osetljivost – proverava se da li je razmatrani relej izvan određenih limita (specificiranih vrednosti unapred zadatih), za ranije određenu vrednost tipa i lokacije kvara. Obično, kvarovi bivaju simulirani na kraju osnovne zone štice. Kada se proverava osetljivost faznih releja, tada se razmatra trolni kratak spoj bez uticaja rezistanse na luk, koji se pojavljuje na mestu kvara. U terminologiji ovi kratki spojevi, poznati su i kao metalni kratki spojevi (*solid*), čime se zanemaruje uticaj rezistanse (otpora) na mestu kvara.

U četvorožičnim distributivnim sistemima, radi uštede osigurača, dve faze pripadaju primarnoj zoni reklozera. Fazni reklozer tada će biti proveren na dvofaze greške na kraju odgovarajućih laterala. Ovo je tipično za severno-američke distributivne mreže, da mogu posedovati veliki broj bočnih izvoda, u terminologiji poznatih kao bočni laterali, ili samo laterali. Kada se razmatra situacija sa

zemljospojnom zaštitom, tada će isto kao i u slučaju fazne logike, biti zanemaren uticaj rezistanse na luk na mestu kratkog spoja.

Provera osetljivosti dalje implicira proveru opterećenja.

Opterećenje – proverava se da li je maksimalno opterećenje elementa u specificiranim limitima. Opterećenje štice elementa mora biti u okviru zadatih ograničenja kako bi se izbeglo nepotrebno delovanje (okidanje) releja u slučaju velikih opterećenja u mreži.

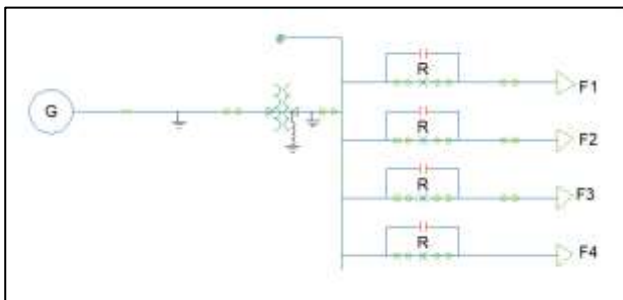
Tajming (okidanje) – proverava da li je vreme odsecanja kvara u specificiranim limitima. Vreme isključenja kvara mora biti takvo da obezbedi čišćenje kvara u što bržem vremenskom periodu, kako bi se obezbedila odgovarajuća koordinacija između zaštitnih uređaja.

Usmerenje se proverava da li je smer delovanja releja podešen na odgovarajući način, posmatrajući topologiju mreže. Provera usmerenja uključuje provere karakteristike releja, ugla releja, za direktan ili inverzan način podešenja, poredeći vrednosti na gore ili na dole, u zavisnosti od oznaka i konfiguracije fidera i reklozera u mreži.

Koordinacija nije odrađena u softverskom paketu za proveru relejne zaštite. Kada su svi uslovi iznad: osetljivost, tajming, opterećenje, kao i usmerenje (ukoliko se specifikira) u zadatim granicama, ne narušavajući prethodno zadate limite, tada će dispečer mreže imati u potpunosti operativnu konfiguraciju mreže u razmatranim održivim opsezima. Postoji opcija u softveru da li operater želi da uključi ili eventualno isključi opciju za usmerenje. Isključenjem opcije za usmerenje, postoji bojaznost koja može dovesti do neselektivnosti, i tada neselektivan rad mreže implicira neispravan rad relejne zaštite.

4.1. Praktičan primer zaštite transformatorske stanice

Na slici 4.1.1. mogu se uočiti distributivni fideri na kojima se nalazi prekostrujna relejna zaštita.



Slika 4.1.1. – Jednopolna šema sa prekostrujnim relejima

Na jednopolnoj šemi nalaze se 4 fidera koji su snabdeveni sa po 4 releja, od toga svaki relej sa po dve vrste zaštite, jednu za međufazne kvarove, i drugu za kvarove sa zemljom, kada je u pitanju prekostrujna zaštita.

Kataloški podaci transformatora, takođe, se ispituju. Broj namotaja, njegove sprege, potencijalna uzemljenja visoke i niskonaponske strane transformatora. Impedansa kratkog spoja, direktnog i nultog režima, kao i broj regulacionih otepa.

Ukoliko se svi podaci poklapaju, dati transformator sa kataloškim nazivom u smislu funkcionalnosti relejne

zaštite i prekidačke opreme zadovoljiće sa aspekta ispravnog rada releja.

Ustanovljeno je da relejna zaštita koja se nalazi na fiderima gledajući sa sekundarne strane transformatora u potpunosti odgovara podešenjima, kao i da podešeni transformator u ispravnom stanju ne remeti rad mreže.

5. ZAKLJUČAK

U radu su ispitane funkcionalnosti relejne zaštite, konkretno njenih najčešće korišćenih tipova u srednjenaponskim nebalansiranim distributivnim mrežama, a to su prekostrujni i distantni releji.

Takođe, dat je kratak osvrt i na uređaje bez kojih je nemoguće zamisliti rad releja u srednjenaponskim nebalansiranim mrežama: transformatori sa specifičnim odnosima transformacije i regulatori napona u izvedenim varijantama. Različiti tretman rada releja uzet je za određene tipove kvarova kod prekostrujne i distantne zaštite. Pristup uz određena aproksimativna zanemarenja kao što je postojanje rezistanse na mestu kvara, rađen je u svrhu ispitivanja koordinacije, osetljivosti, opterećenja, tajminga i usmerenja.

Praktičan deo rada podrazumevao je sve provere kroz softverska rešenja za datu topologiju mreže. Zapisani su specifični elementi sa jednopolnih šema i njihove vrednosti. Upravo ti elementi prolazili su kroz validacije, do zadovoljavajućih kriterijuma odnosno omogućenja nesmetanog rada mreže i pokrivenosti čitavih oblasti kojima upravljaju dispečeri.

Na samom kraju rada uporednom analizom između teorijskih osnova i praktičnog primera, ustanovljeno je da za datu transformatorsku stanicu uz svu snabdevenost elementima pored releja i prekidačke opreme, za date fidera na kojima se nalaze releji, omogućen je nesmetan rad mreže.

6. LITERATURA

- [1] Abdelhay A. Sallam, OM P. Malik – *Electric Distribution Systems Planning and Utilization*, New Jersey 2011,
- [2] Vladimir C. Strezoski – *Analiza elektroenergetskih sistema 1 i 2*, Fakultet tehničkih nauka, Novi Sad 2017,
- [3] Colin Bayliss, Brian Hardy – *Transmission and Distribution Electrical Engineering*, Massachusetts, USA 2007,
- [4] Duško Bekut – *Relejna zaštita* – Fakultet tehničkih nauka, Novi Sad 2009.

Kratka biografija:



Nemanja Mihajlov je rođen u Zrenjaninu 1994. god. Diplomirao je na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi 2017. godine i nakon toga upisao master studije na istom fakultetu.

PROJEKTOVANJE ZAŠTITE U NISKONAPONSKIM INSTALACIJAMA**DESIGN OF PROTECTION IN LOW VOLTAGE INSTALLATIONS**

Jelena Šešlija, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu se razmatra projektovanje zaštitnih uređaja u niskonaponskim instalacijama. Detaljnim opisima uređaja, proračunima i primerima opisana je postupak izbora zaštitnih uređaja kako bi instalacije električne energije bilo što bezbednije za korišćenje.

Ključne reči: Zaštitni uređaji, proračuni

Abstract – The paper deals with the design of protective devices in low voltage installations. Detailed descriptions of devices, calculations and examples describe the procedure for selecting protective devices in order to make electrical installations safer for use.

Ključne reči: protective devices, calculations

1. UVOD

Energetski sistemi su energetski objekti međusobno povezani tako da čine jedinstven tehničko-tehnološki sistem. Elektroenergetski sistemi su posebna klasa energetskih sistema, u kojima se primarna energija u svojoj prirodno raspoloživoj formi, pretvara u električnu energiju, a ova se dalje prenosi i distribuira krajnjim korisnicima. Elektroenergetski sistem u smislu Zakona o energetici [1] čine:

1. Proizvodnja električne energije, koja obuhvata proizvodnju u hidroelektranama, termoelektranama, elektranama – toplanama i elektranama na obnovljive izvore energije ili otpad.
2. Prenosni sistem električne energije, koji čini mreža visokog napona od 400 kV, 220 kV i deo mreže 110 kV, kao i drugi energetski objekti, telekomunikacioni sistem, informacioni sistem i druga infrastruktura neophodna za funkcionisanje elektroenergetskog sistema.
3. Distributivni sistem električne energije, koji čini niskonaponska mreža, sredjenaponska mreža i deo 110 kV mreže, kao i drugi energetski objekti, telekomunikacioni sistem, informacioni sistem i druga infrastruktura za funkcionisanje distributivnog sistema [1].

Da bi se koristila električna energija, potrebno je raspolagati, pored potrošača i izvora, i električnom instalacijom u zatvorenim prostorijama. Napravu koja se za svoj rad koristi električnom energijom se naziva električni prijemnik (potrošač). Električni potrošač u radu mora biti vezan (priključen) na izvor električne energije. Veza potrošača sa izvorom električne energije ima tri dela: prvi deo se nalazi na mestu proizvodnje električne energije, u električnoj centrali (elektrani) i naziva se električnim postrojenjem; drugi deo povezuje mesto proizvodnje sa mestom potrošnje, to je električna mreža; treći se deo nalazi na mestu potrošnje u zgradi i naziva se električna instalacija.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je dr Duško Bekut, red.prof.

Kako bi se električne instalacije projektovale, izvodile i koristile tako da ne ugrožavaju ni ljudske živote, ni opremu ni samu instalaciju potrebno je koristiti zaštitne uređaje koji obezbeđuje sigurno i pouzdano korišćenje električne instalacije. Projektovanjem se mogu baviti lica koja imaju potrebno znanje i dozvolu (licencu Inženjerske komore Srbije). Prilikom projektovanja instalacija izbora zaštitnih uređaja predstavlja jedan od najvažnijih koraka. Iz tog razloga ovaj rad bazira se na izboru i proračunima koji su potrebni za pravilan izbor zaštitnih uređaja u niskonaponskim (NN) instalacijama [2].

Nakon uvoda, u drugoj glavi je objašnjeno je projektovanje elektroenergetskih niskonaponskih instalacija. U trećoj glavi objašnjen je pojam zaštite električnih instalacija kao i vrste zaštite. U četvrtoj glavi detaljno su objašnjeni i prikazani zaštitni uređaji u niskonaponskim instalacijama. U petoj glavi opisan je proračun zaštitnih uređaja kojim se određuju svi potrebni uređaji za bezbedno i sigurno korišćenje instalacija. U šestoj glavi dati su najvažniji članovi standarda za elektroenergetske niskonaponske instalacije. U sedmoj glavi prikazana su dva primera za proračun zaštitnih uređaja. U osmoj glavi iznet je zaključak, a u devetoj glavi je navedena literatura.

2. PROJEKTOVANJE NISKONAPONSKIH INSTALACIJA

Projekat elektroenergetskih instalacija je pisani rad kojim se određuju svi podaci potrebni za izvođenje, a kasnije i za održavanje električne instalacije objekta.

2.1 Vrste projekata elektroenergetskih instalacija

Prema Zakonu o planiranju i izgradnji postoje sledeće vrste projekata:

- generalni projekat,
- idejni projekat,
- glavni projekat,
- izvođački projekat,
- projekat izvedenog objekta.

Generalni i idejni projekat izrađuju se za potrebe utvrđivanja koncepcije objekta. Glavni projekat izrađuje se za potrebe izgradnje objekta i pribavljanja građevinske dozvole. Izrađuje se u skladu sa urbanističkom dozvolom (što za glavni elektroenergetski projekat podrazumeva elektroenergetsku saglasnost nadležnog elektroprivrednog preduzeća). Izvođački projekat sadrži razradu svih neophodnih detalja za građenje objekta prema glavnom projektu. Projekat izvedenog objekta prikazuje izvedeno stanje objekta za potrebe eksploatacije i održavanja objekta [1].

2.2 Sadržaj glavnog elektro projekta

Sadržaj glavnog elektro projekta tehničke dokumentacije nije

određen (propisan) u jednom dokumentu. Projekti pojedinih instalacija (unutrašnjih i spoljašnjih) treba da sadrže: situaciju sa ucrtanim priključcima i spoljašnjom razvodnom mrežom, sve grafičke, računске i opisne priloge potrebne za izvođenje instalacija i za utvrđivanje cene radova. Na osnovu navedene zakonske regulative, projektantska preduzeća su razvila svoj koncept formiranja projekata elektroenergetskih niskonaponskih instalacija. Jedan od najčešćih načina „formiranja” ovih projekata u praksi sadrži sledeće delove:

1. Opšte podatke (naslovna strana, podaci o investitoru, podaci o objektu, itd, [2]).
2. Projektni zadatak (tehnički podaci dobijeni od strane investitora o objektu, mestu građenja, instalisanu snagu svih električnih uređaja [2]).
3. Tehnički opis (način napajanja objekta, izbor vodova, vrste izvođenja instalacije, tip svetiljki, izbor instalacionih sklopki itd. [2]).
4. Opšte i tehničke uslove (izbor najvažnijih delova propisa i zahteva koji su bitni za izvođenje elektro instalacija [2]).
5. Prilog o bezbednosti i zdravlju na radu.
6. Predmer i predračun radova (količine materijala potrebne za ugradnju, potrebna radna snaga za izvođenje radova [2]).
7. Proračuni (određivanja snage potrošača, vršno opterećenje pojedinih delova instalacija i cele instalacije, proračun tipa vodova preseka vodova njihove zaštite i efikasnost zaštite od indirektnog dodira delova pod naponom, proračun uzemljenja, proračun osvetljenja u pojedinim prostorijama [2]).
8. Grafička dokumentacija (situacija, plan električne instalacije, jednopolne šeme, šeme vezivanja, dispozicija, detalji za izvođenje elektro radova).

3. ZAŠTITA ELEKTRIČNIH NISKONAPONSKIH INSTALACIJA

Električna struja, ukoliko teče kroz telo ili deo tela, je opasna po čovečije telo. U svetu su izvršena razna ispitivanja kako električna struja deluje na ljudski organizam.

U svakodnevnom životu svaki čovek se susreće sa električnom strujom (osvetljenje, mali kućni aparati, utičnice), a sve te električne naprave su izvor opasnosti ukoliko nisu pravilno priključene ili su oštećene. Uloga zaštite električnih instalacija je sprečavanja svakog dodira čoveka sa delovima instalacije koji su pod naponom [2].

3.2 Zaštita od prekomernih struja

Provodnici pod naponom moraju biti zaštićeni sa jednim ili više uređaja za automatski prekid napajanja u slučaju preopterećenja i kratkog spoja, osim u slučajevim gde prekomerna struja ne može biti veća od podnosive struje provodnika (npr. određeni transformatori za zvonca i sl.).

3.3 Zaštita od električnog udara

Zaštita od električnog udara postiže se primenom odgovarajućih mera [1]:

- Istovremene zaštite od direktnog i indirektnog dodira.
- Zaštite od direktnog dodira.
- Zaštite od indirektnog dodira.

3.3.1 Istovremena zaštita od direktnog i indirektnog dodira
Primenom istovremene zaštite od direktnog i indirektnog dodira ne ugrožava se čovečiji život ni pri direktnom dodiru delova pod naponom, ni pri kvaru u izolaciji.

3.3.2 Zaštita od direktnog dodira delova pod naponom

Zaštita od direktnog dodira delova pod naponom obuhvata sledeće mere:

- Zaštita delova pod naponom izolovanje.
- Zaštita pregradama ili kućištima.
- Zaštita preprekama.
- Zaštita postavljanjem van dohvata ruke.
- Dopunska zaštita pomoću zaštitnih uređaja diferencijalne struje (ZUDS).

3.3.3 Zaštita od indirektnog dodira delova pod naponom

Zaštita od indirektnog dodira delova pod naponom obuhvata:

- Zaštitu automatskim isključenjem napajanja.
- Zaštitu upotrebom uređaja klase II ili odgovarajućom izolacijom.
- Zaštitu postavljanjem u neprovodne prostorije.
- Zaštitu lokalnim izjednačenjem potencijala.
- Zaštitu električnim odvajanjem.

3.4 Zaštita od prenapona

Prenaponi se mogu pojaviti u niskonaponskim instalacijama kao posledica:

- Atmosferskih pražnjenja.
- Zemljospoja usled kvara na strani visokog napona.
- Sklopnih prenapona.
- Rezonanci.

Prema Pravilniku o tehničkim normativima za električne niskonaponske instalacije, na mestima na kojima atmosferski prenaponi mogu izazvati opasnost, moraju se postaviti odvodnici prenapona.

3.5 Izjednačenje potencijala

Izjednačenje potencijala je električni spoj kojim se razni izloženi i strani provodni delovi dovode na isti potencijal. Galvanskim povezivanjem vodovodnih i drugih instalacija (gromobranska instalacija, grejanje, lift itd.) sa uzemljenjem objekta postiže se izjednačenje potencijala [1].

4. ZAŠTITNI UREĐAJI

Zaštitni uređaju u niskonaponskim instalacijama imaju ulogu da prekinu strujno kolo pre nego što nastane oštećenje instalacije, izvora ili potrošača usled nedozvoljenog opterećenja, bilo namernog ili slučajnog ili opterećenja kratkim spojem. Najzastupljeniji niskonaponski uređaji su: osigurači i prekidači [1].

4.1 Osigurači

Osigurač je uređaj koji služi za zaštitu instalacionih vodova od velikih struja kratkog spoja i preopterećenja [1]. Da bi osigurači svoju namenu u niskonaponskim instalacijama mogli ostvariti na najbolji način, neophodno je da zaštita električnih uređaja bude selektivna tj. da iz strujnog kola

isključi samo onaj deo koji je u kvaru. Selektivnost se postiže tako što ose osigurači postavljaju uvek ispred električnih uređaja koje štite, ali i samim izborom osigurača. To znači da nazivne struje osigurača bližih električnim uređajima su manje od nazivnih struja osigurača koji su udaljeniji.

Osigurač se mora postaviti [3]:

- Na početku svakog neuzemljenog provodnika.
- Na mestima gde se menja presek provodnika ili gde se prelazi na odvojak sa manjim presekom.
- Osigurači se obično postavljaju centralizovano za više strujnih krugova na pristupačnom mestu.

Zabranjeno je postaviti osigurač:

- U provodnik za pogonsko uzemljenje.
- U zaštitne ili nul-provodnike višefaznih vodova.
- U strujne krugove gde bi pregorevanjem osigurača moglo doći do neke štete ili opasnosti.
- U blizini lako zapaljivih predmeta.

4.3 Prekidači

Prekidači su mehanički rasklopni aparati koji mogu da uključuju, provode i prekidaju struju u normalnim uslovima strujnog kola. Kao i da uključuju, provode u određenom vremenu i prekidaju struje u nenormalnom uslovima strujnog kola kao što su one pri kratkom spoju [1].

5. PRORAČUN ZAŠTITE U NISKONAPONSKIM INSTALACIJAMA

Za ekonomično i pouzdano projektovanje instalacije neophodno je na samom početku odrediti najveću snagu instalacije (P_{max}). Pri određivanju najveće snage instalacija u obzir se uzima faktor jednovremenosti (k_j). Brojne vrednosti faktora jednovremenosti određuju se uglavnom empirijski i daju u tabelama.

Jednovremena maksimalna snaga (P_{max}) kojom se električna energija prenosi kroz razvodne vodove predstavlja se jednačinom:

$$P_{max} = k_j \sum_{i=1}^n P_i, \quad (1)$$

gde je:

$$\sum_{i=1}^n P_i \quad - \quad \text{instalirana snaga,}$$

$$k_j \quad - \quad \text{faktor jednovremenosti.}$$

5.1 Izbor provodnika s obzirom na trajno dozvoljenu struju

Stvarna trajno dozvoljena struja kabla izračunava se prema sledećoj jednačini:

$$I_z = k_\theta k_\lambda k_n I_{tr \text{ doz}}, \quad (2)$$

gde je:

$$I_z \quad - \quad \text{stvarna trajno dozvoljena struja kabla [A],}$$

$$I_{tr \text{ doz}} \quad - \quad \text{trajno dozvoljena struja kabla [A],}$$

$$k_\theta \quad - \quad \text{korekcionni faktor za temperaturu,}$$

$$k_\lambda \quad - \quad \text{korekcionni faktor za termičku otpornost tla,}$$

k_n – korekcionni faktor za grupno položena strujna kola.

Trajno dozvoljena struja kabla zavisi od tipa električnog razvoda, broja opterećenih provodnika i vrste izolacije. Osim toga trajno dozvoljena struja se razlikuje za provodnike načinjene od bakra i provodnike od aluminijuma [1]. Vrednosti trajno dozvoljene struje date su tabeli 1.

Tabela 1 – Trajno dozvoljena struja [1]

Bakar		Aluminijum	
Presek (mm ²)	$I_{tr \text{ doz}}$	Presek (mm ²)	$I_{tr \text{ doz}}$
1	13,5	1	11
1,5	17	1,5	14
2,5	23	2,5	19
4	31	4	25
6	40	6	32
10	54	10	43
16	73	16	58
25	95	25	76
35	117	35	94
50	141	50	113
70	179	70	142
95	216	95	171
120	249	120	197
150	285	150	226
185	324	185	256
240	380	240	300

5.2 Proračun zaštite od struja preopterećenja

Zaštitni uređaji moraju biti predviđeni da prekidaju svaku struju preopterećenja koja protiče provodnicima pre nego što prouzrokuju povišenje temperature štetno po izolaciju, spojeve, stezaljke ili okolinu.

Radna karakteristika uređaja koji štiti električni vod od preopterećenja mora da ispuni dva uslova [1]:

- Prvi uslov:

$$I_B \leq I_n \leq I_z, \quad (3)$$

- Drugi uslov:

$$I_2 \leq 1.45 I_z, \quad (4)$$

gde su:

$$I_B \quad - \quad \text{struja za koju je strujno kolo projektovano,}$$

$$I_n \quad - \quad \text{nazivna struja zaštitnog uređaja,}$$

$$I_z \quad - \quad \text{trajno podnosiva struja provodnika ili kabla.}$$

$$I_2 \quad - \quad \text{struja koja određuje pouzdano delovanje zaštitnog uređaja.}$$

5.3 Proračun zaštitnih uređaja na struje kratkih spojeva

Zaštitni uređaji moraju obezbediti prekidanje struje kratkog spoja koja protiče kroz provodnike strujnog kola, pre nego što takva struja prouzrokuje opasnost od toplotnih i mehaničkih dejstava u provodnicima i spojevima

Svaka struja kratkog spoja koja se pojavi u bilo kojoj tački strujnog kola mora biti prekinuta u okviru onog vremena

koje dovodi provodnike do dozvoljene granične temperature. Za kratke spojeve koji traju do 5 s vreme (t) u kojem data struja kratkog spoja podiže temperaturu provodnika od najviše dozvoljene temperature u normalnom radu do granične temperature izračunava se formulom:

$$\sqrt{t} = k \frac{S}{I_{ks}}, \quad (5)$$

gde je:

- t – trajanje [s],
S – presek provodnika [mm²],
I_{ks} – efektivna vrednost struje kratkog spoja [A],
k – faktor čija vrednost zavisi od materijala provodnika, izolacije i ostalih delova.

To znači da proizvod ($k^2 S^2$) mora biti veći od vrednosti propuštene energije ($I_{ks}^2 t$) navedene od strane proizvođača zaštitnih uređaja.

Struja kratkog spoja proračunava se prema sledećoj formuli [2]:

$$I_{ks} = \frac{U_{nom}}{\sum Z_{ks}}, \quad (6)$$

gde je:

- $\sum Z_{ks}$ – suma impedansi kratkog spoja, tj. ukupna impedansa od izvora do mesta kvara tj. kratkog spoja kroz koji protiče struja I_{ks}.

7. PRIMER PRORAČUNA ZAŠTITNIH UREĐAJA U NISKONAPONSKIM INSTALACIJAMA

U ovoj glavi obrađen je primer proračuna zaštitnih uređaja u niskonaponskim instalacijama, koji se odnosi na proračun u stambenom objektu.

7.1 Proračun zaštitnih uređaja u stambenom objektu

Neka se razmatra stambeni objekat ukupne površine 225 m², koji se sastoji od dve etaže. Glavna razvodna tabla objekta nalazi se u prizemlju i označen je sa RT-PR, iz koje će se napajati svi potrošači koji se nalaze u prizemlju. Osim toga iz razvodne table prizemlja napajaće se i razvodna tabla koja se nalazi na drugoj etaži objekta (RT-SP). Od velikih potrošača planirana je jedna trofazna utičnica u kuhinji (5kW).

Od elektrodistribucije tražena je i odobrena maksimalna instalisana snaga za objekat 13,8 kW, kao i faktor snage 0,95. Jednovremena snaga iznosi 11,3 kW koja je izračunata na osnovu formule 1. Nominalna struja objekta iznosi 17,2 A. Na osnovu nominalne struje može se izračunati struju I₂ koja u ovom slučaju iznosi 18 A. Na osnovu iskustva projektanta i na osnovu jednovremene snage pretpostavlja se da je za napajanje objekta (RT-PR) potreban kabel preseka 10 mm², za koji je I_{trdoz}=54. Nakon dobijanja struje I_{trdoz}, na osnovu formule (2) može se proračunati struja I_z koja

iznosi 43,2 A. Nakon toga potrebno je proveriti da li su zadovoljeni uslovi iz relacija 3i 4.

Odnosno u razmatranom slučaju:

$$17,2 < 43,2;$$

$$17,2 < I_n < 43,2.$$

Zaključak je da zaštitni uređaj koji će da štiti strujni krug koji napaja RT-PR treba da bude 25 A. Nakon proračuna provodnika za napajanje RT-PR i zaštitnog uređaja od struja preopterećenja na isti način se proračunavaju i provodnici i zaštitni uređaji za ostale strujne krugove objekta, koji se rade na isti način.

Nakon proračuna padova napona potrebno je uraditi proračune na termičko naprezanje kablova tokom kratkog spoja i proveru efikasnosti zaštite od indirektnog dodira automatskim isključenjem napajanja. Struja kratkog spoja proračunava se po formuli (6). Za strujni krug kojim se napaja objekat struja kratkog spoja iznosi 5.141,00 A, a vreme t=0,05 s.

8. ZAKLJUČAK

U ovom radu obrađena je tema proračuna zaštitnih uređaja u niskonaponskim instalacijama.

Zaštitni uređaju u niskonaponskim instalacijama imaju ulogu da prekinu strujno kolo pre nego što nastane oštećenje instalacije, izvora ili potrošača usled nedozvoljenog opterećenja, bilo namernog ili slučajnog ili opterećenja kratkim spojem. Uloga zaštite električnih instalacija je sprečavanja svakog dodira čoveka sa delovima instalacije koji su pod naponom. Kako bi zaštitni uređaji u potpunosti odradili svoju funkciju i zaštilili kako ljudski život, tako i opremu i instalaciju, oni moraju biti izabrani i postavljeni po propisima.

9. LITERATURA

- [1] G. Dotlić: *Elektroenergetika kroz standarde, zakone, pravilnike i tehničke preporuke*: SMEITS, Beograd, 2013.
- [2] M. Mišković *Električne instalacije i osvetljenje*: Građevinska knjiga a.d, Beograd, 2005.
- [3] <https://www.scribd.com/document/363218526/INSTALACIONI-OSIGURA%C4%8CI>

Kratka biografija:



Jelena Šešlija rođena je u Trebinju 1993. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva –Elektroenergetski sistemi odbranila je 2016. god. Iste godine upisala se na master studije na istom smeru.

**JEZIK ZA PODRŠKU EKSTRAKCIJI PODATAKA IZ NESTRUKTURIRANIH IZVORA
SA VEBA****A LANGUAGE FOR UNSTRUCTURED WEB-BASED DATA EXTRACTION**Filip Frank, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu opisan je DSL (eng. Domain Specific Language) za ekstraktovanje podataka sa veba, napisan uz pomoć textX biblioteke i Python programskog jezika..

Ključne reči: DSL, struganje, Scrapy, textX, ekstrakcija, Python

Abstract – In this paper, DSL for extracting data from Web resources is described, created using textX library and Python programming language.

Keywords: DSL, scraping, Scrapy, textX, extraction, Python

1. UVOD

Osnovna ideja pisanja jezika za podršku ekstrakciji podataka jeste implementirati jasan i koncizan jezik, ograničen na domen veb ekstraktovanja, kojim će se svlačiti određene informacije definisane upitima. Jezik se pritom neće ograničavati na jedan izvor podataka (jednu veb stranicu) već će omogućavati prilagođavanje različitim domenima i strukturama stranice.

Cilj je da programer ima potpunu slobodu u definisanju modela podataka i formata na koji će se ti podaci čuvati. U objektno-orijentisanom svetu, svaki entitet se može opisati odgovarajućom klasom i atributima, pa je bitno identifikovati entitete značajne za proces ekstraktovanja, uzimajući u obzir informacije dostupne u izvoru podataka. Na taj način, definisanje modela će se zasnivati na identifikaciji značajnih elemenata strukture izvora podataka sa stanovišta informacija koje sadrže i njihovo prevođenje na model.

2. TEORIJSKE OSNOVE**2.1 World Wide Web**

World Wide Web, iliti samo Veb, predstavlja informacioni prostor u kojem se dokumenti i drugi veb izvori identifikuju jedinstvenim lokatorima (URL - Uniform Resource Locators) i gde su ti dokumenti povezani hipertekstualnim vezama i dostupni preko Interneta [1]. Sa stanovišta ekstraktovanja i svakodnevne upotrebe Interneta, korišćenjem veb pregledača, komunikacija se odvija korišćenjem HTTP protokola.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Igor Dejanović, vanr. prof.

HTTP je protokol za transport HTML dokumenata između servera i klijenta. Najbitnije karakteristike ovog protokola jeste zahtev/odgovor komunikacija. HTTP je u suštini jednostavan mrežni protokol, baziran na tekstualnim dokumentima (porukama) propisanim po HTTP standardu.

HTML (Hyper Text Markup Language) je jezik koji opisuje format i strukturu veb stranica. Iz ugla HTTP komunikacije, HTML opisuje sadržaj tela HTTP odgovora poslatog od strane HTTP servera. Pošto se upravo u HTML-u nalaze podaci koji su od važnosti za ekstraktovanje, ovaj element je ključan u Internet komunikaciji.

HTML je „markup” jezik, iz razloga što tag može da definiše i strukturu i način na koji će se sadržaj koji tag ima prikazati korisniku.

CSS (Cascade Style Sheet) je jezik za definisanje izgleda veb stranice. Razlika u odnosu na HTML jeste, pre svega, sintaksa. Definiše se kao lista izraza razdvojenih tačkazarezom pri čemu se izrazi navode u obliku *naziv:vrednost*.

CSS selektori su bitni pošto služe za selektovanje odgovarajućih elemenata koristeći CSS osobine HTML elementa.

2.2 Struganje

Termin struganja označava proces automatskog prikupljanja podataka sa Interneta (eng. *scraping*).

Sa stanovišta struganja, API servisi su najbolji izvor informacija. Podaci koje on pruža su potpuni, ažurni, koncizni, standardizovani, u unapred definisanom formatu i dokumentovani, dostupni preko URL-a.

2.3 Struganje veb stranica

Veb stranice su namenjene krajnjem korisniku stoga sadrže najviše informacija koje su nam potrebne, one su ažurne i lako im je pristupiti. Nema dokumentacije niti posebne provere da li su podaci ažurirani i glavni zadatak je pronaći i ekstraktovati elemente koji su nosioci traženih informacija. Svaki sadržaj, koji je prikazan na stranici, može biti ekstraktovan [2].

2.4 Puzanje

Puzanje (eng. *crawl* – puzanje, mleti) predstavlja proces u kom se struganje ne ograničava na jednu stranicu već softver „puzi” kroz više stranica čije su adrese ekstraktovane parsiranjem stranice. U teoriji, puzanje je

termin koji označava sposobnost programa da automatski ekstraktuje stranice i da sam istražuje određeni domen [3]. Dobra praksa je razdvojiti procese puzanja i struganja, tako da je logika pronalaženja novih stranica razdvojena od logike parsiranja stranice.

Nakon uspešnog pronalaska podataka koji se ekstraktuju, potrebno je sačuvati te podatke u nekom obliku. Dobar proces struganja se ogleda u uspešno prikupljenim podacima koji su dostupni za dalje analize nakon što se čitav proces završi. Način na koji će se ti podaci čuvati zavisi dosta od tipa podataka koji se sakuplja i njihove svrhe.

2.4 Jezici specifični za domen

Jezici opšte namene (JON) obuhvataju široku oblast problematike jer je osnovna ideja prilikom njihovog kreiranja upravo sposobnost rad sa problemima iz različitih oblasti.

Za razliku od njih, postoje jezici koji pokrivaju samo usko ograničen domen i čija namena je da se prilagode problematici tog domena u cilju veće efikasnosti i produktivnosti u odnosu na jezike opšte namene. Ovakve jezike nazivamo jezicima specifičnim za domen – JSD (DSL – *Domain Specific Language*) [4].

Glavne karakteristike JSD jezika:

- prilagođenost domenu problema,
- visok nivo apstrakcije,
- konciznost.

3. PREGLED I KORIŠĆENE TEHNOLOGIJE

3.1 Korisni moduli

Python sadrži mnoštvo modula koji podržavaju ovakav tip komunikacije: *urllib*, *urllib3*, *httplib2*, *requests*, *aihttp*, itd [3]. *Requests* modul u potpuno podržava rad sa HTTP komunikacijom, slanje HTTP zahteva, modifikaciju zaglavlja, rad sa formama, slanje kolačića itd.

BeautifulSoup je *Python* biblioteka za parsiranje HTML i XML dokumenata. Pruža splet raznoraznih funkcija za pronalazak elemenata, navigaciju i modifikovanje parsiranog stabla.

Treba spomenuti i jednu od alternativa *BeautifulSoup*-u, *lxml* modul. Glavne karakteristike ovog modula su robustnost, brzina, podrška za *Xpath* selektore. Brzina izvršavanja i veliki broj funkcija je glavna prednost u odnosu na druge alate. Osnovna ideja je ista, programski se kreira stablo na osnovu unetog HTML koda a zatim se pozivaju funkcije za pronalazak i manipulaciju elemenata stabla. Za razliku od *BeautifulSoup*-a, *lxml* podržava mehanizam i CSS i *Xpath* selektora.

Selenium je alat koji služi prvenstveno za testiranje veb stranica ali je između ostalog i snažan alat za ekstraktovanje. Integriše se u postojeće veb pregledače i automatizuje ih u cilju testiranja i prikupljanja podataka ili izvršavanja bilo kakve akcije na stranici. Podržan je od raznih programskih jezika, između ostalog i u *Pajtonu*.

3.1 Scrapy

Scrapy je softversko okruženje, napisano u *Pajtonu*, za puzanje po veb sajtovima i ekstraktovanje podataka koji se mogu koristiti za razne aplikacije, prikupljanje podataka, procesiranje informacija i ostalo [5]. Objedinjuje sve funkcionalnosti potrebne za struganje: ekstraktovanje novih linkova, puzanje po čitavim domenima i pronalaženje elemenata u parsiranim stranicama. *Scrapy* koristi termin pauk (eng. *spider*) za klase koje prikupljaju informacije sa veb stranice.

3.2 Arpeggio

Arpeggio je parser, napisan u *Python* jeziku i njegova osnovna namena je podrška za pisanje JSD jezika [6]. Glavne karakteristike ovog parsera jeste da je on rekursivni silazni parser sa vraćanjem i memorizacijom. Omogućava definisanje gramatike, parsiranje na osnovu definisanih gramatičkih pravila i interpretaciju koda.

3.2 textX

TextX je JSD jezik za definisanje drugih JSD jezika koristeći *Python*. Na osnovu definisane gramatike, *textX* kreira meta-model i parser za definisani JSD jezik [7]. Gramatika se zasniva na *textX* pravilima koja se definišu u *.tx* fajlovima. Osnovna struktura *textX* pravila je naziv pravila (parser prihvata slova, cifre i donje crte), dvotačka, *textX* izraz i na kraju tačka-zarez. U okviru *textX* izraza koristimo ugrađene tipove, druga definisana pravila i *textX* operatore.

3.3 Jinja2

Jinja2 je obrađivač šablona namenjen za *Python* programski jezik [8]. Zgodan je za generisanje bilo kakvog programskog koda, i sličan je *Django* šablonima. Najveće prednosti korišćenja *Jinja2* obrađivača je: automatsko eskejpovanje HTML-a korišćenjem specijalnih karaktera, mehanizam nasleđivanja šablona, sloboda u konfigurisanju sintakse, *sandbox* izvršavanje koje pruža zaštićeno okruženje za automatizovano testiranje programa.

3.4 MongoDB

Za persistenciju podataka sakupljenih struganjem, u implementiranom rešenju koristimo *MongoDB*. *MongoDB* je *open-source* baza podataka u kojoj se podaci čuvaju u obliku dokumenata. Glavne karakteristike su dobre performanse, dostupnost i automatsko skaliranje.

4. IMPLEMENTACIJA SISTEMA

4.1 Definisanje gramatike

Kreirani jezik za definisanje svih izraza i upita u implementiranom rešenju zasniva se na gramatikama napisanim korišćenjem *TextX* alata, tačnije na jednoj gramatici koja definiše konfiguraciju i deklarisanje modela podataka (tipova) i drugoj, koja definiše pisanje jezičkih upita.

Gramatika za deklarisanje entiteta koji se ekstraktuju treba da podrži sledeće funkcionalnosti:

- Definisanje jednog ili više tipova entita.
- Za svaki entitet, definisanje njegovog imena i jednog ili više atributa.
- Za svaki atribut, definisanje CSS selektora koji jedinstveno identifikuju elemenat na željenoj stranici.
- Ukoliko su traženi podaci u okviru tabele ili (ne)uređene liste na veb stranici, mogućnost definisanja takvih elemenata i CSS selektora koji identifikuju ove elemente. Definirati CSS selektor elementa koji će iterirati kroz pomenute strukture.

Koristeći napisanu gramatiku, jezikom je pokriveno definisanje tipova podataka, njihovo povezivanje sa elementima nestrukturiranih izvora (HTML elemenata veb stranice) i navođenje elemenata koji su nosioci više parametara definisanih tipova podataka.

```
Movie (
  title {"div.title_wrapper h1::text"},
  year {"div.title_wrapper h1 span#titleYear a::text"},
  rating {"div.ratingValue strong span::text"},
  duration {"div.subtext time::text"},
  genre {"div.subtext a::text"},
  release_date {"div.subtext a::text"},
  story_line {"div.plot_summary div.summary_text::text"},
  director {"div.credit_summary_item a::text"},
  writers {"div.credit_summary_item a::text"},
  cast {"div.article table.cast_list"}
)
```

Slika 1. Primer deklarisanja tipa

Gramatika za deklarisanje entiteta koji se strugaju treba da podrži sledeće funkcionalnosti:

- Definisanje jednog ili više tipova entita.
- Za svaki entitet, definisanje njegovog imena i jednog ili više atributa.
- Za svaki atribut, definisanje CSS selektora koji jedinstveno identifikuju elemenat na željenoj stranici.
- Ukoliko su traženi podaci u okviru tabele ili (ne)uređene liste na veb stranici, mogućnost definisanja takvih elemenata i CSS selektora koji identifikuju ove elemente. Definirati CSS selektor elementa koji će iterirati kroz pomenute strukture.

```
find Movie where cast = 'Morgan Freeman'
```

Slika 2. Primer definisanja upita

4.2 Podrška za rad sa ekstraktovanim sadržajem

Ideja je da koristeći *Python* definišemo funkcije za rad sa sadržajem ekstraktovanih podataka pri čemu nam se pružaju sve pogodnosti korišćenja *Python* jezika: definisana sintaksa, postojeće funkcije za rad sa stringovima, editori, sintaksno bojenje itd. Na ovaj način, definisani JSD jezik će ostati nepromenjen i potrebno je samo integrisati ručno-pisani i generisani kod, odnosno

Python funkcije vezati za konkretne parametre definisanih tipova. Rezultat izvršavanja svake funkcije biće string koji se dodeljuje vrednosti tog parametara.

```
def Movie_genre(content):
    return ", ".join(content.extract()[::-1])

def Movie_release_date(content):
    return content.extract()[::-1].strip()

def Movie_story_line(content):
    return content.extract_first().strip()
```

Slika 3. Primer *Python* funkcija

Pošto gramatika podržava definisanje elemenata koji sadrži više parametara određenog tipa (tabela, uređena ili neuređena lista), *Python* funkcije se koriste prilikom definisanja načina ekstrakcije informacija iz ovakvih struktura.

Mehanizam traženja parametara se zasniva na pronalasku takvih elemenata na veb stranici korišćenjem definisane gramatike. Pisanjem deklaracije tipova i navođenjem CSS selektora koji ih identifikuju, uspešno pronalazimo na stranici ovakve elemente i elemenat kojim prolazimo kroz strukturu i pronalazimo podatke.

Sledeće što je potrebno definisati su načini prepoznavanja značenja tih podataka, odnosno mapiranje parametara i izvučenih podataka, na osnovu vrednosti određenih elemenata na stranici.

```
Table (
  tag {"div.product-panels-content table.data-table"},
  row {"tr"}
)
```

Slika 4. Primer deklarisanja tabele

```
def Table_name(row_selector):
    return row_selector.css('th::text').extract_first().strip()

def Table_value(row_selector):
    return row_selector.css('td::text').extract_first()
```

Slika 5. Primer *Python* funkcija za rad sa tabelom

Mehanizam „normalizovanja” vrednosti izvučenih iz tabela smanjuje osetljivost programa na odstupanja između naziva parametara i naziva kolona/ređa na način što briše sve specijalne karaktere, razmake zamenjuje donjom crtom (prilagođavanje pravilima gramatike definisanja naziva parametara) i sve karaktere pretvara u mala slova. Ovakvim normalizovanjem izbegavaju se situacije ne poklapanja dveju vrednosti zbog malih i velikih slova ili razmaka.

Mehanizam pseudonima, podržan od definisane gramatike, omogućava definisanje nekoliko alternativnih naziva za isti parametar i program će, ukoliko ne dođe do poklapanja dva naziva, porediti sve pseudonime za dati parametar i pokušati da nadomesti ovakav problem.

4.3 Šabloni

Nakon parsiranja deklaracije tipova i upita na osnovu definisanih gramatika, dobijamo dva modela, po jedna za oba JSD izraza. Da bi iskoristili ove modele za generisanje koda *Scrapy* projekta, potrebni su nam *Jinja2* šabloni. Imajući u vidu strukturu *Scrapy* projekta, kreirani

su šabloni koji odgovaraju imenima ciljanih fajlova: *items, pipelines, settings, middlewares...*

Šabloni za generisanje pauka će biti jedinstveni za svaki domen, za koji radimo struganje. Glavna struktura pauka se svodi na pisanje *start_requests* metode, koja će poslati inicijalni zahtev na specificiranu stranicu, i metode koja će ekstrahovati linkove i slati zahteve ka stranicama koje sadrže podatke o entitetu koji se ekstrahuje. Metode se navode direktno u šablonu, i nepromenjene se generišu u krajnji kod.

```
def start_requests(self):
    url = "https://www.imdb.com/chart/top?ref_=nv_mv_250"
    return [scrapy.Request(url, callback=self.main_page)]

def main_page(self, response):
    links = response.css("table.chart.full-width tbody")
    for link in links.css("tr"):
        yield scrapy.Request("https://www.imdb.com"+
            link.css("td.titleColumn a::attr(href)").extract_first())
```

Slika 6. Primer spider metoda

Šablon za osnovni oblik *parse* funkcije zavisi od karakterističnosti stranice koja se ekstrahuje. U njemu se instancira objekat klase definisane u *items.py* modulu, zajedno sa svim parametrima, kojima se dodeljuje vrednost na osnovu *response* objekta i CSS selektora unetih na ulazu.

```
def parse(self, response):
    movie = Movie()

    title = response.css("div.title_wrapper h1::text")

    if "Movie_title" in dir(custom_module):
        movie['title'] = getattr(custom_module,
            'Movie_title')(title)
    else:
        movie['title'] = title.extract_first()
```

Slika 7. Primer dela parse metode

Način integracije generisanog i ručno pisanog koda, definisanog u prpratnom .py modulu, koji se prilikom generisanja koda smešta u isti folder gde i *spider.py* fajl, može se primetiti na datom primeru. Ukoliko postoji funkcija za taj parametar, definisana u *custom_module*, ona će biti pozvana, u suprotnom, prosleđujemo kompletan sadržaj ekstrahovanog elementa.

Osnova prilikom generisanja koda za *pipelines.py* je nepromenljiva: kod će validirati podatke na osnovu određenih uslova i ukoliko su oni zadovoljeni, čuvaće ih u *MongoDB* bazu. Uslovi za validiranje su određeni upitom, odnosno kriterijumima upita.

Metoda `__init__` će referencirati podešavanja vezana za bazu podataka, definisana u *settings.py*, koja će biti nepromenljiva za sve aplikacije. Metoda *process_item* - sadrži logiku validiranja podataka i u slučaju uspeha, čuvanja podataka u bazu. Šablon će proći kroz sve kriterijume definisane u upitu i smestiti u jedan *if* uslov, tako da za sledeći upit:

U *settings.py* modulu se nalaze sva projektna podešavanja vezana za ponašanje bota, podešavanje zaglavlja, aktiviranje *pipeline* klasa, parametri baze podataka itd.

4.4 Python generator

Namena *Python* generatora je da objedini sve što je navedeno u implementaciji rešenja: upotrebom *textX* da

generiše meta-model i model podataka na osnovu definisanih deklaracija tipova i upita, imajući u vidu gramatiku, i da zatim dobijene modele prosledi spomenutim šablonima koristeći *Jinja2* klase.

5. ZAKLJUČAK

Nedostatak kvalitetnog i konciznog JSD-a koji se bavi struganjem, osnovni je razlog implementiranja ovog rešenja. Upoznati sa prednostima struganja i korišćenja jezika specifičnih za domen, ideja je implementirati jasan i precizan jezik za struganje veb stranica. Sa njim ćemo kreirati entitete i upite koji će se lako prilagođavati različitim strukturama, koji će omogućavati filtriranje željenih podataka i koji će brzo i efikasno rezultirati sa podacima sačuvanim u bazi podataka.

Dalji pravci razvoja jezika su pre svega usmereni na kreiranje modela pa zatim i generisanje koda vezanog za definisanje stranice koja se ekstrahuje i način na koji će se linkovi ekstrahovati (CSS selektori u kojima se nalaze linkovi). Trenutno se ova logika nalazi u konkretnim šablonima za željeni domen.

Pored toga, trenutno se proces ekstrahovanja zasniva na CSS selektorima, pa bi podrška *Xpath* selektorima bila značajna u situacijama kada nam je potrebna sva moć i fleksibilnost *Xpath* jezika.

6. LITERATURA

- [1] Wikipedia, „World Wide Web,“ [Na mreži]. Available: https://en.wikipedia.org/wiki/World_Wide_Web. [Poslednji pristup 10 October 2018].
- [2] H. Brody, The Ultimate Guide to Web Scraping, 2017.
- [3] S. vanden Broucke i B. Baesens, Practical Web Scraping for Data Science, Apress, Berkeley, CA, 2018.
- [4] I. Dejanovic, „Jezici specifični za domen,“ [Na mreži]. Available: <http://www.igordejanovic.net/courses/jsd.html>. [Poslednji pristup 3. oktobar 2018].
- [5] Scrapy, „Scrapy 1.5 documentation,“ [Na mreži]. Available: <https://doc.scrapy.org/en/latest>. [Poslednji pristup 4. oktobar 2018].
- [6] I. Dejanovic, „Arpeggio Documentation,“ [Na mreži]. Available: <http://www.igordejanovic.net/Arpeggio>. [Poslednji pristup 6. oktobar 2018].
- [7] I. Dejanovic, „Arpeggio,“ [Na mreži]. Available: <http://www.igordejanovic.net/courses/tech/arpeggio.html>. [Poslednji pristup 6. oktobar 2018].
- [8] Jinja2, „Jinja2 Documentation,“ [Na mreži]. Available: <http://jinja.pocoo.org/docs/2.10/api>. [Poslednji pristup 7. oktobar 2018].

Kratka biografija:

Filip Frank rođen je u Novom Sadu 1992. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo, odbranio je 2018.god.

RAZVOJNO OKRUŽENJE ZA GENERISANJE PROGRAMSKOG KODA SPRING API REST APLIKACIJA**DEVELOPMENT ENVIRONMENT FOR GENERATING SPRING API REST APPLICATIONS**

Jana Vojnović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu predstavljeno je softversko rješenje za generisanje koda Spring API REST aplikacija. Pomenuto rješenje preuzima metapodatke iz šeme baze podataka i u kombinaciji sa unaprijed definisanim šablonima generiše izvršiv programski kod.

Ključne reči: Generator koda, šablon, metapodaci

Abstract – This paper presents a software solution in favor of code generation for Spring API REST applications. The solution uses predefined templates and metadata extracted from the database to generate runnable programming code.

Key words: code generator, template, metadata

1. UVOD

U svijetu informacionih tehnologija koje prate agilne metodologije razvoja softvera [1], sve više se javlja potreba da se proizvod, u ovom slučaju programski kod koji pruža određenu funkcionalnost, što brže i efikasnije plasira na tržište. U cilju podizanja kvaliteta koda na što viši nivo, kao i brzine kojom se dostavlja isti, pojavile su se tehnike automatskog generisanja programskog koda. Primenom pomenutih tehnika, softverski inženjer piše kod na višem nivou apstrakcije koji služi kao strukturalni šablon i osnova za automatizaciju procesa pisanja programskog koda. Ovaj pristup je usko vezan za inženjerstvom vođenim modelima (engl. MDE) [1], koji podrazumjeva da je prvi korak zapravo kreiranje modela.

Kao jedan od pravaca u kome ide MDE jeste i generisanje koda na osnovu metapodataka šeme baze podataka (BP). U ovom slučaju, struktura šeme BP se koristi kao jedinstven izvor informacija.

Savremeni softverski sistemi najčešće primjenjuju objektno-relacione mapere (eng. ORM) da bi povezali torke iz tabela relacione šeme BP u klase objektno reprezentacije aplikativnog sloja. Uzevši u obzir realne potrebe korisnika, u ovom slučaju softverskog inženjera, osnovna primjena ovog softverskog rješenja jeste podrška u implementaciji procesa objektno relacionog mapiranja. U radu će biti opisano projektovanje i implementacija komponente generatora koda čija je svrha automatizacija repetitivnih procesa prilikom razvoja softvera, a koje se tiču pisanja programskog koda.

NAPOMENA:

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

2. KORIŠĆENE TEHNOLOGIJE

Projektno rješenje u ovom radu predstavlja softverski alat JGen napisan u programskom jeziku Java [3], gdje je Spring Framework [4] odabran kao radni okvir. Dio softverskog rješenja zaduženog za generisanje programskog koda će pristupiti šemi BP na osnovu zadate konfiguracije i iz nje dobiti metapodatke. Za pomenuti proces su kao izvor podataka odabrane relacije BP. Glavni razlog jeste njihova rasprostranjenost u poslovnim sistemima, ali i jasna struktuiranost i uniformnost kroz cijelu šemu BP. Kao sistem za upravljanje bazama podataka odabran je MySQL [5].

Pomenuti metapodaci će uz pomoć šablona napisanog u FreeMarker [6] jeziku, biti osnova za proces generisanja programskog koda koji obuhvata:

- Model klase, koje služe kao reprezentacija tabela iz šeme BP. Ove klase će koristiti Hibernate [7] ORM tako da će sam proces mapiranja torki iz šeme BP biti u potpunosti kompatibilno sa poljima iz klasa;
- JPA repozitorijum, koji služi kao sloj komunikacije sa šemom BP. Za implementaciju ovog sloja je referenciran Spring Data [8] repozitorijum;
- Servis, koji služi kao međusloj između repozitorijuma i kontrolera. U ovom dijelu će biti implementirane osnovne operacije manipulacije nad podacima;
- Kontroler, koji predstavlja krajnji sloj u ovoj aplikaciji. S obzirom da je aplikacija generisana na osnovu MVC dizajn šablona [9], ovaj kontroler će biti implementiran kao REST kontroler;

Izgenerisana aplikacija se oslanja na pomenute tehnologije i alate korištene prilikom implementacije softverskog rješenja zaduženog za njeno generisanje. Treba uočiti da jezik u kome je napisan sam softver za generisanje koda ne mora biti isti kao i izgenerisani kod, u opštem slučaju. Takođe, u izgenerisanoj aplikaciji može se koristiti i bilo koja relaciona baza, pod uslovom da postoji kompatibilna biblioteka za Hibernate ORM koji je implementiran na aplikativnom nivou.

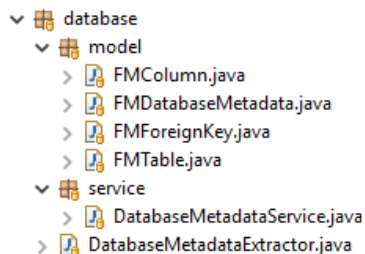
3. IMPLEMENTACIJA RJEŠENJA

Rješenje opisano u ovom radu se sastoji iz više modula koji ujedno predstavljaju i logičke cjeline. Pomenuti moduli, kao i načini njihove implementacije i zaduženja u

procesu generisanja, će biti detaljnije opisani u narednim poglavljima.

3.1. Modul za dobavljanje metapodataka

Modul za dobavljanje metapodataka se nalazi u paketu *database* (Slika 1) i zadužen je da ostvari konekciju na zadatu šemu BP i dobavi informacije iz nje.



Slika 1. Paket modula za dobavljanje metapodataka

Klase definisane u ovom modulu (slika 2) predstavljaju reprezentaciju relacionih metapodataka u ovom projektu. Za dobavljanje svih metapodataka iz šeme BP korištena je JDBC konekcija. Direktno iz pomenute konekcije se poziva metoda *getMetaData* koja dobavlja metapodatke u obliku klase *DatabaseMetadata* [10]. Metode koje su korištene su: metoda za dobavljanje podataka o tabelama, metoda za dobavljanje podataka o kolonama u tabeli, metoda za dobavljanje podataka o stranim ključevima, metoda za dobavljanje podataka o indeksima unutar tabele i metoda za dobavljanje podataka o primarnim ključevima u tabeli.

Kao rješenje u većini savremenih softverskih sistema jeste ograničavanje korisnika na aplikativnom nivou o unosu enumeracije, odnosno nabrojivih vrijednosti. U tom slučaju su ova polja u šemi BP zapravo obična tekstualna polja, a samo na aplikativnom nivou imaju predefinisani skup mogućih vrijednosti. Prilikom izrade ovog rješenja posvećena je posebna pažnja na dobavljanje predefinisanih enum vrijednosti iz šeme BP i kreiranje klase koja ih oslikava na nivou objekata u reprezentaciji Java programskog jezika.

Prilikom preuzimanja svih kolona koje pripadaju referentnoj tabeli koristi se metoda *getColumn* pomenute klase *DatabaseMetadata*. Tokom pomenutog procesa nije moguće razlikovati kolone primarnog ključa od običnih kolona samo na osnovu skupa informacija koje ova metoda dobavlja. Iz ovog razloga je korištena metoda posebno za tu namjenu pod nazivom *getPrimaryKeys*. Pomenuta metoda dobavlja sve kolone primarnog ključa iz prosledene tabele.

Proces dobavljanja informacija o primarnom ključu u ovom projektu je realizovan tako da se prvo dobave sve kolone referentne tabele. Zatim se dobave sve kolone primarnog ključa iste tabele. Pomenute kolone se zatim uporede i ukoliko se poklapaju, zaključak je da referencirana kolona spada pod ograničenje primarnog ključa.

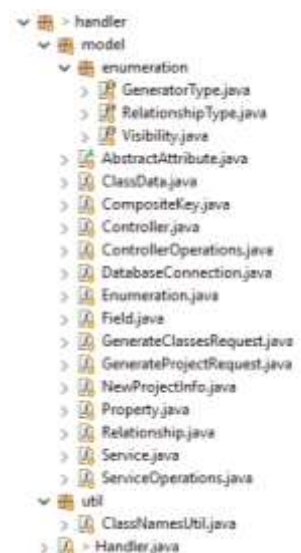
Ukoliko tabela ima više od jedne kolone pod ograničenjem primarnog ključa, to znači da ona ima kompozitni primarni ključ. Pomenute informacije se tretiraju na poseban način prilikom procesa generisanja aplikacije. Na slici 2 je prikazan dijagram klase za modul za dobavljanje metapodataka.



Slika 2. Dijagram klase za database modul

3.2. Modul za transformaciju podataka

Modul za transformaciju metapodataka se nalazi u paketu *handler* (Slika 3) i zadužen je da podatke dobijene iz modula za dobavljanje metapodataka transformiše u oblik koji će više odgovarati objektnoj reprezentaciji. U paketu *model* se nalaze model klase koje po svojoj strukturi predstavljaju metamodel ovog softverskog rješenja.



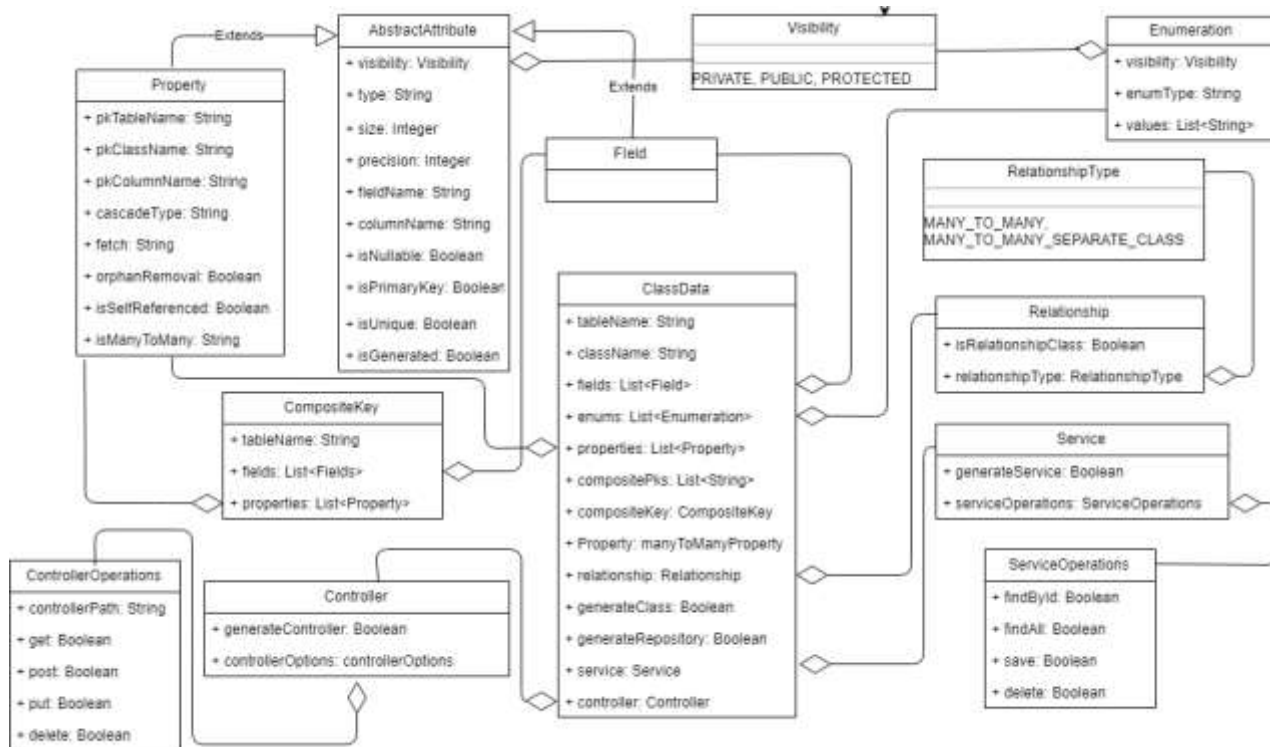
Slika 3. Paket modula za transformaciju podataka

Klasa *Handler* je komponentna klasa ovog modula u kome se obavljaju sve transformacije. Prilikom prosledivanja podataka za generisanje kontroleru koji je povezan sa pomenutom *Handler* klasom, moguće je specificirati dvije vrste generisanja:

- Potpuno novi projekat - u ovom slučaju će biti generisani svi potrebni elementi za novi Spring Boot Maven projekat, koji podrazumjevaju dodatne klase i konfiguracione datoteke;

- Integracija u postojeći projekat – u ovom slučaju će biti generisane samo odabrane klase u MVC dizajn šablonu, tj. model, repozitorijum, servis i kontroler klase u već postojećem Spring projektu.

Na slici 4 prikazan je dijagram klasa za modul za transformaciju podataka.



Slika 4. Dijagram klasa za handler modul

Shodno odabranom tipu generatora razdvojena je i logika unutar samog softverskog rješenja. Svaka tabela preuzeta u modulu za dobavljanje metapodataka u obliku klase *FMTable* se u ovom modulu transformiše u klasu *ClassData*. Slučaj kada tabela iz šeme BP neće biti prevedena u klasu objektno reprezentacije jeste ukoliko je to poveznika tabela sa samo dva strana ključa. Prilikom realizacije same *Handler* klase, ona je izdijeljena u više manjih logičkih cjelina koje su ostvarene kroz privatne metode prateći dizajn šablon razdvajanja odgovornosti.

Nakon inicijalnog kreiranja objekata tipa *ClassData*, obrađuju se i kolone koje će postati njena polja. Sama hijerarhija kolona podrazumjeva da je *AbstractAttribute* apstraktna klasa koja sadrži zajedničke informacije za bilo koji tip polja. Zatim se podjela pomenutih polja vrši na:

- *Field* klase, koje predstavljaju obične kolone iz šeme BP uključujući i kolone primarni ključ, ukoliko je ovo ograničenje samo nad jednom kolonom;
- *Property* klase, koje predstavljaju referencijalne integritete iz šeme BP;
- *Enumeration* klase, koje predstavljaju kolone tipa nabiranja;

- *CompositeKey* klase, koje objedinjuju sva polja ograničenja primarnog ključa, ukoliko ono obuhvata više od jedne kolone;

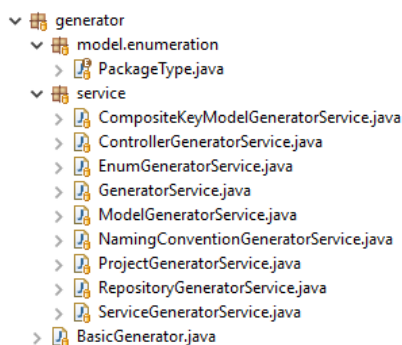
Prilikom implementacija ovog rješenja poseban osvrt je na mapiranje kompozitnih, tj. složenih primarnih ključeva iz relacione u objektnu reprezentaciju. U tom procesu korišćen je *javax.persistence* interfejs na koji se Hibernate mapper oslanja. Važno je napomenuti da je podržan i slučaj ukoliko je strani ključ dio primarnog ključa.

U relacionoj reprezentaciji modela tabele mogu biti povezane vezom više ka više, pri čemu se kreira poveznika tabela koja u sebi sadrži referencijalne integritete povezanih tabela. U objektnoj reprezentaciji poveznika tabela se, većinom, ne mapira na povezniku klasu, osim ukoliko korisnik to eksplicitno ne želi. Mapperi kao što je Hibernate vrlo jasno prepoznaju pomenuti tip veze, pa će se poveznike tabele biti prepoznate i po potrebi generisane na strani šeme BP.

Prilikom implementacije ovog rješenja pojavio se problem prepoznavanja poveznika tabela, kao i njihova prezentacija u objektni domen. Pored tabela koja su jasno poveznike i sadrže samo dvije kolone koje predstavljaju referencijalne integritete poveznika tabela, javljaju se i tabele koje sadrže dodatne kolone. Ovakve pomenute tabele bi trebalo mapirati kao posebne klase, ukoliko želimo da očuvamo postojanost pomenutih kolona.

3.2. Modul za generisanje programskog koda

Modul za generisanje programskog koda se nalazi u paketu *generator* (Slika 5) i zadužen je da iz ulaznih podataka generiše izvršivi programski kod.



Slika 5. Paket modula za generisanje koda

Modul za generisanje koristi model klase iz *handler* paketa uz napomenu na implementirane validacije na poljima i implementaciju posebnog validatora kao u cilju sprečavanja korisnika da unese nevalidne podatke. Uzimajući Java programski jezik za odredišni kod, javlja se potreba za importima (uvozima) u svakoj izgenerisanoj klasi. Njihovo rukovanje je odrađeno dinamički, tako da se neće desiti da neka klasa ima uvoze koje ne koristi. Na taj način se izbjegavaju upozorenja o nekorisćenim uvozima. Opisana logika je implementirana u svakoj servis klasi na osnovu sadržaja prosleđenog generatoru.

Ovaj model je ostvaren preko više geneatora definisanih kao servisi koji obavljaju određeni dio logike za generisanje. Svaki generator servis koristi sopstveni šablon. BasicGenerator predstavlja apstraktni generator koga nasleđuju svi ostali generatori. Svaki učitani šablon se mapira na objekat klase Template iz paketa freemarker, korištenog tokom projektovanja.

Prilikom izrade ovog softverskog rješenja pojavio se problem imenovanja tabela prilikom prevođenja iz objektnih klasa u relacionih oblik tabela. Za ovu konvenciju imenovanja odgovoran je mapper Hibernate. Naime, od verzije 5 Hibernate-a koristi se kao podrazumjevana konvencija po kojoj prevodi sva imena tabela u oblik sa karakterom „_“. Tako će tabela koja se zove npr. customerDetails biti automatski prevedena u naziv customer_details. S obzirom da je ovo podešavanje neželjeno za ciljnu aplikaciju, bilo je potrebno na neki način pružiti korisniku mogućnost reimplementacije metoda koje sadrži klasa *PhysicalNamingStrategyStandardImpl* iz paketa org.hibernate.boot.

4. ZAKLJUČAK

U ovom radu predstavljeno je projektno rješenje sistema za generisanja programskog koda namijenjeno za Java Spring API aplikacije. Softversko rješenje predstavlja aplikaciju namijenjenu softverskim inženjerima kao domenskim stručnjacima omogućavajući im brzo projektovanje samostalne aplikacije. Metpodaci iz šeme BP se preuzimaju i prebacuju u tekstualni oblik JSON sintakse, koji se zatim transformiše u jezik specifičan za domen ovog softverskog rješenja. Korisniku je omogućena izmjena podataka u pomenutom obliku kao i uticaj na sam proces generisanja, uz kontrolu i sprečavanje unosa grešaka ili nevalidnih podataka. Od primljenih podataka projektno rješenje zatim generiše Java Spring API aplikaciju sa model klasama koje odgovaraju tabelama iz šeme BP.

Razmatrani pravci daljeg razvoja su unapređenje generatora tako da ih je moguće nezavisno koristiti. Primjer za to je generisanje samo model klasa ili generisanje samo repozitorijuma ili samo servis sloja. Zatim, mogućnost uticaja korisnika na zavisnosti koje se generišu u *pom.xml* datoteku, kao i mogućnost odabira njihovih verzija.

U pogledu proširenja samog softverskog rješenja, kao sledeći korak u razvoju je projektovanje grafičkog interfejsa koji bi komunicirao za postojećim rješenjem. Na taj način bi korisnik imao pregledniji uvid u podatke, za razliku od sadašnjeg tekstualnog fajla. Kao predlog za ovo je web aplikacija napisana u odabranom programskom jeziku koja komunicira preko REST HTTP protokola. Sa druge strane, može se raditi na proširenju generatora za druge jezike i platforme. U tom slučaju se modul za preuzimanje metapodataka može u potpunosti iskoristiti.

5. LITERATURA

- [1] G. Milosavljević, materijali sa predmeta „Metodologije brzog razvoja softvera“, Fakultet tehničkih nauka, Novi Sad, 2017
- [2] „Model Driven Engineering“ [Na mreži]. Dostupno na: https://en.wikipedia.org/wiki/Model_driven_engineering. [Pristupljeno: 15-Okt-2018].
- [3] „Java Programming Language“ [Na mreži]. Dostupno na: <https://docs.oracle.com/javase/8/docs/> [Pristupljeno: 10-Jun-2018].
- [4] „Spring Framework“ [Na mreži]. Dostupno na: <https://docs.spring.io/spring/docs/current/spring-framework-reference/> [Pristupljeno: 7-Jul-2018].
- [5] „MySQL“ [Na mreži]. Dostupno na: <https://www.mysql.com/> [Pristupljeno: 17-Okt-2018].
- [6] „Apache FreeMarker“ [Na mreži]. Dostupno na: <https://freemarker.apache.org/> [Pristupljeno: 4-Okt-2018].
- [7] „Hibernate Mapper“ [Na mreži]. Dostupno na: <http://hibernate.org/> [Pristupljeno: 22-Okt-2018].
- [8] „Spring Data“ [Na mreži]. Dostupno na: <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/> [Pristupljeno: 20-Sep-2018].
- [9] „Model View Controller Design Pattern“ [Na mreži]. Dostupno na: <https://en.wikipedia.org/wiki/Model-view-controller> [Pristupljeno: 17-Sep-2018].
- [10] „DatabaseMetadata“ [Na mreži]. Dostupno na: <https://docs.oracle.com/javase/7/docs/api/java/sql/DatabaseMetaData.html> [Pristupljeno: 17-Sep-2018].

Kratka biografija:



Jana Vojnović rođena je na Ilidži 1994. Osnovne studije iz oblasti Računarstvo i automatika – Primjenjene računarske nauke i informatika je završila 2016. godine kada je upisala i master studije iz iste oblasti.

INTERPRETER REST API SERVISIA BAZIRAN NA OPENAPI SPECIFIKACIJI REST API SERVICES INTERPRETER BASED ON OPENAPI SPECIFICATION

Sergej Kešelj, *Fakultet tehničkih nauka, Novi Sad*

Oblast- ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad prikazuje jednu implementaciju softverskog sistema za interpretiranje modela i njegovo izvršavanje u obliku generičkih REST API servisa opisanih OpenAPI dokumentom.

Ključne reči: MDE, MDA, REST, OpenAPI

Abstract – This paper features an implementation of a software system for interpreting and execution of models as generic REST API services described by an OpenAPI document.

Keywords: MDE, MDA, REST, OpenAPI

1. UVOD

Od prve pojave veb interfejsa za programiranje aplikacija, novine poput računarstva u oblaku ili platformi prenosivih uređaja, motivisale su promene i u pristupu razvoju interfejsa uslovljenih ovim novim potrebama.

Potreba za standardizacijom protokola i pristupa razvoju uslovila je pojavu REST stila arhitekture softverskih sistema 2000. godine. Mrežni interfejsi za programiranje koji su u skladu sa ograničenjima REST stila arhitekture se nazivaju i RESTful interfejsima. Ovaj stil arhitekture se snažno oslanja na postojeću infrastrukturu interneta, kao što je HTTP (*Hypertext Transfer Protocol*), URL (*Uniform Resource Locator*), internet vrste medija ili HTTP glagoli.

Tehnike brzog razvoja softvera poput MDE (*Model Driven Engineering*) ili agilnih metodologija razvoja, mogu nam teorijski i praktično pomoći u brzom razvoju složenih softverskih sistema. Neke od tehnika brzog razvoja softvera uključuju i generisanje izvršnog koda rešenja koji se dalje, zajedno sa ručno pisanim prilagođenim kodom isporučuje u produkcionu okruženja.

Ovaj rad se bavi implementacijom rešenja za dinamičko izvršavanje modela, korišćenjem tehnika inženjerstva vođenog modelima.

Ovako interpretirane modele, pomoću generičkih RESTful interfejsa za programiranje aplikacija, možemo opisati dinamički generisanim standardizovanim OpenAPI dokumentima (slika 1). Ovo će nam omogućiti da koristimo postojeća rešenja za generisanje klijentskih aplikacija za ovako interpretirane RESTful interfejse, jer ćemo kao ulaznu informaciju moći da koristimo generisan OpenAPI dokument koji opisuje naš interpretirani model, odnosno njegov generički RESTful interfejs (slika 2).

2. RAZVOJ SOFTVERA VOĐEN MODELIMA

Razmatraćemo razvoj softvera vođen modelima kroz sledeće bitne elemente:

1. Inženjerstvo upravljano modelima
2. Arhitekture upravljane modelima
3. Generisanje koda iz modela
4. Izvršavanje, odnosno interpretiranje modela
5. Evoluciju modela.

2.1. Inženjerstvo upravljano modelima

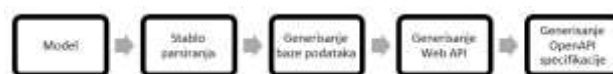
Apstrakcije su jedan od osnovnih principa inženjeringa softvera. Apstrakcije se iz realnog okruženja prevode u računarsko okruženje pomoću modela. Modelovanje i transformacije modela čine srž razvoja upravljanog modelima. Modele je moguće rafinirati i transformisati u konkretne tehničke implementacije, odnosno softverske sisteme.

OMG (*Object Management Group*) konzorcijum za tehnološke standarde opisuje inženjerstvo upravljano modelima kao metodologiju razvoja softvera koja se fokusira na kreiranje i korišćenje domenskih modela, koji predstavljaju konceptualne modele za sve teme vezane za specifičan problem koji se opisuje.

Moglo bi se reći da MDE (*Model-driven Engineering*) ima za cilj uspostavljanje apstraktne reprezentacije znanja i aktivnosti u okviru određenog aplikativnog domena. Ovo je drugačiji pogled na problematiku implementacije softvera koja se oslanja na pristupe okrenute algoritamskim konceptima.

Smatra se da je MDE paradigma uspešna ukoliko proizvedeni modeli imaju smisla za poznavaoce domena, dok istovremeno mogu poslužiti kao osnova za automatizovanu implementaciju sistema. Cilj je da se modeli razvijaju kroz intenzivnu saradnju i komunikaciju između svih strana umešanih u projekat.

Kako je jedan od ciljeva OMG konzorcijuma da svaki standard ili metodologija ima i konkretne alate i principe primene, nastao je pristup razvoju softverskih sistema pod nazivom MDA (*Model-driven architecture*).



Slika1. Proces interpretacije modela na serveru

NAPOMENA:

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



Slika 2. Proces generisanja klijenta

2.2. Arhitektura upravljana modelima

Prvi put 2000. godine OMG konzorcijum pominje MDA (*Model-driven architecture*) kao viziju razvoja softvera koja se oslanja na vezivanje modela u celinu koja bi predstavljala kompletan opis sistema. Ovaj pristup bi trebalo da koristi postojeće tehnologije koje podržavaju postojeće, ali i buduće OMG standarde i kao rezultat toga daju podršku razvoju upravljanim modelima, tako da modeli postanu dobra koja, prilikom promene tehnološke platforme, mogu da se ponovo iskoriste, umesto troška implementacije svakog pojedinačnog sistema od nule.

Danas modeli i dalje najčešće predstavljaju trošak. Jednom kada je model dizajniran, mora biti transformisan u izvršni kod. Ovo je težak, dugotrajan, sklon greškama i pre svega skup proces. Dalje, nakon što je posao apstrakovanja realnog domena u modele završen, samo je transformacija iz izvornog koda u izvršni kod automatizovana.

MDA je rezultat prepoznavanja da je modelovanje, interoperabilnost i nezavisnost od konkretnih tehničkih platformi dobra stvar. Ovaj pristup dozvoljava programerima da izgrade modele bez znanja o drugim modelima koji će činiti krajnji sistem. Ovakvi modeli se kombinuju u konačni sistem tek na kraju procesa, odnosno što kasnije u procesu. Ovo sprečava odluke koje su vezane za dizajn rano u procesu, pa samim tim čini modele nezavisnim od platforme za implementaciju krajnjeg sistema. U krajnjem određenju, aplikacija postaje nezavisna od implementacije, pa može biti kombinovana sa mnogim tehnologijama, kao i drugim aplikacijama i sistemima. Moglo bi se govoriti o nekoj vrsti interoperabilnosti u dizajnu, što naše modele čini vrednostima, a ne troškovima.

2.3. Generisanje koda iz modela

Generisanje koda je dobro poznati način izvođenja implementacije iz modela sistema. Računar izvodi izvršni kod direktno iz modela. Ovaj proces može rezultirati različitim kvalitetom generisanog koda. Često se proizvodi samo osnovni kod. Ovaj osnovni kod čine nekompletni fragmenti koda sa namerom da isti budu kompletirani od strane programera u jeziku generisanog koda.

Ukoliko modeli sadrže kompletne informacije o sistemu moguće je generisati kompletno operativan sistem iz modela. U ovom slučaju generisani kod može da se kompajlira u izvršni sistem bez potrebe za intervencijom programera. Tada, kod najčešće niti može, niti bi trebalo da bude modifikovan od strane čoveka, jer bi to prouzrokovalo problem sa integracijom izmena nazad u modele. Samo je sistem kompletno generisan iz modela u potpunoj saglasnosti sa samim modelom. Kako je očekivano da modeli budu u ekvivalenciji sa sistemom, nije poželjno dozvoliti izmene u generisanom kodu.

Ovakve izmene bi morale biti načinjene i u modelima. Iz tog razloga samo su izmene na modelima poželjne.

Bez obzira na cilj da se izmene rade samo u modelima, većina alata koja generiše kod nije u stanju da generiše kompletan sistem. Često je neophodno modifikovati generisani kod, što implicira da modeli nemaju kompletnu informaciju o sistemu koji je modeliran. Ovo se naročito odnosi na MDA nivo modelovanja sistema.

2.4. Interpretiranje modela

Jedan način sinhronizacije implementacije i dizajna jeste korišćenje modela kao implementacije. Jezik za opis modela bi morao biti dovoljno informativan da pruži dovoljno detalja za implementaciju.

Druga komponenta ovakvog sistema bi bila neka vrsta virtualne mašine koja bi bila u stanju da izvršava ovako opisane modele. Korišćenjem virtualne mašine eliminišemo potrebu da se model transformiše u različite programske jezike korišćenjem generatora koda.

Ovakav pristup pruža nekoliko prednosti. Model postaje direktno izvršiv na svakoj platformi koja obezbeđuje odgovarajuću virtualnu mašinu. Ovo eliminiše dodatni napor potreban da bi se generisao novi izvršni kod svaki put kada se menja ciljanoj platforma ili kada se dodaje neki novi mehanizam u softversko rešenje koje se već izvršava na ciljanoj platformi.

Ovakva ažuriranja se mogu vršiti direktno na virtuelnoj mašini bez potrebe da se menja model koji se izvršava. Dakle, za razliku od pristupa sa generisanim kodom, ne postoji potreba za ponovnim generisanjem za novu platformu ili za novu verziju postojeće platforme.

Ovo je značajno unapređenje, jer sve izmene koje bi uvodili u virtuelno okruženje mogu da se primene na sve postojeće sisteme, prethodno modelirane, a koji se već nalaze u izvršnom stanju. Ovo višestruko smanjuje vreme u toku razvoja. Može se smatrati da je potrošeno vreme minimalno moguće.

2.4. Evolucija modela

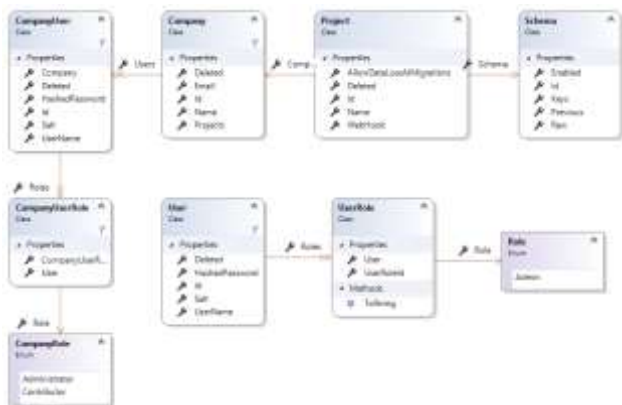
Samo jedna stvar je konstantna u razvoju softverskih sistema, a to je da će biti promena. Ova jednostavna činjenica je uticala na velike izmene u pristupu razvoju softvera i između ostalog dovela nas od *waterfall* modela do agilnog manifesta, pa dalje do agilnih metodologija razvoja softvera.

U skladu sa ovim trendom, može se govoriti o evolutivnom razvoju softvera i svih njegovih komponenti, odnosno slojeva. U kontekstu MDE možemo govoriti o agilnom pristupu evoluciji baza podataka kao najčešćeg krajnjeg određenja podataka većine softverskih sistema. Konkretno pitanje koje se postavlja je kako možemo automatizovati detekciju i primenu izmena modela na postojeće softverske sisteme u svetlu interpretacije modela?

Postoji ograničen skup operacija koje možemo primeniti kako bismo automatizovali razumevanje izmena nad modelom i njihovog uticaja na model baze podataka. Izdvajaju se dva podskupa potencijalnih migracionih promena zavisno od toga da li dolazi do gubitka podataka evolucijom modela.

3. IMPLEMENTACIJA SISTEMA

Projekat je realizovan kao SaaS (Software as a Service) rešenje. Svaka kompanija može da upravlja svojim korisnicima i projektima. Projekat je predstavljen podacima o samom projektu, konfiguracijom projekta i modelom. Model je reprezentovan korišćenjem OpenAPI specifikacije šema objekta i OpenAPI proširenjem kojim se opisuju primarni ključevi modela (slika 3).



Slika 3. Prikaz dijagrama klasa sistema

3.1. Dinamička interpretacija modela

Modeli se u sistemu čuvaju u okviru entiteta *Schema*. Ovaj entitet čuva opis modela u polju *Raw*. Model se opisuje JSON ili YAML dokumentom po OpenAPI specifikaciji za opis OpenAPI info objekta. Primer 1 prikazuje jedan model u polju *Raw*:

```
Pet:
  required:
    - id
    - name
  properties:
    id:
      type: integer
      format: int64
    name:
      type: string
    family:
      type: string
```

Primer 1. Opis modela Pet

Zbog kompletnosti modela bilo je neophodno proširiti OpenApi specifikaciju meta podacima koji opisuju primarne ključeve modela. Za proširenje specifikacije iskorišćeno je svojstvo proširivosti OpenAPI specifikacije. Primer 2 daje primer proširenja specifikacije koja definiše meta podatke o primarnom ključu entiteta definisanog modelom:

```
x-entity-extensions:
  Pet:
    keys:
      - id
```

Primer 2. Opis proširenja meta-modela

3.2. Dinamičko generisanje tipova klasa

Na osnovu stabla parsiranja generišemo .NET klase u vreme izvršavanja aplikacije (slika 4). Ovako generisane

klase možemo iskoristiti da uz pomoć objektno-relacionog mapera kreiramo bazu podataka koja bi odgovarala modelu. Ovaj korak je neophodan kako bismo mogli interpretirati modele i omogućiti njihovo izvršavanje u okviru našeg projekta, a bez međukoraka generisanja koda.

```
public static EntityTypes CompileResultTypes(this IOpenApiDocument document,
                                             string company,
                                             string project)
{
    var result = new ConcurrentDictionary<string, EntityTypeInfo>();
    var schemasWithPrimaryKey = document.Components.Schemas.AsEnumerable();
    var typeBuilders = new Dictionary<string, TypeBuilder>();
    var modelBuilder = GetModelBuilder(company, project);
    var assemblyName = $"TestApiIntegrater.WebApi.Tenants.{company}.Projects.{project}";

    foreach (var schema in schemasWithPrimaryKey)
    {
        foreach (KeyValuePair<string, TypeBuilder> th in typeBuilders)
        {
            Type objectType = th.Value.CreateType();
            result.TryAdd(objectType.Name, new EntityTypeInfo() {
                Type = objectType,
                Keys = FindKeysFromExtensions(document, th.Key),
                Required = FindRequiredFromExtensions(document, th.Key) });
        }
    }
    return new EntityTypes(result, company, project);
}
```

Slika 4. Prikaz koda za dinamičko kreiranje klasa

3.3. Dinamičko generisanje DbContext objekta

Nakon što smo dinamički kreirali klase možemo preći na perzistenciju objekata tako što ćemo dinamički kreirati DbContext objekat. Dinamičko ponašanje smo implementirali tako što smo nasledili DbContext klasu iz EntityFramework biblioteke i izmenili njeno ponašanje tako da šemu baze podataka kreira na osnovu klasa koje smo dinamički izgenerisali i dodali u kontekst. Kako se radi o multi-tenant rešenju, morali smo obratiti pažnju i na to da će svaki projekat imati različitu bazu i različit model baze.

3.4. Migracije šeme baze podataka

Evolucija modela direktno utiče na šemu baze podataka. Kako je evolucija produkcionih softverskih sistema neminovna, iskoristićemo EntityFramework podršku za migracije i omogućiti, zavisno od konfiguracije projekta, migracije sa i bez gubitka podataka.

Ponašanje implementiramo nasleđivanjem i proširivanjem ponašanja klase DbMigrationsConfiguration iz biblioteke EntityFramework.

3.5. Dinamička interpretacija REST API servisa

Potreban nam je generički ASP.NET Web API kontroler koji će prihvatati zahteve klijentskih aplikacija i izvršavati osnovne operacije nad interpretiranim modelima. Parametri akcije nam pomažu da odredimo tačan model koji želimo da izvršimo, a HTTP glagoli određuje vrstu operacije.

3.6. Generisanje OpenAPI dokumenta iz modela

Pošto imamo sve neophodne informacije o modelu i REST API servisu možemo iskoristiti OpenAPI .NET SDK da generišemo OpenAPI dokument koji opisuje REST API servis preko kog pristupamo izvršavanju modela. Ovakav dokument se može koristiti kao ulaz za druge alate, koji mogu generisati dokumentaciju REST API servisa ili klijentske aplikacije.

3.7. Generisanje klijentskih aplikacija

OpenAPI dokument koji opisuje REST API servis sadrži sve podatke potrebne za generisanje dokumentacije ili klijentskih aplikacija. Pošto je OpenAPI standardizovan, već postoji veći broj alata.

Mi ćemo ovde navesti primer korišćenja alata AutoRest. Ovaj alat može da se koristi sa različitih platformi (Windows, MacOS ili Linux) i da se pomoću njega generiše klijentski kod u nekom od ponuđenih programskih jezika poput: C#, Go, Java, Python, NodeJS i drugih.

Kako je alat proširiv i otvorenog koda, za očekivati je da se podrška za jezike i dalje unapređuje.

6. ZAKLJUČAK

Inženjerstvo vođeno modelima i arhitekture upravljane modelima gledaju na problematiku implementacije softverskih sistema kao dobro definisanih modela koji su nezavisni i samodovoljni. Ovako definisani modeli postaju vrednost koja može da se upotrebljava za građenje konačnih softverskih sistema, kao i za ponovnu upotrebu u različitim softverskim sistemima.

Arhitektura upravljana modelima daje dva pristupa problemu transformacije modela u konačni izvršni softverski sistem: pomoću generisanja koda ili pomoću softverskog sistema za interpretaciju modela.

U ovom radu smo se bavili implementacijom softverskog sistema koji bi bio u stanju da izvršava modele tako što bi generisao sve potrebne elemente dinamički u vreme izvršavanja i praktično interpretirao modele bilo kog sistema opisanog modelima.

Rešili smo da sistem implementiramo kao generički REST API koji će biti u stanju da generiše OpenAPI dokument koji opisuje REST API posmatranog modela.

Ovako generisan dokument smo koristili kao ulaz za alate koji na osnovu OpenAPI dokumenta mogu da generišu dokumentaciju ili klijentske aplikacije, poput AutoRest alata.

Tako smo uspešno zaokružili sistem koji omogućava korisniku kreiranje i interpretaciju, odnosno izvršavanje modela, bez potrebe za generisanjem izvornog koda.

Dodatno, ovaj softverski sistem omogućava evoluciju modela pomoću migracija podataka, kao i povezivanje sa drugim sistemima mehanizmima definisanja *WebHook* URL adresa.

Ovaj sistem predstavlja dokaz koncepta i kao takav ima dovoljno prostora za dalja unapređenja. Potrebno je implementirati sigurnosne mehanizme u opis modela. Ovo bi se moglo postići korišćenjem nekih od sigurnosnih mehanizama opisanih OpenAPI specifikacijom, a implementiranih u tehnologijama koje koristimo u ovom projektu. Podrška za evoluciju modela bi mogla da se poboljša uvođenjem bolje kontrole operacija koje se vrše u toku migracija. Još jedan od pravaca unapređenja sistema bi mogao biti bolja kontrola instrumentacije sistema uvođenjem dnevnika aktivnosti koji bi mogao da se konfiguriše na nivou projekta.

7. LITERATURA

- [1] L. Richardson, S. Ruby (2007), RESTful Web Services, O'Reilly Media, Inc. Sebastopol, CA
- [2] Kent S. (2002) Model Driven Engineering. In: Butler M., Petre L., Sere K. (eds) Integrated Formal Methods. IFM 2002. Lecture Notes in Computer Science, vol 2335. Springer, Berlin, Heidelberg
- [3] Beydeda S., Book M., Volker G. (2005) Model-Driven Software Development, Springer-Verlag GmbH ISBN 978-3-540-28554-0
- [4] Stephen J. Mellor, Kendall Scott, Axel Uhl, Dirk Weise (2004) MDA Distilled: Principles of Model-driven Architecture
- [5] Tim Schattkowsky (2003), Model-based Development of Embedded Systems: Executable Models vs. Code Generation
- [6] Scott W. Ambler, Agile Database Techniques—Effective Strategies for the Agile Software Developer
- [7] F. Chong, G. Carraro, R. Wolter (2006), Multi-Tenant Data Architecture, Microsoft Corporation

Kratka biografija:



Sergej Kešelj rođen je u Mostaru 1979. god. Osnovne studije na Prirodno-matematičkom fakultetu u Novom Sadu, iz oblasti Informatike je završio 2006. godine. Od 2016. je student akademskih master studija na Fakultetu tehničkih nauka na smeru Softversko inženjerstvo. Master rad je odbranio 2018. godine.

MODELOVANJE I IZRADA KONTROLERA ZA ŽALUZINE ZASNOVANOG NA ALGORITMU FUZZY KONTROLERA ZA OPTIMIZACIJU UGLA ZAKRIVLJENOSTI**MODELING AND IMPLEMENTATION OF BLINDS CONTROLLER BASED ON ALGORITHM OF FUZZY CONTROLLER FOR BLINDS TILT ANGLE OPTIMIZATION**Marko Paspalj, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Ovim radom je obuhvaćeno modelovanje i izrada kontrolera za upravljanje žaluzinama u cilju boljeg iskorišćenja prirodne svetlosti i uštede energije. Kontroler se zasniva na algoritmu dobijenom pomoću fuzzy logike koji vrši optimizaciju ugla zakrivljenosti, a proračun uštede energije se oslanja na matematički model za proračunavanje uticaja sunčevog zračenja na zagrevanje i dodatno veštačko osvetljenje prostorije. Kontroler je testiran sa različitim setovima ulaznih parametara kako bi se proverio uticaj kontrolera na uštedu energije.

Ključne reči: Kontroler, osvetljenost, ušteda, ugao zakrivljenosti

Abstract – Subject of this paper covers modeling and implementation of blinds controller with aim of better use of natural light and power savings. Controller is based on fuzzy logic algorithm which optimizes tilt angle of blinds, and calculates power savings using mathematical model for solar hit gain and additional artificial lighting of the room. Controller is tested with different sets of input parameters to determine impact of controller on power savings.

Key words: controller, natural light, savings, angle of blinds

1. UVOD

Većina energije koja je dostupna na Zemlji dolazi nam od Sunca, a čak i energija koju crpimo iz drugih izvora (vetar, voda, biomasa...) zapravo potiče od njega. Primenom odgovarajuće tehnologije se može iskoristiti ovaj potencijal imajući u vidu da je Sunce ekološki izvor energije, besplatan i večan.

Broj časova sunčevog zračenja na teritoriji Srbije iznosi između 1.500 i 2.200 časova godišnje. Prosečan intenzitet sunčevog zračenja je od 1,1 kWh/m²/dan na severu do 1,7 kWh/m²/dan na jugu - tokom januara, a od 5,9 do 6,6 kWh/m²/dan - tokom jula. Prosečnavrednost energije zračenja iznosi od 1.200 kWh/m²/godišnje u severozapadnoj Srbiji, do 1.550 kWh/m²/godišnje u jugoistočnoj Srbiji, dok u centralnom delu iznosi oko 1.400 kWh/m²/godišnje [1].

U cilju što bolje iskoristivosti Sunčeve svetlosti i toplote od izuzetnog značaja je i orijentacija objekta:

NAPOMENA:

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

- prostorije okrenute ka jugu će imati dobro dnevno osvetljenje i tokom čitave godine imaće dobitke od sunčevog zračenja

- prostorije okrenute ka istoku će imati dobro jutarnje osvetljenje i toplone dobitke u tom periodu

- prostorije okrenute ka zapadu će imati dobro osvetljenje popodne

- prostorije okrenute ka severu neće imati dovoljno dnevnog svetla tokom čitave godine i zbog toga će imati male toplotne dobiti [2].

U zavisnosti od godišnjeg doba i orijentacije prostorija zavisi i količina toplotne dobiti koja utiče na njeno zagrevanje. Imajući u vidu sve navedene faktore, kontrolom žaluzina se može uticati na što bolje iskorišćenje sunčeve energije za klimatizaciju.

2. MATEMATIČKI MODEL

Solarna toplotna dobit (eng. Solar HeatGain, SHG) se odnosi na povećanje toplote nekog objekta, prostorije ili strukture, koje se javlja kao posledicadelovanja sunčevih zraka. Izračunavanju konačne toplotne dobiti kroz prozor prethodi niz matematičkih jednačina i proračuna vezanih za prateće koeficijente propusnosti, faktore pogleda, faktore refleksija i drugih karakteristika sredina i materijala kroz koje prolazi Sunčev zrak [3].

Modeli za toplotnu dobit pri vedrom i oblačnom danu su predstavljeni izrazima (1) i (2) respektivno:

$$SHG_{\text{clear}} = I_{\text{clear}}^{\text{total}} \cdot SHGC_{\text{win}} \quad (1)$$

$$SHG_{\text{overcast}} = I_{\text{total}}^{\text{overcast}} \cdot SHGC_{\text{win}} \left[\frac{W}{m^2} \right] \quad (2)$$

gde je:

$I_{\text{clear}}^{\text{total}}$ - ukupno zračenje na površinu prozora za vedar dan;
 $SHGC_{\text{win}}$ – koeficijent toplotne dobiti prozora;

$I_{\text{total}}^{\text{overcast}}$ – ukupno zračenje na površinu prozora za oblačan dan.

2.1. Model prostorije

Za određivanje osvetljenosti prostorije veoma su bitne njene karakteristike jer one znatno utiču na promenu osvetljenosti. Prostorija je pravilnog četvorougao oblika sa jednim spoljnim zidom na kome se nalazi zastakljena površina [3].

Koordinate tačke u kojoj se meri osvetljenost računaju se iz izraza (3) i (4):

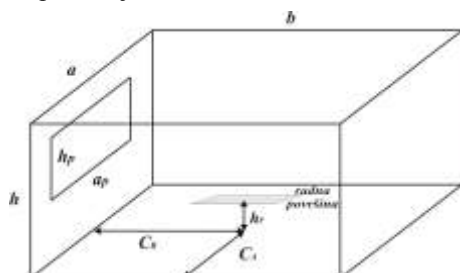
$$C_A = \frac{a}{2} [m]$$

$$C_B = \frac{b}{2} [m]$$

gdeje:

a – širina prostorije;

b – dužina prostorije.



Slika 1. Koordinate tačke u kojoj se meri osvetljenost

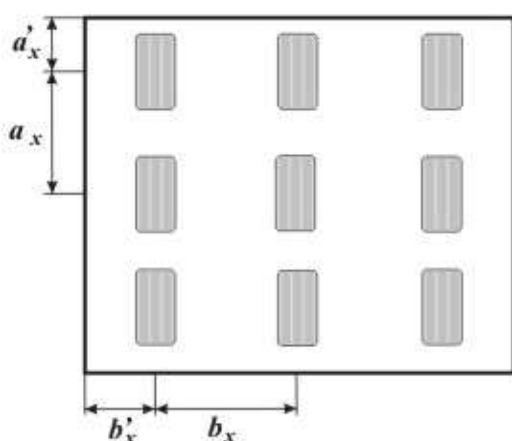
2.2. Proračun veštačkog osvetljenja

Nakon definisanja modela za računanje svetlosne dobiti neophodno je izračunati zahteve za veštakim osvetljenjem. Za željenu vrednost osvetljenosti je izabrano 500 lux, koja je dovoljna za posmatrani prostor i radne površine.

Odabrane svetiljke “MENLOSFT SR 3X24W HFS DMB WL5 L840” snabdevene su sa po tri fluorescente T5 cevi i visokofrekventnom digitalnom prigušnicom koja se nalazi u predspojnom uređaju snage 5W [3].

Da bi se dimenzionisalo veštačko osvetljenje neophodno je izračunati stepen iskorišćenja svetiljke, svetlosni fluks, broj potrebnih svetiljki, stvarnu srednju osvetljenost i maksimalnu potrošnju. Svetiljke je potrebno pravilno rasporediti kako bi se minimizovala senke.

Određivanje rasporeda svetiljki je jednostavnije jer je prostorija pravougaonog oblika, nego što bi bio ako bi prostorija imala neki nepravilan oblik. Primenom jednostavnih relacija na primeru pomatrane prostorije dobija se sledeći raspored svetiljki, prikazan na Slici 1 [3].



Slika 2. Prikaz rasporeda svetiljki

2.3. Proračun uređaja za grejanje/hlađenje prostorije

Kako se pri proračunima uzima 90W/m² kao prosečna godišnja potrošnja po kvadratu prostorije kojom će se zadovoljiti potrebe grejanja/hlađenja, lako se dolazi do toga da je potreban uređaj snage 2700W. Na osnovu toga tražen je fan coil uređaj, sa ugradnjom na podu. Odabran je uređaj firme DAIKIN, serije FVXS-F/RXS-G[3].

3. FUZZY LOGIKA

Koristeći klasičnu logiku, moguće je raditi jedino sa informacijama koje su ili u potpunosti tačne ili u potpunosti pogrešne. Nije moguće upravljati informacijama koje su neprecizne ili nekompletne, iako ovakve informacije mogu pružiti bolje rešenje nekog problema.

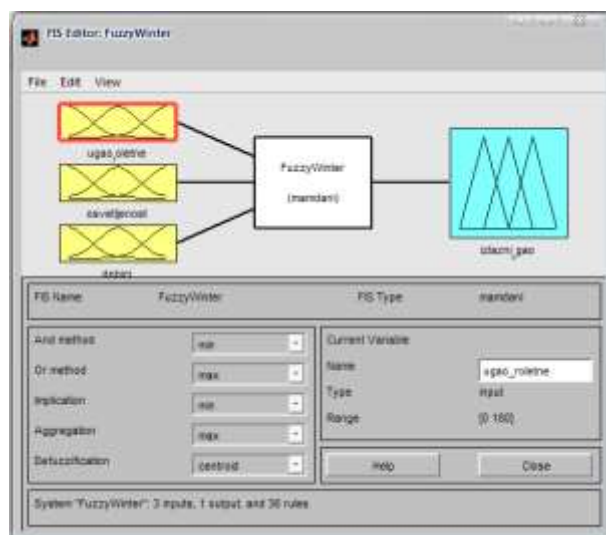
Pojam fazi skupa je u svom radu uveo Zadeh (Zadeh, 1965), koji se smatra i tvorcem fazi logike. Fazi logika je logika koja stoji iza približnog umesto preciznog zaključivanja. Njen značaj i veliki potencijal leže u činjenici da je ljudsko razmišljanje po prirodi aproksimativno [4].

3.1. Fuzzy kontroler za upravljanje žaluzinama

Realizacija fuzzy kontrolera za upravljanje uglom zakrivljenosti žaluzina se zasniva na definisanju sledećih ulaza: trenutni ugao žaluzina, toplotna dobit i spoljašnja osvetljenost. Kontroler generiše upravljanje na osnovu baze pravila i određuje optimalni ugao žaluzina za navedene uslove.

Kriterijumi optimalnosti u ovom slučaju su postizanje željene osvetljenosti i temperature unutar prostorije uz maksimalnu iskoristivost prirodne svetlosti i uštedu energije.

Na Slici 2. je prikazan izgled fuzzy kontrolera za upravljanje uglom zakrivljenosti žaluzina [5].



Slika 3. Fuzzy kontroler za upravljanje žaluzinama

4. ALGORITAM ZA UPRAVLJANJE ŽALUZINAMA ZASNOVAN NA FUZZY KONTROLERU

Sam algoritam za upavljanje žaluzinama predstavljen je pomoću UML dijagrama. UML ili Unified Modeling Language je grafički jezik za vizuelizaciju, specifikaciju, konstruisanje i dokumentovanje sistema programske podrške. Predstavlja kolekciju najboljih uspešnih inženjerskih metoda kojima je, iskustveno dokazano, pojednostavljeno modelovanje velikih i složenih sistema [5].

Najbitniji uslov za sam algoritam jeste spoljašnja temperatura, na osnovu koje se odlučuje da li je potrebno izabrati režim za grejanje ili hlađenje prostorije.

Zatim se na osnovu izmerene spoljašnje osvetljenosti, izračunatih toplotnih dobitaka i trenutnog položaja žaluzina određuje novi ugao zakrivljenosti žaluzina [6].

5. IZBOR I REALIZACIJA LOGIKE KONTROLERA

Analizom algoritma za upravljanje žaluzinama i kriterijuma za izradu fuzzy kontrolera, izbor kontrolera se sveo na AllenBradleyMicroLogix 1100.

Na Slici 3. je prikazan izgled kontrolera MicroLogix 1100 [7].



Slika 4. Kontroler Allen Bradley MicroLogix 1100

Jedan od kriterijuma za izbor je bio i njegova mogućnost komunikacije sa Microsoft Excel-om preko Rockwel Software RS Linx Clasic-a radi simulacije, praćenja rada kontrolera i beleženja dobijenih rezultata.

Microsoft Excel pored toga što nam je svima poznat kao deo Microsoft Office paketa, veoma je koristan za lako modelovanje tabela i grafika, pruža nam mogućnost korišćenja matematičkih funkcija i upotrebu VBA (Visual Basic for Applications), što je obuhvaćeno ovim radom:

- način komunikacije VBA sa kontrolerom
- funkcije koje se koriste
- matematičke formule potrebne za simulaciju kontrolera
- vizualizacija(prikaz HMI interface-a)
- praćenje i prikaz rezultata dobijenih na ovaj način.

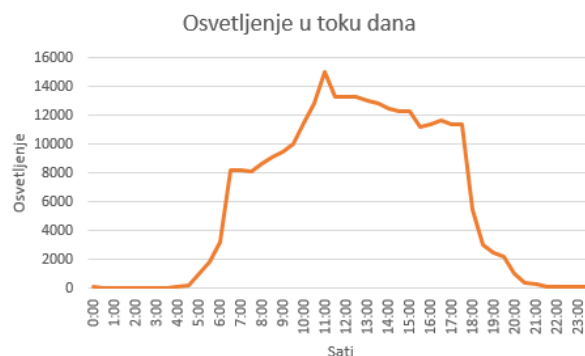
6. ANALIZA REZULTATA

U ovom poglavlju su prikazani i komentarisani rezultati dobijeni korišćenjem tehnologija objašnjenih i razrađenih u prethodnom poglavlju, a u cilju dokazivanja izvodljivosti i primene algoritma za upravljanje žaluzinama zasnovanog na Fuzzy kontroleru.

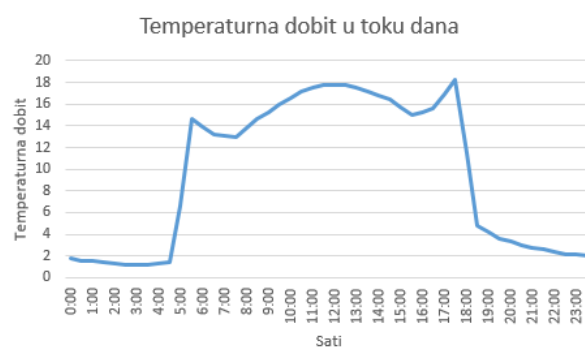
Analiza je praćena adekvatnim graphicima radi boljeg razumevanja dobijenih rezultata.

Kao primer je dat zimski dan, sa orijentacijom prozora ka severu.

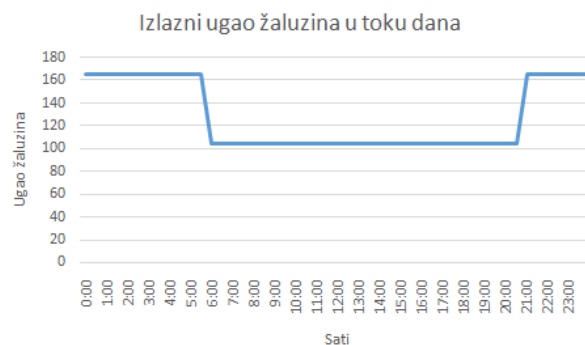
Na slikama (5), (6) i (7) prikazana je promena ugla žaluzina u zavisnosti od promene osvetljenosti i temperaturnih dobitaka u toku 24 sata.



Slika 5. Promena osvetljenosti u toku 24 sata



Slika6. Promena temperaturne dobiti u toku 24 sata



Slika7. Promena ugla žaluzina u toku 24 sata

7. ZAKLJUČAK

Prirodnih resursa je sve manje, a troškovi proizvodnje energije su sve veći. Sa energetske tačke gledišta optimalno korišćenje dnevne svetlosti je neizostavno u svim sferama našeg društva. Što zbog ekonomskih benefita od uštede redukovanjem same potrošnje energije, što zbog smanjenja emitovanja štetnih materija u našu okolinu prilikom proizvodnje iste.

Elektronski upravljanoj zaštiti od sunca moguće je ostvariti benefite pri uštedi energije prilikom grejanja i hlađenja, zbog čega je i sastavni deo moderne arhitekture.

Najveći potrošači električne energije u poslovnim zgradama i fabrikama su klimatizacija i veštačka rasveta koji su neophodan faktor za normalno funkcionisanje zaposlenih u navedenim zdanjima. Iskorišćavanjem dnevnog svetla može se uštedeti 50% do 80% na troškovima osvetljenja [7].

Sve navedeno nas navodi na zaključak da optimalno korišćenje samih žaluzina omogućava uštedu energije, uz dodatni benefit povećanja komfora unutar bilo kog prostora. Korišćenjem odgovarajućih modularnih kontrolera i njihovom implementacijom moguće je unapređivanje i pojednostavljivanje upravljanja radom žaluzina, što je i bio zadatak ovog rada.

8. LITERATURA

- [1] <https://www.energetskiportal.rs/obnovljivi-izvori-energije/energija-sunca/>
- [2] <https://www.gradnja.rs/kako-najbolje-orientisati-prostorije-u-kuci/>
- [3] Danilo Bojović, „Ušteda energije u zgradama primenom algoritma upravljanja roletnama“, Fakultet tehničkih nauka, Novi Sad, Srbija, 2011. godina.
- [4] Fuzzy logika
<http://nardus.mpn.gov.rs/bitstream/handle/123456789/1788/Disertacija.pdf?sequence=1>
- [5] „UML“, Dragan Marković
- [6] Marija Paspalj, „Izrada algoritma za upravljanje žaluzinama zasnovanog na optimizaciji ugla zakrivljenosti primenom veštačke inteligencije“, Fakultet tehničkih nauka, Novi Sad, Srbija, 2012. godina.
- [7] RockwellAutomation
<https://ab.rockwellautomation.com/Programmable-Controllers/MicroLogix-1100>
- [8] Ušteda energije, Korišćenje dnevnog svetla, HELLA
<http://www.hella.info/yu/energiesparen.html>

Kratka biografija:



Marko Paspalj rođen je u Sisku 1986. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Automatika i upravljanje sistemima odbranio je 2018.god.

СИСТЕМ ЗА ПЛАЋАЊЕ УТРОШЕНОГ ВРЕМЕНА**SYSTEM FOR PAYING SPENT TIME**Виолета Новаковић, *Факултет техничких наука, Нови Сад***Област – РАЧУНАРСТВО И АУТОМАТИКА**

Кратак садржај – У раду је описан систем за плаћање утрошеног времена. Специфициран је модел описаног система на основу кога је извршена имплементација. Такође, дат је и преглед сличних решења.

Кључне речи: Е-плаћање, време, QR код, PayPal, Bitcoin

Abstract – This paper presents the System for paying spent time. Detailed description of a model through diagrams is given as well as implementation details and system overview. Similar solutions are presented.

Keywords: E-payment, time, QR code, PayPal, Bitcoin

1. УВОД

Различити модели и технологије система електронског плаћања постају све више заступљени у савременом платном промету. Пре свега захваљујући њиховој ефикасности, погодности и правовремености. Ради се о системима који се константно прихватају и усвајају у финансијским системима како развијених тако и мање развијених земаља са циљем да упрости и олакша пословне трансакције и смањи дуге процедуре као што су одлазак у банку и чекање у понекад веома дугим редовима [1].

Појава информационих и комуникационих технологија је у потпуности променило животе и операције појединаца и организација. Долазак овог развоја у глобалном пословном окружењу изазвао је већину организација да се аутоматски пребаце са конвенционалних новчаних трансакција папирним новцем на електронско плаћање [1].

Неки од видова електронског плаћања доступних корисницима су [2]:

1. Картице са чипом и пином,
2. Картице са безkontakном технологијом,
3. *Prepaid* преносни уређаји,
4. Дигитални новчаници који користе NFC,
5. Дигитални новчаници за плаћање у апликацији и др.

Широко распрострањено прихватање ових начина плаћања нуди бројне могућности као што су [3]:

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Горан Сладић, ванр. проф.

1. Развој нових платних система,
2. Боља заштита потрошача,
3. Смањење сиве економије,
4. Нижи оперативни трошкови и др.

О томе колико је електронско плаћање битно, говори податак да је е-трговина, која је уско повезана са е-плаћањем, једина индустријска грана која је одолела светској економској кризи у протеклих пет година [3].

Тема овог рада се бави апликацијом која представља дигитални новчаник за плаћање утрошеног времена на брз и ефикасан начин. Корисницима је неопходно да поседују паметни уређај, интернет и неки од платних средстава као што су кредитне картице, *PayPal*, *Amazon* и сл. налози, а такође постоји и могућност коришћења дигиталних валута као што су *Bitcoin*, *Etherium* и сл.

Систем за плаћање утрошеног времена настоји да омогући корисницима да самостално одлуче колико времена изискује њихова радња и да плате само онолико колико потроше.

Овај систем је настао из идеје да се карте у градском превозу на јединствен начин плаћају свуда по држави и то на што ефикаснији начин. Та ефикасност би се постигла не само брзином којом би се обављала куповина карата него и давањем могућности корисницима да сами бирају колико времена за возњу ће уплатити. На тај начин би корисници избегли плаћање карте коју не би у потпуности искористили, а због једноставности коришћења и привидно јефтинијим ценама, и већина „падобранаца“ не би оклевала да плати своје време проведено у превозу.

Поред могућности примене овог система за потребе наплатног система у градском превозу, нека од потенцијалних места за примену истог би могла бити: плаћање паркинг места, изнајмљивање разне врсте опреме, спортских терена, свечаних сала, плаћање боравка у високо прометним ресторанима и сл.

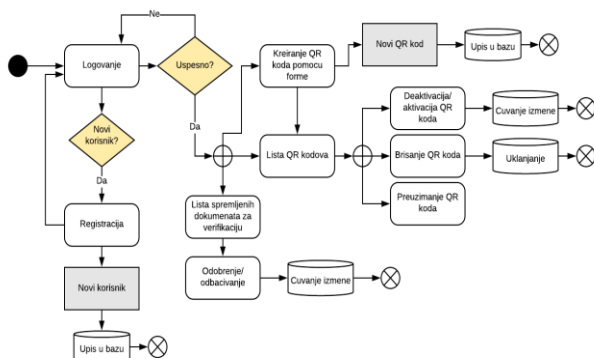
2. ТОК АКТИВНОСТИ СИСТЕМА

На слици 1 је приказан ток активности приликом коришћења веб апликације од стране продаваца.

Ток активности креће од црног круга који представља полазну тачку и из које следи аутентификација корисника, у улози продавца, на систем. Уколико је продавац нов и тек започиње да користи овај систем у оквиру свог бизниса, треба да се региструје, при чему се он заводи у базу података. Након успешног регис-

тровања, може прећи на аутентификацију. Након успешног логовања, продавац има могућност коришћења трију функционалности:

1. Креирање QR кода, путем попуњавања форме са неопходним подацима, након чијег чувања следи упис у базу,
2. Преглед листе постојећих кодова које је продавац креирао, и управљање њима (брисање, деактивирање и преузимање). Измене се чувају у базу,
3. Преглед листе докумената послатих на верификацију од стране корисника система (купаца) и могућност њиховог одобравања или одбијања, након чега купац бива обавештен о исходу.



Слика 1 - Дијаграм тока активности веб апликације

На слици 2 приказан је дијаграм тока активности потрошача на андроид мобилној апликацији.

Ток активности креће од црног круга који представља полазну тачку и из које следи аутентификација корисника. Након корисничког успешног логовања на систем, аутоматски се покреће тајмер за проверу валидности карата, који уколико детектује да је нека од карата истека, освежава базу новим податком, а кориснику шаље обавештајну поруку о томе. Поред ове функционалности која је невидљива на око, корисник може да приступи делу за слање докумената за верификацију профила, а ти се документи чувају на *cloud*-у и бивају доступни администраторима.

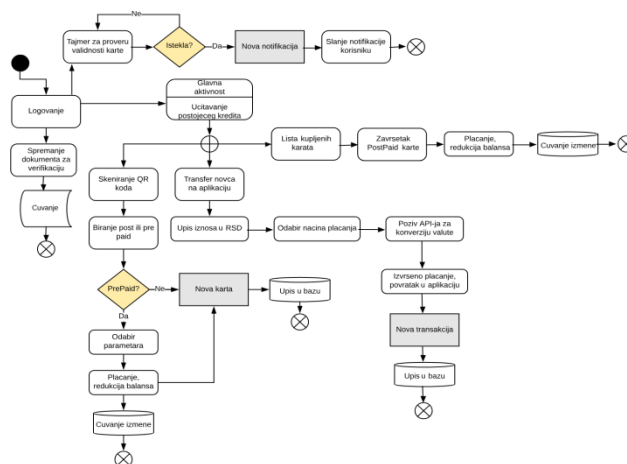
Последња и најбитнија активност која следи након логовања јесте главни екран. На главном екрану се налазе три активности.

Прва прикупља податке о тренутном балансу доступном кориснику, и активира скенер за QR код. Након успешног скенирања кода, корисник бира да ли жели купити *prepaid* или *postpaid* карту. Уколико се одлучи за прву опцију, наставља на активност за уношење параметара, а потом на плаћање, чиме се редукује баланс, ажурира база и креира нова карта, како у бази тако и у листи карата. Уколико се корисник одлучи за другу опцију, односно *postpaid* карту, креира се нова карта која се такође чува у базу и приказује у листи карата.

Друга активност врши трансфер новца на апликацију. Након уноса жељене вредности, бира се једна од понуђених опција за плаћање.

По клику на изабрано дугме, позива се *API* за конверзију валуте, јер РСД валута није једна од подржаних од стране понуђених система за плаћање. По извршеној аутентификацији и потврди трансфера, врши се повратак на апликацију, а нова трансакција бива забележена у бази, а баланс ажуриран.

Трећа активност приказује листу купљених карата и њихов статус. Овде се врши завршетак плаћања *postpaid* карте, измене се ажурирају у бази, као и у апликацији.



Слика 2 - Дијаграм тока активности андроид мобилне апликације

4. ИМПЛЕМЕНТАЦИЈА СИСТЕМА

За аутентификацију на мобилну апликацију, коришћена је подршка за логовање помоћу *Google* налога. На листингу 1 је приказан део кода којим се остварује веза за аутентификацију. Да би било могуће имплементирати овај *API* у апликацији, неопходно је да се креира одређени токен и укључи у пројекат.

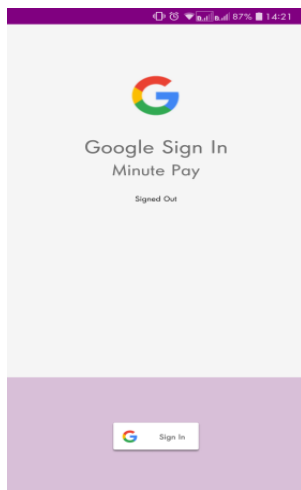
```
GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignIn
nOptions.DEFAULT_SIGN_IN)
```

```
.requestIdToken(getString(R.string.serv
er_client_id))
.requestEmail()
.build();
mGoogleSignInClient =
GoogleSignIn.getClient(this, gso);
Intent signInIntent =
mGoogleSignInClient.getSignInIntent();
startActivityResult(signInIntent,
RC_SIGN_IN);
```

Листинг 1 - Позив за *Google* аутентификацију

На слици 3 приказан је екран помоћу ког корисници врше аутентификацију на мобилну апликацију путем свог *Google* налога.

У оквиру овог рада коришћено је *express checkout* дугме, које омогућава брзу и једноставну куповину. Кликом на *express checkout* дугме се отвара *PayPal* веб страница, на коју корисник треба да се улогује, а потом обави плаћање.



Слика 3 - Приказ екрана за логовање

Листинг 2 показује како се обавља позив за преусмеравање везе ка *PayPal*-у. *mBraintreeFragment* представља инстанцу *BraintreeFragment* и за њено креирање је неопходан токен, који продавци морају претходно креирати.

```
PayPalRequest request = new
PayPalRequest(fromRSDtoUSD (RSDamount))
.displayName(sSharedPreferences.getString("paypal_display_name", null))
.currencyCode("USD")
.intent(PayPalRequest.INTENT_AUTHORIZE)
PayPal.requestOneTimePayment(mBraintreeFragment, request);
```

Листинг 2 - Позив ка *PayPal*-у

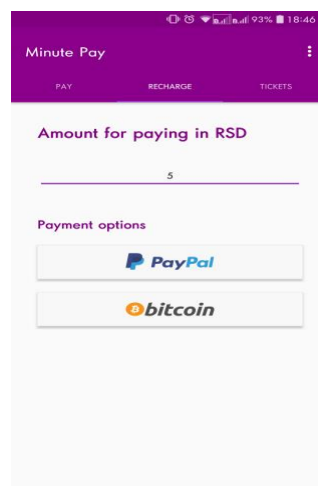
Листинг 3 приказује како се остварује позив ка *Bitcoin* новчанику корисника система.

Варијабла *bitcoinReceiveAddress* представља адресу новчаника власника апликације, на чији рачун достижу све уплате. Да би се успоставила веза између овог система и дигиталног новчаника, корисник треба да поседује инсталирани дигитални новчаник на свом уређају.

```
String walletAddress =
"2NFDMpbQtj1XCTc6GEJCKNs6yd6HcLoDdse";
String bitcoinSchema = "bitcoin:";
String bitcoinReceiveAddress =
walletAddress; //your address from wallet
String bitcoinAmount = "?amount=" +
amount+"&label=Minute-
Pay&message=Balance%20Recharge";
//Amount in BTC, 1$
String bitcoinURI = bitcoinSchema +
bitcoinReceiveAddress + bitcoinAmount;
Intent bitcoinIntent = new
Intent(Intent.ACTION_VIEW);
bitcoinIntent.setData(Uri.parse(bitcoinURI));
try {
startActivityForResult(bitcoinIntent, 1);
} catch (ActivityNotFoundException e) {
Toast.makeText(getActivity().getApplicationContext(), "You don't have any Bitcoin Wallet installed.", Toast.LENGTH_LONG).show();}
```

Листинг 3 - Позив ка *Bitcoin* новчанику

На слици 4 приказан је екран за вршење трансакције новчаних средстава са једног од понуђених налога, на апликацију.



Слика 4 - Приказ екрана за плаћање

Листинг 4 показује директиву која генерише код на основу прослеђеног објекта. Ова директива је део *Angular* кода, којим је изграђена веб апликација.

```
<ngx-qrcode [qrc-element-
type]="elementType" [qrc-value] =
"value" qrc-errorCorrectionLevel =
"L"></ngx-qrcode>
```

Листинг 4 - Генератор QR кода

Да бисмо били у могућности да користимо ову директиву, неопходно ју је инсталирати помоћу следеће конзолне команде:

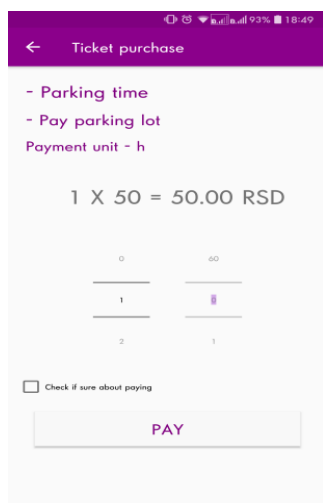
```
$ npm install ngx-qrcode3 -save
```

Систем детектује баркод преко камере, а листинг 5 показује како се детектор креира, и како исти поставља ослушкивач након извршене детекције, и прослеђује очитане информације ако није дошло до неке грешке.

```
barcodeDetector.setProcessor(new
MultiProcessor.Builder<>(new
MultiProcessor.Factory<Barcode>()) {
@Override
public Tracker<Barcode> create(final
Barcode barcode) {
getActivity().runOnUiThread(new
Runnable() {
@Override
public void run() {
try {
mCallback.onScanComplete(barcode.displayValue);
} catch (NoSuchAlgorithmException e) {
e.printStackTrace();
} catch (InvalidKeyException e) {
e.printStackTrace();
} catch (SignatureException e) {
e.printStackTrace();
}
}
});
return new Tracker<>();
}).build();}
```

Листинг 5 - Креирање детектора баркода

На слици 5 приказан је екран једне верзије очитаног QR кода. Кориснику је омогућено да изабере жељено време у сатима, а потом плати своју карту.



Слика 5 - Приказ очитане *prepaid* карте

5. ЗАКЉУЧАК

Електронско плаћање из године у годину постаје све распрострањеније средство новчане интеракције како на локалном тако и на глобалном нивоу.

У оквиру рада је описан модел система кроз дијаграм тока података. У делу имплементације система издвојене су неке од коришћених технологија и приказани занимљиви делови кода, као и неколико *screenshot*-ова саме апликације

Овај систем у тренутном стадијуму свога развоја располаже неким основним функционалностима, али и као такав даје могућност даљем развоју и неким иновативним надоградњама и проширењима.

Неки од примера проширења би могли бити: опције за логовање на систем, оне би се могле проширити на логовање нпр. преко апликације *Facebook*, путем имејла и сл. Затим, могућности плаћања би се могле проширити са још неким платним системима као што су *Amazon*, *Google pay*, директно плаћање кредитним картицама и сл.

Што се тиче надоградње система за плаћање утрошеног времена, једна од идеја јесте креирање уређаја које би поседовали продавци и који би омогућавали завршавање куповине *postpaid* карата без интеракције корисника.

Пример тога би био: уколико се овај систем користи за наплату карта у градском превозу, претходно споменути уређаји би могли бити постављени на излазе из превоза, а при изласку путника, он би својим сигнаlima позвао завршетак плаћања одговарајуће *postpaid* карте.

5. ЛИТЕРАТУРА

- [1] - <https://www.researchgate.net/publication/303329794>
- [Adoption of e-Payment Systems A Review of Literature](https://www.researchgate.net/publication/303329794)
- [2] - <https://www.ft.com/content/91b5efe2-70bb-11e5-9b9e-690fdae72044>
- [3] - <https://europa.rs/images/publikacije/elektronska-i-mobilna-placanja-u-svetu-i-u-srbiji.pdf>
- [4] - <https://angular.io/guide/architecture>
- [5] - <https://research.ijcaonline.org/rtfem2016/number1/rtfem45109.pdf>
- [6] - <https://firebase.google.com/>
- [7] - <https://about.gitlab.com/what-is-gitlab/>
- [8] - <https://www.qr-code-generator.com/qr-code-marketing/qr-codes-basics/>
- [9] - <https://www.paypal.com/us/webapps/mpp/about>
- [10] - <https://blockgeeks.com/guides/what-is-bitcoin/>

Кратка биографија:

Виолета Новаковић рођена је 23.5.1993. године у Сомбору. Основне академске студије је завршила 2016. године, смер рачунарство и аутоматика на ФТН-у. Мастер академске студије је уписала исте године, такође смер рачунарство и аутоматика. Испунила је све обавезе и положила је све испите предвиђене студијским програмом.

INDUSTRIJSKI MODUL ZA MIKROKONTROLERE I SBC RAČUNARE INDUSTRIAL MODULE FOR MICROCONTROLLERS AND SBC COMPUTERS

Goran Tanasić, Vladimir Rajs, *Fakultet Tehničkih Nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu predstavljen je razvoj PCB modula koji se koristi kao industrijski interfejs za mikrokontrolere i SBC računare kao što je „Raspberry Pi“, „Orange Pi“ itd. Modul sadrži 7 relejnih izlaza, 8 digitalnih ulaza, RS232 i RS485 interfejs, mikroBUS podršku, RTC/EEPROM modul i 24-bitni AD konvertor. Napajanje ploče, digitalni ulazi i izlazi su projektovani tako da rade na standardnom industrijskom naponu od 24V. Takođe, svi sklopovi osim mikroBUS-a i RTC/EEPROM modula su galvanski odvojeni što doprinosi robusnosti uređaja.

Ključne reči: mikroBUS, Raspberry Pi, industrija.

Abstract – In this paper the development of a PCB module which can be used as an industrial interface for microcontrollers and SBC computers like Raspberry Pi, Orange Pi etc. has been presented. The module consists of 7 relay outputs, 8 digital inputs, RS232 and RS485 interfaces, a mikroBUS expansion connector, an RTC / EEPROM module and a 24-bit AD convertor. The power supply, digital inputs and outputs are designed to work with standard industrial voltage of 24V. Also, all blocks except mikroBUS and RTC / EEPROM module are galvanically isolated which contributes to the module's robustness.

Key words: mikroBUS, Raspberry Pi, industry

1. UVOD

Industrijsko upravljanje je oduvek predstavljalo velik motiv za razvoj industrijske upravljačke elektornike. Ranije su namenski razvijane elektronske jedinice koje su bile vezane isključivo za određeni tip uređaja, tako da je bilo teško iskoristiti jednom razvijeni sistem za upravljanje različitim mašinama i procesima. Sa pojavom mikrokontrolera došlo je do razvoja PLC-ova (*programable logic controller*) koji su neverovatno olakšali industrijsko upravljanje. Ovi kontroleri omogućavaju korisniku brz razvoj upravljačkog softvera, velik izbor periferija, komunikacionih protokola i izvršavanje složenih zadataka. Svaki PLC se sastoji od centralne procesorske jedinice koja je spregnuta sa nekim ulazno-izlaznim modulom.

Često postoji komunikacija sa korisničkim interfejsom koji je zasnovan na ekranu osetljivom na dodir. Postoje i određeni PLC-ovi na tržištu koji u sebi već imaju ugrađen HMI (*human machine interface*). U ovom radu je opisan mikrokontrolerski modul koji može da se koristi za upravljanje manjim industrijskim procesom gde je korisniku omogućen izbor procesorske jedinice.

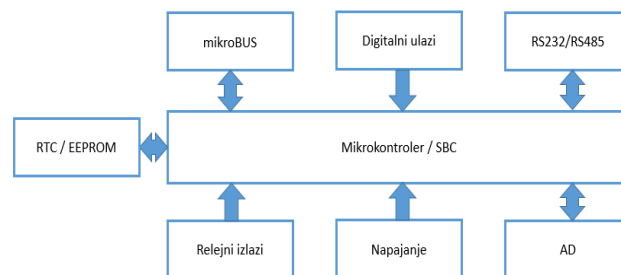
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Rajs, docent.

Samim tim korisnik može da bira programski jezik koji želi da upotrebi za programiranje logike ovog mini upravljačkog sistema.

2. RAZVOJ HARDVERA

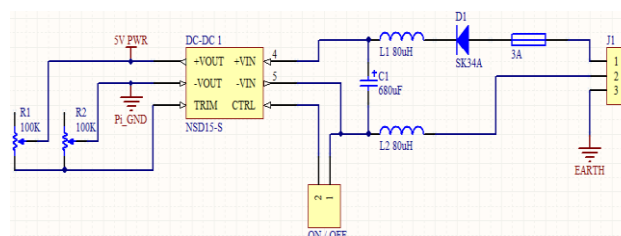
Šematski i PCB dizajn modula je izveden u programskom paketu „Altium designer 2009“. PCB ploča je širine 107 mm što odgovara standardnom industrijskom kućištu za PCB ploče. Ovo omogućava jednostavnu ugradnju modula u industrijske upravljačke ormane. Ploča sadrži *polygon-pour* na oba sloja koji su vezani na EARTH konektor. Blok šema modula data je na slici 1.



Slika 1. Blok šema modula

2.1. Napajanje

Modul se napaja sa 24V što predstavlja industrijski standard. Galvanska izolacija je postignuta upotrebom DC-DC konvertorskog modula. Napojni sklop sadrži izmenjivi topljivi osigurač čija se nominalna jačina struje koristi u zavisnosti od odabranog tipa mikrokontrolera odnosno SBC računara. Takođe je ugrađena zaštita od pogrešnog polariteta napona, LC filter, podešavanje izlaznog napona kao i izvodi za uključivanje / isključivanje modula. Električna šema bloka napajanja data je na slici 2.



Slika 2. Električna šema bloka napajanja

2.2. mikroBUS

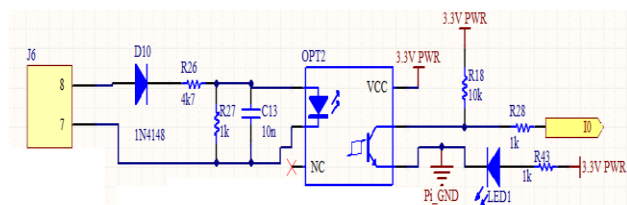
MikroBUS [1] je otvoreni standard koji je razvila kompanija „Mikroelektronika“ iz Srbije. Zasniva sa SPI, UART i I2C protokola. Ovaj standard omogućava dodavanje raznovrsnih dodatnih periferija na modul. Modul ne podržava AN, RST, PWM i INT signale. Modul podržava do dva mikroBUS uređaja. U slučaju korišćenja dva

2.3. Digitalni izlazi

[illegible]

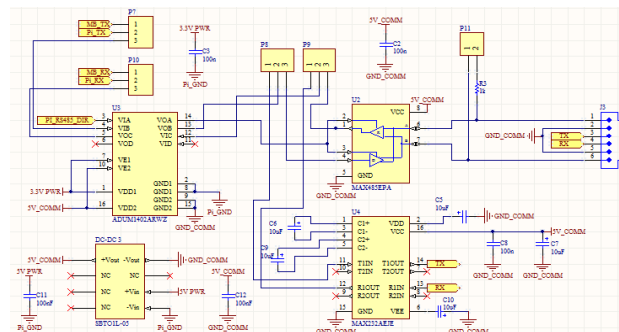
2.4. Digitalni ulazi

Ulazno kolo se zasniva na *RC* filteru i *Schmitt trigger* optokapleru *PC900*. Električna šema jednog digitalnog ulaza data je na slici 5.



2.5. RS485 / RS232

ženski RJ45 konektor sa 6 kontakta. Električna šema RS485 / RS232 bloka data je na slici 6.



2.6. RTC / EEPROM

Wiring diagram for the TinyRtc module:

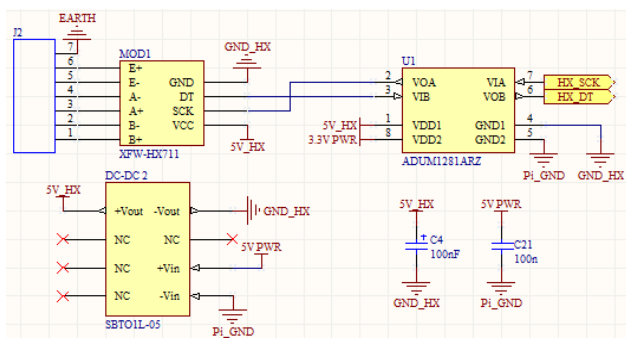
- Left Side (Pi Connections):**
 - Pi_GND
 - 5V PWR
 - Pi I2C SDA
 - Pi I2C SCL
- Right Side (Module Connections):**
 - Bat VCC
 - GND
 - VCC
 - SDA
 - SCL
 - DS
 - DQ
- Module Labels:**
 - RTC1 (top)
 - TinyRtc (bottom)
- Connections:**
 - Pi_GND to GND
 - 5V PWR to VCC
 - Pi I2C SDA to SDA
 - Pi I2C SCL to SCL
 - DQ to DS
- Not Connected (Red X marks):**
 - Pi_GND to Bat VCC
 - 5V PWR to GND
 - Pi I2C SDA to VCC
 - Pi I2C SCL to SDA
 - DS to SCL

A blue PCB I2C module with two I2C modules connected via a ribbon cable. The module features various components including resistors, capacitors, and integrated circuits. Labels on the PCB include 'I2C modules', 'DS', 'SCL', 'SDA', 'GND', 'VCC', 'BAT', 'R1', 'R2', 'R3', 'R4', 'R5', 'R6', 'R7', 'R8', 'R9', 'R10', 'R11', 'R12', 'R13', 'R14', 'R15', 'R16', 'R17', 'R18', 'R19', 'R20', 'R21', 'R22', 'R23', 'R24', 'R25', 'R26', 'R27', 'R28', 'R29', 'R30', 'R31', 'R32', 'R33', 'R34', 'R35', 'R36', 'R37', 'R38', 'R39', 'R40', 'R41', 'R42', 'R43', 'R44', 'R45', 'R46', 'R47', 'R48', 'R49', 'R50', 'R51', 'R52', 'R53', 'R54', 'R55', 'R56', 'R57', 'R58', 'R59', 'R60', 'R61', 'R62', 'R63', 'R64', 'R65', 'R66', 'R67', 'R68', 'R69', 'R70', 'R71', 'R72', 'R73', 'R74', 'R75', 'R76', 'R77', 'R78', 'R79', 'R80', 'R81', 'R82', 'R83', 'R84', 'R85', 'R86', 'R87', 'R88', 'R89', 'R90', 'R91', 'R92', 'R93', 'R94', 'R95', 'R96', 'R97', 'R98', 'R99', 'R100'.

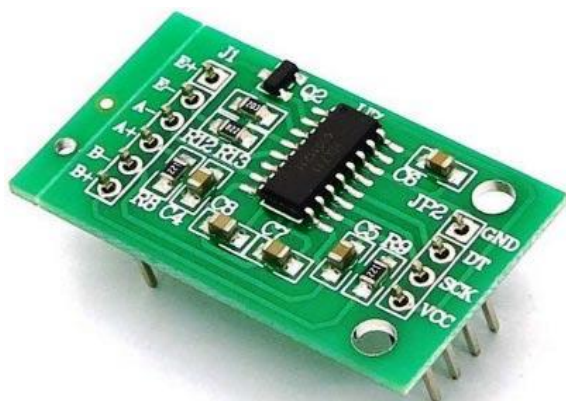
2.7. AD

1077

DC-DC konvertora SBT01L-05 proizvođača Meanwell. Električna šema AD bloka data je na slici 9, dok je izgled modula prikazan na slici 10.



Slika 9. Električna šema AD bloka



Slika 10. Izgled AD modula

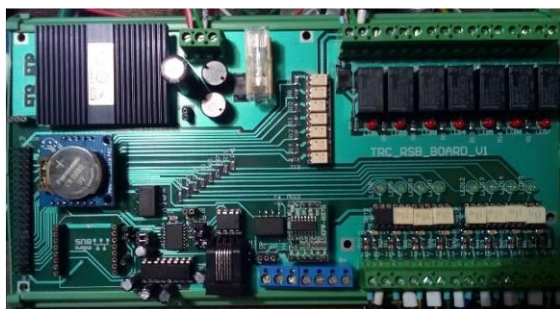
2.8. Štampana ploča

Štampana ploča je izrađena u dvoslojnoj tehnologiji. Ploča pored navedenih blokova, sadrži kontakte za testiranje kritičnih tačaka kao što su naponi DC-DC konvertora i signali komunikacionih linija.

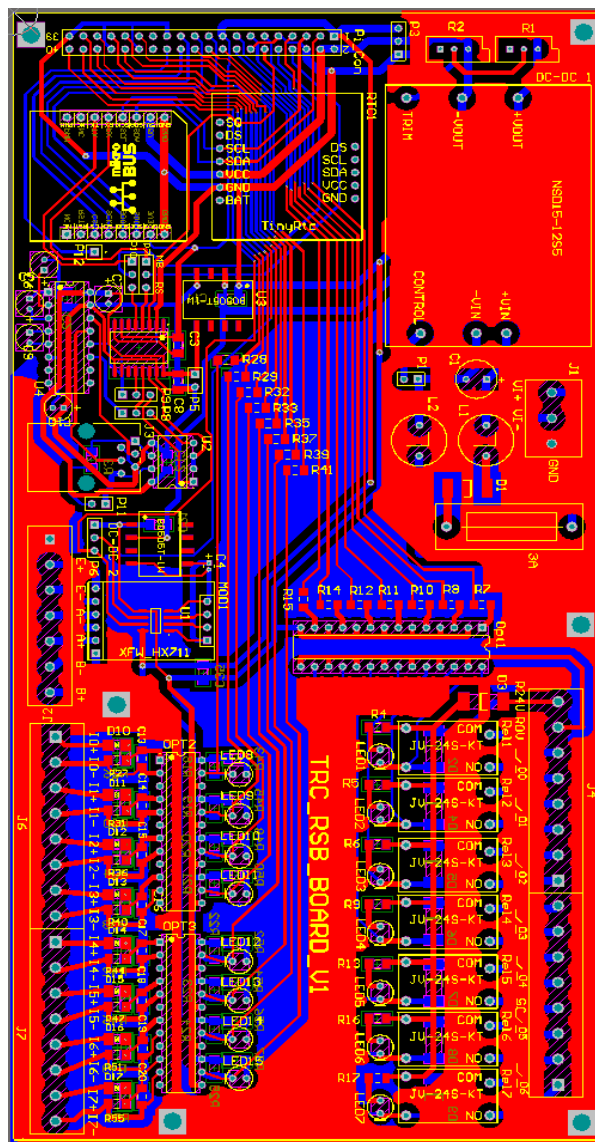
Svi konektori koji se koriste za povezivanje modula sa perifejama su industrijskog tipa.

Takođe su dodati bušeni otvori za ugradnju odstopnika u slučaju drugačijeg načina montiranja od industrijskog PCB kućišta za koji je ploča projektovana. Integrisana kola koja mogu zakazati (HX711, PC900, MAX232, MAX485, TLP627) u slučaju pogrešnog povezivanja sa okruženjem ili usled velikih EM smetnji, montiraju se u PDIP podnožja.

Na ovaj način je omogućeno brzo servisiranje modula u slučaju kvara jer u tom slučaju nema potrebe za prelemljavanjem. Izgled izrađene ploče dat je na slici 11, dok je izgled dizajna štampane ploče dat na slici 12.



Slika 11. Izgled izrađene štampane ploče



Slika 12. Dizajn štampane ploče

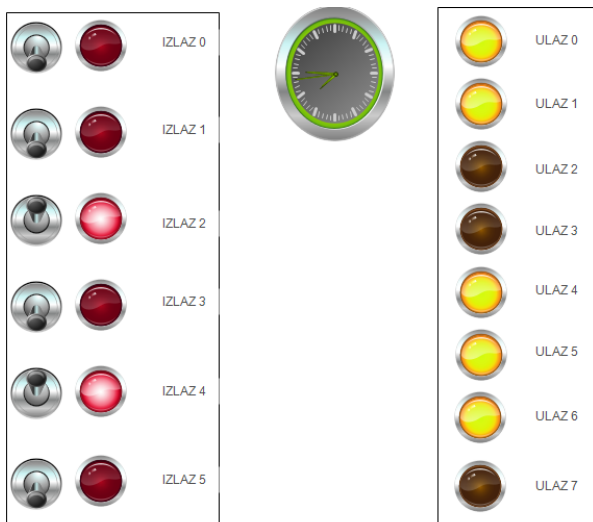
3. RAZVOJ SOFTVERA

Programiranje logike ovog modula zavisi od odabranog tipa mikrokontrolera/SBC računara. Korisniku je omogućen širok izbor programskih jezika u slučaju korišćenja Raspberry Pi SBC računara kao logičke jedinice. Najčešće korišćeni programski jezici za programiranje ovog SBC računara su Python, C, C++, Java, Codesys (podržava ladder logiku i vizuelizaciju) itd.

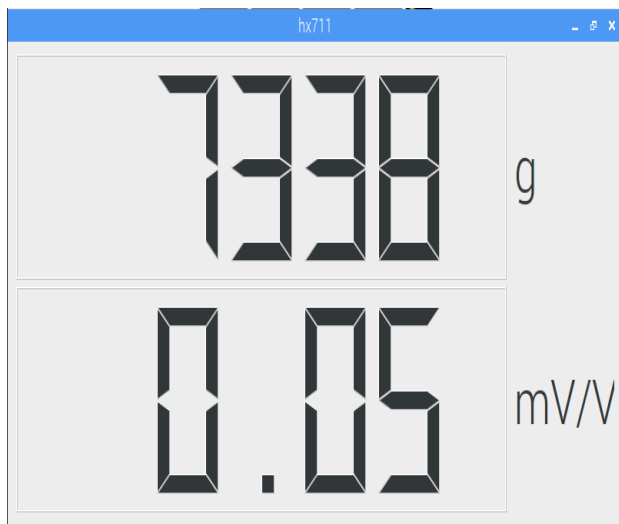
U slučaju korišćenja nekog od mikrokontrolera, najčešće korišćeni programski jezik je C, koji se uglavnom koristi sa razvojnim okruženjem (IDE-om) proizvođača mikrokontrolera. Testiranje ovog modula je izvršeno upotrebom Raspberry pi SBC računara.

Za različite funkcionalnosti modula razvijene su aplikacije koje su napisane u različitim programskim jezicima kako bi se demonstrirale mogućnosti razvoja upravljačke logike. Aplikacija razvijena u Codesys programskom paketu se koristi za demonstraciju funkcionalnosti digitalnih ulaza, izlaza i RTC modula, dok se aplikacija razvijena upotrebom Python-a i QT5 programskog paketa koristi za demonstraciju rada AD modula sa mernom ćelijom.

Obe aplikacije su razvijene sa vizualizacijom prilagođenom za 7" ekran osjetljiv na dodir proizvođača *Adafruit*. Vizualizacije obe aplikacije date su na slikama 13 i 14.



Slika 13. Codesys aplikacija



Slika 14. Python-QT5 aplikacija

4. PRIMENA

Praktična primena modula je prvenstveno industrijska, a može biti i edukativna. U industrijskog primeni, ovaj modul je idealan za manje dozirne sisteme jer podržava korišćenje mernih ćelija. Zahvaljujući RS485 i RS232 interfejsima omogućena je sprega sa raznim industrijskim uređajima koja podržavaju ovaj način komunikacije. U slučaju korišćenja *Raspberry pi SBC* računara kao logičke jedinice, moguće je implementirati većinu industrijskih protokola koji su zasnovani na *ethernet* interfejsu. Pored toga moguće je dodati i ekran osjetljiv na dodir za prikaz relevantnih informacija procesa i unos parametara od strane operatera.

Ovo su funkcionalnosti koje moderni PLC-ovi trenutno podržavaju. U edukativnoj primeni ovaj modul može da predstavlja razvojni system za obuku studenata, đaka, hobista itd. Podržava većinu interfejsa koji standardni mikrokontroleri podržavaju (*I2C*, *UART*, *SPI*, *bit bang*,

digitalne ulaze i izlaze) kao i širok izbor samih mikrokontrolera. Upotreba različitih logičkih jedinica podrazumeva izradu adaptera za spregu sa modulom. Korisnici bi upotrebom ovog modula mogli relativno brzo i efikasno da upravljaju različitim aktuatorima i obrađuju signale sa raznih senzora koji se koriste u industriji.

5. ZAKLJUČAK

Trend automatskog upravljanja sve više i više raste što podstiče razvoj malih upravljačkih sistema koji su modularni i jednostavni za upotrebu. U ovom radu je predstavljena izrada jednog takvog sistema, opisane su njegove mogućnosti, raznovrsnost i primena. Projektovani modul ispunjava sve funkcionalnosti projektnog zadatka. Testiranja pojedinačnih blokova modula su izvršena upotrebom *Raspberry pi SBC* računara u laboratorijskim uslovima.

Slede testiranja u konkretnoj primeni u industrijskom okruženju. Upotrebom ovog industrijskog modula, korisnici mogu sami da odaberu logičku jedinicu pa samim tim i programski jezik za razvoj logike. Modul u sprezi sa logičkom jedinicom predstavlja uređaj koji može da parira savremenim PLC-ovima. Pristupačna cena ovog modula u poređenju sa cenom nekog od PLC-ova u sličnoj klasi performansi i ulazno izlaznih opcija predstavlja veliku prednost. Dalji razvoj ovog projekta bi podrazumevao objedinjavanje modula sa ugrađenom logičkom jedinicom, ekranom osjetljivim na dodir kao i analognom izlaznom jedinicom.

6. LITERATURA

- [1] Internet sajt, specifikacija *mikroBus-a*
<https://www.mikroe.com/mikrobus/>, septembar 2017
- [2] Internet sajt, specifikacija *UART* protokola
<http://ece.colorado.edu/~ecen2120/Manual/uart/UART.html>, septembar 2017
- [3] Internet sajt, specifikacija i demo aplikacija Tiny RTC modula.
https://www.elecrow.com/wiki/index.php?title=Tiny_RTC, septembar 2017.
- [4] Internet sajt, *datasheet* HX711 modula
https://cdn.sparkfun.com/datasheets/Sensors/ForceFLEX/hx711_english.pdf, septembar 2017.

Kratka biografija:



Goran Tanasić rođen je u Sremskoj Mitrovici 1990. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Primenjena elektronika odbranio je 2014.god.



Vladimir Rajs rođen je 1982. godine u Apatinu. Diplomirao je 2007, a doktorirao 2015. godine na Fakultetu tehničkih nauka u Novom Sadu. Od 2016. godine zaposlen je kao docent na Departmanu za elektroniku, energetiku i telekomunikacije FTN-a. Oblasti interesovanja su mu elektronika i primenjena elektronika

IMPLEMENTACIJA SPOTIFY VEB KLIJENTA U REACT I DJANGO RADNOM OKRUŽENJU

IMPLEMENTATION OF A SPOTIFY WEB CLIENT IN REACT AND DJANGO FRAMEWORK

Žana Bilbija, Fakultet tehničkih nauka, Novi Sad

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – Prezentovan je kratak pregled alata sa Spotify Developer portala sa ciljem boljeg upoznavanja sa istim. Osnove integracije alata sa softverskim rešenjima su predstavljene kroz primer implementacije veb klijenta.

Ključne reči: Spotify, React, Django Rest, OAuth

Abstract – Presents a short overview of tools located on the Spotify Developer portal, with the goal of an introduction to the aforementioned tools. The basics of integrating these tools into future software solutions are presented through an implementation of a web client.

Keywords: Spotify, React, Django Rest, OAuth

1. UVOD

Kako se razvija digitalni svet i kako se sve više širi domen interneta, povećava se broj servisa pristupačnih na internetu. Prateći povećanje broja servisa, takođe se povećava i obogaćuju tipovi usluga koji dotični servisi nude korisnicima. Jedna od popularnih usluga jeste mogućnost reprodukcije muzike.

Postoji značajan broj sajtova, veb servisa i veb aplikacija koje pružaju upravo tu uslugu, ali ovaj rad se koncentriše na jednu od njih – **Spotify**. Osim usluge reprodukcije muzičkih fajlova, ovaj veb servis nudi i mogućnosti rukovanje putem izdvajanja muzičkih fajlova u svoje arhive, ili pak grupisanja muzičkih fajlova u playliste ili pravljenja imitacije radio stanice, gde sam veb servis dopunjuje playlistu sa sličnim pesmama [1].

Količina usluga za rukovanje muzičkim fajlovima, velika pristupačnost veb servisa kao i postojanje alata za integraciju Spotify-a u druga softverska rešenja su razlozi radi čega je ovaj veb servis izabran kao tema ovog rada

2. REACTJS BIBLIOTEKA

React jeste javno dostupna Javascript biblioteka namenjena implementaciji korisničkih interfejsa [2]. Razvijena je od strane kompanije Facebook [3]. Inicijalno je React razvijen kao alternativa XHP-u, radno okruženje za PHP. zasnovano na komponentama [4], prvenstveno namenjeno smanjivanju broja XSS napada [3, 5].

NAPOMENA:

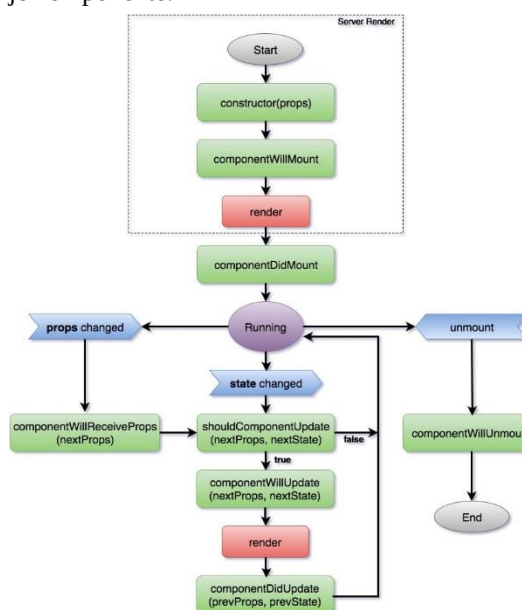
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinać, vanr. prof.

Osnovna strukturna jedinica u React-u jeste komponenta [6]. Komponente omogućuju da se korisnički interfejs segmentiše u nezavisne, relativno nezavisne delove. Kao takve, otvara se prilika da se već implementirane komponente ponovo iskoriste, te se izbegava dupliranje koda.

Komponente je moguće definisati u vidu funkcija ili kao klase koje nasleđuju *React.Component* klasu, i svakoj komponenti je moguće proslediti podatke u vidu *props* objekta. Bitna napomena kod *props* jeste da su oni objekti iz kojih komponenta samo može da čita vrednosti. Znači komponenta ne sme da menja sopstvene *props*-ove. Dakle može se posmatrati da su React komponente jedan vid tzv. čistih funkcija u odnosu na svoje *props*.

Da bi se znalo kada treba koju komponentu ponovo prikazati, mora se znati kada je komponenta bila izmenjena u odnosu na njeno ranije stanje. Stoga se uvodi stanje u komponentu. Stanje predstavlja objekat koji sadrži sve podatke o kojima komponenta mora da vodi računa.

Tek kada se stanje u komponenti izmeni, onda React ponovo prikazuje tu komponentu tako što samo njen deo unutar DOM stabla osvežava sa novim podacima koji su zabeleženi u njenom objektu stanja. Takođe metode životnog ciklusa komponente u kojima se može ažurirati stanje komponente.



Slika 1. Životni ciklus React komponente i metode koje se pozivaju [7]

JSX poznat i kao *JavaScript eXtension* jeste dodatak na sintaksu Javascripta. Sam oblik JSX-a podseca na XML sintaksu, i nudi alternativni, koncizniji način pisanja prikaza komponenti u kontrastu čistog Javascript pristupa. Osim kao alternativa strukturiranju komponenti unutar *render* funkcije, JSX omogućava i umetanje promenljivih u sam HTML kod. Pored promenljivih, JSX prihvata i pozive funkcija unutar vitičaste zgrade.

3. DJANGO REST RADNO OKRUŽENJE

Django REST je implementiran da podrži MVC dizajn šablon. Model može biti jednostavan kao već iskonfigurisana *SQLite* fajl baza podataka, ili konkretna relaciona baza podataka ili pak *NoSQL* baza podataka. Kontroler prati koncepte REST komunikacije, dok *View* varira od osnovne HTML stranice do eksternog veb klijenta poput React i sl. Da bi se ostvarila komunikacija sa Django REST-om, neophodno je definisati URL putanje koje će osluškivati unutar *urls.py* fajlu, kao i definisati koji kontroleri (preciznije, same metode u kontrolerima) će odreagovati kada se pošalje zahtev na zadatu adresu.

U definiciji samih funkcija se ne navode putanje koje osluškiju, ali se mogu navesti tipovi zahteva koji funkcije očekuju, putem anotacija. Serijalizacija i deserijalizacija podataka se vrši pravljenjem posebnih klasa za modele koje je potrebno pripremiti za slanje, te i za svaki atribut unutar klase je potrebno naznačiti kako se pretvaraju podaci. Moguće je precizirati da li je u pitanju serijalizacija jednog objekta ili niza objekata putem atributa pri pravljenju objekta klase serijalizatora.

4. SPOTIFY WEB API

Određeni podaci koje WEB API vraća se mogu tretirati kao podaci lične prirode, te je neophodno registrovati se na Spotify Developer sajtu da bi se održala bezbednost dotičnih podataka. Prvi korak u procesu registracije jeste da korisnik ili napravi Spotify nalog ili se uloguje u isti.[8] Dalje nakon prijave preko Spotify naloga, neophodno je registrovati konkretno softversko rešenje. Rezultat registracije jeste da Spotify zna koje konkretno softversko rešenje pristupa WEB API-ju, putem jedinstvenog identifikatora. Ovim se ograničava pristup i predstavlja dodatnu meru bezbednosti za korisničke podatke.

Osim registracije softverskog rešenja na sajtu, neophodno je i dobijene kredencijale podesiti u sklopu softverskog rešenja. Potrebno je navesti dobijeni identifikator, tajni ključ koji se generiše po registraciji kao i URL na koji će se slati token za autorizaciju zahteva [8].

4.1. Tipovi autorizacije

Kad je u pitanju autorizacija softverskog rešenja, postoje dva načina. Prvi podrazumeva da se rešenje autorizuje pristup Spotify platformu što podrazumeva pristup API-ju i drugim resursima za razvoj i integraciju. Drugi je pak autorizacija putem korisničkih kredencijala, gde se korisnički kredencijali šalju pri slanju zahteva ka API-ju, i zauzvrat se dobija token na ograničen vremenski period [9]. Svaki vraćeni token ima određen domen pristupa, a postoje i krajnje putanje API-ja koje ne zahtevaju token za pristup.

Postoje tri toka zahtevanja autorizacije [9]:

- Korisnička autorizacija koja se osvežava - podrazumeva da krajnji korisnik dopusti softverskom rešenju da pristupa podacima, a kao rezultat se dobija token za pristup koji može da se osveži. Razmena obuhvata i slanje tajnog ključa, te se mora voditi računa o tipu komunikacije.
- Privremena autorizacija korisnika - namenjen klijentskim aplikacijama implementiranim u JavaScript-u i pokrenutim u korisnikovom veb pretraživaču. Ovaj pristup ne zahteva slanje tajnoj ključa, nego se korisnik usmerava ka posebnom servisu za korisničke naloge- Kao rezultat prijave na taj servis dobija se token za pristup. On je kratkog životnog veka, i nemoguć osvežiti.
- Aplikaciona autorizacija koja se osvežava - ovaj pristup se koristi kada se vrši autentifikacija između dva servera, i omogućava pristup krajnjim putanjama koji je rukovode korisničkim podacima. Token se dobija prijavom slanjem identifikator softverskog rešenja i tajni ključ.

```
//Authorization Code Flow
{
  "access_token": "NgCXRK...MzYjw",
  "token_type": "Bearer",
  "scope": "user-read-private user-read-email",
  "expires_in": 3600,
  "refresh_token": "NgAagA...Um_SHo"
}

//Client Credentials Flow
{
  "access_token": "NgCXRKc...MzYjw",
  "token_type": "bearer",
  "expires_in": 3600,
}
```

Slika 2 – Primer tokena za pristup za Korisničku autorizaciju i Aplikacionu autorizaciju

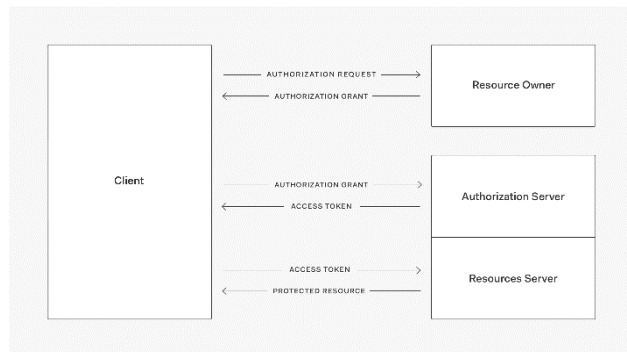
Postoji nekoliko domena koju su vezani za konkretan token pristupa, formirani na osnovu tipa podataka kojima rukovode. Svaka krajnja putanja API-ja spada u neki domen pristupa, a pojedini mogu pripadati nekoliko domena. Neki od domena su:

- Čitanje pesme koja se reprodukuje
- Ažuriranje liste sačuvanih pesama
- Ažuriranje stanja drugih Spotify klijenata
- Rukovanje drugim Spotify klijentima [10]

4.2. OAuth

OAuth je specifikacija koja definiše protokol za delegaciju, i korisno za prikazivanje odluke o autorizaciji u mreži veb-orijentisanih aplikacija, servisa i API-ja. Najčešće je korišćen pri autentifikaciji samih korisnika, ali sam OAuth nije protokol za autentifikaciju. Zapravo OAuth se koristi unutar protokola za autentifikaciju, i njene komponente se pozivaju iz konkretnog protokola za autentifikaciju. Konkretno OAuth se razvio kao način da korisnici interneta dozvole sajtovima da pristupe njihovim

informacijama na drugim sajtovima bez da im korisnici daju svoje kredencijale. Aktuelna verzija OAuth protokola je OAuth 2.0 koji definiše tok autorizacije za veb aplikacije, desktop aplikacije, mobilne telefone i pametne uređaje.



Slika 3 – Tok OAuth autorizacije

4.3. Analiza podataka

Podaci koje vraćaju krajnje putanje Spotify WEB API-ja su organizovani u strukturu koja prati definisani objektni model [11]. Dotični objekti se vraćaju u obliku JSON objekata. Putanje su grupisane po modelu objekta koji vraćaju, a pojedine vraćaju pojednostavljene verzije objekata. Pojednostavljene verzije objekata ne poseduju sve attribute, nego samo one neophodne za identifikaciju, one koje nose osnovne informacije i link ka putanji gde se nalaze potpuni podaci o objektu. Ovim putem se smanjuje veličina odgovora koji se vraća tako što se izbegava ugnježdavanje velikih objekata unutar podjednako ili pak većih objekata.

Ukoliko se vraća kolekcija objekata, i gde broj konkretnih objekata varira, kolekcija objekata se dodatno smešta unutar objekta podeljenog na stranice. Time se olakšava dobavljanje ostalog niza podataka u kolekciji, i smanjuje veličina odgovora, te se ne zatrpava komunikacija. Ovo je naročito važno kod softverskih rešenja sa tokenima za pristup kratkog životnog ciklusa. Objekti podeljeni na stranice imaju attribute koji sadrže URL putanje do krajnjih putanja gde se nalaze potpuni oblici traženog tipa objekta.

5. IMPLEMENTACIJA

Softversko rešenje koje je tema ovog rada se sastoji iz dva dela. Prvi deo jeste server koji je napisan u Django REST radnom okruženju i komunicira kako sa veb klijentom, tako i sa API-jem posredstvom biblioteke *Spotipy*. Drugi deo je sam veb klijent implementiran pomoću *ReactJS* biblioteke kao i pojedinih dodatnih biblioteka za izgled samog veb klijenta i dodatnu biblioteku za razrešenje putanja u klijentu.

5.1. Spotipy biblioteka

Spotipy je biblioteka napisana u Python programskom jeziku i koja se može integrisati u softversko rešenje bez dodatne konfiguracije [12]. Koristeći *Spotipy* biblioteku, nudi se mogućnost pristupa svim krajnjim putanjama Spotify WEB API-ja. Takođe nudi funkcionalnosti putem kojih se izvršava autorizacija softverskog rešenja u koje je integrisana. Da bi se uopšte započela komunikacije, neophodno je instancirati *Spotify* objekat, definisan unutar samog *spotipy* modula. Svi tipovi zahteva se prosleđuju

posredstvom *Spotify* objekta, pozivajući adekvatne metode. Metode korespondiraju tipu zahteva. Biblioteka podržava dva vida autorizacije: Korisnička autorizacija koja se osvežava i Aplikaciona autorizacija koja se osvežava. Da bi se koristila Aplikaciona autorizacija, potrebno je koristiti klasu *SpotifyClientCredentials* iz modula *oauth2*, i objekat te klase se navodi kao parametar pri instanciranju objekta *Spotify* klase.

Ukoliko se pak ide putem Korisničke autorizacije (kao što je slučaj u ovom radu) onda je neophodno navesti klijentski identifikator, tajni ključ i URL za preusmeravanje.

Za slanje zahteva za vid Aplikacione autorizacije poziva se posebna metoda *prompt_for_user_token* koja pored nabrojanih parametara očekuje i parametar za domen za koji se zahteva autorizacija. Generisani token se prosleđuje na URL koji je naveden za preusmeravanje i kopira u terminal da bi biblioteka mogla učitati i privremeno arhivirati.

Klasa *SpotifyOAuth* iz modula *oauth2* vodi računa o isteku tokena, i ima sačuvanu putanju do Spotify-jevog servisa za autorizaciju. Potom se dobijeni token navodi pri instanciranju objekta klase *Spotify*.

```

import spotipy
import spotipy.util as util

scope = 'user-library-read'
util.prompt_for_user_token(username, scope, client_id='clientID',
    client_secret='clientSecret', redirect_uri='app-redirect-url')
if token:
    sp = spotipy.Spotify(auth=token)
    results = sp.current_user_saved_tracks()
    return results

```

Slika 4 – Poziv funkcije za slanje zahteva za autorizaciju

5.2. Arhitektura rešenja

Kao što je bilo spomenuto softversko rešenje se sastoji iz servera implementiranog u Django REST radnom okruženju i veb klijenta pisanog u ReactJS biblioteci. Arhitektura servera je grupisana vertikalno po tipu korisničkog zahteva koji ispunjavaju, tj. po domenu funkcionalnosti koje obuhvata. Veb klijent je organizovan na osnovu komponenti od koji se sačinjava, te su i one dalje grupisane na osnovu toga koja komponenta je enkapsulirana drugom. Kako se svi podaci dobivljaju iz Spotify-jevom API-ja, ne postoji konkretna baza podataka za skladištenje podataka, izuzev generisanja Microsoft Word dokumenta sa spiskom korisnikovih sačuvanih pesama.

Naravno ovo ne negira potrebu za sopstvenim modelima, štaviše veličina pojedinih podataka te i međusobna povezanost istih nameće obradu podataka na modele koji sažeto enkapsuliraju potrebne podatke shodno korisničkim zahtevima.

Parsiranje JSON odgovora sa API-ja u odgovarajuće klase modela se vrše u servisima. U servisima su takođe raspoređeni pozivi ka API-ju, te i provera da li je već dobijeni token za pristup istekao ili ne, tako što se porede vreme kada je token dobijen i trenutno vreme. Maksimalni životni ciklus tokena je jedan sat.

Konekcionni podaci potrebni za API su skladišteni u JSON fajlu. Ovim se drže podaci na jednom mestu, te je pogodnije za buduće izmene u konfiguraciji.

Pri instanciranju servisa na serveru, učitavaju i dodeljuju se podaci neophodni za autorizaciju zahteva ka API-ju. funkcionalnost arhiviranja spiska svih korisnikovih sačuvanih pesama u Microsoft Word dokument je implementirana pomoću modula za rukovanje dokumentima.

Obuhvata pravljenje novog dokumenta, podešavanje putanje za isti, dodavanje podataka u sam dokument koristeći metode za generisanje paragrafa i čuvanje istog na zadatoj putanji.

Pri pokretanju veb klijenta, neophodno je da se korisnik uloguje sa svojim Spotify nalogom. Uneseno korisničko ime se dalje prosleđuje server i koristi pri autorizaciji zahteva ka API-ju, sve do momenta kada se korisnik odjavi sa veb klijenta. Posle uspešne prijave korisnik ima opciju da izlista sve svoje sačuvane pesme, da izlista nekoliko poslednjih pesama koje su se puštale, izlista sve izvođače ili pak njihove albume. U donjem delu svakog od ovih prikaza se nalazi plejer, sa opcijama za kontrolu puštanja/pauziranja pesme.

Za slanje zahteva ka serveru se koristi funkcija *fetch()*. Potrebno je samo proslediti URL, tip zahteva, zaglavlja i podatke u JSON formatu URL putanje u veb klijentu su iskonfigurisane pomoću *React Router* dodatka [13]. Ova komponenta kao atribut prihvata URL putanju u obliku teksta, koja može da ima definisane parametre u sebi.

Da se naznači eksplicitno koja komponenta se prikazuje kada se preusmeri na određeni URL, u atribut *component* u *Route* komponenti se navodi naziv željene komponente

6. ZAKLJUČAK

U ovom radu opisana je implementacija softverskog rešenja veb klijenta za *Spotify*. Veb klijent je implementiran u vidu servera koji komunicira sa eksternim servisom, konkretno Spotify-jevim WEB API-jem, i u vidu veb klijenta sa kojim korisnik direktno stupa u interakciju u svom veb pretraživaču. U komunikaciji se razmenjuju JSON fajlovi.

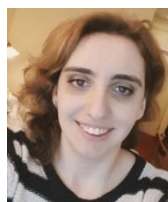
U daljem razvoju bi se dobavljanje tokena za pristup prebacilo da sam server dobavi token umesto trenutno da se ručno mora kopirati u terminal sa URL-a na koji je preusmereno. Takođe bi se razmatrala mogućnost prelaska isključivo na veb klijenta u *ReactJS* biblioteci kako se već razvija dodatna biblioteka koja komunicira sa WEB API-je.

Ovo bi ubrzalo performanse jer svakako je brže komunicirati sa API-jem direktno nego preko servera kao posrednika. I na kraju ostaje mogućnost nabavke plaćenog naloga, čime se omogućuje pristup krajnjim putanjama za kontrolu drugih *Spotify* uređaja putem API-ja.

7. LITERATURA

- [1] Spotify, <https://www.spotify.com/int/about-us/contact/>
- [2] React – A Javascript library for building user interfaces, <https://reactjs.org/>
- [3] JavaScript's History and How it Led To ReactJS, <https://thenewstack.io/javascripts-history-and-how-it-led-to-reactjs/>
- [4] XHP, <https://github.com/hhvm/xhp-lib>
- [5] XSS, OWASP, [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [6] React Components and Props, <https://reactjs.org/docs/components-and-props.html>
- [7] React Component Lifecycle Diagram, <https://levelup.gitconnected.com/componentlifecycle-explanation-393dcb19e459>
- [8] Spotify WEB API Quick Guide, <https://developer.spotify.com/documentation/web-api/quick-start/>
- [9] Spotify WEB API Authorization Flows and Guide, <https://developer.spotify.com/documentation/general/guides/authorization-guide/#authorizaton-code-flow>
- [10] Spotify WEB API Scopes, <https://developer.spotify.com/documentation/general/guides/scopes/>
- [11] Spotify WEB API Object Model, <https://developer.spotify.com/documentation/web-api/reference/object-model/>
- [12] Spotipy Documentation, <https://spotipy.readthedocs.io/en/latest/>
- [13] React Router Documentation, <https://reacttraining.com/react-router/web/guides/philosophy>

Kratka biografija:



Žana Bilbija rođena je 26.08. 1994. godine u Novom Sadu, Republike Srbije. Završila je Gimnaziju Svetožar Marković 2013. godine, i iste godine je upisala Fakultet Tehničkih Nauka, isto u Novom Sadu, na odsek Računarstvo i automatika. Diplomirala je 2017. godine, završivši usmerenje Primenjene računarske nauke i informatika, i sa radom „Implementacija kontrole pristupa zasnovane na korisničkim ulogama u sistemu platnog prometa.“ Posle toga je upisala master akademske studije na programu Računarstvo i automatika, takođe na Fakultetu Tehničkih Nauka u Novom Sadu. Ispunila je sve obaveze i položila je sve ispite predviđene studijskim programom.

ANALIZA SOLID PRINCIPA U PROGRAMSKOM JEZIKU JAVASCRIPT**SOLID PRINCIPLES ANALYSIS IN JAVASCRIPT**Milica Kapetina, *Fakultet tehničkih nauka, Novi Sad***Oblast – Elektrotehničko i računarsko inženjerstvo**

Kratak sadržaj – Pružen je opis SOLID principa i dati su primeri njihove primene u JavaScript-u, uz osvrt i na primenu u TypeScript-u. Opisani su i dizajn šabloni koji se mogu koristiti pri implementaciji SOLID principa.

Ključne reči: SOLID principi, dizajn principi, dizajn šabloni, JavaScript, TypeScript, objektno orijentisani programski jezici

Abstract – A description of SOLID principles with examples on how to apply them in JavaScript, and TypeScript also. Design patterns that help apply SOLID principles are also described.

Keywords: SOLID principles, design principles, design patterns, JavaScript, TypeScript, Object oriented programming languages

1. UVOD

JavaScript je slabo tipiziran jezik i nema koncepte interfejsa i klase, kakvi postoje u objektno orijentisanim programskim jezicima. Zbog toga je primena SOLID [1] principa, koji su pisani za objektno orijentisane programske jezike, u JavaScriptu drugačija od njihove primene u objektno orijentisanim jezicima. Da bi se neki od principa primenili potrebno je imitirati interfejs, ali tako da suština i efekat principa ostanu isti. SOLID principi su principi koji govore o tome kako dizajnirati softver da bude što čitljiviji i lakši za održavanje. Primenom ovih principa se takođe smanjuju greške pri kodiranju i omogućava se mnogo lakše menjanje delova koda u budućnosti kada nastane potreba za izmenama. Ako se nikakvi principi programiranja ne primenjuju, tada je kod vrlo podložan greškama, delovi softvera su međusobno jako zavisni i svaka izmena nekog dela bi zahtevala i izmene u drugim delovima koji inače ne bi trebalo da se menjaju.

2. DIZAJN PRINCIPA I ŠABLONI PROGRAMIRANJA

Dizajn principi i šabloni programiranja [2] pomažu u pisanju kvalitetnijeg koda. To nisu zahtevi koji se moraju poštovati već smernice koje bi bilo dobro, ako mogu, da se primene u pisanju softvera. Ne treba stremiti tome da se primene svi principi koji postoje, već je vrlo važno dobro razumeti sve principe, njihove mane i prednosti,

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusina, vanr. prof.

efekte koje izazivaju i naspram tog znanja i poznavanja problema koji se rešava, odlučiti se za one principe koji zaista najviše odgovaraju i koji će doneti dobre efekte svojom primenom. Često se dešava da, preteranim pokušavanjem da se iskoriste neki principi, se zapravo postigne suprotan efekat, to jest, dobije se prekompleksan kod koji nije čitljiv i vrlo ga je teško održavati jer je teško snalaziti se u njemu. SOLID principi su jedni od najčešće primenjivanih principa, ali osim njih postoje i drugi poznati kao što su KISS (eng. Keep it simple, stupid), YAGNI (eng. You aren't going to need it), DRY (eng. Don't repeat yourself) i tako dalje. Svi ti principi, na svoj način, doprinose čitljivijem kodu, sa manje grešaka i koji je lakši za testiranje.

Akcenat je na pisanju što modularnijih delova koda, koji su međusobno nezavisni, tako da svaki deo koda ima svoju jednu odgovornost. Ako bi se desila greška, mnogo bi se lakše otkrila u modularnom kodu nego u kodu gde su svi delovi međusobno isprepleteni i čvrsto povezani. Takođe, dizajn šabloni nude načine na koje se treba programirati tako da se dobija što modularniji i čitljiviji kod. Dizajn šabloni se dele u tri glavne grupe: konstrukcioni, strukturalni i biheviroistički.

Konstrukcioni šabloni daju razne načine za kreiranje objekata, neki od njih su Konstruktor, Prototip, Fabrika, Singleton...

Strukturalni šabloni govore o kompoziciji objekata i veza između njih, tako da kada se deo sistema promeni, ne mora da se menja ceo sistem. Neki od njih su Dekorator, Façade, Adapter...

Biheviroistički šabloni se koriste za bolje osposobljavanje što bolje i kvalitetnije komunikacije između objekata. To su na primer, Observer, Iterator, Mediator...

3. SOLID PRINCIPCI

SOLID je akronim koji se sastoji iz 5 principa: Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation i Dependency Inversion. Ovih 5 principa se najčešće koriste i imaju veoma velike pozitivne efekte na softver i kod u kome se primene.

3.1. Single Responsibility princip

Princip jednostruke odgovornosti govori o tome da svaka klasa treba da ima samo jednu odgovornost. To ne znači da treba da ima samo jednu funkciju. Ona može imati i više funkcija ako one zajedno izvršavaju jedan zadatak i zajedno čine samo jedan razlog za promenu te klase. Na primer, klasa koja ima funkcije za slanje i prijem poruka i funkcije za otvaranje i zatvaranje konekcije, nema ispoštovan ovaj princip. Princip jednostruke odgovornosti bi se ispoštovao ako bi se razdvojila klasa na dve klase,

jednu koja prima i šalje poruke, i drugu koja otvara i zatvara konekciju. Time bi svaka klasa imala po jednu odgovornost i jedan razlog za promenu.

3.2. Open-Closed princip

Otvoren-zatvoren princip govori o tome da klasa treba biti otvorena za proširenja, a zatvorena za izmene. To znači da ako je potrebno uneti neke nove funkcionalnosti, na primer, onda ne treba menjati potrebnu klasu, već je proširiti. U ovom kontekstu proširenje klase bi podrazumevalo nasleđivanje te klase i implementacija njenih funkcija onako kako nove funkcionalnosti zahtevaju. Ovaj princip se postiže upotrebom apstrakcija i polimorfizmom.

3.3. Liskov Substitution princip

Princip Liskove zamene kaže da sve podklase neke nadklase mogu da se zamene svojom nadklasom a da se pri tome ponašanje programa ne promeni. Na mestu gde se koristi objekat podklase može se staviti da je tip objekta zapravo tip nadklase i da se ponašanje programa ne promeni i ne pokvari. Na taj način sve podklase neke nadklase mogu da se koriste na potpuno isti način, samo što će svaka podklasa imati svoju implementaciju određenih funkcija i njih izvršavati na različite načine. Poštovanjem ovog principa dobija se i ispoštovan Otvoren-zatvoren princip.

3.4 Interface Segregation princip

Princip razdvajanja interfejsa potencira to da se ne kreiraju interfejsi sa puno funkcija, već taman onoliko koliko će ih, klasa koja nasledi taj interfejs, implementirati. U suprotnom klasa koja implementira taj interfejs će imati i funkcije koje nikada neće koristiti. Zbog toga ovaj princip savetuje prvo dobru analizu interfejsa i klase koje će ga nasleđivati, pa tek onda kreiranje samog interfejsa. Time se dobija mnogo čitljiviji kod jer nema viška linija koda koje se nigde neće koristiti.

3.5. Dependency Inversion princip

Princip inverzije zavisnosti kaže da moduli višeg nivoa ne treba da zavise od modula nižeg nivoa, a takođe i apstrakcije ne treba da zavise od detalja, već detalji od apstrakcija. To zapravo znači da treba da se kreiraju apstrakcije ili interfejsi koji predstavljaju ugovor. Apstrakcija samo govori šta neka klasa treba da radi, ne i kako. Klase koje će taj interfejs implementirati treba da znaju šta sve taj interfejs ima od funkcija i na taj način zapravo detalji zavise od apstrakcija, a apstrakcije ne zavise od detalja.

4. OBJEKTI U JAVASCRIPT-U

JavaScript je ujedno i objektno orijentisan programski jezi i funkcionalan programski jezik. Postoji mnogo polemika oko toga kojoj tačno vrsti pripada, međutim on na određene načine, može se reći, pripada obema vrstama. JavaScript je objektno orijentisan jezik jer, iako nema koncept klase, koristi objekte. Za kreiranje objekata u JavaScriptu mogu se koristiti određeni dizajn šabloni kao što su konstruktorski šabloni, singleton, šabloni prototipskog nasleđivanja, šabloni modula i tako dalje.

Singleton šablon predstavlja kreiranje objektnog literala koji ima samo jednu instancu. Ako se želi više instanci, potrebno je kopirati isti deo koda više puta za svaku instancu. Na slici 1 prikazan je objektni literal u JavaScript-u.

```
var User = {  
  name: "John",  
  age: 20,  
  printUser: function() {  
    return "I am John";  
  }  
}
```

SLIKA 1. OBJEKTNI LITERAL

Konstruktorski šablon predstavlja korišćenje konstruktorske funkcije za kreiranje objekta. Konstruktorska funkcija je funkcija koja se ponaša kao konstruktor, to jest, vraća kreirani objekat kao povratnu vrednost. Šabloni modula su slični konstruktorskim šablonima, samo što mogu da imaju i privatna polja i funkcije kojima se ne može pristupiti izvan objekta. Takođe postoji i šablon otkrivajućih modula, koji pokazuje da se kreiraju objekti sa privatnim poljima i javnim funkcijama generičkog naziva koje će ta polja čitati ili menjati. Šablon prototipskog nasleđivanja pokazuje kako se mogu naslediti objekti u JavaScriptu nasleđivanjem prototipa objekta. Prototipskim nasleđivanjem kreira se *Prototype Chain*, to jest, lanac prototipa koji povezuje prototipove svih nasleđenih objekata. Vrhovni objekat je tipa *Object* i njega svaki objekat nasleđuje, to jest, njegov prototip. Nasleđivanjem prototipa objekat nasleđuje i svojstva drugog objekta, pa i njegov tip. Ovaj šablon je vrlo važan za shvatanje primene SOLID principa u JavaScriptu. Na slici 2 prikazan je primer prototipskog nasleđivanja gde „klasa“ *Square* nasleđuje „klasu“ *Rectangle*.

```
function Rectangle(length, width){  
  this.length = length;  
  this.width = width;  
}  
Rectangle.prototype.getArea = function() {  
  return this.length * this.width;  
}  
  
function Square(size){  
  this.length = size;  
  this.width = size;  
}  
Square.prototype = new Rectangle();  
Square.constructor = Square;  
  
var rect = new Rectangle(5, 10);  
var square = new Square(6);
```

SLIKA 2 PROTOTIPSKO NASLEĐIVANJE

ECMAScript2015 donosi velike novine u JavaScript, u obliku koncepta klase i nasleđivanja klase, koje i dalje u pozadini zapravo radi prototipsko nasleđivanje. Ovo donosi povećanje čitljivosti koda i razumljivosti jer sintaksno JavaScript od tada postaje mnogo bliža i sličnija objektno orijentisanim programskim jezicima. Šablon

fabrika predstavlja funkciju koja omogućava kreiranje objekata bez poznavanja kog tipa će oni biti.

Podklase pri instanciranju biraju koju klasu žele da instanciraju. Ovo je vrlo rasprostranjen šablon, koji se koristi u raznim vrstama softvera.

5. PRIMENA SOLID PRINCIPA U JAVASCRIPT-U

Primena SOLID principa u JavaScript-u je dosta zanemarena. S obzirom na to da je JavaScript slabo tipiziran jezik i da je veoma fleksibilan, ne forsira primenu nikakvih principa i vrlo lako može doći do neke greške u kodu. Na primer, česte greške se dešavaju kada objekat promeni svoj tip i više nema polja koja su se očekivala.

Takođe, pošto nema koncepte interfejsa i apstraktnih klasa, primena nekih SOLID principa je otežana, međutim interfejs se može na određene načine imitirati i tako uspeti u pokušaju da se primene SOLID principi. TypeScript je, na primer, programski jezik koji je nastao od JavaScripta i u njemu ovih problema nema. On podržava statičko tipiziranje objekata i takođe ima koncepte interfejsa. Samim tim dosta je lakše i intuitivnije primeniti SOLID principe u TypeScript-u.

5.1. Single Responsibility Principle

Princip jednostruke odgovornosti se može lako primeniti u JavaScriptu. Da bi se primenio, potrebno je dobro razumeti problem koji se rešava datim objektom. Potrebno je pokušati, na osnovu znanja o objektu, kreirati takav objekat da ima samo jednu odgovornost u sebi, to jest, samo jedan razlog da se promeni. Problem kod ovog principa nastaje kada se previše pokušavaju smanjiti objekti i time se dobija ogroman broj sitnih objekata čime se mnogo poveća kompleksnost koda.

Zbog toga je vrlo važno razumeti šta taj objekat treba da radi i dobro razmisliti da li je poštovanje ovog principa zaista potrebno u tom slučaju. Ako je objekat mali i njegova funkcionalnost zanemarljiva, pri čemu se zna da se neće u budućnosti menjati, onda uglavnom nema potrebe da se vodi računa o ovom principu.

5.2. Open-Closed Principle

Otvoren-zatvoren princip se postiže apstrakcijom klase koja treba da bude otvorena za proširenja i zatvorena za izmene. Ovo se dobija kreiranjem klase koja nema implementirane funkcije, a pri čemu će klase koje nasleđuju tu klasu da implementiraju na svoje načine te funkcije.

Kao što je već spomenuto, u JavaScriptu se nasleđivanje može izvršiti prototipskim nasleđivanjem tako što se prototip nadklase kopira u prototip podklase, ili korišćenjem ECMAScript2015 standarda gde postoji ključna reč „*extends*“ kojom se može naslediti neka klasa. Time bi se moglo, za svaki novi potreban tip kreirati klasu koja će naslediti nadklasu.

5.3. Liskov Substitution Principle

Kao što je već spomenuto, princip Liskove zamene se svodi na to da objekat podklase može da se zameni objektom nadklase a da se pri tom ponašanje programa ne promeni. U JavaScript-u se može postići prototipskim nasleđivanjem jer tada objekat podklase prima tip

nadklase koju nasleđuje. Međutim, ovaj princip je više primenljiv u TypeScript-u gde postoje tipovi i provere tipova. Moglo bi se, na primer, u funkciji, koja može da primi bilo koji tip podklase koje nasleđuju jednu nadklasu, staviti da prima objekat tipa nadklase kao ulazni parametar. Tada bi se funkciji mogao proslediti bilo koji objekat podklase i ne bi došlo do greške.

Ovime se štedi mnogo linija koda, jer se dobija generalizacija funkcije, i umesto više funkcija ili umesto više uslova provere unutar jedne funkcije, može se iskoristiti samo jedna generička funkcija. Takođe, ovo je mnogo čitljivije, jer uglavnom nije toliko važno koje sve podklase postoje, koliko je važno koje su ponašanje nasledile od nadklase i koje funkcije postoje u nadklasi.

5.4 Interface Segregation Principle

Princip razdvajanja interfejsa se može postići u JavaScriptu tako što će se objekti koji imitiraju interfejs kreirati tako da sadrže što manje funkcija, to jest, tačno onoliko koliko će objekti koji implementiraju taj interfejs koristiti. Ovo je dosta korisno u TypeScript-u koji zaista ima interfejsa i može dosta doprineti smanjenju viška koda koji se nikad ne izvršava i ne koristi.

Takođe, klase koje implementiraju taj interfejs, neće morati da implementiraju i metode koje im neće nikad trebati. Ovde opet treba u potpunosti razumeti ulogu i funkciju interfejsa, kao i zadatak koji treba da izvršava, da bi se moglo planirati unapred kakve će sve izmene biti moguće u budućnosti.

5.5 Dependency Inversion Principle

Princip inverzije zavisnosti može se u JavaScriptu postići prototipskim nasleđivanjem objekata koji imitiraju interfejs, ili pomoću funkcija višeg reda. Funkcije višeg reda su funkcije koje primaju druge funkcije kao ulazni parametar (eng. *callback function*) ili kreiraju novu funkciju koju vraćaju kao povratnu vrednost.

Na ovaj način dobija se apstraktna funkcija (funkcija višeg reda) koju definišu, druge funkcije, funkcije koje joj se proslede kao ulazni parametar. Time se postiže to da apstrakcija ne zavisi od detalja, već detalji od implementacije, što je i suština ovog principa. Samim tim povećava se iskoristivost ove funkcije u mnogo više slučajeva upotrebe.

6. ZAKLJUČAK

Iz priloženog može se zaključiti da su SOLID principi u JavaScriptu poprilično zanemareni, jer da bi se primenili, potrebno je malo dodatnog truda. Međutim, primena SOLID principa je od velikog značaja jer doprinosi kreiranju čitljivijeg koda i koda koji je lak za održavanje. Kada se u budućnosti budu unosile neke izmene mnogo ih je lakše uneti ako se unose na jednom mestu, a to bi bio slučaj ako su SOLID principi ispoštovani.

Ako nisu, onda je potrebno menjati veliki deo koda, jer su mnogi delovi čvrsto povezani i to može koštati mnogo i izazvati dosta grešaka u kodiranju.

Takođe, kada se izmene unose posle dužeg vremena, čak i autor koda zaboravi šta je sve i gde radio, pa je i zbog toga vrlo važno da je kod čitljiv i da je lako snalaziti se u njemu iako se ne poznaje u potpunosti ceo kod.

Osim povećane čitljivosti i razumljivosti koda, poštovanjem SOLID principa dobija se i veća otpornost na greške. Modularnost koda dovodi do toga da manje dolazi do grešaka jer su svi delovi koda modularni, to jest, izolovani.

Čak i ako se desi greška, mnogo ju je lakše pronaći kada je kod modularan, jer se tačno zna koji deo koda ima koju odgovornost. Ako se zna u kojoj odgovornosti se desila greška, znaće se i deo koda u kom je greška nastala. Korišćenjem interfjesa, nasleđivanjem i polimorfizmom dobija se dosta generalizacije i razdvajanja zavisnosti između delova koda, što doprinosi smanjenju grešaka, lakšem testiranju, ali i boljoj čitljivosti koda. SOLID principi nisu zahtevi već saveti kako bolje programirati, uz manje grešaka i lakše menjanje u budućnosti. Ne treba se uvek truditi da se po svaku cenu implementiraju svi SOLID principi, jer to često može dovesti do prekompleksnog koda.

Zapravo, treba prvo dobro shvatiti sve SOLID principe, njihove mane, prednosti, načine primene i efekte koje proizvode. Potom, treba dobro razumeti koji se problem rešava, kakve su njegove osobine i kako će se on menjati u budućnosti. Tek onda treba dobro razmisliti koji SOLID principi bi mogli najviše pomoći u rešavanju tog problema, bez suvišnog komplikovanja i nagomilavanja koda.

S obzirom na to da se JavaScript ubrzano i stalno razvija, i to ne samo JavaScript već i drugi jezici i radni okviri koji su nastali od JavaScript-a, vrlo je verovatno da će on sve više vremenom ličiti na objektno orijentisane programske jezike i da će se u potpunosti moći koristiti svi principi koje koriste i čisti objektno orijentisani jezici.

7. LITERATURA

- [1]Robert C. Martin (2000), „Design Principles and Design Patterns“, (pristupljeno u septembru 2018. godine)
- [2]Addy Osmani (2012), “Learning JavaScript Design Patterns”, (pristupljeno u septembru 2018. godine)

Kratka biografija:



Milica Kapetina rođena je 31.05.1994. godine u Sarajevu. Završila je osnovnu školu “Prva vojvodanska brigada” u Novom Sadu. 2009. godine. Srednju medicinsku školu “7. april” završila je 2013. godine i iste se upisala na Fakultet tehničkih nauka u Novom Sadu, odsek Računarstvo i automatika. Osnovne studije na Fakultetu tehničkih nauka završila je 2017. godine i iste godine se upisala na master studije, takođe na Fakultetu tehničkih nauka.

UTICAJ SOLARNIH PANELA NA PODEŠENJE ZAŠTITE NA IZVODU IMPACT OF SOLAR GENERATORS ON FEEDER PROTECTION SETTING

Filip Surla, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je razmatran problem relejne zaštite sa injektiranim solarnim panelima po potrošačkim područjima. Iznete su teorijske osnove iz aspekta relejne zaštite izvoda, model konzuma i solarnog panela. Potom je izvršena analiza rada zaštite u inicijalnom stanju i u stanju sa injektiranim strujama solarnih izvora.

Ključne reči: *Relejna zaštita, zaštita izvoda, solarni paneli*

Abstract – *This paper discusses the problem of relay protection with injected solar panels in consumer areas. Theoretical bases from the aspect of relay protection of the outlet, model of consumption and solar panels are laid out. Then, it was analyzed relay protection operation in the initial state and in the state with the injected currents from solar sources.*

Keywords: *Relay protection, feeder protection, solar generators*

1. UVOD

U ovom radu obradiće se tema uticaja solarnih panela instalisanih na individualna domaćinstva na prepodešenje relejne zaštite na izvodu. Izneće se teorijske osnove o zaštiti izvoda, solarnih panela i adekvatan matematički model sa verifikacijom. U glavi broj dva definiše se teorijske osnove o kvarovima u elektroenergetskim sistemima sa akcentom na distributivne mreže. U glavi broj tri definišu se osnovni zadaci relejne zaštite kao i vrste releja koji se koriste. U glavi broj četiri biće reči o solarnim panelima, kako se modeluju, koje su sve veličine koje figurišu pri odabiru i proračunu solarnog panela, zatim će se navesti kako se solarni paneli štite od eventualnih kvarova. U glavi broj pet predstaviće se svi pozitivni i negativni aspekti instalisanja distribuiranih izvora energije iz aspekta relejne zaštite. U glavi broj šest je izvršio se proračun pojedinačnog konzuma, inicijalno podešenje zaštite kao i maksimalan broj domaćinstava po konzumu sa solarnim panelima bez uticaja na relejnu zaštitu. U glavi broj sedam je donet zaključak na osnovu svega gore navedenog.

2. KVAROVI U DISTRIBUTIVNIM MREŽAMA

Kratak spoj je slučajan ili nameran spoj provodnika, preko malog otpora ili impedanse, između dve ili više tačaka, koje su na različitim potencijalima [1].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je dr Duško Bekut, red. prof.

Karakteristike uređaja koji obavljaju tu funkciju zavise od veličine struja koje prekidaju, odnosno od vremenskog trenutka u kome isključuju kvar. Zato je poznavanje odvijanja prelazne pojave i veličine struje kvara koja se prekida od iznimnog značaja za podešavanje zaštitnih releja i izbor parametara prekidača i druge rasklopne opreme, odnosno rezultati proračuna krakatih spojeva se koriste kako pri projektovanju tako i pri eksploataciji i planiranju elektroenergetskog sistema. U distributivnim mrežama, kao i u elektroenergetskim sistemima uopšte, analiziraju se sledeće vrste kvarova: jednopolni kratki spoj, dvopolni kratki spoj bez zemlje, dvopolni kratki spoj sa zemljom i troploni kratak spoj sa i bez zemlje.

3. ZAŠTITA RADIJALNIH MREŽA

Električne mreže predstavljaju deo elektroenergetskog sistema koji služi za prenos električne energije. Pod pojmom mreže se prvenstveno misli na vodove (kablovske i vazdušne) [2]. Za zaštitu vodova se koriste sledeće vrste zaštita:

- prekostrujna,
- distantna,
- diferencijalna i
- zaštita osiguračima.

Izbor za zaštitu radijalnih mreža jeste prekostrujna zaštita sa nekim svojim izvedbama a to su:

- trenutna prekostrujna zaštita $J >>$ (tzv. kratkospojna zaštita),
- prekostrujna zaštita $J > i$
- zaštita sa relejom nulte komponente – zemljospojna zaštita $J_0 >$.

Primenom gore navedenih zaštita vrši se obezbeđenje od međufaznih kratkih spojeva i od zemljospojeva kojih ima procentualno i najviše u distributivnim mrežama. Zaštita je pozicionirana na početku izvoda, odnosno u samom izvodnom polju transformatorske stanice 110/x/y kV/kV/kV. U američkim mrežama koristi se drugi pristup, odnosno, zaštite se pozicioniraju i u dubini mreže zbog veće selektivnosti.

3.1 PREKOSTRUJNA ZAŠTITA

Prekostrujni releji sa strujno nezavisnom karakteristikom se stavljaju u stanje pobude kada struja premaši podešenu vrednost, ali njihovo delovanje nije trenutno, već dolazi posle određenog vremenskog perioda. Konstrukcija trenutnih prekostrujnih releja (često se nazivaju i kratkospojni) je dosta jednostavnija u odnosu na obične prekostrujne releje, jer ne sadrže vremenski član.

Strujno podešenje releja se bira na osnovu izraza (1):

$$I_p = \frac{k_{sigurnosti} \cdot k_{spoja} \cdot I_{pogonsko,max}}{a \cdot p_i}, \quad (1)$$

gde je:

$k_{sigurnosti}$ – koeficijent sigurnosti koji uzima vrednosti 1.1 ili 1.2,

k_{spoja} – koeficijent spoja koji je za spoj na fazne struje jednak jedinici a kod spoja razlika struja,

p_i – prenosni odnos strunog transformatora.

3.2 ZEMLJOSPOJNA ZAŠTITA

Zemljospojna zaštita se realizuje preko releja nulte komponente. Ovi releji se koriste i u slučaju kada je po intezitetu struje teško razlikovati normalno radno stanje ili stanje kvara. Nulta komponenta se dobija pomoću sumarnog spoja strujnih mernih transformatora. Ovakav spoj se naziva i Holmgrin spoj.

Princip rada se zasniva na detektovanju sume struja koja je različita od nule. U normalnom pogonu, suma tri fazne struje bi trebala da bude nula ili veoma bliska nuli. Ovakvo razmatranje se odnosi na trofazne uravnotežene mreže. U slučaju kvara (zemljospoja), nesimetrija postaje znakovitija, pa je i suma struja različita od nule, te relej detektuje takvo stanje.

3.3 DISTANTNA ZAŠTITA VODOVA

U ovom radu nije stavljen akcenat na podešenje i impelenciju distante zaštite. Distantna zaštita se prevashodno koristi na nivoima 110 kV i više ali neretko se nalazi i na nivou distribucije. Smisao uvođenja distatne zaštite jeste da se ne traži struja kvara ili napon na mestu kvara već impedansa kvara od mesta ugradnje releja do mesta kvara i na takav način dolazi se do reagovanja zaštite.

4. MODEL SOLARNIH PANELA

Solarni paneli predstavljaju skup više solarnih ćelija vezanih u celinu i mogu se za potrebe matematičkog modelovanja prikazati šematski električnim veličinama. Modeluje se kao idealni strujni generator sa dodatim otpornostima R_p i R_s koje se u grublim aproksimacijama mogu zanemariti.

HDKR model (Hay, Davies, Klucher, Reindl) je jedan od najboljih modela za određivanje količine sunčevog zračenja koje dolazi na neku površinu koja je pozicionirana na površini zemlje.

Ukupna količina Sunčevog zračenja koje pada na solarni panel računa se kao:

$$G_{Tt} = (G_{bt} + G_{dt} \cdot A_i) \cdot R_b + G_{dt} \cdot (1 - A_i) \cdot \left(\frac{1 + \cos\beta}{2} \right) \cdot \left[1 + f \sin \frac{\beta^3}{2} \right] + G_t \cdot \rho_g \cdot \left(\frac{1 - \cos\beta}{2} \right), \quad (2)$$

gde je:

G_{Tt} – ukupno ekstraterističko horizontalno zračenje [kW/m^2],

G_{bt} – direktno ekstraterističko horizontalno zračenje [kW/m^2],

G_{dt} – rasuto ekstraterističko horizontalno zračenje [kW/m^2],

R_b – odnos direktne komponente zračenja na nagnutnu površinu u odnosu na horizontalu,

A_i – indeks anizotropije koji je mera propusnosti direktnog zračenja kroz atmosferu,

f – indeks kojim se uzima u obzir da je svetlost koja dolazi sa horizonta više rasuta,

ρ_g – refleksija okoline i

β – nagib površine generator [$^\circ$].

Izlazna snaga solarnog panela se definiše kao:

$$P_{PV} = Y_{PV} \cdot f_{PV} \cdot \left(\frac{G_{Tt}}{G_{Tt,STCt}} \right) \cdot [1 + \alpha_p \cdot (T_c - T_{c,STC})] \quad (3)$$

gde je:

Y_{PV} – snaga koju daje solarni panel pri standardnim uslovima [kW],

f_{PV} – faktor redukcije [%],

$G_{T,STCt}$ – količina radijacije pri standardnim uslovima testiranja [kW/m^2],

α_p – temperaturni koeficijent snage [%/ $^\circ\text{C}$],

T_c – temperatura solarnog panela za dati vremenski korak [$^\circ\text{C}$] i

$T_{c,STC}$ – temperatura solarnog panela pri standardnim uslovima testiranja [$^\circ\text{C}$].

4.2 ZAŠTITA SOLARNIH PANELA

Zaštita od kratkih spojeva se izvodi pomoću prekostrujnih okidača prekidača (na niskonaponskoj strani). Takođe ovakav uređaj služi i za zaštitu od preopterećenja. Ugrađuje se u razvodnim ormanima niskog napona.

5. UTICAJ SOLARNIH PANELA NA ZAŠTITU IZVODA

USN-mrežama koje sadrže uglavnom nadzemne vodove većinakvarova su privremene prirode, pa prvo zahteva zaštitne strategije radijalnog sistema odnosno uklanjanje svih grešaka. U praksi se ističe nekoliko problema pri povezivanju DG na radijalne izvode, a iz aspekta relejane zaštite: povećanje struje kratkog spoja, zaslepljivanje zaštite inehsinhronizirani rad uređaja distributivne automatike [3].

5.1 POVEĆANJE STRUJE KVARA

Doprinos ukupnoj struji kvara od jednog malog distributivnog generatora nije veliki. Ipak, kada se razmatra skup više malih jedinica, njihov doprinos može značajno uticati na povećanje struje kratkog spoja, a time na greške u koordinaciji između zaštitne opreme, što dalje dovodi do ugrožavanja sigurnosti i pouzdanosti distributivnih sistema. Distribuirani generatori koji su povezani na distributivnu mrežu menjaju intenzitet ismer struje kvara kroz distributivne vodove, i nakon priključenja mogu dovesti do poremećaja koordinacije postojeće zaštite.

5.2 ZASLEPLJIVANJE ZAŠTITE

Ovaj problem najviše utiče na rad relejne zaštite. Kao što je gore navedeno, struja kvara u radijalnim mrežama se povećava ako su u mreži dodati distribuirani izvori električne energije. Takođe je poznato da je relejna zaštita pozicionirana na glavi fidera odnosno da se prekostrujna zaštita nalazi na prekidaču u izvodnom polju.

Struja kvara jednaka je zbiru struja koju vidi relej (struja iz mreže) i struje koju daje distributivni generator. Što znači da kroz relej teče struja:

$$I_R = I_F - I_{DG}, \quad (4)$$

gde je:

I_R – struja kvara koju vidi relej [A],

I_F – stvarna struja kvara [A] i

I_{DG} – struja kvara koju doprinosi DG [A].

Zaključuje se da će relej videti manju struju od stvarne te da postoji mogućnost da neselektivno deluje.

5.3. NESINHRONIZOVANI RAD UREĐAJA DISTRIBUTIVNE AUTOMATIKE

U srednjenaponskim i niskonaponskim radijalnim distributivnim mrežama, zbog jednostavnosti i niske cene, često se mogu sresti sistemi zaštite bazirani na "rekloserima" i osiguračima. Ovaj koncept može ostvariti dovoljno dobru selektivnost, osetljivost i pouzdanost sistema zaštite jedne distributivne mreže, bez velikih ekonomskih investicija u vidu srednjenaponskih prekidača i mikroprocesorskih releja.

U slučaju da se u mrežu doda distribuirani izvor energije, raspodela struja se može radikalno promeniti. U ovakvom slučaju struja kroz osigurač se povećava dok se struja kroz reklozer smanjuje.

Zavisno od toga koliko struje distribuirani izvor ubacuje u mrežu toliko će se koordinisanje reklozer-osigurač poremetiti.

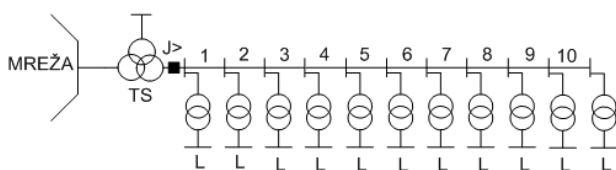
5.4 FAULT RIDE THROUGH

Većina razvijenih zemalja ima jasno definisana Pravila o pogonu prenosnih i distributivnih mreža. Između ostalih zahteva, u ovim pravilima su definisani i posebni zahtevi „Fault Ride Through – FRT“ (ili opštije – „Low Voltage Ride Through – LVRT“).

Ovi zahtevi se odnose na sposobnost DER da ostanu u pogonu pri sniženim naponima. Zahtevima FRT se jasno utvrđuje kako generatori zasnovani na uređajima energetske elektronike (DER tipa 3-4 u distributivnim mrežama) reaguju u slučaju smanjenih napona u čvorovima u kojima su oni priključeni na mrežu, koji su prouzrokovani kvarovima bilo gde u mreži [4].

6. PRIMER UTICAJA SOLARNIH PANELA NA RELEJNU ZAŠTITU

Primer uticaja solarnih panela na prepodešenje zaštite prikazaće se na jednom srednjenaponskom izvodu. Na šemi je uočena samo prekostrujna zaštita jer trenutna prekostrujna zaštita ne može da odreaguje na bilo koju promenu zbog FRT-a. Razmatra se 20 kV izvod prikazan na slici 6.1.



Slika 6.1 – Razmatrani srednjenaponski izvod

6.1 MODEL POTROŠAČKOG PODRUČJA

Potrošačko područje se napaja preko transformatora instalirane snage od 1000 kVA, samim tim se proračunava koliko se može opteretiti ovakav transformator. Iz ovakve transformatorske stanice izlazi ukupno osam izvoda, pa se opterećenje ravnomerno deli.

Za ovakvo područje usvajaju se sledeći parametri koji su neophodni za proračun opterećena distributivnog transformatora: $\cos\phi = 0.95$, $k=0.8$, $P_{\max, \text{izvodu}} = 95$ [kW], $j_\infty = 0.25$, $j = 0.6$, $P_{\text{dom}} = 11.25$ [kW] pa se ukupan broj domaćinstava po izvodu dobije 38 a na celom trafo području – 304 domaćinstva. Potrebno je još izračunati koliko aktivnu snagu odaje jedan solar postavljen na krov jednog domaćinstva. Na osnovu površine krova, faktora redukcije i unapred poznate snage koju prosečan solarni panel daje izračunava se maksimalna izlazna snaga, a samim tim i odata struja. Tako da za usvojenih $S_{\text{krova}} = 90$ m², $G_{z, kv} = 120$ [W/m²], $f_{\text{redukcije}} = 0.75$ dobija se $I_{\text{solara}} = 0.24$ [A] svedeno na srednjenaponski nivo.

6.2 PODEŠAVANJE RELEJNE ZAŠTITE BEZ SOLARNIH PANELA

Za razmatrani izvod prikazan na slici 6.1 vrši se podešavanje trenutne prekostruje zaštite, prekostrujne zaštite i kratkospojne zaštite. Inicijalno podešenje zaštite se vrši pomoću ukupne struje potrošnje koje ide kroz izvod. Strujna podešenja trenutne prekostruje, prekostruje i zemljospojne zaštite respektivno iznosi:

- $I_{\text{podešenja, kratkospojno, zaštite}} = 1732$ [A],
- $I_{\text{podešenja, prekostrujno, zaštite}} = 519.615$ [A] i
- $I_{\text{podešenja, zemljospojno, zaštite}} = 60$ [A].

Za prekostrujnu zaštitu treba izvršiti proračun koeficijenta osetljivosti, koji se definiše kao odnos struje dvopolnog kratkog spoja bez zemlje i struje podešenja prekostruje zaštite. Za izvod koji je prikazan na slici 6.1 proračunate su ekvivalente impedanse direktnog i inverznog redosleda, kao i adekvatna struja dvopolnog kratkog spoja I_{LL} , koji se simulira na kraju izvoda. Odnos struje dvopolnog kratkog spoja i struje podešenja prekostruje zaštite dobija se vrednost koeficijenta osetljivosti koja je u ovom slučaju 2.34, što je dovoljno da bi se održala selektivnost zaštite.

6.3. UTICAJ SOLARNIH PANELA NA OSETLJIVOST RELEJNE ZAŠTITE

Posle podešenja zaštite u inicijalnom stanju, i utvrđivanju koeficijenta osetljivosti za prekostrujnu zaštitu, u mrežu se dodaju solarni paneli koji su montirani na krovove individualnih potrošača (domaćinstava). Prvo treba definisati koliko uticaj ima jedan solarni panel sa svakog konzuma na struju koju vidi relej. Posle toga treba utvrditi iterativno koji je maksimalan broj domaćinstava sa solarnim panelima dozvoljen a da se ne ugrozi koeficijent osetljivosti zaštite. Svi solarni paneli u konzumu se modeluju kao idealni strujni generator koji može da uzme vrednost od 0 A do maksimalne struje koju odaju svi solarni paneli po konzumu. U čvor koji se injektira konzum, jasno je da struja kroz impedansu transformatora neće biti degradirana.

Struju dvopolnog kratkog spoja u ovakvom slučaju treba rešavati metodom superpozicije, odnosno rešavati dva kola, jedno sa naponskim generatorima, a drugo kolo sa strujnim generatorima. Potrebno je proračunati kolika struja kvara, koja potiče od solarnih panela, prolazi sa strane releja, odnosno koliko ta struja degradira struju I_{LL} , i dovodi do zaslepljenja zaštite. Da bi se kompletirao uticaj svakog solarnog panela po konzumu, nevezano na koji je čvor priključen, potrebno je definisati dve promenljive – Z_{desno} i Z_{levo} . Pomoću ove dve impedanse za svaki čvor ponaosob se računa ekvivalentna impedansa levo i desno gledajući iz određenog čvora. Smisao uvođenja ove promenljive jeste određivanje struje koja ide levo, odnosno koja će uticati na smanjenje struje kroz relej. Problem je iskonvergirao sa sledećim rešenjima: broj domaćinstava – $n = 182$, struja solara injektirana u mrežu $-I_{solara} = 451$ [A], struja releja $-I_{releja,novo} = 812.43$ [A] i koeficijent osetljivosti – $k_{osetljivosti} = 1.5047$.

Što implicira tome da je 59.87% domaćinstava kritičan procenat gde se treba razmišljati o prepodešenju zaštite. Ovakav rezultat praktično nije moguć jer bi u tom slučaju od svakog konzuma tekla snaga 40% veća od nominalne snage transformatora. U slučaju kada bi se u svakoj transformatorskoj stanici instalisao još jedan transformator snage 1000 kVA i da rade u paraleli, ovakav rezultat bi imao smisla.

6.4 UTICAJ SOLARNIH PANELA NA POVEĆANJE STRUJE KVARA

Razmatra se srednjenaponski izvod kao sa slike 6.1. Po čvorovima su i dalje injektirani isti transformatori srednji na niski napon, ali je zato odata snaga solara smanjena na 5 kW i broj dodatih solarnih panela po konzumu je 15%, što predstavlja realnije stanje sistema nego kao u prethodnom testu. Simuliranjem trolnog kratkog spoja na kraju izvoda dobijeno je da je porasla struja kvara za 5% dok je koeficijent osetljivosti pao, što potvrđuje fenomen porasta struje kvara i pada osetljivosti releja.

6.5 UTICAJ SOLARNIH PANELA NA RAD REKLOSERI I OSIGURAČA

Za zadati problem, analiziraće se slučajevi: kada instalisani distributivni generatori imaju uticaj na osigurač koji štiti distributivni transformator SN/NN i kada se štiti srednjenaponska deonica. Pri nominalnom radu, kada nema priključenih solarnih generatora sa strane niskog napona, iz mreže dolazi po 28 A za svaki konzum. Predpostavlja se da je na 15% domaćinstava instalisan solarni panel, kao u poglavlju 6.4. Tada će kroz osigurač teći struja reda veličina 20 A.

Za ovakvo stanje u sistemu, birali bi se drugačiji osigurači. Međutim treba imati na umu da se ovakva proizvodnja po konzumima dešava u određenim terminima (leti i preko dana), tako da bi svako menjanje osigurača koji štite konzum bilo nepotrebno. Takođe ako se uključi DG u bilo koji čvor pre osigurača može doći do povećanja struje osigurača i smanjenje struje reklosera i ugroziti selektivnost.

6.6 POSLEDICE POSTOJANJA DODATIH SOLARNIH PANELA NA STANJE U MREŽI

Kroz gornja poglavlja analizirali su se uticaji instalisanih solarnih panela sa strane niskog napona na relejnu zaštitu pozicioniranu na izvodu. Kroz nekoliko testova zapaženo je sledeće: pri velikom broju dodatih solarnih panela po konzumima, duž celog izvoda, može se doći do gubitka selektivnosti zaštite na izvodu, da pri manjem broju dodatih solarnih panela po konzumu, dolazi do povećanja struje kvara, te se može dovesti u pitanje odabir prekidačke opreme, prilikom dodavanja solarnih panela sa strane niskog napona, tokovi snaga se drastično mogu promeniti te se u momentima najveće proizvodnje solarnih panela može doći do smanjenja struja koje teku kroz osigurače, čime se smanjuje njihova selektivnost i ako se neke deonice štite osiguračima i rekloserima u paru, prilikom dodavanja solarnih panela može narušiti koordinisanje.

7. ZAKLJUČAK

U ovom radu obrađena je tema uticaja solarnih panela instalisanih na individualna domaćinstva na prepodešenje relejne zaštite na izvodu. Iznete su teorijske osnove o zaštiti izvoda, solarnih panela i adekvatan matematički model sa verifikacijom gde je ustanovljeno da se po konzumu nalazi 304 domaćinstva sa srednjom instalisanom snagom, kao i da je maksimalan broj domaćinstava sa instalisanim solarnim panelima 182 bez uticaja na prepodešenje zaštite. Takođe, donet je zaključak da pri niskim instalisanim snagama solarnih panela po konzumima, dolazi do povećanja struje trolnog kratkog spoja i da se menja opterećenje osigurača koje štite konzume.

8. LITERATURA

- [1] Strahil Gušavac, *Osnovni principi projektovanja u mrežama srednjeg i niskog napona*, Fakultet Tehničkih Nauka Novi Sad, 2004.
- [2] Duško Bekut, *Relejna zaštita*, Fakultet Tehničkih Nauka Novi Sad, 2009.
- [3] E. Coster, J. Myrzik, W. Kling, *Effect of distributed generation on protection of medium voltage cable grids*, CIRED Vienna 2007
- [4] L. Strezoski, *Proračun kompleksnih katkih spojeva neuravnoteženih distributivnih mreža sa distribuiranim energetske resursima*, Doktorski rad, Fakultet tehničkih nauka Novi Sad, 2017

Kratka biografija:



Filip Surla rođen je u Novom Sadu 1994. godine. Master rad je odbranio 2018. godine na Fakultetu tehničkih nauka iz Elektrotehnike i računarstva – Elektroenergetski sistemi.

**UPOTREBA SAVREMENIH METODA MODERNIZACIJE POSTOJEĆIH ASP.NET
APLIKACIJA SA CILJEM RADA NA CLOUD PLATFORMI****USAGE MODERN METHODS FOR MODERNIZATION EXISTING ASP.NET
APPLICATION WITH GOAL OF WORKING ON THE CLOUD PLATFORM**

Miloš Gagović, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – PRIMENJENO SOFTVERSKO
INŽENJERSTVO**

Kratak sadržaj – Ovaj rad ima za cilj da prikaže upotrebu microservisne arhitekture u modernizaciji monolitskih ASP.NET aplikacija, sa osvrtom na Cloud Native koncepte razvoja, i upotrebu Docker kontejnera kao jedinicu isporuke servisa.

Ključne reči: Docker, Mikroservis, ASP.NET, Servis Fabrik klaster, Azure

Abstract – This paper aims to demonstrate the use of microservice architecture in the modernization of monolithic ASP.NET applications, with a focus on Cloud Native development concepts, and the use of Docker containers as a service delivery unit.

Keywords: Docker, Microservice, ASP.NET, Service Fabric cluster, Azure

1. UVOD

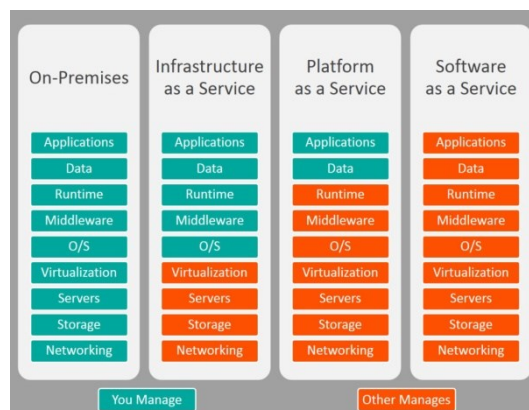
Svedoci smo da je današnje korišćenje aplikacija baziranih na WWW tehnologijama gotovo nemoguće zamisliti bez upotrebe termina cloud computing (računarstvo u oblaku). Moderni razvoj aplikacija podrazumeva korišćenje tehnologija zasnovanih na „cloudu“, tj platformama i infrastrukturi koje nude cloud provajderi. Zbog ubrzanog rasta i razvoja IT industrije javlja se potreba za pristupačnim, sigurnim, i pouzdanim softverom koji se lako održava i brzo isporučuje. U vezi sa tim, u daljem tekstu dat je pregled mogućnosti koje pruža cloud computing, kao i način dizajniranja, migracije i modernizacije aplikacije u cilju optimalnog iskorišćenja računarskih resursa.

2. RAČUNARSTVO U OBLAKU

Računarstvo u oblaku (cloud computing) predstavlja deljenje i korišćenje dostupnih računarskih resursa preko interneta. Velike kompanije kao što su Google, Microsoft i Amazon poseduju cloud infrastrukturu koju nude svojim korisnicima u zamenu za novac.

U zavisnosti od vrste korisnika, kao i od namene resursa, cloud computing možemo podeliti na četiri osnovne grupe, gde prva predstavlja deljenje resursa na zahtev korisnika, a bez učešća provajdera (Slika 1) :

- IaaS (infrastructure as a service) [1] je servis koji pruža korisniku odabir čitave vitruelizovane infrastrukture operativnog sistema, njegovo održavanje i ažuriranje. Cloud provajder je taj koji obezbeđuje hardver na kom će biti smeštena infrastruktura, i osigurava rad računarske mreže. Najčešći korisnici ovakvih servisa jesu administratori.
- PaaS (platform as a service) je servis koji pruža korisniku platformu za razvoj, odnosno nudi mu razvojno okruženje za njegove aplikacije i servise. Provajder je zadužen za ispravnost infrastrukture na koju se platforma oslanja, tj da obezbedi operativni sistem i čitav hardver na kome se on izvršava. Najčešći korisnici ovakvih servisa jesu developeri, a kao primer PaaS je Microsoft Azure. U daljem radu biće najviše fokusa posvećeno razvoju aplikacija korišćenjem ove cloud usluge.



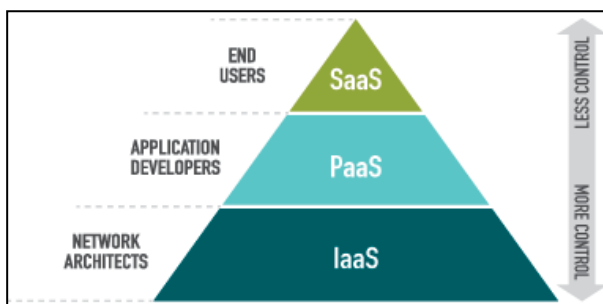
Slika 1. – Podela odgovornosti korisnika na cloudu

- SaaS (Software as a service) je servis koji nudi krajnjem korisniku korišćenje gotovih aplikacija u vidu servisa, a za kompletan njihov rad zadužen je provajder.

Na slici 2 prikazana je raspodela korisnika i njihovih odgovornost prilikom korišćenja servisa na oblaku.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Milosavljević, red. prof.

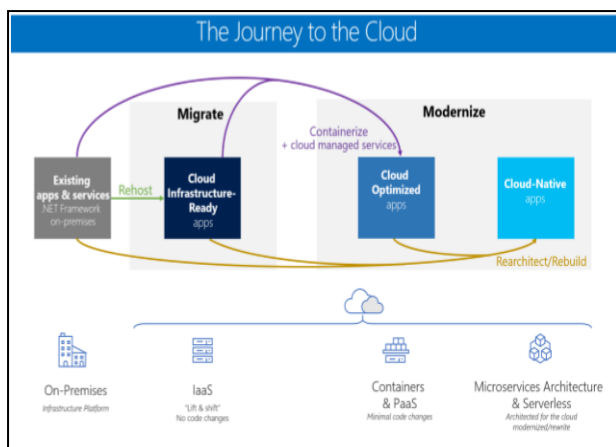


Slika 2. – Raspodela korisnika po vrstama servisa

3. MIGRACIJA APLIKACIJE NA „CLOUD“

Kada se donese odluka da se modernizuje web aplikacija ili usluge, i prebace na oblak, ne mora se nužno potpuno re-arhitektovati aplikacija. Ponovna arhitektura aplikacije pomoću naprednog pristupa kao što su mikroservisi nije uvek opcija zbog ograničenja troškova i vremena. U zavisnosti od vrste aplikacije, ponovna arhitektura aplikacije možda neće biti neophodna. Na to najviše utiče početni dizajn i kompleksnost, kao i vreme utrošeno na ponovnu izradu iste.

U tu svrhu izdvojena su tri nivoa migracija sistema na cloud koji predstavljaju razvojni put aplikacije u zavisnosti od stepena razvijenosti (slika 3).



Slika 3. Migracioni nivoi ka cloudu

Nivo 1 : Cloud infrastructure ready (aplikacije spremne za oblak). U ovom pristupu migracije aplikacija se jednostavno premesti ili se ponovo pokrenu lokalni servisi na infrastrukturi kao servisnoj platformi (IaaS). Aplikacija ima gotovo istu kompoziciju kao i ranije, ali sada je postavljena na virtuelne mašine na oblaku. Ova jednostavna vrsta migracije je u industriji poznata kao "Lift & Shift".

Nivo 2: Aplikacije optimizovane za cloud. Na ovom nivou se još uvek može preći na cloud bez ponovnog redizajna ili značajne izmene koda. Ovo se može postići upotrebom savremenih tehnologija poput kontejnera i dodatnih usluga koje nudi provajder. Na ovaj način se poboljšava agilnost u razvoju aplikacija i skraćuje se vreme potrebno za njihovu isporuku. To se postiže korišćenjem tehnologija kao što su Windows kontejneri, koji se zasnivaju na Docker Engine-u. Kontejneri obezbeđuju izolovanu sredinu u kome se izvršava aplikacija, i postavljaju se na IaaS ili PaaS.

Dodatna prednost ovakvog načina razvoja ogleda se u poboljšanju kontinualne integracije/ kontinualne isporuke (CI / CD) softvera, što ga čini konkurentnijim na tržištu.

Nivo3: Cloud-Native aplikacije. Ovaj migracioni pristup obično podrazumeva poslovne potrebe i cilja na modernizaciju aplikacija sa kritičnom misijom. Na ovom nivou, koriste se PaaS usluge za premeštanje aplikacija na PaaS računarske platforme. Primenjuje se arhitektura mikroservisa kako bi evoluirali aplikacije, povećali agilnost i pomerili granice razvoja. U većini slučajeva se piše novi kod, posebno kada su aplikacije zasnovane na mikroservisima. Ovaj pristup može pomoći pri dobijanju pogodnosti koje se teško postižu u monolitnom i lokalnom aplikacijskom okruženju.

Na osnovu navedenih migracionih pristupa na cloud, uvode se pojmovi kontejnera, Docker Engine-a, kao i mikroservisne arhitekture. U cilju prikaza pogodnosti migracionih postupaka postojeća monolitna ASP.NET aplikacija sa WEB API dizajnom biće premeštena na cloud. Koristiće se optimizovan pristup koji se zasniva na kontejnerima, i cloud nativ pristup baziran na mikroservisnoj arhitekturi.

4. DOCKER I DOCKER KONTEJNERI

Docker je kompjuterski program otvorenog koda koji automatizuje primenu aplikacija kao prenosivih, samoodrživih kontejnera koji mogu da se pokreću u oblaku ili na lokalnim serverima. Docker [2] je kompanija koja promoviše i razvija ovu tehnologiju. Kompanija radi u saradnji sa cloud provajerima (uključujući i Microsoft), a podržava operativne sisteme kao što su Linux i Windows.

Šta su kontejneri? Kontejneri (Linux ili Windows) su način za izvršavanje aplikacija u sopstvenom izolovanom procesu. U svom kontejneru aplikacije ne utiču na aplikacije ili procese koji postoje izvan kontejnera. Sve od čega aplikacija zavisi pokreće se kao proces unutar kontejnera. Gde god se kontejner premesti, zahtevi aplikacije za referencama će uvek biti ispunjeni, u smislu direktnih zavisnosti, jer dolazi u kompletu sa svime što je potrebno da radi (reference na biblioteke, vreme trajanja ...).

Glavna karakteristika kontejnera je ta što okruženje čini istim kroz različite faze isporuke, jer sam kontejner ima sve potrebne zavisnosti. To znači da možete pokretati aplikaciju na vašoj mašini, a zatim je isporučiti na drugu, i dobiti jednako okruženje za rad.

Kontejner je instanca slike (Image) kontejnera . Slika kontejnera je način za pakovanje aplikacije ili usluge, a zatim ga rasporediti na pouzdan i ponovljiv način. Moglo bi se reći da Docker nije samo tehnologija - to je i filozofija i proces. Pošto kontejneri svakodnevno postaju sve češći u upotrebi, oni postaju industrijska "jedinica raspoređivanja" (unit of deployment).

Za nekoga ko je upoznat sa virtuelnim mašinama, izgleda da su kontejneri izuzetno slični. Kontejner pokreće operativni sistem, ima sistem datoteka i može mu se pristupiti preko mreže, baš kao i fizički ili virtuelni računarski sistem.

Međutim, tehnologija i koncepti iza kontejnera su znatno različiti od virtuelnih mašina. Sa stanovišta programera, kontejner mora biti tretiran više kao jedan proces.

Zapravo, kontejner ima jednu ulaznu tačku za jedan proces. Docker kontejneri mogu raditi na Linux i Windows operativnim sistemima.

Kada se koriste redovni kontejneri, Windows kontejneri se mogu pokrenuti samo na Windows hostovima (host server ili VM), dok se Linux kontejneri mogu pokrenuti samo na Linukskim hostovima.

Međutim, u najnovijim verzijama Windows servera i Hiper-V kontejnera, kontejneri za Linux se takođe mogu pokrenuti na Windows Server-u koristeći Hiper-V tehnologiju izolacije koja je trenutno dostupna samo na Windows Server kontejnerima. U bliskoj budućnosti, mešovita okruženja koja imaju i Linux i Windows kontejnere biće moguća, čak i česta.

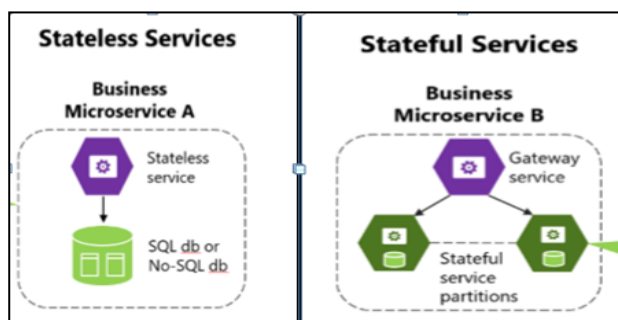
4.1. Poređenje kontejnera i virtuelne mašine

Na slici 4 prikazano je poređenje rada virtuelnih mašina i docker kontejnera. Virtuelne mašine uključuju aplikacije i potrebne biblioteke za njihov rad, kao i čitav operativni sistem na kome se one izvršavaju.

Suprotno tome, kontejneri se izvršavaju kao zasebni (izolovani) procesi koje kontroliše container engine. Svi procesi dele resurse operativnog sistema na kome su pokrenuti, a njihovo pokretanje traje znatno kraće. Zbog toga što zahtevaju manje resursa (ne treba im ceo operativni sistem) lakše se isporučuju i pružaju mogućnost pokretanja više servisa korišćenjem istog hardvera.

Neželjeni efekat pokretanja na istom operativnom sistemu je taj da se postiže manja izolacija procesa u odnosu na virtuelne mašine.

Za Docker se može reći da nije samo tehnologija, već i filozofija razvoja softvera. Upotrebom docker kontejnera izbegava se česta sintagma kod developera „Program funkcioniše na mojoj mašini, zašto ne radi u produkciji“. Ovo se jednostavno izbegava i dovoljno je reći da radi upotrebom dockera. Docker aplikacija se može izvršiti na bilo kom docker okruženju, i izvršiće se na načinu za koji je namenjena (DEV, QA, Staging, Product).



Slika 4. Poređenje virtuelne mašine i docker kontejnera

Kao osnovna jedinica isporuke u oba migraciona modela (Cloud Optimized i Cloud Native) odabrani su docker kontejneri zbog optimalnog iskorišćenja računarskih resursa.

5. MIKROSERVISNA ARHITEKTURA

Razumevanje mikroservisa i njihov način rada su važni prilikom planiranja o premeštanju softvera na cloud.

Arhitektura mikroservisa je napredni pristup koji se može koristiti za aplikacije koje su kreirane od nule ili razvijane primenom cloud native principa. Razvoj ovakvog tipa softvera iziskuje poseban dizajn kao i ponovnu implementaciju.

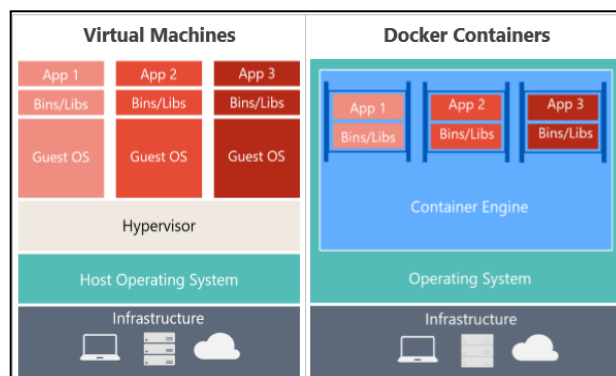
Međutim, mikroservisi nisu obavezni za bilo kakvu novu ili modernu aplikaciju. Mikroservisi [3] ne predstavljaju idealno rešenje, i nisu samo jedan, najbolji način za stvaranje svake aplikacije. Kako i kada koristite mikroservise zavisi od vrste aplikacije koju treba izgraditi.

Arhitektura mikroservisa postaje preferirani pristup za distribuirane i velike ili kompleksne kritične aplikacije koje se zasnivaju na višestrukim, nezavisnim podsistemima u obliku autonomnih usluga. U arhitekturi zasnovanoj na mikroservisima, aplikacija je izgrađena kao skup usluga koji se može samostalno razvijati, testirati, verifikovati, raspoređivati i skalirati. Ovo može uključivati bilo koju povezanu, autonomnu bazu podataka po mikroservisu.

5.1 Podela mikroservisa prema nameni

Podela mikroservisa implementiranih u Service Fabric [4] tehnologiji je na Stateless (bez pamćenja stanja) i Statefull (imaju sposobnost pamćenja stanja). Svaki mikroservis mora posedovati sopstveni domenski model podataka, i ne sme biti nezamenljiv. Na slici 5 prikazane su šeme ove dve vrste mikroservisa.

Glavna osobina statefull servisa u odnosu na stateless je ta da se podaci repliciraju preko više nezavisnih instanci servisa. Okruženje u kome se vrši praćenje replikacije i distribucije podataka obezbeđuje platforma za orkestriranje servisa. Service Fabric klaster kao Microsoftov predstavnik nudi mogućnost praćenja rada servisa, kao i prikaz stanja servisa prilikom rada.



Slika 5. – Prikaz razlike između mikroservisa

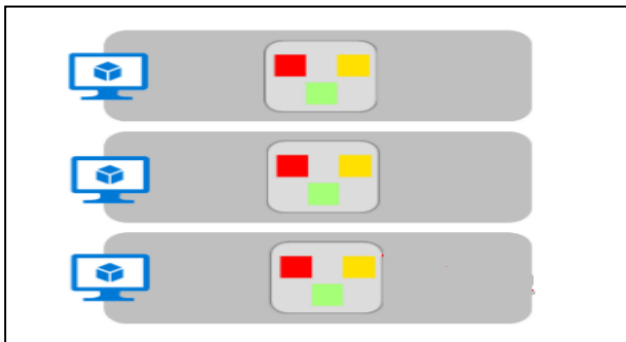
Pristup kreiranju stateless servisima je lakši za implementaciju. Ovakav način je sličan tradicionalnim obrascima za kreiranje. Podaci se čuvaju u eksternoj bazi i nema replikacije po particijama. Nasuprot tome, statefull servisi se „podižu“ na oblak u vidu više nezavisnih particija. Jedna od njih predstavlja primarnu, dok se na sve ostale podaci repliciraju i imaju mogućnost da postanu primarne ukoliko se desi nepredviđen ispad. Sadrže pomoćni mikroservis u vidu gateway-a koji balansira saobraćaj između particija, i u trenutku ispada menja namenu particijama.

6. IMPLEMENTACIJA MIGRACIONIH REŠENJA I POREĐENJE PERFORMANSI RADA

Za potrebe implementacije migracionih rešenja iskorišćena je ASP.NET aplikacija i WEB API tehnologija. WEB API je odabran iz razloga što se razmena podataka obavlja u JSON formatu upotrebom HTTP protokola, i mogu mu pristupiti sve aplikacije koje podržavaju ovaj protokol. Kao monolitna ASP.NET aplikacija upotrebljen je servis za rezervaciju smeštaja u hotelima. Primenom migracionih postupaka Cloud Modernized i Cloud Native došlo se do rešenja pogodnog za izvršavanje na cloud platformi. U nastavku će biti prikazane pogodnosti koje ova rešenja nude, kao i razlike u kompleksnosti dizajna i arhitekture.

Rešenje implementirano upotrebom mikroservisa i servis fabric klastera zasniva se na upotrebi više instanci mikroservisa postavljenih na virtuelnim mašinama na cloudu, ali tako da svaka instanca predstavlja jedan proces unutar docker kontejnera.

Na slici 6 prikazan je izgled aplikacije na cloudu upotrebom Cloud Modernized metode. Uočavaju se tri nezavisna servisa postavljena na tri virtuelne mašine u vidu docker kontejnera, s tim da komunikacija između servisa na različitim mašinama nije ostvarena. Na ovaj način smo relativno lako preneli čitavu arhitekturu na cloud, gubeći pri tom mogućnost skalabilnosti, i smanjujući stepen otpornosti na otkaz.

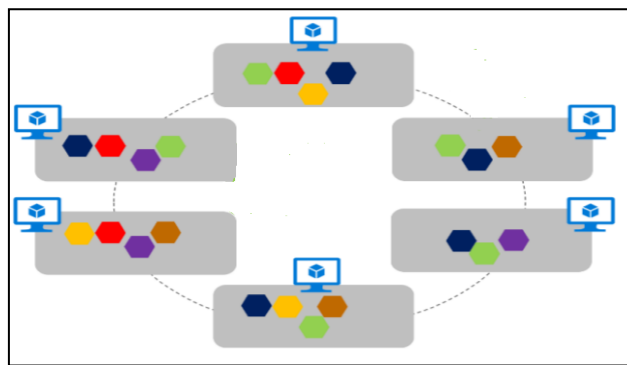


Slika 6. – Cloud modernized migracija na cloud

Suprotno ovome, na slici 7 prikazana je arhitektura rešenja upotrebom mikroservisa u obliku kontejnera, i service fabric klastera kao orkestratora. Svaka instanca servisa postavljena je na više različitih okruženja, između kojih postoji komunikacija preko API-a. Pomoću njega se razmenjuju i repliciraju podaci.

U zavisnosti od potrebe, klaster u kome su smešteni zahtevaće veću ili manju promenu u skaliranju procesa unutar kontejnera. To znači da im obezbeđuje onoliko resursa koliko je stvarno potrebno za njihov rad, ili čak da kreira nove ili gasi stare procese.

Ovo je od velikog značaja prilikom rada na cloudu, jer se cena rada formira na osnovu iskorišćenih resursa. Postiže se veći stepen otpornosti na otkaze, što je jedan od preduslova za dobar i siguran rad distribuiranih aplikacija.



Slika 7. – Cloud native migracija na cloud

7. ZAKLJUČAK

Fokus ovog rada baziran je na primeni migracionih tehnika pomoću kojih se tradicionalne ASP.NET aplikacije mogu postaviti na PaaS ili SaaS okruženje na cloudu. Za tu svrhu upotrebljen je koncept docker kontejnera i mikroservisna arhitektura.

Upotrebom ovih tehnologija, značajno se obezbeđuje stabilan i optimalan rad servisa, kao i mogućnost daljeg rada na razvoju novih funkcionalnosti. Odluka da se software podigne na cloud zahteva veća zalaganja ako su u pitanju mikroservisi, ali obezbeđuje duži i pouzdaniji rad, kao i veću konkurentnost na tržištu.

8. LITERATURA

- [1] Cesar de la Torre, “Modernize existing .NET applicatios with Azure cloud and Windows Containers”, Microsoft Corporation, 2018.
- [2] Cesar de la Torre, “Containerized Docker Application Lifecycle with Microsoft platform and Tools”, Microsoft Corporation, 2018.
- [3] Cesar de la Torre, Bill Wagner, Mike Rousos “ .NET Microservices: Architecture for Containerized .NET Applications”, Microsoft Corporation, 2018.
- [4] <https://msdn.microsoft.com/en-us/library/ms971499.aspx>, datum pristupa septembar 2018.

Kratka biografija:



Miloš Gagović rođen je u Vrbasu 24.2.1994. Diplomirao je na Fakultetu tehničkih nauka na smeru Elektroenergetski softverski inženjering 2017. god. Master rad iz oblasti Elektroenergetski softverski inženjering odbranio je 2018. god. Kontakt: gagovicmilosgagovic@gmail.com

POTENCIJAL KORIŠTENJA GEOTERMALNIH PUMPA NA PODRUČJU GRADA BIJELJINE**POTENTIAL OF THE GEOTHERMAL HEAT PUMPS IN BIJELJINA**Branko Adžić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu predložene su metode eksploatacije geotermalne tople vode na području Bijeljine. Izrađena je tehnološko-ekonomska analiza grejanja poslovnih zgrada sa geotermalnim toplotnim pumpama u Bijeljini. Predložen je način projektovanja toplotnih pumpi za zagrevanje objekata i izračunat udeo električne energije u tome.

Ključne reči: Geotermalna energija, Geotermalne elektrane, Toplotne pumpe

Abstract – This paper proposed ways of exploiting geothermal hot water in Bijeljina. A techno-economic analysis of the heating of business buildings with geothermal heat pumps in Bijeljina was done. The method of designing heat pumps for heating buildings is proposed and electricity share is calculated.

Keywords: Geothermal energy, Geothermal power plants, Geothermal heat pumps

1. UVOD

Geotermalni resursi su široko rasprostranjeni u svim delovima sveta, ali je korišćenje nedovoljno prisutno uprkos ekološkim pogodnostima i dobrim energetske bilansom. Višestruke mogućnosti primene pokazuju da se energetske značaj u proizvodnji toplote i električne energije kontinuirano povećava sa smanjenjem rezervi uglja, nafte i gasa. Upotrebljivost toplotne geotermalne energije se smatra velikom šansom za pripremu potrošne tople vode u domaćinstvima, za distributivni sistem zagrevanja stanova, poslovnih i javnih objekata, a onda i proizvodnih staklenika, plastenika, akvakulturu, sušenje poljoprivrednih proizvoda i mnogih drugih oblasti privređivanja.

U ovom radu će se razmatrati mogućnost primene geotermalne energije za zagrevanje objekata u Bjeljini, ali i udeo električne energije u ovom procesu.

2. GEOTERMALNA ELEKTRANA

Prema proračunima do kojih je došla Evropska Komisija za istraživanje energije, toplota Zemljine unutrašnjosti može obezbediti veoma stabilan i dugotrajan izvor energije. Samo jedan kubni kilometar užarene mase može predati 30 MW električne energije u periodu od 30 god.. Da bi se ta energija iskoristila, razvijene su mnoge tehnologije, ali pojednostavljeno mogu se izdvojiti dve vrste

korištenja geotermalne energije: Direktno, tj. upotreba vruće vode koja izbija (ili se ispumpa) iz podzemlja ili Indirektno korišćenje geotermalne energije, tj. pretvaranje u električnu energiju (geotermalna elektrana).

Geotermalna elektrana u stvari je kao svaka druga termoelektrana, s tim što se para ne proizvodi gorenjem nekog fosilnog goriva, već se crpi iz zemlje. Vrela voda i para iz dubine zemlje može se koristiti za proizvodnju električne energije tako što se izbuše bunari u zemlji i cevi se spuštaju u vrela bazena sa podzemnom vodom. Vrela voda ili para penje se tim cevima na površinu, gde se dodatno obrađuje i pod velikim pritiskom vodi na parnu turbinu. Turbina obrće sinhroni generator i generiše se električne energije, koja se zatim sprovodi u elektroenergetski sistem preko prenosnih transformatora.

3. GEOTERMALNA ENERGIJA NA PODRUČJU GRADA BIJELJINA

Geotermalni resurs Semberije otkriveni su 1957. godine posle izrade istražne bušotine u Dvorovima kada je došlo do erupcije termalne vode sa temperaturom od 75°C. Površina rezervoara procenjuje se na oko 2.000 km², a njegova dubina minimalno 500 m do 2.500 m, u proseku oko 1.000 m. Dobijeni rezultati utvrđeni su izradom više bušotina, od toga pet dubokih bušotina, dubine od 1.280 - 2.500 metara urađeno je u Semberiji.

Dosadašnja primena evidentnih resursa je međutim veoma skromna s obzirom da se u Dvorovima iz bušotine koristi samo 7 l/sek termalnih voda pri čemu se 2/3 njene geotermalne energije baca u kanalizaciju jer nema potrošača koji bi tu energiju iskoristili. Samo za toplifikaciju grada Bijeljine u sadašnjoj veličini, postojeća energija je dovoljna za minimalno narednih 200 godina, pod uslovom da se geotermalni sistem ne obnavlja. Pošto je geotermalni sistem obnovljiv, to je perspektiva geotermalne energije na ovom području velika i imaće značajno mesto u budućem razvoju grada i neposredne okoline.

4. TOPLOTNE PUMPE

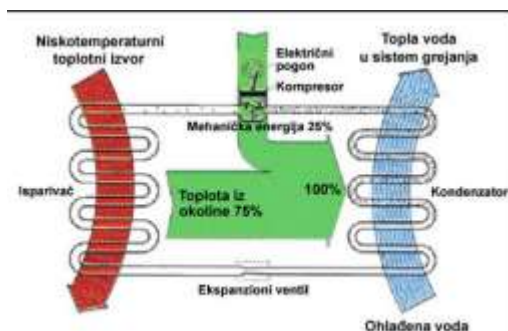
Toplotne pumpe su uređaji pomoću kojih se toplotna energija iz jedne sredine prenosi u drugu sredinu. Za taj prenos toplotne energije takođe je potrebno da se utroši određena energija koja je nekoliko puta manja od prenete energije (slika 1)

4.1. Tipovi toplotnih pumpi

Postoje dva osnovna tipa sistema geotermalnih toplotnih pumpi, GTP s otvorenim i zatvorenim krugom. Ova dva tipa se mogu podeliti na sledeće podsisteme:

NAPOMENA:

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



Slika 1. Odnos uložene električne energije i iskorišćene geotermalne energije [2]

1) Sistem sa zatvorenim krugom:

- vertikalni
- horizontalni
- zatvoreni sistem s površinskom vodom

2) Sistem s otvorenim krugom: dve bušotine (proizvodna i utisna, ili izliv)

4.2 Osnovni delovi toplotne pumpe

U osnovi, po svojim nazivima i funkciji, skoro sve važne komponente toplotnih pumpi istovetne su sa rashladnim elementima. To su:

- kompresori sa pogonskim motorom,
- izmenjivači toplote,
- sudovi pod pritiskom (separatori, odvajači, skupljači rashladnog fluida),
- automatika,
- armature i cevovodi,
- elektrokomandni uređaji,
- rashladni fluid.

4.3 Električna snaga i energija za toplotne pumpe

Za proračun energije dobijene od toplotnih pumpi uzima se obračun potrošnje cirkulacione pumpe kao i samog kompresora toplotne pumpe - SPFH₂ sa slike 2 obeleženo crvenom linijom.

Prema tome, dobija se:

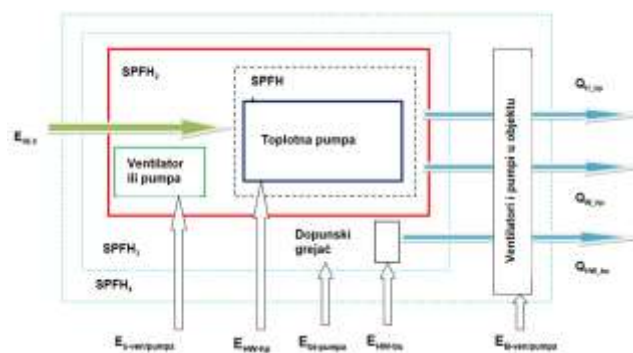
$$E_{RES} = Q_{upo} - E_{s_ven/pumpa} - E_{HW-hp} = Q_{upot} \times (1 - 1/SPF) \quad (1)$$

$$Q_{upo} = Q_{H_hp} + Q_{W_hp} \quad (2)$$

gde je:

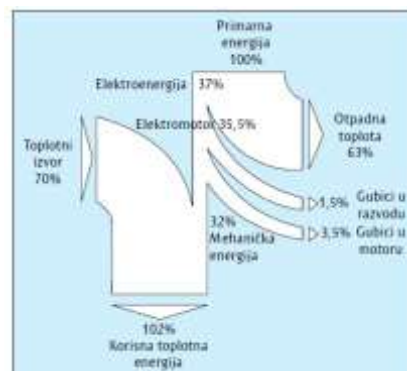
- E_{RES} - obnovljiva energija iz vazduha, zemlje ili vode (toplotnog izvora) preuzeta sa strane toplotne pumpe,
- Q_{upo} - ukupna korisna toplota koju daje toplotna pumpa,
- $E_{s_ven/pumpa}$ - energija upotrebljena za pogon ventilatora ili pumpe za cirkulaciju rashladnog fluida,
- E_{HW-hp} - energija za rad same toplotne pumpe,
- Q_{H_hp} - toplota isporučena (oduzeta) iz izvora toplote preko toplotne pumpe,
- Q_{W_hp} - toplota dobijena iz mehanicke energije koja se koristi za pogon toplotne pumpe

Iz granice sistema proizlazi da proračun obnovljive energije dobijene od toplotne pumpe zavisi od same toplotne pumpe, a ne od sistema za grejanje u čijem sastavu je toplotna pumpa. Neefikasno korišćenje energije toplotne pumpe stoga je pitanje energetske efikasnosti, i zato ne utiče na proračun obnovljive energije isporučene od strane toplotne pumpe.



Slika 2. Energija toplotne pumpe [5]

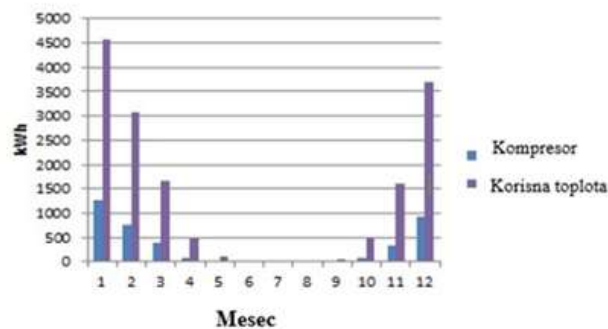
Ako se toplotna pumpa poredi sa drugim izvorima toplote za grejanje, onda njena potrošnja primarne energije, npr. električne energije, iznosi 25% do 30% u odnosu na druge sisteme za grejanje (slika 3). Za pogon toplotne pumpe najčešće se koriste elektromotori. Toplotne pumpe sa elektromotornim pogonom mogu da obezbede temperature vode do 75 °C.



Slika 3. Energetski bilans toplotne pumpe [6]

Kako se povećava temperaturna razlika između ulaza i izlaza toplote, podrazumeva se da je potrošnja kompresora koja je potrebna za njegovo napajanje veća, pa se COP smanjuje. Stoga je važno razumeti temperaturu za koju je predviđena toplotna pumpa i opseg temperature u kojoj će se koristiti toplotna pumpa.

Prema projektu "WP-Effizienz" [4] sprovedenom od 2005. do 2010. godine je merena potrošnja energije i efikasnost toplotnih pumpi u stambenim objektima. Podno grejanje je bilo najčešći tip sistema distribucije toplote. Prosečna potrošnja grejanja u ovim objektima u 2009. godini iznosila je 72 kWh/m² za prosečnu površinu grejanja površine 199 m².



Slika 4. Potrošnja kompresora i dobijena toplotna energija [4]

Studija je ilustrirala dobar primer potrošnje energije toplotne pumpe i energetske efikasnosti. U samoj studiji dobili smo da se na osnovu ispitivanja i merenja za jednu godinu, da za malu ulaznu energiju, koja se troši na pokretanje kompresora, dobijamo dosta veću izlaznu toplotnu energiju kao što je prikazano na slici 4. Efikasnost toplotne pumpe obično se prikazuje tako što za svaki 1 kW električne energije, koji se potroši za pokretanje kompresora dobije preko 3 kW toplotne energije, dajući "efikasnost" od 300 % i više. Termodinamički je nemoguće imati efikasnost veću od 100 %, jer bi to podrazumevalo da se proizvede više energije nego što ulazi u sistem.

5. PROJEKTOVANJE TOPLOTNIH PUMPI

Sa revolucijom informacionih tehnologija došlo je do razvoja raznih softvera za modelovanje i odabir toplotnih pumpi. Preko softvera se može simulirati rad toplotnih pumpi. To znači da su važni radni parametri izračunati na osnovu raznih standarda u projektovanju objekata i toplotnih pumpi prema preporukama od proizvođača. Postoji veliki broj dostupnih softvera za proračun toplotnih pumpi a najpopularniji su :

GeoT*SOL program, HOMER Pro, The Easy RSS Heat Pump Design, Polysun Simulation Software, RETScreen, Heat Pump Pre-Screening Tool, ETU Software GmbH, TRACE 700, Ground Loop Design GLD, GLHEPRO, Geo Hourly

GeoT*SOL program je specijalizovani alat za planiranje i projektovanje sistema toplotnih pumpi. Dinamična simulacija u toku godine daje sledeće parametre za izabrani sistem toplotne pumpe: Godišnji koeficijent performansi za toplotnu pumpu, gde je to pogodno, solarnu frakciju, količinu energije koju generišu toplotna pumpa na godišnjem nivou, godišnju električnu energiju potrebnu za toplotnu pumpu, godišnju generisanu energiju za toplotnu pumpu i solarni ciklus, godišnje gubitke rezervoara i cevi za skladištenje, rezime finansijske analize sa svim parametrima i rezultatima, kao i korisne grafike.

HOMER Pro softver predstavlja globalni standard za optimizaciju dizajna u svim sektorima. HOMER obavlja tri osnovna zadatka: simulaciju, optimizaciju i analizu osjetljivosti. HOMER eksplicitno ne modeluje toplotnu pumpu, ali jednostavno može modelovati i simulirati električno opterećenje na toplotnoj pumpi i dodati to primarnom električnom opterećenju ili ga odrediti kao drugo primarno opterećenje.

The Easy RSS Heat Pump Design je softver za dimenzionisanje i specifikaciju sistema toplotnih pumpi. Easy RSS Heat Pump dizajn softver uključuje:

Određivanje veličine toplotne pumpe, izbor rezervnog izvora, izlaz sertifikata usklađenosti, izračunavanje buke, dimenzioniranje podzemne petlje, izračunavanje troškova, izračunavanje RHI

Polysun Simulation Software je idealan alat za projektovanje solarnih, toplotnih i fotonaponskih sistema, kao i kombinovanih sistema. Softver uključuje proračun ekonomske održivosti, omogućava i na osnovu projektovanog dizajna se dobija proračun ušteda. U

softver su integrisani svetski vremenski podaci na osnovu koji se proračunavaju pouzdane simulacije.

RETScreen softver predstavlja program upravljanja obnovljivim izvorima energije za analizu izvodljivosti energetske efikasnosti kao i tekuće analize energetske performansi. Program, dostupan svima besplatno, može se koristiti širom sveta za procenu proizvodnje energije i ušteda, troškove tokom životnog veka projekata, smanjenje emisija, finansijsku isplativost i rizike za razne tipove tehnologija energetske efikasnosti i tehnologija obnovljive energije.

Heat Pump Pre-Screening Tool je softversko rešenje koje omogućava brzu procenu sistema toplotnih pumpi za određeni skup građevinskih parametara sa minimalnim ulaznim parametrima. HPAC ima za cilj pružanje podrške konsultantima za upravljanje energijom i građevinskim inženjerima u opravdanju sveobuhvatne studije izvodljivosti projekta toplotne pumpe. Program upoređuje opcije sistema za toplotnu pumpu sa predviđenim sistemom HVAC sistema za nove komercijalne i institucionalne građevinske konstrukcije ili glavne naknade. Pruža najbolje opcije na bazi smanjenja emisije staklene bašte.

ETU je softver koji omogućava vremenske simulacije toplotnog ponašanja zgrade i potražnje za grejanjem, na osnovu karakteristika zgrade, profila korišćenja i lokalnih klimatskih uslova. Izračunavanje predviđenih performansi toplotne pumpe. Finansijski obračun uključujući povraćaj investicije. Izmenljive korisničke profile za izračunavanje grejanja i potrebe za toplom vodom. Upoređivanje toplotnih pumpi vazduha/vode, toplotne pumpe vode/vode ili toplotne pumpe zemaljskog izvora/vode.

TRACE 700 je softver za dizajn i analizu koji pomaže osobama HVAC-a da optimizuju dizajn sistema za grejanje, ventilaciju i klimatizaciju na osnovu korišćenja energije i životnog ciklusa troškova. Softver TRACE 700 omogućava da se uporedie energetski i ekonomski uticaj izbora vezanih za gradnju, kao što su arhitektonske karakteristike, HVAC sistemi, HVAC oprema, toplotne pumpe, korišćenje zgrada i finansijske opcije.

Ground Loop Design GLD softver predstavlja integrisan skup moćnih, fleksibilnih i prilagodljivih alata za projektovanje komercijalnih i stambenih zatvorenih petlji za sisteme toplotne pumpe. GLD omogućava dizajneru da brzo i tačno:

Odrede dinamičku mehaničku opremu, izračuna zahteve za bušenje/iskopavanje, odredi dužine cevi, vrste i izgleda, optimizujte konfiguracije cevovoda, obezbedi efikasnu cirkulacionu pumpu, dizajnira optimizovani i stabilan sistem, proceni troškove životnog ciklusa, predvidi emisiju CO2

GLHEPRO se koristi kao pomoć u dizajniranju vertikalnih izmenjivača toplote sa zupčastom bušotinom. Zahteva tri osnovna skupa ulaznih podataka: opterećenje mesečnog grejanja i hlađenja na toplotnoj pumpi, opis izmenjivača toplote sa zateznom petljom - uzorak, radijus bušotine i dubinu bušotine, performanse toplotne pumpe - sa zadanim vrednostima.

Kada se unesu svi ulazni podaci, GLHEPRO će izvršiti simulaciju ili odrediti dubinu bušotine.

Nakon toga je prikazan kratak rezime rezultata. Program sadrži detalje o dizajnu petlje i minimalne i maksimalne temperature fluida koje ulaze u bušotine.

Geo Hourly je softver za stambeni i komercijalni geotermalni dizajn za modelovanje petlje u zemlji, analize troškova energije, hibridni dizajn kao i solarni dizajn. Softver omogućava jednostavnu optimizaciju analitike s kojom možemo dobiti operativne troškove nasuprot dizajnu petlje. Mogu se postaviti razni dizajni na osnovu realnih uslova rada što kao posledicu može smanjiti realnu cijenu sistema.

6. ELEKTRIČNA ENERGIJA I GREJANJE POSLOVNE ZGRADE TOPLOTNOM PUMPOM

Novosagrađena poslovna zgrada u Bijeljini ima ukupnu površinu poslovnog prostora od 2.000 m². Nasuprot konvencionalnom sistemu grejanja gasnim kotlom koji godišnje troši 28.781 m³ prirodnog gasa ili 204 MWh energije grejanja, plan je da se instalira sistem bušotinskih izmenjivača toplote odnosno geotermalna toplotna pumpa. Zgrada će, zbog veće termodinamičke iskoristivosti, imati sistem podnog grejanja (nisko-temperaturno grejanje) te sistem prisilne konvekcije za potrebe hlađenja. Za kompletni tehnokonomski proračun odabran je programski paket RETScreen koji sadrži sveobuhvatnu svetsku bazu podataka proizvođača toplotnih pumpi i klimatoloških parametara [3].

Za klimatološke prilike grada Bijeljine proračunate su veličine stepen dana grejanja (pri nominalnoj vrednosti temperature od 16°C) te na temelju podataka o tipu i termalnim karakteristikama tla, geotermalnom gradijentu, te ukupno potrebnoj instaliranoj snazi za pokrivanje vršne potrošnje u ciklusu grejanja, proračunata je potrebna dubina bušotinskog izmenjivača toplote.

Predloženi sistem grejanja toplotne pumpe s bušotinskim izmenjivačem toplote prema programskom paketu RETScreen [3] ima sledeće karakteristike:

- Potrebna zemljana površina oko zgrade: 613 m²
- Ukupno potrebna dubina izmenjivača toplote: 2.115 m (15 bušotina×141 m)
- Snaga cirkulacijske pumpe: 3,2 kW
- Volumen vode u izmenjivaču toplote: 0,37 m³
- Potreba dužina polietilenskih cevi izmenjivača: 4.231 m
- Cena goriva (električna energija): 51,0 €/MWh
- Ukupni kapacitet grejanja odabranog modela toplotnih pumpi: 129,0 kW (104% pokrivena vršne potrošnje)
- Ukupna ostvarena godišnja energija grejanja: 204 MWh (100 % pokrivena ukupno godišnje potrebne energije)
- Stepenn iskorištenja toplotne pumpe: COP = 5,79
- Proizvođač i model: Viessmann GmbH, Vitocal300 WW232 (3 jedinice) [1]
- Utrošena godišnja el. energija kompresora: 35,2 MWh
- Godišnji trošak električne energije: 1795,2 €

Kada se upoređi isplativost sistema grejanja sa toplotnom pumpom u odnosu na konvencionalni sistem grejanja na prirodni gas dolazi se do informacije da sistem sa toplotnom pumpom ima bolju isplativost. U sistemu sa konvencionalnim sistemom grejanja je za godinu dana potrošeno 204,0 MWh energije dok je preko sistema sa toplotnom pumpom za grejanje potrošeno 35,2 MWh električne energije. Finansijski prikazano za konven-

cionalni sistem imamo godišnju potrošnju od 9.843 € (za cenu gasa 0,342 €/m³) dok za sistem za toplotnom pumpom potrebno 1.795,2 €, što predstavlja veliku uštedu. Ipak, treba napomenuti da sistem troši dosta električne energije za pogon kompresora (35,2 MWh), odnosno u proseku oko 3.000 kWh mesečno, što zahteva posebnu instalaciju, pa ima dodatne troškove (angažovana snaga, posebna tarifa i sl.).

7. ZAKLJUČAK

Toplotne pumpe smatraju se jednim od najučinkovitijih uređaja za dobijanje toplotne energije, a često se upotrebljavaju u bogatijim zemljama s razvijenom visokom ekološkom svesti. Važan podsticaj u razvijenim zemljama je i uređeno zakonodavstvo, pre svega na području obnovljivih izvora energije, te državna podrška za obnovljivim izvorima energije. U BiH takve podrške za toplotne pumpe još ne postoje, a niti zakonodavstvo nije uređeno na tom području.

Cene sistema s toplotnim pumpama u BiH su na žalost još uvek jako visoke. Razloga ima više, a oni najizraženiji su: nedostatak državnih podsticaja za primenu obnovljivih izvora energije, relativno visoke cene uređaja i radova koje su rezultat malog broja proizvođača toplotnih pumpi i edukovanih izvođača radova. Bez obzira na brojne poteškoće i nelogičnosti u sistemu, toplotne pumpe ipak i u BiH doživljavaju sve veću ekspanziju. U razvijenim zemljama Evrope obnovljivi izvori energije zauzimaju značajno mesto u energetskej politici. Za očekivati je da će se i kod nas težiti istom trendu. Do tada, ušteda energije i zaštita životne sredine primenom obnovljivih izvora energije ostaje na nivou razvijenosti naše savesti i na našim finansijskim mogućnostima.

8. LITERATURA

- [1] <https://www.viessmann.rs/>
- [2] R. Bertani, „Geothermal Energy: An Overview on Resources and Potential“, Inter.Conf.on National Development of Geothermal Energy Use, Slovakia, 2009
- [3] RETScreen, <http://www.retscreen.net/>, Canada
- [4] <https://www.ise.fraunhofer.de/en/research-projects/wp-monitor.html>
- [5] <http://kgh-kongres.rs/index.php/en/>
- [6] <https://www.automatika.rs>

Kratka biografija:



Branko Adžić, dipl.ing. rođen je 1987. god. u Sarajevu. Srednju tehničku školu Mihajlo Pupin, završio je u Bijeljini, 2006. god. Fakultet tehničkih nauka, studijski program Energetika, elektronika i telekomunikacije (OAS) upisao je školske 2006/2007. Na studijama se opredelio za modul Elektro-energetika-energetska elektronika i električne mašine i diplomirao 2012. god. Kontakt: adzicbranko@gmail.com

**PROFESIONALNI DETEKTOR BEŽIČNIH SIGNALA (PSD-100) –
PREVENCIJA VISOKOTEHNOLOŠKIH PREVARA****PROFESSIONAL SIGNAL DETECTOR (PSD-100) –
PREVENTING OF HIGH-TECH CHEATING**Bojan Lungulov, Vladimir Rajs, Vladimir Milosavljević, *Fakultet tehničkih nauka, Novi Sad***Oblast - ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu predstavljena je implementacija uređaja za detekciju bežičnih komunikacija. Uređaj meri jačinu elektromagnetnog zračenja u opsegu od 1 Mhz do 10 GHz i ima širok opseg podešavanja osetljivosti. Projektovani uređaj uspešno detektuje aktivnosti prisutne u mobilnoj mreži u sva tri glavna opsega rada, kao i ostale tipove bežične komunikacije kao što su: radio komunikacija, Bluetooth (Blutut), Wi-Fi i druge. Ovaj uređaj predstavlja unapređenu verziju uređaja UD – 100. Rezultati merenja se prikazuju korisniku na grafičkom displeju čime se korisniku omogućava jednostavan i pregledan prikaz rezultata kao i jednostavan i intuitivan način upravljanja uređajem. Uređajem se upravlja uz pomoć četiri tastera koji su osetljivi na dodir, uređaj je baterijski napajan, portabilan i veličine takve da može stati u svaki džep.

Ključne reči: Detektor signala, GSM, WiFi, Bluetooth.

Abstract – In this paper the implementation of the device for wireless communication detection is presented. The system measures RF electromagnetic field strength for frequencies from 1 MHz up to 10 GHz and has wide range of adjustable sensitivity. Implemented device provides detection of cellular phone activities in all three main operating bands, and detection of other types of wireless communication such as radio communications, Bluetooth, Wi-Fi and so on. The device presents an improvement of a previous version of the detector UD-100. Measurement results are displayed on graphical display providing easy user readings and device operating. Device is operated by four touch sensitive keypads, the system is battery powered, portable and can fit in any pocket.

Key words: Signal detector, GSM, WiFi, Bluetooth.

1. INTRODUCTION

With technology development and miniaturization of devices in telecommunications, problem of citizen privacy became more significant.

The problem of mobile phone use on places where their use is forbidden such as exams and prisons is also increasing. A device that is mostly used for this purpose is mobile phone with miniature headphones called “bugs” that communicate with the phone via Bluetooth connection.

NOTE:

This paper arose from master thesis whose mentor was Vladimir Rajs, Ph.D, Assistant Professor.

For citizen privacy, the biggest threat are small devices equipped with a GSM modem that can live transmit audio recording and therefore can violate person's privacy.

This combination without detector, which is proposed here, represents ideal solution for using mobile phone in places where their use is forbidden.

2. METHODS OF HIGH-TECH CHEATING

The problem of cheating by using wireless communications with micro earphones has rapidly increased, as a consequence of price decrease of those systems, their easy use, impossibility to detect them with the naked eye and their great expansion on the market [1]. The price of renting such equipment is about 20-30€ per day, what in fact represents the price of passing an exam. Nowadays, the price of the whole wireless communication system is from 80€ to 300€, so more and more students can afford to buy that system, and later rent it or share with their friends. This caused an expansion of use of such systems for cheating in the school system.

First, it started at universities, and lately it expanded into secondary schools. There is a reasonable risk of spreading this phenomenon in primary schools as well. Cheating on the written exams is the easiest (pass rate 90%), while some sellers claim that there is also great pass rate on the oral exams (over 80%).

Many professors are under the misapprehension that they will prevent smuggling the test out of the room by forbidding students to take the test with them out of the classroom or leaving the room during the exam. There are many ways of smuggling the test out of the exam room [2]:

- Smuggling the exam questions out of the exam room
- Dictating questions in the miniature microphone
- Taking photo of the test by using mobile phone and sending that photo to a colleague outside the building
- Shooting the exam questions by using a camera of a button size or by using a micro camera in a pen (Figure 1).



Figure 1. Micro camera in a pen

Using the system for cheating is quite easy (Figure 2):

- It is necessary to connect a receiver (in most cases it is a mobile phone) to an adapter with antenna (which is put under the clothes, most often around the neck).
- There is a wireless communication between the antenna and the invisible micro earphone placed in the ear. Nothing can be noticed from the outside.
- The colleague on the other side of the phone line dictates the answers, and these can be only heard by the person with the micro earphone.



Figure 2. System for cheating on exams

Another problem is using devices for electronic surveillance. Miniature device with GSM modem embedded in them and equipped with a microphone that are battery powered can now be bought almost everywhere. Those small devices can be used for illegal surveillance, secretly eavesdropping certain meetings or even eavesdrop on your friends, wife and so on. Those devices are having SIM card in it and simply by calling that mobile number person who is calling can listen everything that is happening in that room where device is placed without anyone knowing that. Because those devices are working via cell network it is easy to detect them with this kind of detector. Those listeners can be hidden on number of places such as power extension cords, key chains, power outlets, lamps and others. Places like power outlet are providing power so device can work non-stop but it is not portable, and device that is like key chain is easily portable but is battery operated so use of this kind of device is limited by battery life.

3. PROJECT REQUIREMENTS AND HARDWARE IMPLEMENTATION

Mobile phones today operate on three frequencies – 900 MHz, 1800 MHz and 2100 MHz, but it is still possible to detect activities of such devices, despite their small dimensions and relatively low radiation. There are certain requirements that the device for detection of mobile phone activities should satisfy. First of all, such device should be light weight and small in dimensions so it can be handheld. Another requirement that must be satisfied is ease of use. Device must be simple to use because it is intended for wide population of people who work in various professions. The visual representation of the information about detected signal level should be provided in the simplest form. In this implementation, the visual representation is given through a number of light

emitting diodes or LEDs and also on graphical display. The diodes flash when the device detects RF signal. The number of flashing LEDs indicates the signal strength. Also detected signal strength is displayed with a bar graph on graphical display, so user can choose way the detected signal level is displayed, only LEDs, only bar graph on the display or both. On the display there are separate bars graphs for displaying 2G and 3G signal strength.

The goal of this project was to make a small portable device that can detect transmitted RF waves in wide frequency range. Range from 100 MHz up to 2500 MHz is accepted as adequate because it covers FM range, both GSM and 3G ranges, as well as 433 MHz and 2400 MHz bands, which are used by great number of wireless devices. Also there are Wi-fi modems that operate at 5GHz frequency that also can be detected as chip used for detection can detect signals up to 10GHz.

Photograph of PSD-100 Professional Signal Detector is displayed on Figure 3. On front panel LED bar is placed, providing to operator clear and fast reading of current detection signal strength. Under the LED bar there is graphical display with resolution of 84x48 pixels which is sufficient for displaying detected signal strength and other information such as menu for settings and battery level indicating, connectivity indicating and other things. Under display there are four touch sensitive keypads (UP, DOWN, BACK and OK) that are used for moving through menu.

On the top of device there is connector for SMA antenna which is used for improving signal detection. On the bottom there are reset switch and micro USB connector for charging device and for connection with a PC.

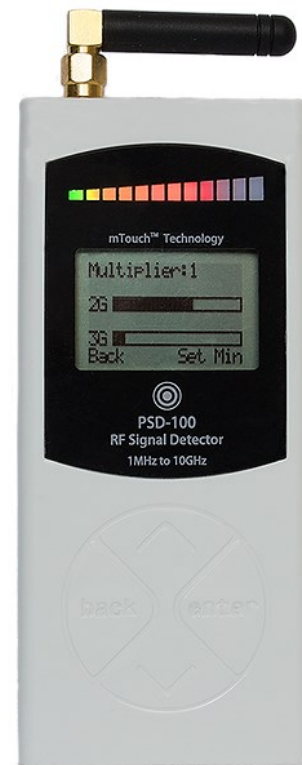


Figure 3. PSD – 100

Whole device is battery powered via Lithium-ion battery which enables long working time and easy charging, device can be easily charged from any mobile phone

charger which is connected to power outlet or from PC or laptop or external battery pack used for charging mobile phones.

4. SIGNAL PROCESSING BLOCKS INSIDE DETECTOR

PSD-100 can be separated into several functional blocks that are displayed on Figure 4.

MCU module is Microchip controller PIC16F707.

Power block consists of battery and chip which regulates charging of battery also there is switch that is controlled from MCU that cuts power to other modules when they are not needed in order to save power.

Boost module consists of DC-DC boost converter and it is used for generating voltage for display. Logic circuits on the display are using 3,7V from battery, but crystal on the display needs 7V or more to function properly. External boost is used because using build-in booster on the display was not reliable.

External memory is used to store images that are displayed on display and for remembering user settings in case that battery fully discharges. EEPROM used is AT25256AN from Atmel, size of memory is 256kB.

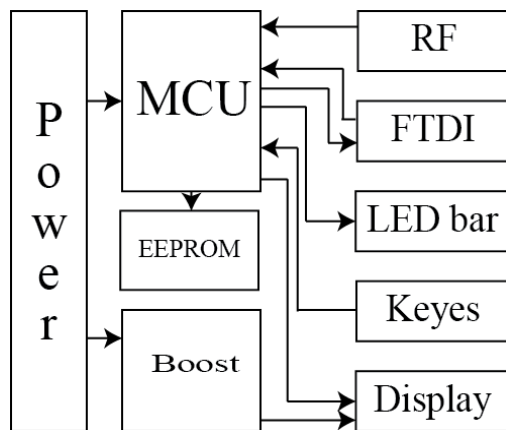


Figure 4. Block diagram of PSD-100

RF module consists of Analog Devices AD8317 chip which is used for detecting RF signals. This module will be explained in greater detail later.

FTDI module consists of FT232RL chip which is used for communicating with PC. Device can communicate with PC in order to save history of detection or to monitor current signal level on computer or even via internet from remote location.

Led bar consist of 12 light emitting diodes. First diode is green and represents that device is powered on and if that diode is only lit than there is no relevant RF signal detection. Other 11 diodes are mixed yellow and red and are placed in a way that they create easy way of reading detected signal level.

Keys represent four touch sensitive touch pads which are used for operating device. MCU uses Microchip M-touch technology.

Display is 84x48 pixel graphical display which is used for displaying relevant information to user. Display is having backlight so user can clearly see information on the display in low light situations.

5. POWER SUPPLY AND RF BLOCK

Block diagram of power supply is given in Figure 5.

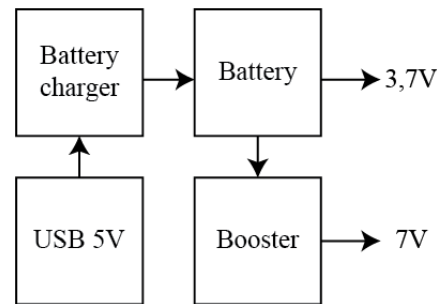


Figure 5. Power supply block diagram

Micro USB connector is used for charging device and also for communicating with PC. Battery management chip is MCP73831 from Microchip, that chip is linear charge management controller, it includes integrated pass transistor, integrated current sensing, and reverse discharge protection. Chip is controlling battery charging and protects battery from short circuit, over and under voltage protection. Chip also has dedicated pin that signals charging state, that pin is connected to MCU so information about charging is presented to user on display.

Battery is Lithium-ion with nominal voltage of 3,7V. Battery capacity is 2000mA/h which is enough for device to stay on for couple of days.

For generating 7V voltage for display chip ADP1612 from Analog Devices is used. ADP1612 is step-up dc-to-dc switching converter with an integrated power switch capable of providing an output voltage as high as 20 V. With a package height of less than 1.1 mm, the ADP1612 is optimal for space-constrained applications such as portable devices or thin film transistor (TFT) liquid crystal displays (LCDs).

Block diagram of RF module is given on Figure 6.

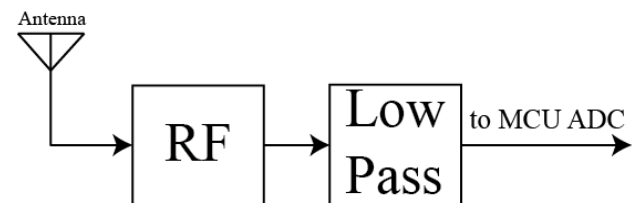


Figure 6. RF module block diagram

On top side of device is SMA connector for antenna, user can choose from wide variety of available antennas on the market in order to increase sensitivity in certain range. For instance if someone wants to detect GSM (2G) network then they can use antenna that is optimized for use in that range and gives better performance in that particular range. That is also case for using optimized antenna for detection of 3G signals or for Wi-Fi and so on.

Chip used for detection is AD8317 from Analog Devices. The AD8317 is a demodulating logarithmic amplifier, capable of accurately converting an RF input signal to a corresponding decibel-scaled output. It employs the progressive compression technique over a cascaded amplifier chain, each stage of which is equipped with a detector cell. The AD8317 maintains accurate log

conformance for signals of 1 MHz to 8 GHz and provides useful operation to 10 GHz. The input dynamic range is typically 50 dB with error less than ± 1 dB. The AD8317 has 8 ns/10 ns response time (fall time/rise time) that enables RF burst detection to a pulse rate of beyond 50 MHz.

Signal from AD8317 is filtered with low pass filter in order to reduce noise and to get useful signal information. After filter signal is taken to 8 bit analog digital converter of MCU.

It is possible to simultaneously detect 2G and 3G signals because of their different way of functioning, 2G signal create peaks of max value which corresponds to detected signal level and in that case mean value stays almost the same. On the other hand 3G signal is opposite of 2G signal because detected signal level affects mean value not max value, so if 3G signal is detected max value stays almost the same, and mean value increases according to detected signal level. Algorithm for calculating this is implemented in MCU. This way of functioning enables more control to end user because user can fine tune certain settings in order to get better results in range and type of signal that he intends to detect.

6. THE USE OF WIRELESS COMMUNICATION DETECTOR PSD – 100

The device is primarily intended for using in the classrooms and amphitheatres in order to detect active mobile phones as the sources of a high-frequency magnetic field. In the following part the use of detector will be described under the assumption that we are in some classroom and that we are looking for an active mobile phone. When we enter the classroom, we put the device on the table and we turn on device. When we see that the green diode is turned on that is the sign that the device is working. On the basis of length of the classroom we adjust the device sensitivity. The device sensitivity is adjusted when device is in detection mode with UP and DOWN touch pads. It is hard to detect correct distance that corresponds to measured signal level because the radiation of a mobile phone depends on its distance from the base station. More it is distant from the base station, more it radiates.

The question arises: "How do we come exactly to the location of the mobile phone?"

Since all diodes are turned on that is the sign that one of the mobile phones in the classroom is active. Then we lower the detector sensitivity. After that, we take the device in our hand and walk across the classroom. As we approach to an active mobile phone more diodes turn on. When we come to an active mobile phone all diodes will turn on and in this way we know the location of the searched mobile phone. In the given description the assumption was that the mobile phone was searched for in the classroom, and of course the device can be used in other places for detection of other sources of high-frequency electromagnetic radiation. This method of detecting can also be used for checking is there a secret listener device in conference room for instance. If there is classified meeting going on and if we want to be sure that nobody is listening.

The answer on the question why it was decided to make a detector, and not a jammer or some similar device is following:

- Using jammer is not legal, because it is forbidden to block public frequencies
- Jammers of great power have a bad influence on human organism because of their high level of radiation
- Jammers are not reliable for many kinds of wireless communications which are not permitted (they only block narrow range of frequencies)
- Jammers have big dimensions and their transport is not simple in comparison to detector of this kind
- Jammers are much more expensive

7. CONCLUSION

In spite of PSD-100 low price and relatively simple design it represents reliable and flexible system with possibility of further development. High measurement sensitivity enables monitoring of bigger rooms such as amphitheater by using only the one detector PSD-100. Device has been tested and shown good results. In city conditions we have detected mobile phone activity on distance of 20 m and with optical visibility up to 45 m which is enough for one conference room or amphitheater.

8. REFERENCES

- [1] S. Campbell, "Perceptions of mobile phones in college classrooms: ringing, cheating, and classroom policies," *Communication Education*, vol. 55(3), pp. 280-294, 2006.
- [2] V. Rajs, M. Olujic, M. Slankamenac, V. Milosavljevic, M. Živanov, "Problem of High – Tech cheating with bugs," *KSI Transactions on KNOWLEDGE SOCIETY*, Vol. 3, pp. 172- 174, 2008.
- [3] V. Milosavljevic, V. Rajs, M. Slankamenac, M. Živanov, "Hardware system MD-100 for cellular phones activity and other HF electromagnetic field sources," *INFOTEH 2007*, Vol. 6, Ref. B-13, pp. 112-113, March 2007.
- [4] V. Milosavljevic, V. Rajs, M. Slankamenac, M. Živanov, "A system for detection a cellular phones activities and other HF electromagnetic field sources," *Research people and actual tasks on multidisciplinary sciences*, Lozenec, Bulgaria, Vol 2, pp. 291- 293, June 2007.
- [5] V. Rajs, V. Milosavljevic, M. Slankamenac, M. Živanov, "Upgraded version of the detector for cellular phones activity," *INFOTEH 2008*, Jahorina, pp. 195- 197, March, 2008.

KORELACIJA FIZIČKOG I MATEMATIČKOG MODELA PRENOSNE MREŽE

CORRELATION BETWEEN PHYSICAL AND MATHEMATICAL MODEL OF THE TRANSMISSION NETWORK

Bojan Kuzmić, Vladan Krsman, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu su prikazane aproksimacije koje se vrše nad prenosnom mrežom, njenom fizičkom modelu pri čemu se ne gube suštinske osobine, a sve u cilju povećanja efikasnosti izvršavanja matematičkih proračuna. Prikazani su konkretni praktični primeri koji u potpunosti opisuju važnost aproksimiranja. Takođe, navedene su i određene validacije koje je potrebno uvesti da bi se mogao kreirati što kvalitetniji matematički model prenosne mreže.

Ključne reči: Fizički model prenosne mreže, Matematički model prenosne mreže, Aproksimacije nad prenosnom mrežom, Korelacija podataka

Abstract – This paper presents the approximations that are performed over the transmission network, its physical model, without the essential features being avoided, all in order to increase the efficiency of the execution of mathematical calculations. Specific real-life examples are presented that fully describe the importance of approximating. Also, certain validations have to be introduced in order to create the highest quality mathematical model of the transmission network.

Keywords: Physical model of the transmission network, Mathematical model of the transmission network, Approximation of the transmission network, Data correlation

1. UVOD

Elektroenergetika kao jedna od podoblasti elektrotehnike se bave proizvodnjom, prenosom i distribucijom električne energije. Oblast koja se bavi elektroenergetskim sistemima se dalje grana u dva osnovna pravca. S jedne strane, velika pažnja se poklanja izučavanju stacionarnih (ili statičkih) pojava u sistemu, što podrazumeva neizbežnu upotrebu modela zasnovanih na algebarskim jednačinama.

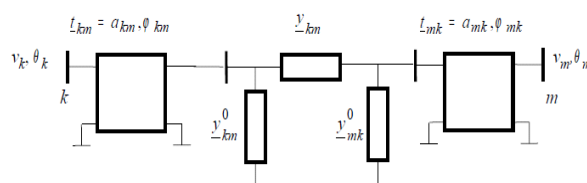
Ovde se proračunavaju moduli i fazni uglovi fazora napona u čvorovima, kao i druge sistemske promenljive. Potrebne su razne aproksimacije kojima se fizički model uprošćuje da bi se dobio što tačniji i lakši matematički model.

NAPOMENA:

Ovaj rad je proistekao iz master rada čiji mentor je dr Vladan Krsman, docent.

2. OSNOVE MATEMATIČKOG MODELA PRENOSNE MREŽE

Matematički modeli elemenata elektroenergetskog sistema, koji se deli na proizvodnju, prenos i distribuciju, definisani su za opštu (fiktivnu) granu („unified branch model“). Statički model opšteg nesimetričnog π – elementa prikazan je na slici 1. Šema praktično važi za sve moguće modele elemenata.



Slika 2.1 – Statički model opšteg nesimetričnog π – elementa

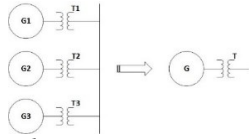
v_k, θ_k – moduo i fazni stav fazora napona u čvoru k ,
 v_m, θ_m – moduo i fazni stav fazora napona u čvoru m ,
 t_{km}, t_{mk} – kompleksni regulacioni prenosni odnos sa strane k -tog, odnosno m -tog čvora,
 a_{km}, a_{mk} – realni (nenominalni) deo prenosnog odnosa sa strane k -tog, odnosno m -tog čvora,
 ϕ_{km}, ϕ_{mk} – fazni pomeraji sa strane k -tog, odnosno m -tog čvora,
 y_{km} – admitansa između tačaka k i m ,
 y_{km}^0, y_{mk}^0 – otačna admitansa sa strane k -tog, odnosno m -tog čvora.

3. KOMPARATIVNA ANALIZA MATEMATIČKOG I FIZIČKOG MODELA PRENOSNE MREŽE

Modeli prenosnih vodova, tj. aproksimacije prenosnih vodova su klasifikovane po njihovoj dužini. Ove klasifikacije su:

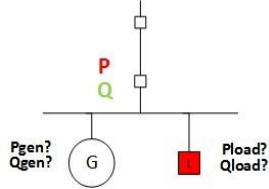
- *aproksimacija kratkog voda* za sve vodove koji su kraći od 80 km,
- *aproksimacija voda srednje dužine* za sve vodove čija je dužina između 80 i 250 km,
- *aproksimacija dugih vodova* za sve vodove koji su duži od 250 km.

Aproksimacija proizvodnje je predstavljena u nekoliko slučajeva koji će biti prikazani ispod. Kada se posmatra elektrana kao izvor, koja sadrži određeni broj proizvodnih jedinica, tada se vrši aproksimacija jednom proizvodnom jedinicom.



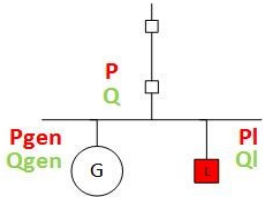
Slika 3.1 – Aproksimacija više proizvodnih jedinica

Na lici 3.2 prikazan je sličan slučaj kada postoje merenja na bližem prekidaču, samo što u ovom slučaju nije zanemarljiva dužina voda.



Slika 3.2. Specifičan slučaj kada postoje merenja samo na bližem prekidaču

Jedna od varijanti je da se zanemari opterećenje i da se samo duplira merenja sa prekidača i pridruži generatoru sa inverznim signalom. U još boljem slučaju ako znamo krive potrošnje, proizvodnju snage generatora, može se aproksimirati snaga koja se detektuje kao opterećenje generatora (na slici označeno sa L). Snaga opterećenja generatora obično predstavlja od 1 do 6% vrednosti snage generatora (u zavisnosti koja je vrsta generatora). Na slici 6 je predstavljena situacija.



Slika 3.3. Računanje sopstvenog opterećenja generatora na osnovu njegove snage

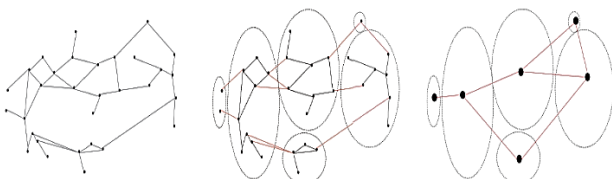
$$\begin{aligned} P &= P_{gen} - P_l, (a) \\ P &= P_{gen} - 0.06 \cdot P_{gen}, (b) \\ P_{gen} &= P/0.94. (c) \end{aligned} \quad (1)$$

$$\begin{aligned} Q &= Q_{gen} - Q_l, (a) \\ Q &= Q_{gen} - 0.06 \cdot Q_{gen}, (b) \\ Q_{gen} &= Q/0.94. (c) \end{aligned} \quad (2)$$

Ovo je dobar način za povećanje redundanse bez stvarnih merenja.

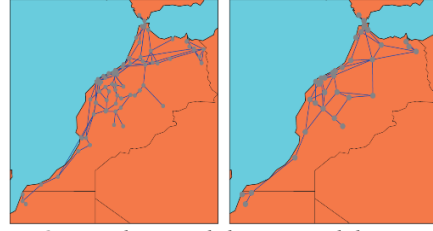
Dodatni vid aproksimacije prenosne mreže može se dobiti koristeći **Kron**-ovu redukciju uvažavajući limite vodova, koja je široko korišćena metoda za pojednostavljenje originalnog modela. Zasniva se na sukcesivnoj eliminaciji čvorova.

Primer jedne konkretne redukcije dat je na slici 3.4.



Slika 3.4 – Primer redukcije jednog prenosnog sistema

Na slici 3.5 je predstavljen redukovani sistem prenosne mreže Maroka (200 kV i više). Originalan model sadrži 857 čvorova (leva strana), dok redukovani sadrži samo 40 (desna strana).



Slika 3.5 – Originalan i redukovani model prenosne mreže Maroka

4. STATIČKA ESTIMACIJA STANJA

Izraz vektora promenljivih stanja su:

$$\mathbf{x} = [V \quad \theta \quad a_i \quad \phi_i]^T \quad (4.1)$$

Vektor merenja ($\mathbf{z}$) povezan je sa vektorom promenljivih stanja ($\mathbf{x}$) preko nelinearne jednačine [3]:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}, \quad (4.2)$$

gde je:

$\mathbf{e}$ – M-dimenzioni slučajni vektor grešaka merenja.

Merenja koja se koriste u proračunu statičke estimacije stanja su:

$$\mathbf{z} = [P_i^T \quad Q_i^T \quad I_i^T \quad P^T \quad Q^T \quad I^T \quad V^T \quad a_i^T \quad \phi_i^T]^T, (4.3)$$

Sva merenja se mogu podeliti na telemetrisana merenja, pseudo-merenja i virtuelna merenja.

Metod minimuma sume otežanih kvadrata odstupanja - WLS (Weighted Least Squares) metod, je najčešće korišćeni algoritam u statičkoj estimaciji stanja.

Ovako definisani problem u matematičkoj formi može se izraziti kao:

$$\min_{\mathbf{x}} \{J(\mathbf{x}) = [\mathbf{z} - \mathbf{h}(\mathbf{x})]^T \mathbf{R}^{-1} [\mathbf{z} - \mathbf{h}(\mathbf{x})]\}, \quad (4.4)$$

ukupna suma kvadrata reziduala merenja koja se minimiziraju.

Struktura jednačine tokova snaga omogućava da se u cilju verifikacije validnosti većine podataka sprovede niz testova konzistentnosti. Primeri ovih testova za razne tipove merenja korišćena u ovom radu su sledeći:

- Za merenje injektiranja i tokova snaga na istom i/ili na susednom povezanom čvoru,
- Za merenja injektiranja i napona i struja na jednom kraju grane (validnost merenja na krajevima grana),
- Za tokove snaga merenje na oba kraja grana (validnost merenja na krajevima grana)

Postoji 8 mogućih podataka o analognim merenjima (sa SCADA sistema u realnom merenju). Testiranje će se vršiti tako da postoje 3 od pomenutih 8. Broj mogućih kombinacija merenja je 56. U okviru poglavlja Primena je na konkretnom primeru mreže pokazano je validiranje po datim stavkama.

5. PREDLOG NOVIH PRAVILA ZA VERIFIKACIJU KVALITETA MATEMATIČKOG MODELA PRENOSNE MREŽE

U svrhu detekcije loših podataka i poboljšanja proračuna u prenosnoj mreži, moguće je uraditi određene provere fizičkog modela na osnovu kojih će se izvršiti modifikacija samog matematičkog modela prenosne mreže. U tekstu ispod će biti predstavljene neke od njih.

- **Granica između prenosnog i distributivnog dela mreže**

Na graničnim čvorovima treba da postoji „granični element“, koji u fizičkom modelu ne postoji, nego samo modelski služi za beleženje napona i faznih stavova od kojih počinju proračuni distributivne mreže. Prenosni deo mreže „vidi“ granične tačke kao PQ čvorove tako da se u proračunima u prenosnom delu koriste vrednosti za P i Q dobijene iz distributivnog dela mreže dok se u proračunu u distributivnom delu koriste vrednosti za V i θ dobijene iz proračuna za prenosni deo mreže.

- **Transformatori sa istom nominalnom snagom na primaru, sekundaru i tercijaru**

Slučaj opisan u naslovu ovog paragrafa definitivno može da pravi problem u proračunima i tačnošću rezultata. Može da se desi da importom podataka stignu ovakve vrednosti u model, pa je potrebno da postoji određeni vid validacije za ovakvo stanje, jer je fizički nemoguće da transformator ima iste nominalne snage (S_{n1} , S_{n2} , S_{n3}) na različitim namotajima.

- **Detekcija transformatora sa očigledno lošim podacima oglada kratkog spoja i praznog hoda**

Vrednosti za napon kratkog spoja dvonamotajnog transformatora u_k [%], struja praznog hoda i_o [%], gubici u bakru P_{Cu} [J] i gubici u gvožđu P_{Fe} [J] su osnovni podaci koje obično dostavlja proizvođač. Pomoću njih se određuju vrednosti za Z i R. Ukoliko vredi izraz:

$$|Z| < R, \quad (5.1)$$

potrebno je detektovati na koji se transformator odnose dati početni podaci. Takođe, ista priča se odnosi i na parametre Y i G. Ukoliko vredi izraz:

$$|Y| < G, \quad (5.2)$$

potrebno je detektovati na koji se transformator odnose dati početni podaci.

- **Poravnatost broje otecepa na namotaju transformatora i položaja regulacione sklopke na njemu**

Postoje dve moguće opcije neslaganja broja otecepa na namotaju transformatora i položaja regulacione sklopke na njemu:

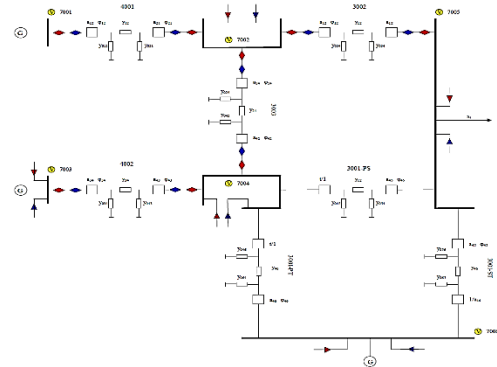
- $RS > T$

- $RS < T$.

U prvom slučaju broj otecepa na namotaju transformatora (T) je manji od pozicije regulacione sklopke (RS), tako da postoje neupotrebljivi položaji regulacione sklopke na njemu. U drugom slučaju je obrnuta situacija, tako da postoje „mrtvi“ otecepi na namotaju transformatora.

6. PRIMENA

Testiranje merenja će se vršiti na zadatom jednostavnom elektroenergetskom sistemu (test mreži) čija je šema prikazana na slici 6.1.



Slika 6.1. Izgled test mreže sa svim parametrima

Legenda:

- ▶ - oznaka da u odgovarajućem čvoru postoji merenje injektirane aktivne snage (P_{inj}),
- ▶ - oznaka da u odgovarajućem čvoru postoji merenje injektirane reaktivne snage (Q_{inj}),
- ◆ - oznaka da po odgovarajućoj grani postoji merenje toka aktivne snage (P_{line}),
- ◆ - oznaka da po odgovarajućoj grani postoji merenje toka reaktivne snage (Q_{line}),
- - oznaka da u odgovarajućem čvoru postoji merenje napona (V).

Kada se može izvršiti samo jedan test

Neka je u skupu merenja izvršena simulacija lošeg merenja napona, gde je umesto izmerenog napona u čvoru sa šifrom 7001 (čvor broj 1), sa vrednošću od 240kV, simulirano loše merenje u vrednosti od 200kV.

TEST 1

U ovom slučaju izmerena vrednost napona je manja od donje dozvoljene termičke granice i testiranje se završava, odnosno merenje napona u čvoru sa šifrom 7001 je označeno kao pogrešno.

$$V_{mereno} \leq V_d \quad (6.1)$$

tj. Nije zadovoljen uslov da se napon nalazi u unapred dozvoljenim granicama.

$$V_d \leq V_{mereno} < V_g \quad (6.2)$$

Kada se mogu izvršiti dva testa

Ako bi se pretpostavilo da je u skupu merenja izvršena simulacija lošeg merenja napona, gde je umesto

izmerenog napona u čvoru sa šifrom 7003 (čvor broj 3), sa vrednošću od 420kV simulirano loše merenje u vrednosti od 430kV.

TEST 1

Test je negativan jer je zadovoljena jednačina $V_d \leq V_{mereno} < V_g$ tj. simulirani izmereni napon se nalazi u okviru tehničkih granica i testiranje može da se nastavi dalje.

TEST 2

Test u ovom slučaju nije moguće izvršiti jer ne postoji merenje injektirane struje i injektirane aktivne i reaktivne snage u čvoru pod šifrom 7003.

TEST 3

Izvršava se provera stanja dela mreže između čvora sa šifrom 7003 i čvora sa šifrom 7004, akcenat je na grani 4002 između pomenutih čvorova. Pod pretpostavkom da imamo merenja V_{7003} , P_{7003} i Q_{7003} , pozivajući se na kombinaciju 1, dobiće se stanje dela mreže na grani pod šifrom 4002. Onda se vrši poređenje promenljivih koje su dobijene proračunom iz kombinacije 1 na osnovu poznavanja merenja V_{7001} , P_{7001} , Q_{7001} , i to P_{7002} , Q_{7002} , V_{7002} sa merenim vrednostima datih promenljivih. Test je pozitivan, tj. zaključak je da je u skupu merenja V_{7003} , P_{7003} i Q_{7003} neko od merenja sumnjivo.

Kada se mogu izvršiti sva tri testa

Pod pretpostavkom da je u skupu merenja izvršena simulacija lošeg merenja napona, gde je umesto izmerenog napona u čvoru sa šifrom 7002 (čvor broj 2), sa vrednošću od 235kV, simulirano loše merenje u vrednosti od 220kV.

TEST 1

Test je negativan jer je zadovoljena jednačina $V_d \leq V_{mereno} \leq V_g$, tj. simulirani izmereni napon se nalazi u okviru tehničkih granica i testiranje može da se nastavi dalje.

TEST 2

Test u ovom slučaju moguće je izvršiti jer postoje sva potrebna merenja potrebna za izvršenje ovog testa. Pod pretpostavkom da postoje merenja na osnovu kojih se izračuna injektirana struja u čvoru pod šifrom 7003 i poredi se ta vrednost sa stvarnim merenjem injektirane struje I_{7002} .

$$I_{izračunato} = \frac{\sqrt{P_{7002}^2 + Q_{7002}^2}}{V_{7002}}; \quad (6.3)$$

Dobije se razlika između prave merene vrednosti I_{7003} i izračunate vrednosti. Test je pozitivan, pa se dolazi do zaključka da u skupu merenja V_{7001} , P_{7001} , Q_{7001} postoji neko sumnjivo merenje.

TEST 3

Izvršava se provera stanja dela mreže između čvora sa šifrom 7002 i čvora sa šifrom 7004, odnosno akcenat se stavlja na granu 3003 između njih. Pod pretpostavkom da postoje merenja V_{7003} , P_{7003} i Q_{7003} , pozivajući se na kombinaciju 1, dobiće se stanje dela mreže na grani pod šifrom 3003. Nakon svega, vrši se poređenje promenljivih koje su dobijene proračunom iz kombinacije 1 i na osnovu poznavanja merenja V_{7003} , P_{7003} , Q_{7003} , i to P_{7002} , Q_{7002} , I_{7003} sa merenim vrednostima datih promenljivih. Test je pozitivan, pa se dolazi do zaključka da je u skupu merenja V_{7001} , P_{7001} , Q_{7001} neko od merenja sumnjivo.

7. ZAKLJUČAK

U radu je objašnjeno koliko su bitne aproksimacije u modelovanju prenosne mreže. U primeni raznih aproksimacija koje su objašnjene ponaosob jasno se može videti koliko one same pojednostavljaju i ubrzavaju proračun.

Pri svemu tome, model se uopštava ali ne gubi na svojoj suštini. Poseban akcenat je dat i validacijama koje je potrebno uvesti da bi se kreirao što kvalitetniji model. Zaključak je da najveći problem u kreiranju kvalitetnog modela mreže predstavlja kvalitet podataka koji se dobija od klijenta.

8. LITERATURA

- [1] Vladimir C. Strezoski: Osnovi elektroenergetike, FTN Novi Sad, 1996.
- [2] Harald G. Svendsen: *Grid model reduction for large scale renewable energy integration analyses*, SINTEF Energy Research, Trondheim, 2015.
- [3] A. Monticelli, „State estimation in electric power systems, a generalized approach“, University of Campinas, Norwell, 1999.

Kratka biografija:

Bojan Kuzmić je rođen u Doboju 1989. godine. Master rad je odbranio 2018. godine na Fakultetu tehničkih nauka iz Elektrotehnike i računarstva – Elektroenergetski sistemi.

Vladan Krsman rođen je u Sarajevu 1985. godine. Doktorsku disertaciju odbranio je 2017. godine na Fakultetu tehničkih nauka iz oblasti Elektroenergetskih sistema.

ODREĐIVANJE POLOŽAJA PREKIDAČA ZA 24h NAKON REKONFIGURACIJE MREŽE**DETERMINATION OF THE SWITCH POSITION FOR 24 HOURS AFTER NETWORK RECONFIGURATION**

Nikolina Modraković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu razmatrana je trošak/dobit analiza optimalne rekonfiguracije mreže, na test distributivnoj mreži koja se sastoji od 25 čvorova, u toku 24 sata. Algoritam koji se koristio pri proračunu je algoritam “izmene grana”.

Ključne reči: rekonfiguracija mreže, tokovi snaga.

Abstract – This paper presents Branch Exchange algorithm for optimal network reconfiguration of distribution network systems. Verification of their possibilities is realized on the test case of a distribution network consisting of 25 nodes in period of 24 hours.

Key words: network reconfiguration, power flow.

1. UVOD

Osnovni deo DMS-a čine DMS funkcije, odnosno energetski proračuni za nadzor, analizu i optimizaciju pogona distributivne mreže [1,2]. Rekonfiguracija je jedna od najvažnijih DMS funkcija, a koja se koristi za optimizaciju rada distributivnih mreža. Cilj rekonfiguracije distributivnih mreža jeste postizanje njihove optimalne konfiguracije. Primenom ove funkcije značajno mogu da se smanje tehnički gubici aktivne snage. U radu je opisana sama rekonfiguracija mreže, proračuni tokova snaga, primer proračuna rekonfiguracije mreže za 24h, kao i trošak/dobit analiza.

2. DISTRIBUTIVNE MREŽE

Njih možemo definisati kao ‘sponu’ između prenosne mreže i konkretnog potrošača, tj. predstavlja deo mreže koji raspodeljuje električnu energiju konkretnim potrošačima. Gledajući po nivoima transporta električne energije, distributivna mreža predstavlja poslednji stepen u isporuci. Distributivne mreže su uglavnom radijalne, sa malim brojem izuzetaka.

3. PRORAČUN TOKOVA SNAGA

Proračun tokova snaga predstavlja jednu od najznačajnijih i najšire korišćenih energetskih funkcija u upravljanju distributivnim mrežama. Proračuni tokova snaga sami za sebe retko se koriste, ali su zato osnovni model uključeni u matematičke modele za rešavanje gotovo svih ostalih

problema na koje se može naići u analizi, upravljanju i planiranju elektroenergetskih sistema [3].

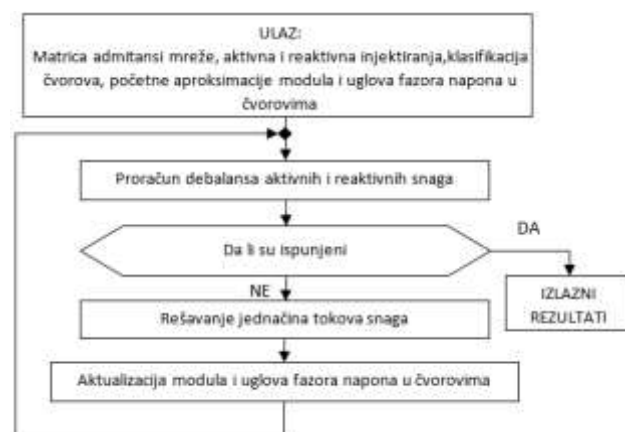
Jedan od algoritama za proračun tokova snaga orijentisan ka granama jeste Shirmohammadi-ev algoritam [4]. Ovaj algoritam predstavlja iterativni postupak za proračun tokova snaga u radijalnim distributivnim mrežama.

3.1 Algoritam sumiranja struja

Algoritam počinje sa inicijalizacijom postupka, odnosno učitavanjem podataka o distributivnom mreži, numeraciji grana i čvorova po lejerima i setovanja indeksa iteracija na početnu vrednost $h=1$. Nakon same inicijalizacije započinje iterativni postupak. Svaka iteracija sastoji se od tri koraka: 1) proračun injektiranih struja za svaki čvor; 2) proračun struja po granama („zamena unazad“); proračun napona u čvorovima („zamena unapred“). Nakon toga, vrši se proračun debalansa aktivnih i reaktivnih snaga i ispitju se uslovi konvergencije. Ukoliko su uslovi konvergencije ispunjeni, iterativni postupak se završava. U suprotnom, prelazi se na sledeću iteraciju.

3.2 Newton/Raphson-ov iterativni postupak

Newton/Raphson-ov postupak sastoji se od linearizacije sistema nelinearnih jednačina koje se rešavaju oko unapred izabrane tekuće (h -te) aproksimacije njihovog rešenja [5]. Tako dobijen sistem linearnih jednačina rešava se po korekcijama tekuće aproksimacije rešenja. Osnovni blok dijagram Newton/Raphson-ovog postupka, prikazan je na slici 3.2.



Slika 3.2 – Osnovni blok dijagram Newton/Raphson-ovog algoritma [6]

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Goran Švenda, red. prof.

Dakle, nakon formiranja ulaza vrši se proračun debalansa aktivnih i reaktivnih snaga u čvorovima. Potom se ispituju uslovi konvergencije i u slučaju da su ispunjeni, proračun se završava. U suprotnom, prelazi se u sledeću iteraciju.

4. POTROŠAČI I NJIHOVE KARAKTERISTIKE

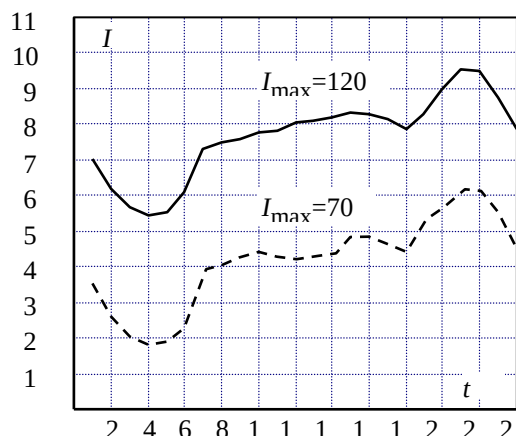
U ovom poglavlju su date definicije osnovnih pojmova, navedene su osnovne karakteristike SN distributivne mreže, kao i opis karakterističnog potrošača i tipovi potrošača.

4.1 Karakterističan potrošač

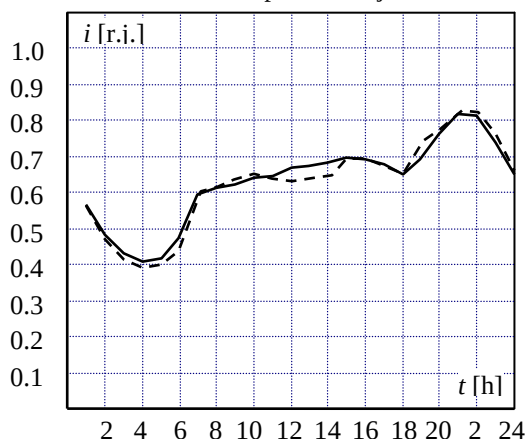
Karakteristični potrošač je fiktivni potrošač određen bar sa dva dnevna hronološka dijagrama (DHD) potrošnje. Ti dijagrami mogu biti normalizovani dnevni hronološki (NDH) dijagrami:

1. modula struja i faktora snaga, ili
2. aktivnih i reaktivnih snaga

Ovi dijagrami mogu da budu sa apsolutnim, odnosno normalizovanim vrednostima. Dnevni hronološki dijagram (u apsolutnim vrednostima) za dva slična potrošača, prikazani su na slici 4.1a), ako se DHD normalizuju na maksimalne vrednosti opterećenja, dobijaju se NDH dijagrami opterećenja prikazani na slici 4.1b).



Slika 4.1a - DHD u apsolutnim jedinicama



Slika 4.1b - DHD u relativnim jedinicama

Dakle, karakteristični potrošač je predstavnik (reprezent) grupe sličnih potrošača čije se funkcije dnevnih hronoloških dijagrama potrošnje (struje, aktivne i

reaktivne snage) mogu dobiti jedni iz drugih množenjem konstantom, a dijagrami faktora snage su im isti [7].

U ovom radu su od interese tri tipa potrošača: Tip 1 – Industrija; Tip 2 – Domaćinstva u individualno izgrađenim objektima, sa grejanjem na gas i zagrevanjem vode sopstvenim akumulacionim bojlerima; Tip 3 – Domaćinstva u individualno izgrađenim objektima.

5. OPTIMALNA REKONFIGURACIJA DISTRIBUTIVNIH MREŽA

Problem određivanja optimalne konfiguracije mreže je kompleksan, kombinatoran, nelinearan i diskretan optimizacioni problem [4]. Rekonfiguracija distributivne mreže se može koristiti i kao planerska funkcija i kao alatka za upravljanje u realnom vremenu, pri nastanku kvarova i u normalnim uslovima rada [8].

Za rešenje ovog problema koriste se: optimizacione metode kombinatorno pretraživanje i heurističke. U praksi se najčešće koriste heurističke metode, čija je svrha da se maksimalnim poznavanjem fizike problema na najkraći način dođe do kvalitetnih radijalnih konfiguracija. Heuristički algoritmi se mogu podeliti u dve grupe algoritam "najmanjih struja" i algoritam "izmene grana". U ovom radu je od interesa algoritam "izmene grana".

5.1 Algoritam izmene grana

"Izmene grana" zapravo znači "izmena mesta" normalno otvorenog (NO) rasklopnog uređaja sa njime spregnutim normalno zatvorenim (NZ) rasklopnim uređajem. Algoritam započinje inicijalizacijom postupka, a zatim se brojač NO rasklopnih uređaja postavlja na vrednost 1. Za prvi NO rasklopni uređaj proračunava se vrednost kriterijumske funkcije i proverava se da li postoji bar jedan NZ rasklopni uređaj sa kojim može da se ostvari bolja vrednost kriterijumske funkcije. Ako ne postoji, ide se u sledeću iteraciju. Ukoliko postoji, bira se onaj rasklopni uređaj za koji je vrednost kriterijumske funkcije maksimalna. Tada se prelazi u novu konfiguraciju izmenom mesta NO i NZ rasklopnih uređaja. Dalje se vrši proračun tokova snaga čiji rezultati predstavljaju ulaz u novu iteraciju. Proračun završava kada je dobijena radijalna konfiguracija. Osnovna mana ovog algoritma je da se u zavisnosti od početnog, dobija različito konačno rešenje.

5.2 Kriterijumi za ocenu performansi distributivnih mreža

U rekonfiguraciji distributivne mreže najčešće se koristi sledećih 6 optimizacionih kriterijuma za ocenu performansi radijalne konfiguracije [4]: Minimalni gubici aktivne snage; Debalans opterećenja na VN/SN transformatorima; Debalans opterećenja na izvodima; Kritični pad napona; Pouzdanost napajanja; Troškovi manipulacija. Najčešće se koristi kriterijum minimalnih gubitaka aktivne snage i kao takav je korišćen i u ovom radu. Pored njega koristio se i kriterijum za troškove manipulacija.

5.2.1 Kriterijum minimalnih gubitaka aktivne snage

Gubici aktivne snage u radijalnoj konfiguraciji (h), definisani su kao [13]:

$$IG^{(h)} = \sum_{i=1}^{n_{izv}} \sum_{j \in \alpha_{d_i}^{(h)}} r_{ij} \left(J_{ij}^{(h)} \right)^2, \quad (1)$$

gde je:

- n_{izv} – ukupan broj izvoda,
- r_{ij} – rezistansa grane (j) koja pripada izvodu (i),
- $\alpha_{d_i}^{(h)}$ – skup indeksa grana izvoda (i) u radijalnoj konfiguraciji (h),
- $J_{ij}^{(h)}$ – moduo aktuelne struje grane (j) koja pripada izvodu (i) u radijalnoj konfiguraciji (h).

Ako je sa supskriptom (h) označena konfiguracija nakon izmene mesta NO/NZ rasklopnih uređaja, a sa (h-1) konfiguracija pre te izmene, vrednost kriterijuma je:

$$IG^{(h)} = IG^{(h-1)} - \Delta IG \quad (2)$$

$$\Delta IG = 2 \cdot \text{Re} \left[\left(J_j^{(h-1)} \right)^* \left(\Delta V_k^{(h-1)} - \Delta V_m^{(h-1)} \right) \right] - \left| J_j^{(h-1)} \right|^2 \cdot R_{petlje} \quad (3)$$

gde su sa $IG^{(h)}$, $IG^{(h-1)}$ i ΔIG naznačeni gubici aktivne snage u (h), odnosno (h-1) konfiguraciji i promena te vrednosti, respektivno; sa $J_j^{(h-1)}$ struja grane (j) u radijalnoj konfiguraciji (h-1); sa $\Delta V_k^{(h-1)}$ i $\Delta V_m^{(h-1)}$ padovi napona u čvoru (k) i (m) u rezistivnoj mreži, u radijalnoj konfiguraciji (h-1) i sa R_{petlje} rezistansa konture između dva napojna čvora koja sadrži granu u kojoj se nalazi NO rasklopni uređaj. Bitno je napomenuti da je pri izvođenju algoritma „izmene grana” za proračun optimalne rekonfiguracije dovoljno proračunati samo ΔIG (relacija 3).

Ako je ta vrednost veća od nule, tada su izmenom mesta razmatranog spregnutog para NO/NZ rasklopnih uređaja gubici aktivne snage smanjeni.

5.2.2. Troškovi manipulacija

Troškovi manipulacija za prelazak u radijalnu konfiguraciju (h) su definisani sledećom relacijom:

$$IC^{(h)} = \sum_{i \in \alpha_c^{(h)}} c_i \quad (4)$$

gde su: c_i – troškovi manipulacije (i); $\alpha_c^{(h)}$ – skup neophodnih manipulacija za prelazak u radijalnu konfiguraciju (h). Gornjom relacijom je data suma troškova manipulacija.

6. TEST DISTRIBUTIVNA MREŽA I PRORAČUN

U ovom delu dat je opis test distributivne mreže, kao i primer proračuna.

6.1 Opis test distributivne mreže

Test mreža se sastoji od 25 potrošačkih čvorova (transformatorske stanice SN/NN).

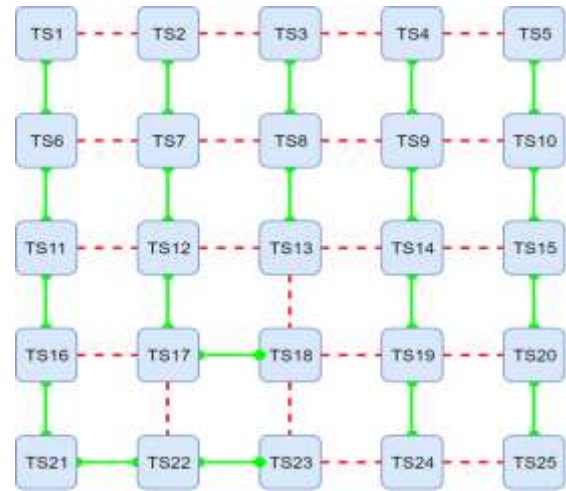
Potrošnja je raspoređena na 5 izvoda koji se napajaju sa SN sabirnica u postrojenja VN/SN. Ukupna prividna snaga potrošnje za test DM je različita za svaki sat. Dužine izvoda su takođe različite.

6.2 Optimalna rekonfiguracije DM za 24 sata

Inicijalna topologija test mreže je različita, takva da se sa izvoda napaja različit broj potrošačkih čvorova.

Različita raspodela opterećenja i različiti broj NO rasklopnih uređaja, kao i različite dužine vodova uzrokuje različite topologije mreža nakon izvršavanja optimalne rekonfiguracije u cilju smanjenja gubitaka aktivne snage, ostvarenja što veće dobiti, a na rezultate će uticati i različito modelovanje potrošnje.

U radu će biti analizirani ovi efekti i izvršeno poređenje rezultata dobijeno pomoću metode „izmene grana”.



Slika 6.2 – Test distributivna mreža

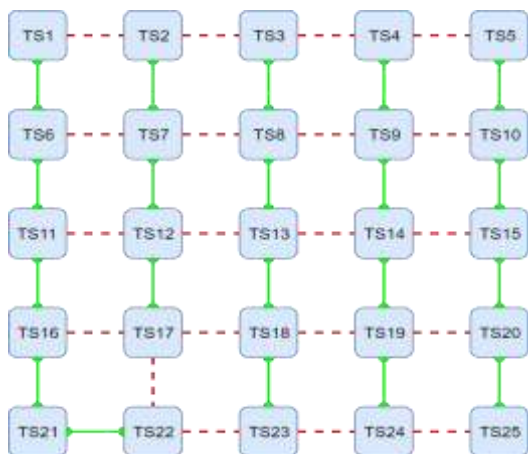
6.3 Primer proračuna

Test DM je prikazana na slici 6.2. Kao što se sa slike može videti, topologija mreže se sastoji iz 5 izvoda, koji su različito opterećeni. Osim što su različito opterećeni, napajaju različit broj potrošača. Drugi, četvrti i peti izvod napajaju po 5 potrošača, dok prvi izvod napaja 7, a treći 3 potrošača.

6.1 Prvi scenario

Prvi scenario proračuna rekonfiguracije mreže za 24 sata, na pomenutoj test distributivnoj mreži je rađen tako što se rekonfiguracija radila svaki put kada smo imali dobit za taj sat. Odnosno, kada smo imali manje gubitke nakon rekonfiguracije. Svaki sat je vršen proračun, tako što se za svaki NO prekidač razmatrao njemu spregnuti NZ prekidač i ukoliko se ostvari dobit položaj prekidača se menjao, ukoliko nije stanje ostaje nepromenjeno.

Posle izvršenih promena iz sata u sat, kada je ostvarena dobit, dobija se test distributivna mreža kao sa slike 6.5. Ovakav vid rekonfiguracije iz sata u sat se ne isplati.

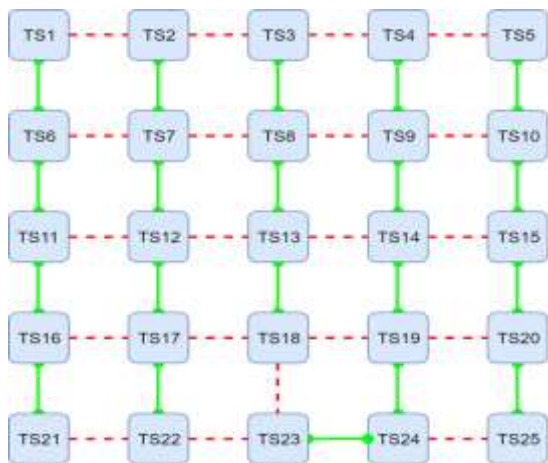


Slika 6.5 – Test distributivna mreža nakon rekonfiguracije

6.3.2 Drugi scenario

U drugom slučaju posmatra se primer kada nije urađena nijedna prekidačka manipulacija, odnosno proračunati su gubici za svaki sat, kada je inicijalno stanje mreže postavljeno 24 sata. Cena samih prekidačkih operacija značajno je prevazilazila dobiti dobijene od smanjenja gubitaka aktivne snage. Trošak/dobit analizom je utvrđeno da se ne isplati primeniti rekonfiguraciju mreže na ovakav način.

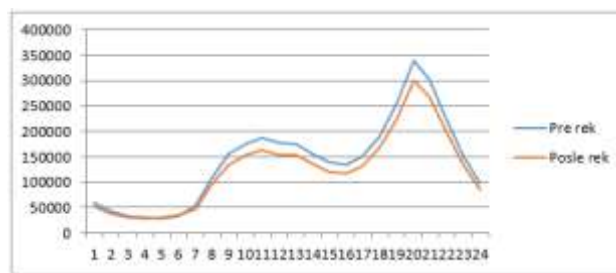
U sledećem slučaju se posmatra trenutak iz sredine dana. Tada je mreža srednje opterećena pa je taj trenutak odabran da se izvrši rekonfiguracija mreže, tj. promeni uklopno stanje prekidača. Nakon takve primene, dobija se topološka struktura mreže kao na slici 6.6.



Slika 6.6 – Test distributivna mreža posle jedne promene uklopnog stanja

6.4 Trošak/Dobit analiza

Nakon što se primenila rekonfiguracija mreže za 24 sata, odnosno algoritam „izmene grana“ na opisanu test distributivnu mrežu, suma gubitaka je iznosila 2993kW, odnosno 3389 kW pre rekonfiguracije. Nakon rekonfiguracije sume gubitaka za 24 sata je za 396 kW manje. Što znači da postoji ušteda u gubicima. Rekonfiguracija je urađena kada se postigla dobit od približno 13%.



Slika 6.7 – Grafički prikaz gubitaka aktivne snage pre i posle rekonfiguracije

7. ZAKLJUČAK

Izmena uklopnog stanja prekidačke opreme iz sata u sat se nikada ne isplati jer je su prekidačke manipulacije mnogo skuplje od ostvarene dobiti. U drugom slučaju je prikazano da može da se isplati ukoliko se jednom primeni novo topološko stanje u toku dana. Zaključak trošak/dobit analize je da se postigla ušteda od 25.98\$ na gubicima, dok su troškovi manipulacije iznosili 6\$, tako da se postigla konačna ušteda od 19.98\$ u 24 sata.

8. LITERATURA

- [1] R.C.Dugan, R.F.Artritt, T.E.McDermott, S.M. Brahma, K.Schneider: Distribution system analysis to support the smart grid; *IEEE Power and Energy Society General Meeting*, pp. 1-8, July 2010.
- [2] D.Popović, D.Bekut, V.Treskanica: *Specijalizovani DMS algoritmi*; DMS Group, Novi Sad, 2004.
- [3] G.S.Švenda: *Osnovi elektroenergetike: Matematički modeli i proračuni*, Fakultet tehničkih nauka, Novi Sad, 2007.
- [4] D.Popović, D.Bekut, V.Dabić: *Specijalizovani DMS Algoritmi*; Prosveta, Novi Sad, 2011.
- [5] V.Strezoski: *Osnovi elektroenergetike (elektroenergetski sistemi)*; radni materijal, školska 2002/2003. godina
- [6] P.Vidović, *Nesimetrični tokovi snaga distributivnih mreža*, Doktorska disertacija, Fakultet tehničkih nauka, Novi Sad, 2015.
- [7] H.C.Kuo, Y.-Y.Hsu: Distribution System Load Estimation and Service Restoration Using a Fuzzy Set Approach; *IEEE Trans. on Power Systems*, Vol.8, No.4, May 1993, pp. 1950-1957
- [8] S.Civanlar, J.J.Grainger, H.Yin, S.S.H.Lee: Distribution Feeder Reconfiguration for Loss Reduction, *IEEE Trans. on Power Delivery*, Vol. 3, No. 3, July 1988, pp 1217-1223.

Kratka biografija:



Nikolina Modraković rođena u Bijeljini 1992. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranila je 2016. god. Iste godine upisala se na master studije na istom smeru.

INDUSTRIJSKA IMPLEMENTACIJA MODBUS PROTOKOLA**INDUSTRIAL IMPLEMENTATION OF MODBUS PROTOCOL**Igor Tot, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu opisani su osnovi SCADA sistema, industrijskih protokola komunikacije kao i detalji Modbus protokola. Predstavljeno je rešenje programske implementacije Modbus protokola u okviru SCADA sistema, u svrhu komunikacije sa udaljenim uređajima - koje zadovoljava realne industrijske potrebe, spram konteksta korišćenja.

Ključne reči: SCADA, RTU, Modbus

Abstract – This paper describes the fundamentals of SCADA systems, industrial communication protocols and specifics of the Modbus protocol. A software solution for implementing the Modbus protocol as means of communication with remote units is presented with regards to realistic industrial needs and constraints.

Key words: SCADA, RTU, Modbus

1. UVOD

Familija Modbus protokola je namenjena za potrebe jednostavne komunikacije sa opremom za automatizaciju. Iako se može koristiti u različitim aplikacijama, najčešća primena ovih protokola je za povezivanje programabilnih logičkih kontrolera, ulazno-izlaznih modula i sl.

Ovaj rad je namenjen da ispita problematiku implementacije Modbus protokola u jedno realno industrijsko okruženje, sa ciljem predloga softverske implementacije koja je arhitekturno dovoljno sveobuhvatna da predvidi poteškoće razvojnog ciklusa takvog sistema. Iz tih razloga se sagleda specifičnost implementacije Modbus protokola u proizvoljnom SCADA sistemu. Distribuirani SCADA sistem podrazumeva dovoljno kompleksnu softversku podršku, koja pruža mogućnost posmatranja problema implementacije Modbus protokola sa akcentom na skalabilnost, robusnost i softversku generičnost. Na temelju znanja o detaljima funkcionalnosti SCADA sistema i ulozi Modbus protokola u takvom okruženju - moguće je definisati problem i predložiti softversko rešenje implementacije Modbus protokola sa obzirom na potrebe životnog ciklusa takvog softvera.

2. SCADA SISTEM I ULOGA INDUSTRIJSKIH PROTOKOLA

Skup računarskih modula koji je prostorno distribuiran i međusobno povezan sa ciljem ostvarenja akviziciono-upravljačkih funkcija nad određenim fizičkim procesom u realnom vremenu - opisuje jedan SCADA (engl. *Supervisory Control And Data Acquisition*) sistem.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branislav Atlagić, docent.

Ovakvi sistemi su specifične namene i strukture, i kao takvi obuhvataju širok spektar hardverske opreme i softverskih komponenti u svrhu ostvarenja tražene funkcionalnosti. Ključne komponente jednog SCADA sistema su: Udaljene RTU (engl. *Remote Terminal Unit*) ili PLC (engl. *Programmable Logic Controller*) jedinice, Nadzorno-Upravljačka MTU (engl. *Master Terminal Unit*) stanica i komunikaciona infrastruktura. Arhitektura SCADA sistema (slika 2-1) opisuje ulogu svake od ovih komponenti spram stanovišta akviziciono-upravljačke funkcije. Procesni ulazi u tom smislu podrazumevaju podatke digitalnog ili analognog tipa prikupljene putem senzora, dok procesni izlazi jesu komande nad raznim aktuatorima koji menjaju stanje sistema. Udaljeni uređaj jeste računar namenjen za rad u industrijskim uslovima (ekstremne temperature, pritisak, i sl.), dok nadzorno-upravljačka stanica kao centralna tačka SCADA sistema komunicira sa velikim brojem takvih računara posredstvom podsistema za komunikaciju. Nadzorno-upravljačka stanica je opremljena ozbiljnim PC računarom (ili grupom računara) na kojem se izvršava SCADA softver koji pruža nivo automatizacije procesa i estimaciju sistema spram obrađenih podataka sa polja.



Slika 2-1 – Arhitektura SCADA sistema

Pored funkcionalnih zahteva akvizicije i upravljanja podacima nad posmatranim fizičkim procesom, SCADA sistem zadovoljava i par osnovnih osobina kritičnog softverskog sistema. Kritičnost znači da greške pri radu sa ovakvim sistemom mogu da imaju katastrofalne posledice, pa iz tog razloga SCADA sistem podrazumeva minimizovano srednje vreme između ispada sistema, uz replikaciju podataka - visok nivo pouzdanosti. Sa druge strane, ovaj sistem mora biti dostupan u slučaju otkaza hardverskih ili softverskih komponenti, što podrazumeva visok nivo raspoloživosti. SCADA takođe zadovoljava i princip rada u realnom vremenu - odnosno, sve akcije nad sistemom podrazumevaju pojam logičke i vremenske ispravnosti; drugim rečima, ukoliko se željene akcije nad sistemom ne izvršavaju u predodređenim vremenskim intervalima - smatramo ih neispravnim, čak iako se tražena akcija logički uspešno izvršava. Konačno, SCADA sistem mora biti prostorno distribuiran po pitanju računarskih resursa.

SCADA sistem kao distribuirani računarski sistem razmenjuje poruke vezane za nadzor i upravljanje

posmatranim fizičkim procesom. Skup pravila i procedura koje kontrolišu tok komunikacije između komponenti distribuiranog računarskog sistema definiše komunikacioni protokol. Ta pravila su vezana za format i redosled poruka koje se razmenjuju, kao i svaku akciju koja se inicira prijemom određene poruke. Osnovne funkcije komunikacionih protokola su kontrola grešaka i upravljanje tokom podataka u mreži.

Budući da telekomunikacioni sistemi imaju strogo hijerarhijsku strukturu - realizovanu kroz više slojeva, komunikacioni protokol mora odrediti nivo na kojem se izvršava. Ova slojevitost definisana je ISO/OSI (engl. *Open Systems Interconnection Basic Reference Model*) modelom.

Prvi nivo protokola je fizički, i definiše električne i mehaničke karakteristike prenosnog kanala. Iznad njega su programski slojevi protokola. Svaki od njih proširuje skup usluga, oslanjajući se u potpunosti na funkcije prethodnog nivoa. Sprega između pojedinih slojeva protokola na istoj stanici vrši se tako da viši sloj koristi isključivo usluge prethodnog nivoa. [1]

Klasa protokola specifičnih za SCADA sisteme podrazumevana je terminom industrijski protokoli i industrijske mreže. Ovaj naziv naglašava njihovu primenu u industrijskim postrojenjima i odvaja ih od protokola iz oblasti računarstva i telekomunikacija. Industrijski protokoli se mahom oslanjaju na prvi (fizički) i sedmi (aplikacioni) sloj ISO/OSI modela i orijentisani su ka zadovoljenju aplikativnih zahteva akviziciono upravljačkog sistema.

3. MODBUS PROTOKOL

Industrijski komunikacioni protokol razvijen i predstavljen od strane kompanije Modicon (sada Schneider Electric) 1979. godine, primarno za komunikaciju sa programabilnim logičkim kontrolerima - Modbus - postao je *de facto* standardni oblik komunikacije pri povezivanju industrijskih uređaja. Modbus standard se odnosi na aplikativni sloj OSI modela i zasnovan je na klijent-server arhitekturi. Danas se koristi za komunikaciju između više uređaja iz iste mreže, gde se smatra jednostavnim za postavku i održavanje.

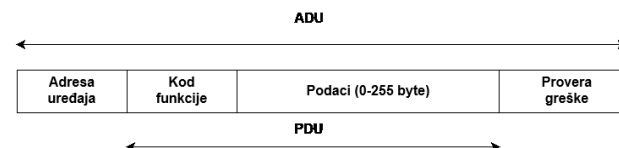
Modbus protokol postao je vrlo popularan usled otvorene prirode protokola i specifikacije. Od 2005. godine razvojem i usavršavanjem Modbus protokola bavi se organizacija Modbus-IDA na koju je prenet sva neophodna nadležnost. Inicijalno je Modbus protokol bio zamišljen za komunikaciju preko asinhronne serijske magistrale, ali je osposobljen i za potrebe lokalnih mreža, posredstvom TCP/IP transportnog sloja.

3.1 FORMAT MODBUS PORUKE

Modbus je tipičan nebalansirani protokol, gde sa ugla akviziciono-upravljačkih sistema: PLC igra ulogu servera (engl. slave) koja na upit SCADA klijenta (engl. master) šalje odgovore u vidu željenih poruka. Opšti format poruke definisan je Modbus protokolom kao aplikativna jedinica podataka (engl. *Application data Unit - ADU*) čije jezgro definiše jednostavan jedinični podatak protokola (engl. *Protocol Data Unit - PDU*) koji je uvek

prisutan nezavisno od varijante Modbusa. Na slici 3.1-1 je prikazan sadržaj Modbus ADU-a. [2]

Adresa uređaja može biti u opsegu 1-247, dok je 0 rezervisana za *broadcast* poruke koje prima i obrađuje svaka stanica - bez da šalje povratnu poruku kao odgovor. Master uređaj sa druge strane, po pravilu nema svoju adresu. Kod funkcije je enkodiran u jedan bajt, pa raspolaže opsegom 1-255, gde su vrednosti opsega 128-255 rezervisane za odgovore grešaka. Svrha koda funkcije jeste da pri komunikaciji, master saopšti serveru koju akciju treba da izvrši. Podaci PDU-a, sa druge strane sastoje se od dodatnih informacija koje server koristi pri izvršenju akcije kod funkcije.



Slika 3.1-1 Opšta struktura poruke Modbus protokola [2]

Na slici 3.1-2 je predstavljen bazni skup poruka neophodan za uspešnu komunikaciju sa udaljenim uređajem posredstvom Modbus protokola. Vertikalno posmatrano svaki od upita ima odgovarajući odgovor, gde su isti grupisani u PDU jedinice. Poruke se sastoje iz dva glavna dela: koda funkcije i podataka sadržanih u upitu - odnosno odgovoru. Prva 4 bita upita i odgovora su uvek rezervisane za kod funkcije, dok su narednih 16 pri upitima uvek rezervisane za adresu registra (registara) koji je neophodan (su neophodni) za izvršenje akcije. *Read* upiti (funkcije 1, 2, 3 i 4) narednih 16 bitova rezervišu za broj registara koje treba pročitati, dok komandni (*write*) zahtevi iste rezervišu za vrednost koju treba upisati. [1]

	Read Discrete Inputs	Read Coils	Write Single Coil	Read Input Register	Read Holding Registers	Write Single Register	Write Multiple Registers
Kod Funkcije	2	1	5	4	3	6	16

Upit	2	1	5	4	3	6	16
1	Input	Coil	Coil	InReg	HoldReg	HoldReg	HoldReg
2	Adr	Adr	Adr	Adr	Adr	Adr	Adr
3	Num	Num	Value	Num	Num	Value	Num
4							
5							ByteCount
6							Registers Values
...							Num*2

Odgovor	2	1	5	4	3	6	16
1	ByteCount	ByteCount	Coil	ByteCount	ByteCount	HoldReg	HoldReg
2			Adr			Adr	Adr
3	Inputs	Coils				Value	Num
4	Num/8	Num/8		Input Registers	Holding Registers		
5				Num*2	Num*2		
6							
7							
...							

Slika 3.1-2 Bazni skup Modbus poruka [1]

3.2 MODBUS MODEL PODATAKA

Model podataka je u jezgru svakog protokola. On određuje tipove podataka i aplikativne funkcije koje

protokol podržava. Kao takav on predstavlja spoljnu sliku uređaja koji ga implementira. Sa obzirom da je Modbus dizajniran krajem 1970-ih, u svrhu komunikacije sa PLC-ovima, tipovi podataka kojima barata protokol su ograničeni na one koje su PLC-ovi tada razumeli. [3]

Modbus logički model (po specifikaciji *Modbus Register Map*) podrazumeva četiri grupe registara, dužine 1 ili 16 bita. Oni predstavljaju najvažnije PLC podatke - sa stanovišta SCADA sistema: vrednosti procesnih ulaza/izlaza analognog odnosno digitalnog tipa. Ovi registri su prikazani slikom 3.2-1 gde se vide dozvoljene operacije komandovanja odnosno čitanja svakog od njih.

Oznaka	Dužina	Pristup	Opis
Discrete Outputs (Coils)	1 bit	Read/Write	Digitalni izlazi
Discrete Inputs	1 bit	Read	Digitalni ulazi
Input Registers	16 bit	Read	Analogni ulazi i Brojači
Holding Registers	16 bit	Read/Write	Analogni izlazi

Slika 3.2-1 Modbus model podataka

Format podataka u registrima vidljivim klijentu je binaran, neoznačen i po pravilu - *big endian*. Budući da Modbus usled inicijalnog dizajna predviđa prenos sirovih podataka u praksi potrebno primeniti rešenja koja odstupaju od osnovnog standarda kako bi se preneli 32-bitni integer, float ili string tipovi podataka. Oni se smeštaju korišćenjem dve lokacije u zoni internih registara. Podatak se deli na dva dela od po 16 bita i smešta na dve uzastopne adrese.

3.3 VARIJANTE MODBUS PROTOKOLA

Prva verzija Modbusa bila je dizajnirana kao tekstualni protokol, za rad preko serijskog UART kanala - Modbus ASCII. Tekstualni format nije optimalan spram vremena prenosa i dužine poruke, pa je zamenjen binarnom verzijom koja je i dan danas aktuelna - Modbus RTU. Ova varijanta protokola je orijentisana ka telemetrijskim sistemima i RTU kontrolerima. Obe verzije protokola imaju istu logičku strukturu poruke, ali razlikuju se po pitanju kodiranja sadržaja (ASCII ili binarno).

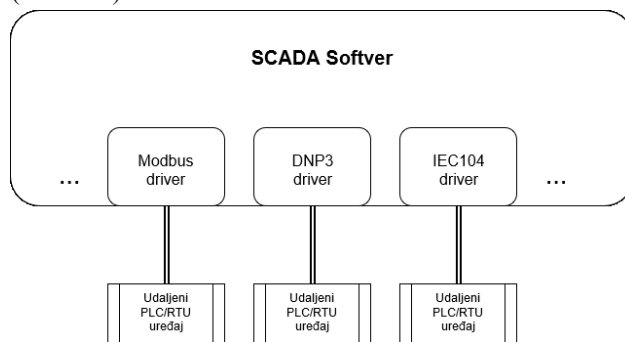
Kodiranje Modbus ASCII podataka pri prenosu se svodi na slanje dva ASCII znaka, od kojih svaki predstavlja jednu heksadecimalnu cifru (0..9, A..F). Prednost ASCII formata se ogleda u tome da postoji mogućnost vizuelne kontrole, što je nekada bilo od koristi u praksi. Modbus ASCII poruka započinje znakom :, a završava slanjem CRLF sekvence (engl. *carriage return-line feed*). RTU varijanta protokola binarnim kodiranjem svodi poruku na minimalan obim podataka kako bi ostvario maksimalnu efikasnost pri brzini prenosa podataka preko sporih analognih modemskih veza brzine 600 - 2400 bit/s. Obe varijante Modbus protokola definišu poruke koje završavaju redundantnom proverom - mehanizmom kojim se utvrđuje ispravnost prenetog sadržaja.

Modbus TCP je mrežna varijanta Modbus protokola. Slično Modbus RTU varijanti - koristi binarno kodiranje podataka, ali i TCP/IP mehanizam za pouzdan prenos poruka u okviru mreže. Modbus TCP je konfekcijski orijentisan protokol, što znači da su moguće istovremene veze sa više slave uređaja (ili više veza sa istim uređajem). Slično RTU varijanti: TCP/IP *Timeout* je

karakteristika TCP verzije protokola koja definiše vreme u slučaju neuspešnog prenosa, nakon kojeg se veza zatvara i ponovo otvara u cilju uspešnog nastavka komunikacije. Budući da ova varijanta protokola uvodi mogućnost komunikacije sa više servera, potrebno je voditi računa o integritetu upravljačkog sistema. [2]

4. KONCEPT REŠENJA

Implementacija Modbus protokola u SCADA sistem podrazumeva realizaciju softverske podrške komunikacionog podsistema, u cilju komunikacije sa udaljenim uređajima posredstvom Modbus protokola. Programsko rešenje opisano u ovom radu nastoji da zadovolji karakteristike modernog SCADA sistema, pa je od izrazite važnosti osmisliti adekvatnu arhitekturu softverske podrške. SCADA softver jednog modernog SCADA sistema uslove generičnosti i skalabilnosti zadovoljava programskim komponentama koje se najčešće nazivaju rukovaocima (engl. *driver*). Ideja je da svaka rukovaoc komponenta ima odgovornost upravljanja perifernim jedinicama, prenosa poruka konstrukcije paketa i obrade grešaka - gde svaki od njih komunicira sa udaljenim uređajima po unapred definisanom protokolu (slika 4-1).



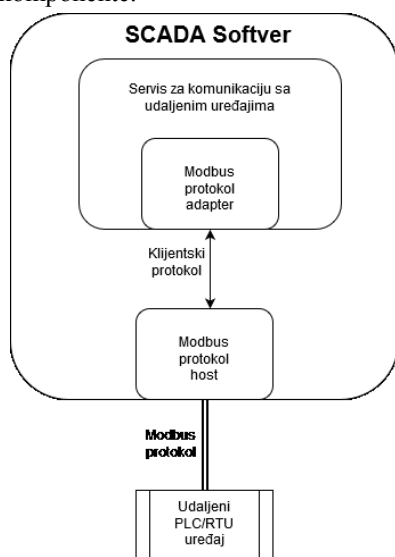
Slika 4-1 Generičnost rešenja

Predloženo rešenje se konceptualno sastoji od dve ključne komponente: servis za komunikaciju sa udaljenim uređajima i Modbus protokol host. Servis za komunikaciju sa udaljenim uređajima je nezavisan od konkretnog protokola komunikacije i kao takav sadrži Modbus protokol adapter koji servisu omogućuje komunikaciju sa Modbus protokol hostom. Protokol host se direktno povezuje sa udaljenim uređajima sa kojima i komunicira putem Modbus protokola. Protokol host i protokol adapter komponente su zamišljene kao servisi koji komuniciraju putem nekog klijentskog protokola što istovremeno omogućava podizanje Modbus protokol hosta na različitim računarima. Arhitektura ovog koncepta je prikazana slikom 4-2.

Servis za komunikaciju sa udaljenim uređajima korisniku omogućava da preko konzole izda naredbe, odnosno očita vrednosti sa udaljenog uređaja nezavisno od industrijskog protokola na jedinstven i konzistentan način. Pored toga što komunicira sa modulom korisničkog interfejsa sa jedne strane, ovaj servis sa druge strane posredstvom odgovarajućeg adaptera obezbeđuje komunikaciju sa protokol hostovima. Servis iz biblioteke adaptera dobavlja Modbus protokol adapter, koji ostvaruje komunikacionu spregu sa Modbus protokol host-om. Dve ključne komponente ovog modula jesu Akviziciono-upravljački servis i Dispečer zahteva. Posredstvom ovih komponenti

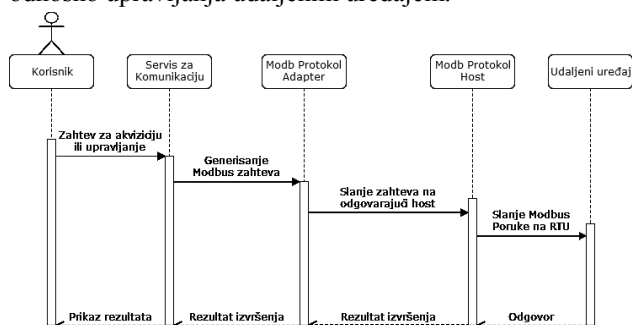
ostvaruje se generisanje zahteva za akviziciju, odnosno generisanje komandi - spram prethodno učitane konfiguracije i dospelog ulaza sa korisničkog interfejsa.

Modbus protokol host je komponenta koja direktno komunicira sa udaljenim uređajima po Modbus protokolu. Broj senzora i aktuatora na udaljenim uređajima potencijalno može biti velik sa porastom samih uređaja u sistemu. To znači da komunikacioni podsistem potencijalno mora da podrži prenos izrazito velikog broja poruka u određenom trenutku, pa je potrebno da SCADA softver omogući konkretnom protokol adapteru komunikaciju sa više protokol hostova kako bi se izbeglo zagušenje komunikacije. Teoretski, predloženo rešenje ima mogućnost skalabilnosti po pitanju broja protokol hostova, ali je zarad jednostavnosti realizovan i opisan samo jedan Modbus protokol host kako bi se sagledala suština te komponente.



Slika 4-2 Arhitektura rešenja

Na slici 4-3 prikazana je razmena poruka između prethodno opisanih komponenti u slučaju akvizicije odnosno upravljanja udaljenim uređajem.



Slika 4-3 – Tok komunikacije pri akviziciono-upravljačkom procesu

5. IMPLEMENTACIJA REŠENJA

Programsko rešenje razvijeno je u programskom jeziku C#, pomoću Microsoft-ovog Visual Studio 2015 razvojnog okruženja. Korišćen je .NET 4.6.2 framework pri razvoju rukovaoca, a za potrebe komunikacije je simuliran RTU uređaj pomoću Modbus PLC Simulatora [4].

Pokretanjem procesa Servisa za komunikaciju, korisniku će se prikazati upravljačka konzola gde je moguće izdati neku od mogućih naredbi akvizicije odnosno upravljanja.

Pri pokretanju servisa vrši se konfiguracija modela podataka SCADA sistema. Po izdavanju zahteva za akviziciju, odnosno upravljanje – proces se oslanja na model podataka sistema kako bi odredio o kojem protokolu je reč, u svrhu dobavljanja adekvatnog Protokol Adaptera. Adapter komponenta je pomoću MEF (engl. *Managed Extensibility Framework*) tehnologije enkapsulirana i izmeštena u zasebnu biblioteku u svrhu postizanja modularnosti. Izdati zahtev se prenosi ka Modbus Protokol Hostu, koji na osnovu njega kreira odgovarajuću Modbus poruku kako bi iskomunicirao zahtev sa simulatorom. Simulator vraća odgovor i odgovarajuće podatke koji se nakon ažuriranja u Redis in-memory bazi podataka Protokol Hosta šalju nazad ka servisu za komunikaciju. Korisnik spram prikaza na konzoli nije svestan detalja komunikacije, te bi prikaz bio isti u slučaju implementacije nekog drugog protokola i simulatora. Servis za upravljanje sa udaljenim uređajem i protokol host realizovani međusobno komuniciraju posredstvom WCF (engl. *Windows Communication Foundation*) tehnologije. Modbus PLC Simulator je javno dostupan simulator PLC uređaja koji podržava RTU i TCP varijante Modbus komunikacije. Iskorišćen je u svrhu simulacije udaljenog uređaja sa Modbus protokol host komponentom, kako bi se adekvatno sagledala funkcionalnost programskog rešenja.

7. ZAKLJUČAK

Tematika problema implementacije Modbus protokola u SCADA sistem je dovoljno kompleksna, a rešenje dovoljno jednostavno da se dočara jasna slika o detaljima problema uvođenja protokola komunikacije u jedan ovakav sistem. Rešenje podrazumeva slanje zahteva za akviziciju i upravljanje nad pojedinačnim tačkama Modbus PLC simulatora. Arhitekturno, rešenje je zamišljeno sa idejom proširivosti, tako da bi eventualan dalji pravac razvoja pokrio preostale Modbus funkcije koje je teoretski moguće izvršiti nad udaljenim uređajem. Takođe, moguće su izmene na postojećoj arhitekturi u cilju uvođenja manuelnog konfigurisanja sistema, izmeštanja konzolne aplikacije u zaseban WCF servis, kao i poboljšanje performansi komunikacionih kanala u slučaju uvođenja više udaljenih uređaja.

8. LITERATURA

- [1] Branislav Atlagić, Softver sa kritičnim odzivom, Fakultet tehničkih nauka, Novi Sad, 2015
- [2] MODBUS Application Protocol Specification, Version 1.1b3, 2012
- [3] <https://en.wikipedia.org/wiki/Modbus>
- [4] <https://sites.google.com/site/plcsimulator/>

Kratka biografija:

Igor Tot rođen je 14.05.1993. u Novom Sadu gde je i završio Gimnaziju „Jovan Jovanović Zmaj” 2012. godine. Upisao je osnovne akademske studije Računarstva i Automatike na Fakultetu Tehničkih Nauka iste godine. Po završetku osnovnih studija, 2017. upisao je master akademske studije Primenjenog Softverskog Inženjerstva na istom fakultetu.

DETEKCIJA I ANALIZA DUPLICIRANOG JAVASCRIPT KODA UPOTREBOM JSINSPECT ALATA**DETECTION AND ANALYSIS OF DUPLICATED JAVASCRIPT CODE USING JSINSPECT TOOL**

Tamara Letić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je opisan JavaScript jezik, njegovo trenutno mesto u svetu softverskog inženjerstva, kao i njegov potencijal u budućnosti. Pokrivena je tema kvaliteta koda kao uvod i potreba za postojanjem alata za njegovu statičku analizu. Detaljno je obrađena problematika pojave dupliciranog koda i istražen je već postojeći alat za njegovu detekciju kroz implementaciju veb aplikacije u koju je alat integrisan i analizu rezultata dobijenih prilikom analize.

Ključne reči: *Java Script, kvalitet koda, statička analiza, duplirani kod, JSInspect*

Abstract – *This paper describes the JavaScript language, its current place in the world of software engineering, as well as its potential in the future. The theme of the quality of the code is covered as an introduction and the need for the existence of a tool for its static analysis. The problem of the duplicate code appearance has been elaborated in detail, and an already existing tool for its detection has been explored through the implementation of the web application into which the tool is integrated and analysis of the results obtained during the analysis.*

Keywords: *Java Script, code quality, static analysis, duplicate code, JSInspect*

1. UVOD

Ovaj projekat ima za cilj da ukratko uvede pojavu kvaliteta koda kroz dinamički otkucani jezik kao što je JavaScript, da analizira rad algoritma za pretragu - Apstraktno sintaksko stablo i alat JSInspect, kroz poređenje sa postojećim algoritmom i alatom za pronalaženje dupliciranog koda. Da definiše dobar skup metrika i problematičnih delova kodova koji su specifični za JavaScript jezik.

Da bi se procenio alat za analizu implementirana je veb aplikacija kroz koju je alat integrisan tako da je omogućen korisnički interfejs, obuhvata se i obrađuje velika kolekcija JavaScript koda prikupljenih iz projekata otvorenog koda i iz zbirke popularnih veb stranica.

2. JAVASCRIPT JEZIK I KVALITET SOFTVERSKOG KODA

JavaScript je objektno orijentisan programski jezik. Kao skriptni jezik, dizajniran je za kontrolu i automatizaciju postojećeg sistema. Ovaj postojeći sistem naziva se host okruženjem. Host okruženje otkriva objekte sa svojstvima i funkcijama kojima se može pristupiti iz JavaScript-a. JavaScript jezik ne pruža nikakav ugrađeni mehanizam za generički ulaz ili izlaz, već se oslanja na objekte koje host okruženje obezbeđuje za komunikaciju sa hostom ili sa eksternim sistemima.

2.1 Code smells

Termin "kvalitet softvera" je skup više različitih ideja. Relevantne teme istraživanja u oblasti kvalitet softvera su identifikacija "smrada koda", određenih anti-paterana koji ukazuju na nizak kvalitet koda i "klonova koda", identične ili slične fragmente koda koji se javljaju na više lokacija u okviru istog projekta. Prisustvo smrada i klonova koda često ukazuje na to da će refaktorizacija koda ili redizajniranje programa koristiti sveukupnom kvalitetu softvera. Donekle povezano, takođe postoji velika količina posla u određivanju i definisanju 'metrika kodova' kao sredstvo za kvantifikaciju različitih aspekata izvornog koda.

Tehnike statičke analize mogu se koristiti za otkrivanje smrada koda, pronalaženje kodnih klonova i izračunavanje metrika koda prilikom pisanja softvera, pomažući programerima da spreče često pravljene greške. Ove tehnike postaju naročito korisne kada su programerima obezbeđeni alati za automatsko analiziranje njihovog koda i pomoć pri pisanju softvera, integrisanog u njihov IDE ili kao deo kontinuiranog sistema integracije.

2.2 Code clones

Softverski inženjeri govore o kod klonovima kada nailaze na više fragmenata koda (npr. datoteke, moduli, funkcije, blokovi) koji su potpuno ili delimično identični. Klonovi kodova se kreiraju kad god programer kopira određeni deo koda i *paste*-uje ga negde drugde, eventualno napravi neke male modifikacije nakon toga. Klonovi kodova takođe mogu da se "pojave" kada, na primer, dve osobe nezavisno napišu veoma sličnu klasu ili metodu u drugom fajlu. Ponekad je njihovo postojanje legitimno, ali u mnogim slučajevima, ovi klonovi su neželjeni i, sa određenom refaktorizacijom i redizajniranjem koda, ih treba pažljivo spajati nazad u jednu jedinstvenu celinu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Čapko vanr.prof.

2.2.1 JavaScript klonovi koda

Ne postoji razlog zašto različite tehnike otkrivanja dupliciranog koda ne bi funkcionisale kod JavaScript-a. U praksi, međutim, čini se da postoji malo istraživanja koja se eksplicitno usmeravaju na klonove u JavaScript kodu.

Syntyssky, Cordy i Dean [1] opisuju pristup za pronalaženje kloniranog koda na veb stranicama koje se fokusiraju na HTML fragmente, ali koje se bave i *inline* i eksternim JavaScript kodom (preko `<script>` oznaka bez ili sa atributom `src`). Oni priznaju da koriste veoma jednostavan algoritam detekcije klona, jer otkrivanje klona nije glavni cilj njihovog istraživanja. Čini se da su uključene datoteke i *inline* kod blokovi tretirani kao pojedinačni fragmenti koda između kojih se otkrivaju klonovi.

Calefato [2] je objavio članak o detekciji kloniranih funkcija u veb aplikacijama. Oni tretiraju JavaScript funkcije kao fragmente koda, najmanjim jedinicama poređenja. Kandidati za klonove koda se detektuju pronalaskom duplikata imena funkcija u bazi koda. Oni tvrde da klonove JavaScript koda kreiraju programeri koji kopiraju i *paste*-uju funkcije iz jednog fajla u drugi, eventualno menjajući telo funkcije, ali zadržavajući ime funkcije netaknuto. Ovi kandidati za klonirane funkcije se ručno upoređivani i klasifikovani.

Softver PMD sadrži copy/paste detektor, skraćeno kao CPD. Koristi algoritam koji podudaranja stringova upoređujući string hash vrednosti i radi na leksičkom nivou. Uključeni CPD u verziji 4.3 PMD sadrži podršku za JavaScript (sa opcijom ECMAScript jezika), ali to nije izloženo u svim delovima interfejsa.

CPD takođe omogućava korisnicima da dodaju podršku za nove jezike pomoću dodataka, u obliku definicije jezika i kreatora tokena. Pronađena su dva open-source projekta koja koriste ovu metodu za analizu JavaScript koda: jedan *standalone* JavaScript modul i drugi modul kao deo Sonar open-source platforme za upravljanje kvalitetom koda.

Komercijalni ECMAScript CloneDR alat koristi sintaksičke metode za detekciju klona pomoću AST-ova. Ira Baxter, koja je napisala tekst "Detekcija klonova koristeći apstraktna sintaksna stabla" [3], je CTO kompanije koja nudi ECMAScript CloneDR alat.

3. STRUKTURA ALATA JSINSPECT I INTEGRACIJA KROZ VEB BAZIRANO REŠENJE

U prethodnom delu ovog rada je opisano zbog čega je došlo do potrebe za pojavom statičke analize koda, kao i o problemima takve analize jezika kao što je JavaScript. U delu 2.2.1 su pomenuti neki postojeći alati za pronalazak dupliciranog koda, ovo poglavlje baviće se novijim open-source alatom JSInspect. Zbog potrebe korisničkog prikaza ovakvih alata, koji se inače pokreću isključivo preko konzole, kreirana je veb aplikacija koja će obezbediti lak unos parametara potrebnih za analizu, kao i prikaz rezultata.

3.1. Opis kreirane veb aplikacije

U cilju olakšavanja korisničkog iskustva sa alatom JSInspect i sličnim alatima napravljena je veb aplikacija. Server je osmišljen tako da se pridržava RESTful principa. Pomoću Express *framework*-a koji se oslanja na Node.js kreiran je Web API.

Potrebno je naglasiti da se podaci ne čuvaju na serveru, interni serverski model služi isključivo za obradu zahteva. Server prima objekat opcija i na osnovu prosleđenih opcija vrši obradu zahteva. Nakon obrade formira čitav model i JSON objekat vraća kao odgovor.

Zbog obezbeđivanja fleksibilnosti servera za podršku integracije različitih alata za pretragu dupliciranog koda bilo je potrebno kreirati adaptore. Cilj adaptera je da kada stigne takav zahtev da zadovoljava interni serverski model, vrši mapiranje pristiglih opcija radi dobijanja formata opcija za odabrani alat. Poziva se funkcija alata koja vrši analizu i vraća rezultat. Taj rezultat ponovo prolazi kroz adapter i kao takav dolazi kao odgovor na klijent.

Zbog klijentske potrebe pregleda strukture odabranog foldera u obliku stabla na serveru je implementiran kontroler. On na osnovu relativne ili apsolutne putanje proveriti sve fajlove koji se nalaze u traženom direktorijumu, kreira strukturu stabla i vrati klijentu.

Da bi klijentu bilo omogućeno da vidi fajlove u kojima se nalaze duplikati takođe je obezbeđeno da na odabir određenog objekta rezultata analize, server na taj zahtev odgovara sadržajem tražene datoteke.

Klijentski deo aplikacije je pisan u JavaScript programskom jeziku, koristeći React biblioteku.

Inicijalno je postavljen testni projekat čija je struktura izlistana na levoj strani, ako korisnik želi da promeni folder za analizu može to uraditi preko opcije **File** -> **Open folder** preko navigacije na vrhu aplikacije. Selekcijom nekog čvora strukture i klikom na dugme **Inspect folder** bira se putanja nad kojom će alat vršiti pretragu i otvara se modalni dijalog radi prikupljanja preostalih neophodnih informacija za analizu.

Prvo polje dijaloga predstavlja combobox preko kog korisnik odlučuje koji alat za analizu želi da koristi. Spram odabranog alata se renderuju različita polja za unos ne bi li se prikupile informacije koje taj specifični alat zahteva.

Nakon popunjavanja i validacije polja za izabrani alat korisnik ima mogućnost da zahtev za analizom šalje na server ili da odustane od akcije.

3.2. JSInspect alat

JSInspect je alat za detekciju dupliciranog koda sa podrškom za JavaScript ekosistem: ES6, JSX i Flow. On koristi AST za pravljenje strukture izvornog koda. To znači da je kod predstavljen preko stabla, gde svaki čvor predstavlja konstrukciju kao što je blok izjava, deklaracija promenljivih itd. Zbog ovoga, JSInspect bi trebalo da bude dobar u pronalaženju strukturno sličnog koda.

Kroz alat je pružena mogućnost da se specificira *threshold* koji određuje najmanji podskup čvorova za analizu. Ovo će identifikovati kod sa sličnom strukturom, baziranom na AST tipovima čvorova. Inicijalno, on traži čvorove sa podudarajućim identifikatorima i literalima za copy/paste orijentisanu detekciju, ali ovo može biti isključeno. Zbog konteksta, identifikatori uključuju imena varijabli, metoda, svojstva itd. Dok literalne predstavljaju stringovi, brojevi i ostalo.

Alat prihvata listu putanja za parsiranje i vraća ispis svih pronađenih podudaranja. Kroz sve direktorijume na prosleđenim putanjama se prolazi rekurzivno i samo .js i .jsx fajlovi bivaju analizirani. Takođe, postoji mogućnost da se eksplicitno proslede putanje koje uključuju drugačije ekstenzije.

3.2.1 Parsiranje

Apstraktno sintaksko stablo (AST) je reprezentacija apstraktnih sintaktičkih struktura izvornog koda napisanog u programskom jeziku. Svaki čvor stabla predstavlja konstrukciju koja se javlja u izvornom kodu. Sintaksa je apstraktna u smislu da ne predstavlja bas svaki detalj koji se javlja u pravoj sintaksi, nego samo strukturne detalje, vezane za sadržaj. Npr. grupisane zagrade su implicitne u strukturi stabla, a sintaktička konstrukcija kao što je *if-uslov-then* izraz može biti obeležena kao jedan čvor sa tri grane.

Ovo pravi razliku između apstraktno sintaksnih stabala i konkretnih sintaksnih stabala, tradicionalno korišćenih stabala za parsiranje, koja tipično kreira parser prilikom prevođenja izvornog koda i kompajliranja. Nakon kreiranja, dodatne informacije se dodaju u AST kao proizvod rekurzivne obrade, npr. kontekstualne analize.

AST se naširoko koristi u kompajlerima kako bi se predstavila struktura programskog koda. [4]

JSInspect alat nema svoj parser, već koristi postojeći Babylon, JavaScript parser koji se koristi u Babel-u. Babylon generiše AST prema Babel AST formatu. On je zasnovan na ESTree spec sa sledećim devijacijama:

- **Literal** token se zamenjuje sa `StringLiteral`, `NumericLiteral`, `BooleanLiteral`, `NullLiteral`, `RegExpLiteral`
- **Property** token se zamenjuje sa `ObjectProperty` i `ObjectMethod`
- **MethodDefinition** se zamenjuje sa `ClassMethod`
- **Program** i **BlockStatement** sadrži dodatno `directives` polje sa `Directive` i `DirectiveLiteral`
- **ClassMethod**, **ObjectProperty** i **ObjectMethod** vrednosti `property` se dovode u čvor glavne metode

3.2.2 Analiza

U ovom delu biće opisan način rada alata od dobijanja sadržaja .js fajlova u obliku niza AST objekata do vraćanja rezultata analize.

Glavna klasa alata `Inspector`, koja nasleđuje `EventEmitter`, kroz konstruktor kreira novu instancu koja prima niz putanja i opcije sa bilo kojom kombinacijom postojećih opcija. Instanca emituje tri event-a: *start*, *match* i *end*.

Run je metoda koja pokreće inspekciju na datim putanjama, obezbeđenim u konstruktoru. Na samom početku emituje *start event*, nakon čega radi parsiranje sadržaja svakog fajla sa prosleđenih putanja i pretvara ga u niz AST objekata.

Za svaki AST objekat se radi prolazanje kroz njegovo stablo i pravi se niz čvorova koji zadovoljavaju *threshold* prosledjen *Inspector*-u preko opcija. Kad se pronađu svi takvi čvorovi, konačan niz se prosleđuje dalje metodi koja generiše ključ baziran na kombinovanim tipovima svakog prosleđenog čvora. Niz stavlja u drugi niz na mesto tog

ključa u mapi. Čvorovi se *update*-uju da bi se održala referenca na sve njihove pojave u mapi.

Sledeći korak prolazi kroz ključeve na kojima se čuvaju različiti čvorovi. Ključ koji sadrži niz od više od jednog upisa označava mogući pogodak. Čvorovi se zatim grupišu ako je podudaranje identifikatora moguće. Rezultati podudaranja u relevantnim čvorovima se uklanjaju iz bilo kakvih budućih rezultata. Ovo odsecanje obezbeđuje da je samo najveći zajednički roditelj uključen u set podudaranja. Kraj analize označava ispaljivanjem *end event*-a.

Alat vraća niz objekata koji sadrže imena fajla, *start*, *end* i linije koje su povezane sa svim instancama pogotka. Taj niz dalje ima mogućnost da se konvertuje u neki od ponuđenih i spomenutih reportera.

3.2.3 Reporting

Osim inicijalnog reportera, dostupni su i JSON i stil PMD CPD XML reporteri. U JSON obliku su primenjeni uvlačenje i formatiranje. Dalje, `property` id koji je dostupan u ovim reporterima je koristan za parsiranje rezultata od strane automatskih skripti kako bi se utvrdilo da li se duplicirani kod promenio između *build*-ova.

Korisniku se rezultati vraćaju kao niz objekata koji u sebi sadrže po jedno podudaranje, odnosno pronađeni duplikat u dva fajla koji ga sadrže. Kako bi se ovakvi rezultati pregledno prikazali, preko kreirane aplikacije, osmišljeno je da se iz liste dobijenih objekata, klikom otvaraju oba fajla koja taj objekat obuhvata, a njihov duplikat označava posvetljavanjem linija na kojima se nalazi. Sumirani rezultati u vidu broja klonova, duplikata, pretraženih fajlova, procenta dupliciranosti koda i broja dupliciranih linija predstavljeni su komponentom u donjem levom uglu aplikacije.

4. EVALUACIJA RADA ALATA ZA PRONALAZENJE DUPLICIRANOG KODA

Kako bi performanse alata JSInspect bile zaista evaluirane korišćen je još jedan alat za pronalazjenje dupliciranog koda JSCPD, koji radi na potpuno drugačiji način. JSCPD alat radi na principu najslabijem Rabin-Karpovom algoritmu [5]. Na osnovu tokena kreira heš vrednosti. Prvo prepozna sve fajlove koji bi trebalo da budu pročitani. Zatim prolazi kroz svaki fajl, čita liniju po liniju iz koje prepoznaje tokene. Nakon što su prepoznati tokeni, u svakom koraku iterator se pomera za jedan token. Uzima se onoliko tokena, u zavisnosti od parametra koji je pristigao kao minimalan broj tokena koji se prepoznaje kao duplikat. Zatim se od tih tokena kreira heš vrednost, koja se poredi sa mapom koja sadrži sve dotadašnje heš vrednosti. Ukoliko takav heš ne postoji, u mapu se dodaje novi heš. Heš funkcija je kreirana tako da može da razlikuje ključne reči od identifikatora. Ukoliko se neke dve funkcije ponašaju isto, sa različitim imenima varijabli, njihov heš kod biće identičan.

4.1 Prikupljanje

U svrhu analize rada ovih alata bilo je potrebno pribaviti par projekata sa velikim brojem JavaScript fajlova. Ovaj zadatak je lako urađen zbog velike količine *open-source* projekata na internetu.

Tabela 1: Projekti korišćeni za analizu i njihove veličine

Projekat	Veličina	Broj fajlova (total/JS)	Broj linija
Projekat1	12KB	4/4	84
Projekat2	21.8MB	1611/1600	314.575
Projekat3	473MB	15.001/6977	523.120

4.2 Analiza i rezultati

S obzirom na to da alati rade na potpuno drugačiji način prvo je trebalo utvrditi pod kojim parametrima vraćaju slične rezultate. U ovu svrhu analiziran je prvo folder od 7 fajlova, od kojih je 4 sa .js ekstenzijom.

Merenja su vršena sa default-nim parametrima oba alata i sa minimalnim parametrima (JSInspect: threshold = 10, JSCPD: minLines = 5, minTokens = 5) za koje je utvrđena najveća sličnost u vraćenim rezultatima. Rezultati merenja su prikazani tabelarno za svaki projekat pojedinačno, pokretana su oba alata za svaki odabrani ulaz. Vreme obuhvata period od klika na dugme modalnog dijaloga do krajnjeg prikaza rezultata na veb stranici. Broj fajlova predstavlja broj fajlova u kojima su pronađeni duplikati, a broj linija broj onih linija koje duplikati obuhvataju.

Tabela 2: Sumiranje rezultata za Projekat1

Alat	Parametri	Vreme [ms]	Broj fajlova	Broj linija
JSInspect	10	310	4	127
JSCPD	5/5	750	4	69
JSInspect	30	310	0	0
JSCPD	5/70	560	3	42

Tabela 3: Sumiranje rezultata za Projekat2

Alat	Parametri	Vreme [min:sec]	Broj fajlova	Broj linija
JSInspect	10	01:50	693	124.717
JSCPD	5/5	01:50	763	150.003
JSInspect	30	02:02	350	55.670
JSCPD	5/70	02:05	273	41.036

Tabela 4: Sumiranje rezultata za Projekat3

Alat	Parametri	Vreme [min:sec]	Broj fajlova	Broj linija
JSInspect	10	03:49	1475	198.031
JSCPD	5/5	03:01	1533	260.195
JSInspect	30	02:26	846	85.727
JSCPD	5/70	01:48	665	62.814

Nakon obavljenih merenja u predstavljenim tabelama se vidi da su alati slični što se tiče vremena izvršavanja, međutim tokom rada nad velikim projektima primećena je velika mana JSInspect alata što se tiče zauzeća memorije. JSCPD alat je nad svakim projektom radio bez problema na default-nom memorijskom zauzeću od 512MB koje Node.js nudi, međutim, zbog prethodno pomenutog problema JSInspect alata, memorija za izvršavanje aplikacije je proširena na 4GB. Daljim tumačenjem dobijenih rezultata, primećena je mnogo veća preciznost pronađenih klonova, kao i veći broj pogodaka kod JSCPD alata. Iz prethodne tri tabele se vidi da je

JSInspect alat skoro uvek vraćao veći broj dupliranih linija koda, ovo se dešavalo zbog toga što njegova podudaranja često obuhvataju iste linije koda unutar istog fajla, što je još jedna mana ovog alata. Zaključak analize, na osnovu svih prethodno iznetih stavki, bi bila preporuka da se JSInspect alat koristi isključivo na manjim projektima, dok kod velikih projekata bi bilo korisnije upotrebiti analizu JSCPD alata.

5. ZAKLJUČAK

U ovom radu ukratko je napravljen uvod u JavaScript programski jezik kako bi se ukazalo na to kako je došlo do povećanja problema kakav je pojava dupliciranog koda, a spram toga i kako je došlo do pojave sve većeg broja alata za statičku analizu koda. Za dalju analizu odabran je alat čiji je cilj nalaženje klonova koda u isključivo JavaScript fajlovima, pa je spram toga opisan ukratko i njegov način rada. Jedan deo rada je posvećen i AST predstavi koda kao izrazito važnom delu ovakve analize. Kreirana je veb aplikacija u cilju integracije više alata za statičku analizu ne bi li se napravilo poređenje performansi alata kreiranih na različit način, primenom različitih algoritama, ali sa istim, zajedničkim ciljem: nalaženjem dupliciranog koda.

6. LITERATURA

- [1] N.Synytsky, J.R.Cordy, and T.Dean. "Resolution of static clones in dynamic web pages". In: Web Site Evolution, 2003. Theme: Architecture. Proceedings. Fifth IEEE International Workshop on. IEEE, 2003.
- [2] F. Cafeto, F. Lanubile, and T. Mallardo. "Function clone detection in web applications: A semiautomated approach". In: Journal of Web Engineering 3 , pp. 3–21, 2004.
- [3] Ira D. Baxter, Andrew Yahin, Leonardo Moura, Marcelo Sant'Anna, and Lorraine Bier. "Clone Detection Using Abstract Syntax Trees". In: Proceedings of the International Conference on Software Maintenance. ICSM'98. Washington, DC, USA: IEEE Computer Society, 1998.
- [4] Douglas Thain, "The Abstract syntax tree", In: Compilers and language design, <https://www3.nd.edu/~dthain/courses/cse40243/fall2016/chapter6.pdf> , 2016.
- [5] Robert Sedgewick, Kevin Wayne, "Algorithms, 4th Edition" , 2011.

Kratka biografija



Tamara Letić rođena je 05.05.1994. u Novom Sadu, Srbiji. Završila je Osnovnu školu "Jovan Jovanović Zmaj" u Rumi, posle čega upisuje Gimnaziju "Stevan Puzić", društveno-jezički smer. Fakultet tehničkih nauka, smer Elektroenergetski softverski inženjering, upisuje školske 2013/2014. godine. Osnovne studije je završila 2017.godine, nakon čega je upisala master studije, smer Primenjeno softversko inženjerstvo.

METODE OPTIMIZACIJE SERVISA ZA UPRAVLJANJE ALARMIMA U PROCESNOJ INDUSTRIJI**OPTIMIZATION METHODS OF ALARM MANAGEMENT SERVICE IN THE PROCESS INDUSTRY**

Aleksandar Duduković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je prezentovana ideja pametne alarmne supresije u elektroenergetskom sistemu kao i način primene veštačke inteligencije kao podrške operateru tokom ispada mreže

Ključne reči: Pametno alarmiranje, Alarmna supresija, Procesna industrija, Mašinsko učenje, Veštačka inteligencija

Abstract – The document presents the idea of smart alarm suppression in the electric power system as well as the method of applying artificial intelligence to support the operator during the failure of the network.

Keywords: Smart alarming, Alarm suppression, Process industry, Machine learning, Artificial intelligence.

1. UVOD

Rad će se bazirati na alarmiranju u procesnoj industriji, tačnije u elektroindustriji. Osvrnuće se i na sam posao operatera u elektrodistributivnoj SCADA (Supervisory Control and Data Acquisition) stanici koji nadgleda električnu mrežu i kako da se taj rad i olakša. Rad će takođe predstaviti primenu veštačke inteligencije u procesnoj industriji sa osvrtom na automatizaciju upravljanja prilikom ispada mreže iz normalnog radnog stanja.

Alarmi i sami alarmni sistemi su neophodni u svakom procesnom postrojenju kako bi se sprečio otkaz uređaja koji bi kasnije mogao kulminirati havarijom. Kako je tehnologija napredovala tako su i uređaji postali „pametniji“ i složeniji, dobili su više funkcionalnosti, fleksibilniji su, spram toga je i teže upravljati njima. Najveći prioritet što se tiče bezbednosti (eng. *safety*) u industriji je sačuvati ljudski život i prirodu, sprečiti ekološku katastrofu i oštećivanje kritičnih infrastrukture.

Havarija nije jedini razlog zašto se alarmi implementiraju u procesnim industrijama, iako predstavlja najprioritetniji, nego i da produži rok trajanja samom uređaju jer će se boljim uvidom u njegovo stanje optimizovanje rukovati samim uređajem pa i celokupnim sistemom. Oprema koja se koristi u elektroindustriji je veoma skupa, a nekad je i unikatno pravljena samo za projektovanu mrežu, tako da je optimalno upravljanje skupocenim uređajima veoma bitno.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Čapko, vanr. prof.

2. ANALIZA PROBLEMA

Da bi se objasnili problemi koji nastaju usled neadekvatno postavljenog alarmnog sistema i načini na koje ove probleme rešiti mora se prvo definisati šta je to alarm i alarmno stanje.

Po ISA-18.2 standardu [1] alarm je: zvučni i/ili vizuelni indikator operateru da postoji neispravnost u opremi, devijacije u procesima ili da su se pojavili abnormalni uslovi koji zahtevaju blagovremeni odgovor operatera.

Problemi se mogu javiti kada postoji premalo ili previše alarma. Kada postoji premalo alarma, operater nema pravi uvid u sistem koji nadgleda, a kada ih ima previše, dešava se pojava koja se naziva „alarm flood“ (alarmna poplava) i tada je operater sprečen da odreaguje blagovremeno na trenutnu situaciju jer je vreme između prijave dva alarma veoma kratko. Realan problem loše postavke alarmnog sistema može se videti u eksploziji „Texaco“ rafinerije (Milford Haven) 24. jula 1994. godine [2].

Dva operatera koja su radila na nadziranju sistema dobili su, pre same havarije, 275 različitih alarma u 11 minuta. EEMUA (Engineering Equipment and Materials Users Association) [3] je u svom dokumentu EEMUA 191 propisala *de facto* standard za broj alarma koji je prihvatljiv da se dogodi u sistemu za određen vremenski interval da bi operater mogao da ih isprocesuirati, a to je jedan alarm na svakih 10 minuta.

Iz njihovih podataka mogu se izvući zaključci da operateri u Texaco rafineriji nisu mogli brže i bolje da reaguju na toliko alarmnih događaja i da prepoznaju koji alarm je posledica, a koji je uzrok drugog alarma. Ranija mišljenja eksperata i ljudi u vladi bila je da je greška uvek u operaterima, tj. da je havarija posledica ljudskih grešaka, ali je EEMUA 191 jasno stavila do znanja da ako alarmna komponenta nije uspela da redukuje učestanost prijave alarma onda se operater ne može okriviti zato što je propustio važan alarm. Zbog toga je alarmna supresija veoma bitan faktor u alarmnoj komponenti, isto koliko i samo definisanje alarma.

3. METODE SUPRESIJE

Tipovi supresije koji mogu biti implementirani radi optimizacije alarmnog servisa su:

- Dinamička prioritizacija alarmnih stanja po njihovoj važnosti (eng. *importance*),
- *Out of service* supresija,

- Objedinjavanje različitih tipova alarma kreiranjem topoloških ostrva,
- Obeležavanje alarma po njihovom uzroku nastanka/posledica (eng. *root cause*),
- Ručna, kroz HMI (Human Machine Interface) i
- Supresija nivelacionih alarma.

Dinamička prioritizacija alarma omogućava da alarm koji već postoji u sistemu promeni njegovu važnost i ozbiljnost (eng. *importance and severity*) usled uticaja drugih akcija u sistemu koji direktno ili indirektno utiču na element i/ili povezane elemente na kojim se nalazi alarm. Kombinovanjem ove i metode koja objedinjuje alarme po uzroku nastanka/posledica najbolje se može pokazati na primeru kompresora u OGS industriji (Oil and Gas System). Kompresori imaju mnogo delova u sebi koji mogu da rezultuju alarmom i da ukažu na neke abnormalne uslove koji se dešavaju u njemu kao na primer nizak pritisak ulja, nizak pritisak usisavanja, temperatura ležaja itd.

Operator ne treba da vidi sve alarme u slučaju da kompresor počne da otkazuje i da na kraju prekine svoj rad kompletno. Ono što je tada bitno operateru je da ima alarm koji govori da je kompresor „ispao“ iz mreže bez pojedinačnih alarma elemenata koji se nalaze u kompresoru [4]. Ovim ne samo da se odradila supresija *root cause* metodom već alarmni servis može da promeni ozbiljnost (eng. *severity*) alarma i da je poveća, ukoliko ispad kompresora ugrožava postrojenje i ljude.

Out of service supresija bi trebala da postoji u svakom softveru koji nadgleda procesne sisteme. Kada se planira popravak jednog uređaja ili jednog dela sistema operater je skoro uvek obavešten o tome. Postavljanjem odgovarajućeg tag-a u softveru na uređaj ili deo sistema *Out of service* supresija može biti uključena. Tada operater, a i softver znaju da je uređaj odnešen na remont stoga alarmi za taj element operateru ne predstavljaju nikakvu novu informaciju i samo se gomilaju u listi alarma što otežava rad operatera.

Postoje događaji u sistemu koji nisu čisto alarmno stanje vezani za merenje, ali operateru itekako daju informacije sa terena. Jedan tip takvih događaja je i incident, koji pripada OMS-u (Outage Management System) i predstavlja planirane popravke, neplanirane i planirane ispade u sistemu i omogućava operaterima bolji pregled stanja na terenu i omogućava slanje ekipa za popravke (eng. *working crews*). Objedinjavanje više odvojenih sistema (SCADA i OMS u ovom slučaju) je moguće preko logičkog kreiranja topološko deenergizovanih ostrva kao elemenata u sistemu.

U elektroenergetskoj mreži ispadanjem jednog prekidača dobija se niz pojedinačnih alarma na elementima nizvodno od „problematičnog“ prekidača ukazujući na nedostatak napona, i tako se gomilaju isti alarmi u aktivnoj alarmnoj listi. Ako se uvede pojam topološko deenergizovanog ostrva, tada „problematični“ prekidač usled ispada postaje koren novog stabla koje je deenergizovano. Pre prijave incidenta na sam prekidač kao izolovan objekat, treba se proveriti da li je taj element pripadnik nekog deenergizovanog ostrva, i ako jeste treba

naći njegov koren i tu napraviti uniju svih incidenata i alarma i označiti ga kao „Problem“. Problem bi tada bio kompleksan događaj koji u sebi sadrži više događaja, ali bi predstavljao čvorište svih problema koji su posledica ispada tog elementa i tako bi operater imao sažetiji pregled sistema. Logičkim dodavanjem topološkog deenergizovanog ostrva koje ne postoji kao fizički element u mreži, dobija se mogućnost klasterizacije pojava na pojedinačnim elementima u ostrvu.

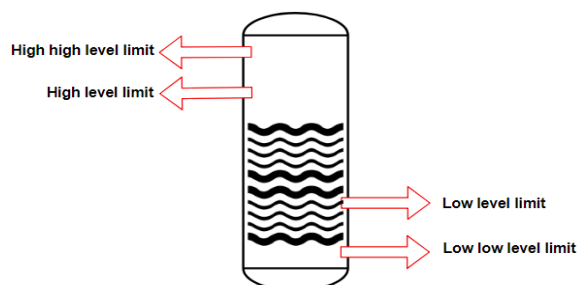
Ručan vid supresije je takođe bitan faktor u softveru koji omogućava operateru da definiše svoja pravila po kojima će se supresija dešavati. Kriterijumi u ovom vidu supresije mogu biti razni i zavise isključivo od sposobnosti operatera da primeni odgovarajuće filtere u datom trenutku.

U toku „alarmne poplave“ operater može da izabere da mu prvo stižu alarmi sa većim prioritetom, da vidi prvo alarme od prioritetnijih potrošača itd. Ovakav pristup bi se mogao i automatizovati ako je softver u stanju da prepozna kada je došlo do navale alarma, što može da se postigne merenjem učestanosti prijave alarma.

Da bi se smanjio broj nepotrebnih alarma bitno je da softver ima supresiju alarma koji su vezani za alarmna ograničenja na pojedinim senzorima/tačkama. U raznim OGS, vodovodnim, petrohemijskim industrijama postoje rezervoari koji rade sa tečnim sirovinama i u njih je zbog bezbednosti ugrađeno više bezbednosnih senzora za merenje nivoa tečnosti (Slika 1).

Ako se desi scenario gde će se nivo tečnosti kontinualno podizati jer se pumpa za punjenje zaglavila, prvo će se generisati High level limit alarm da upozori operatera na visok nivo tečnosti u rezervoaru. Ako se pumpa ne zatvori u neko predefinisano vreme, tečnost će nastaviti da raste i aktivirati *High high level limit* (Slika 1) da upozori na sada već kritičan nivo u rezervoaru i mogućnost izlivanja tečnosti u postrojenje.

U ovakvom scenariju operater ima dva aktivna alarma od kojih je jedan zastareo i podrazumevan jer postoji isti alarm većeg prioriteta koji nosi informaciju o dostizanju većeg alarmnog nivoa. U ovakvim situacijama mora se odraditi rekalkulacija alarma i skloniti podrazumevani alarmi da se slični njima ne bi gomilali u listi aktivnih alarma. U elektroindustriji je ovaj vid supresija takođe mnogo bitan jer tamo mogu biti definisani i po više od četiri alarmna nivoa za jedno merenje.

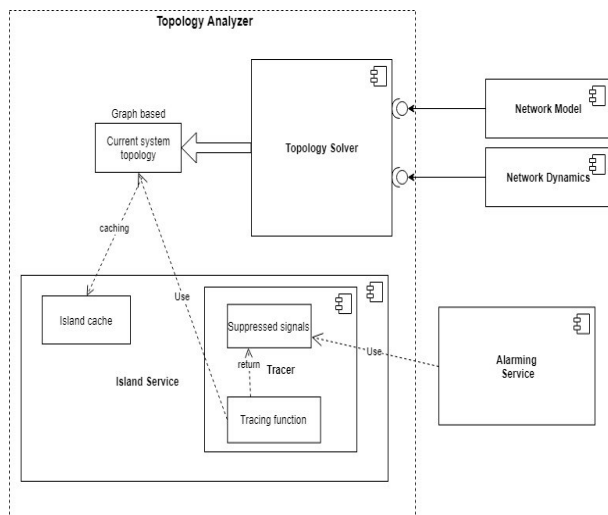


Slika 1 – Rezervoar sa definisanim alarmnim limitima

Softver sa automatskom supresijom ne bi smeo da sakrije supresovana alarmna stanja već treba da omogućući operateru uvid u takve alarme ali tako da ta lista i/ili grafički prikaz ne bude u prvom planu.

4. ARHITEKTURA REŠENJA

Struktura rešenja prikazana je na slici (Slika 2).



Slika 2 – Arhitektura rešenja

Osnovne komponente od značaja za supresiju su:

1. Topology solver
2. Island service
3. Tracer
4. Alarming service

Topology solver komponenta prima statičke i dinamičke podatke o entitetima u mreži i na osnovu konektivnosti pravi graf o aktuelnom uklopnom stanju mreže. *Island service* koristi aktuelno stanje mreže i kešira podatke o deenergizovanim ostrvima pomoću *Tracer* komponente koja aktivira *Tracing* funkciju i popunjava ili pristupa *Island* keš podacima, koja potom pristupa svim deenergizovanim entitetima u ostrvu i kupi njihove signale, kao i signale opreme koja se nalazi u entitetima (npr. zaštitna oprema na prekidačima, rastavljačima, osiguračima itd.). Ti signali se šalju na *Alarming service* gde će posle biti supresovani.

5. PRIMENA VEŠTAČKIH NEURONSKIH MREŽA U PROCESNOJ INDUSTRIJI

Predstaviće se idejno rešenje i arhitektura restauracije mreže prilikom ispada korišćenjem neuronskih mreža. U ovom radu će se analizirati mogućnost primene *OpenAI Five* algoritma [5] koji je implementiran u video igri Dota 2.

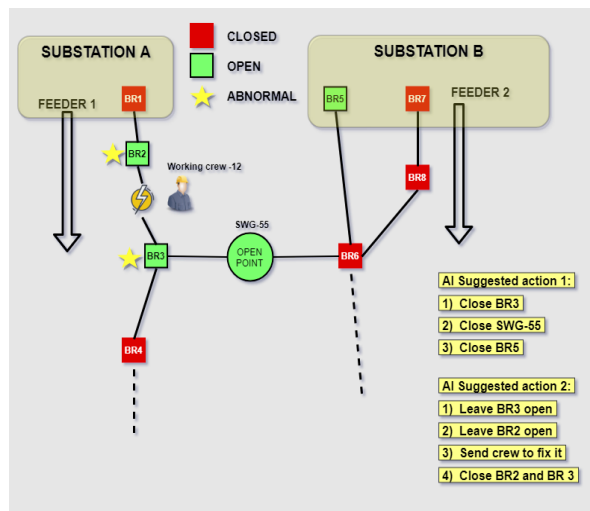
Krajnji, ultimativni cilj koji algoritam treba da održi, odnosno da ispuni je prazna lista alarma. Kada nema nijednog alarma to znači da je sistem, u ovom slučaju mreža, u normalnom radnom stanju. Supresijom je pokušano da se što manje nebitnih alarma pojavi u aktivnoj listi alarma ostavljajući samo one najprioritetnije. Ovo pomaže neuronskim mrežama da rade sa najprioritetnijim alarmima dajući im preciznije informacije o mestu i uzroku grešaka. Zatim treba definisati agente (igrače) koji treba da izvršavaju akcije u sistemu.

Ova stavka može da se mapira na više entiteta u mreži, a neki od primera su:

- Svaki glavni izvod je jedan agent,

- Svaka podstanica je jedan agent,
- Svaki region je jedan agent.

Ovo je moguće samo iz razloga što se pokazalo da agenti u *OpenAI Five*-u znaju da sarađuju međusobno ka ispunjenju dugoročnog cilja. Ako se uzme slučaj da je svaka podstanica jedan agent u sistemu, tada bi svakoj podstanici dodelili upravljanje određenim brojem izvoda koji izlaze iz te podstanice. Kratkoročni cilj svake podstanice je da održava svaki izvod živim, odnosno energizovanim. Algoritam bi trebao da nauči, kao i heroji u Dota 2, da žrtvuju svoje akcije zarad konačnog cilja. Primer toga je ako se definišu neki izvodi kao pomoćni u podstanici i puštanjem u rad tih izvoda bi svakoj podstanici proizvodilo penale od sistema. Ali ako susedna podstanica prepozna da ima problem sa energizacijom jednog dela svog izvoda tada algoritam treba da prepozna da ako energizuje izvod prve podstanice i zatvori prekidače (koje inače ne bi smeo) tada bi se penali od sistema kompenzovali time što je glavni cilj algoritma ispunjen, tj. nema alarmnih stanja u aktivnoj listi alarma (Slika 3). Tada bi ta akcija dobila manje penale i ostala bi zapamćena kao jedan od koraka koji su proizveli kasnije dobitke.



Slika 3 – AI preporuke za reenergizaciju mreže

Slika 3 prikazuje jedan od načina kako bi AI (Artificial Intelligence) mogao da asistira operateru da što brže vrati mrežu u normalno stanje. Slika 3 prikazuje dva izvoda (FEEDER 1 i FEEDER2) koji pripadaju različitim podstancama, A i B. Oba izvoda su energizovana pod normalnim uslovima rada, a prekidač BR5 je otvoren i on predstavlja pomoćni izvod podstanice B. Dešava se ispad na izvodu "FEEDER 1" između prekidača BR2 i BR3, prekidačka oprema se otvara i nizvodno od prekidača BR3 izvod je deenergizovan. Podstanica A javlja ostalim podstancama da joj je potrebna pomoć jer ne može sama da energizuje izvod. Podstanica B prepoznaje da može da pomogne. Akcije koje algoritam može da izvrši mogu da budu samo preporuka operateru ako u softveru ne postoje kontrolne sekvence, ako postoje onda bi algoritam mogao i sam da izvrši korake koje je definisao u preporuci sa tim da bi izabrao povoljniju akciju.

Prva akcija bi se primenjivala ako se na prvom izvodu nalazi priključena bolnica jer je tada bitno odmah reenergizovati izvod u što kraćem roku. Prva akcija

zahteva od algoritma da bude svestan opterećenja koja mogu da se dese ako se samo prespoji BR3 na BR6. Zato je potrebno aktivirati još jedan izvor napajanja. Ako se na prvom izvodu nalaze manje bitniji potrošači, onda će algoritam birati opciju 2 da se kvar na izvodu odmah popravi.

Postavlja se pitanje kako algoritam da prepozna ko je bitniji potrošač. Kako se u video igri AI vraćao da brani bazu i da ostale manje bitne akcije, koje su do tog trenutka (napada na bazu) bile najprioritetnije, ostaju zapostavljene, tako bi se ponašao AI i u elektroindustrijskim sistemima tako što će se reći u okruženju da su sve bolnice baze koje treba braniti, jer će nagrada za to biti najveća u sistemu.

6. ARHITEKTURA IDEJNOG REŠENJA OPEN AI FIVE ALGORITMA U ELEKTROINDUSTRIJI

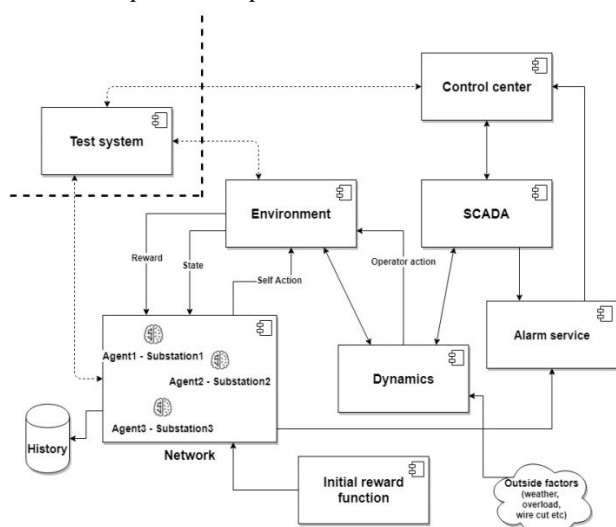
Idejno arhitekturno rešenje prikazano je na slici 4. Podstanice su izabrane kao agenti sistema gde svaka podstanica predstavlja zasebnu neuronsku mrežu. Na početku svim agentima je prosleđena lista nagrada kao težinski faktori grana kojima njihove neuronske mreže treba da se vode.

Statički se odredi normalno stanje sistema i u neuronske mreže agenata se unose podaci: sa kojim izvodima upravlja agent, sa kojim podstanicama mogu da se prespoje, šta im je glavni cilj itd.

Ovakav skup mreža radi zasebno ali sa istim glavnim i dugoročnim ciljem, ostaviti aktivnu listu alarma praznu. Svaki napredak koji naprave može se sačuvati u bazi podataka kako bi bio kasnije dostupan tokom evaluacije koraka koji se će biti primenjeni na sistem odnosno okolinu (*Environment*).

Environment ustvari predstavlja sve čime podstanice rukuju: izvode, prekidače, potrošače itd. i svi ti entiteti mogu biti komandovani od strane tri učesnika:

- Same neuronske mreže,
- Operatera i
- Spoljnih faktora kao što su vremenske neprilike, neplanirani ispadi itd.



Slika 4 – Arhitektura optimizacije primenom neuronskih mreža

Svaki put kada se mreža izmeni okruženje agentima daje novo stanje i nagradu za njihovu akciju. Nagrada će biti nula ako agenti nisu primenili nikakvu akciju već je rezultat promene mreže došao od operatera ili spoljnih faktora. Testnim sistemom operater može da isprobava predložene akcije neuronskih mreža u realtime-u.

7. ZAKLJUČAK

Rad je analizirao i detektovao probleme koji se događaju prilikom velikih alarmnih “poplava” u industriji, posredstvom vremenskih nepogoda i/ili loše iskonfigurisanog alarmnog sistema, i to demonstrirao kroz primer Texaco rafinerije. Takođe je predstavljeno jedno od rešenja pametne alarmne supresije koja će objediniti alarme po njihovom uzroku nastanka i dati operateru uniformnu informaciju problema kreiranjem topološko deenergizovanih ostrva. Pored ove metode prikazano je i više metoda supresije koje mogu da se primene u softveru kako bi se smanjio broj alarma u aktivnoj alarmnoj listi.

Konceptualno je predstavljeno novo i inovativno rešenje u polju veštačke inteligencije sa njegovom mogućom primenom u elektroindustriji. Prikazano je kako bi veštačka inteligencija mogla da pomaže operaterima ili čak i sama da održava sistem. Data je idejna arhitektura takvog sistema i lista nagrada koju bi neuronske mreže (agenti) trebalo da koriste tokom trening procesa.

Mogući pravci daljeg razvoja predstavljenih rešenja sastoje se u tome da se fizički isproba idejno rešenje veštačkih neuronskih mreža. Takođe, da se pronađu još neki slučajevi njihove primene radi uštede operatorskog odziva na abnormalno stanje sistema, koji ne moraju biti orjentisani ka redukciji alarma.

8. LITERATURA

- [1] ANSI/ISA-18.2-2016, Management of Alarm Systems for the Process Industries, 2016
- [2] Dustin Beebe, Darwin Logerot, Steve Ferrer, Examination of Events that Occur During an Alarm Flood, 2015
- [3] The Engineering Equipment and Materials Users Association (EEMUA) publication No. 191, Alarm Systems: A Guide To Design, Management and Procurement, 2013
- [4] Charlie Fialkowki, Automation IT: Using alarm suppression, 2012
- [5] OpenAI – blog.openai.com

Kratka biografija:



Aleksandar Duduković je rođen 24.06.1994. godine u Somboru. Osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu završio je 2017. godine na smeru Primenjeno softversko inženjerstvo. Iste godine, na istom fakultetu, upisuje master akademske studije, smer Primenjeno softversko inženjerstvo.

Kontakt: acans94@gmail.com

MOGUĆNOSTI PRIMJENE MODELA PIKO HIDROELEKTRANE POSSIBILITY OF APPLICATION OF PICO HYDRO POWER PLANT

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

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu urađeno je idejno rešenje piko hidroelektrane koja proizvodi električnu energiju za sopstvenu potrošnju jednog domaćinstva. U sklopu idejnog rešenja obrađeni su svi ključni elementi koji se rade prilikom projektovanja. Biće izložen hidrološki potencijal Srbije. Tehnička izvodljivost projekta se analizira kroz softverski paket HOMER. Bavi se ciljevima izgradnje malih hidroelektrana i njihovim uticajem na okolinu.

Ključne reči: piko hidroelektrana, sopstvena potrošnja, modelovanje sistema

Abstract – In this paper, the preliminary design of the pico hydro power plant that produces electricity for own consumption of one household. As part of the preliminary design, all the main elements that are being worked out during the design process are processed. The hydrological potential of Serbia will be exposed. The technical feasibility of the project is analyzed through the HOMER software package. It deals with the objectives of building small hydropower plants and their impact on the environment.

Key words: pico hydropower, own consumption, system modelling

1. UVOD

U okviru energetskog sistema Republike Srbije obavlja se eksploatacija domaće primarne energije (uglja, nafte, prirodnog gasa, obnovljivih izvora energije), uvoz primarne energije (prije svega nafte, prirodnog gasa i uglja), proizvodnja električne i toplotne energije, eksploatacija i sekundarna prerada uglja, kao i transport i distribucija energije i energenata do krajnjih potrošača finalne energije. Ukupna potrošnja finalne energije u Srbiji u 2014. godini iznosila je 7,67 Mtoe [1]. Posmatrano po sektorima potrošnje široka potrošnja učestvuje sa 36%, industrija sa 29%, saobraćaj sa 23%, poljoprivreda 2% i ostale delatnosti 10% ukupne potrošnje finalne energije. U strukturi potrošnje finalne energije dominiraju derivati nafte sa 32% i električna energija sa 26%, zatim slede ugalj sa 11%, prirodni gas sa 12 %, toplotna energija sa 8%, dok obnovljivi izvori energije (ogrevno drvo) učestvuju sa 11%. Bitna razlika u strukturi potrošnje finalne energije, u odnosu na Evropsku uniju, je visok udeo potrošnje u domaćinstvima u Srbiji i

mного viši udeo potrošnje energije u saobraćaju u EU [1]. Geomorfološke i hidrološke karakteristike svrstavaju Srbiju u red zemalja bogatih vodom, te ukupni raspoloživi vodni potencijal nije za zanemarivanje. Ukupni hidroenergetski potencijal svih srpskih vodotoka je procijenjen na 31.000 GWh godišnje [1].

2. HIDROELEKTRANA

Da bi se izvršila podjela hidroelektrana (HE), bitne su sledeće karakteristike: snaga, visina pada vode i način korišćenja vode. HE je skup različitih elemenata koji su u takvom međusobnom odnosu da daju bolji konačni rezultat od onog kojeg mogu pojedinačno dati njegovi dijelovi [2]. U zavisnosti od snage razlikuju se velike HE (>100 MW), male HE (500 kW-10 MW), mini HE (100 kW-500 kW), mikro HE (5 kW-100 kW) i piko HE (<5 kW). Samo su mikro i piko HE predviđene za samostalan rad, dok ostale proizvedenu energiju isporučuju u elektroenergetski sistem.

3. PIKO HIDROELEKTRANA

Naziv ovog sistema potice iz riječi *pico*, što znači "vrlo malo". Tehnički rečeno, to je sistem koji pretvara energiju kretanja vode u električnu snagu maksimalne vrijednosti 5 kW. Princip rada je isti kao za sve HE, a oprema je tehnički dimenzionisana za malu snagu. Gabaritne dimenzije su takve da imaju minimalan uticaj na uslove sredine. Konstruiše se kao protočna HE, bez potrebe za akumulacijom, tako da voda direktno ide do turbine, a potom se vraća u svoj tok. Postoji mnogo faktora koji određuju podobnost sistema. Ovo uključuje:

1. Količina snage koja je dostupna na izlazu cijevovoda. Ovo je određeno zapreminom dostupne vode, pritiskom vode i gubicima na trenje u cijevovodima.
2. Tip turbine i tip i kapacitet generatora.
3. Tip i kapacitet opterećenja piko HE.
4. Troškovi razvoja projekta i rada sistema

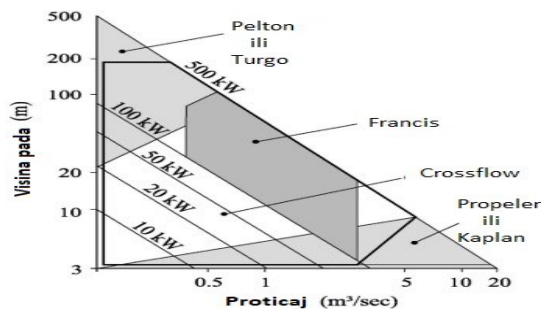
Za mjerenje zapremine protoka vode, izdvojene su dvije najjednostavnije metode za čije izvođenje nisu neophodni stručni mjerni instrumenti: metoda kofe („*bucket method*“) i metoda plovka. Prilikom određivanja visine pada vode treba uzeti u obzir bruto (statičku) i neto (dinamičku) visinu pada.

Bruto visina je vertikalno rastojanje između vrha cjevovoda i tačke gdje voda udara u turbinu.

Neto visina je razlika bruto visine pada i gubitaka nastalih zbog trenja i turbulencija u cjevovodu. Procjena visine pada i protoka vode kod piko hidro-sistema dalje utiče na izbor turbine. Odnos visine pada, količine proticaja, snage i tipa turbine prikazan je na slici:

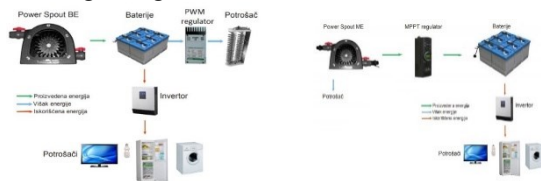
NAPOMENA:

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



Slika 1. Primjenjeni tip turbine u zavisnosti od raspoloživog pada i proticaja [3]

Za realizaciju sistema piko HE moguće je upotrijebiti hidroturbine proizvođača PowerSpout. Dijelovi hidroturbina su pretežno izrađeni od recikliranih materijala. Power Spout BE (Battery Enabled) i Power Spout ME (MPPT Enabled) hidroturbine su predviđene za sisteme koji rade samostalno, nezavisno od elektroenergetskog sistema.



Slika 2. Način primjene PS BE i ME hidroturbine [4]

Opisan je model piko hidroelektrane koji je realizovan preko improvizovanih radnih dijelova prikazan je na slici 3. Ovaj sistem piko HE ima snagu od 1,5 kW. Hidroelektrana koja je predstavljena pripada grupi protočnih piko hidroelektrana sa malim padom. Sistem čine sledeći dijelovi: turbina, remenica, mjenjac, alternator (generator jednosmjerne struje), akumulator i invertor. Električna energija koju proizvodi prezentovana piko hidroelektrana koristi se za osnovne potrebe jednog domaćinstva: osvetljenje i napajanje standardnih kućnih aparata (radio, frižider, punjač za mobilni telefon).



Slika 3. Dijelovi i povezivanje u sistem hidroelektrane [5]

4. MODELOVANJE PIKO HE

Maseni protok može se izraziti kao proizvod zapreminskog protoka Q u m^3/s i gustine ρ u kg/m^3 , te se dobija izraz za snagu vodene mase na ulazu u turbinu:

$$P = Q * \rho * g * H_{net} \quad [W] \quad (1)$$

Raspoloživa snaga na vratilu turbine manja je od snage vode na ulazu u turbinu za iznos gubitaka (hidraulički, zapreminski i mehanički), koji nastaju uslijed strujanja vode kroz turbinu. Odnos snage P_t dobijene na vratilu turbine i snage vode P na ulazu u turbinu predstavlja stepen efikasnosti energetske transformacije:

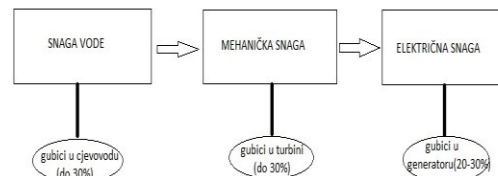
$$\eta_t = \eta_h * \eta_Q * \eta_m \quad (2)$$

$$\eta_t = \frac{P_t}{P} \quad (3)$$

Stepen efikasnosti vodnih turbina se nalazi u granicama od 0,7 - 0,9, pa se raspoloživa snaga na vratilu turbine računa kao:

$$P_t = Q * \rho * g * H_{net} * \eta_t \quad [W] \quad (4)$$

Snaga hidroenergetskog sistema pretvorena je iz jednog oblika u drugi, a takav process podrazumijeva i određene gubitke. Slika 4 ilustruje udio gubitaka.



Slika 4. Raspodjela gubitaka u procesu pretvaranja.

Snaga koju proizvodi generator je proizvod stepena efikasnosti generatora i snage na vratilu turbine. Stepenn efikasnosti generatora za male hidroturbine je u opsegu 0,7-0,8, pri čemu vrijednost 0,8 postiže pri idealnim uslovima. Sledeća relacija predstavlja proračun za proizvedenu električnu snagu:

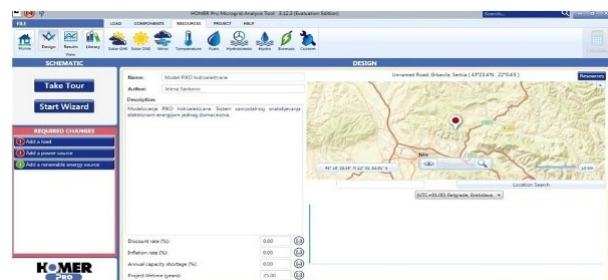
$$P_e = \eta_g * P_t \quad [W] \quad (5)$$

Ako se izvrši integraljenje proizvedene snage u određenom vremenskom ograničenju, dobija se vrijednost proizvedene električne energije:

$$W_e = \int_0^t P_{e(t)} dt \quad [kWh] \quad (6)$$

5. SOFTVERSKI PAKETI ZA PROJEKTOVANJE MALIH I PIKO HIDROELEKTRANA

HOMER (Hybrid Optimization of Multiple Electric Renewables) je kompjuterski program nastao 1992. god. u američkoj Laboratoriji za obnovljive izvore energije NREL (National Renewable Energy Laboratory, Kolorado, SAD). Danas predstavlja globalni standard za optimizaciju dizajna mikro mreža u svim sektorima.



Slika 5. Početna strana programa HOMER

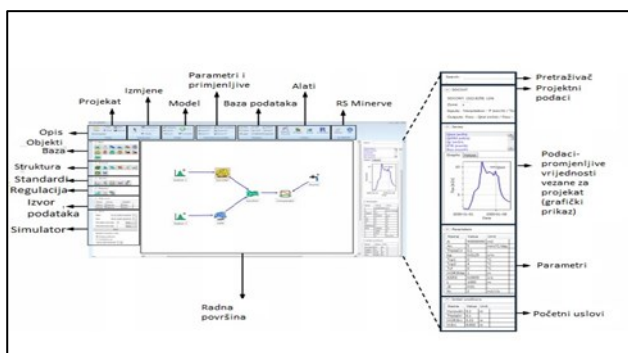
RETScreen je programski paket koji je razvila Vlada Kanade i prva verzija je dostupna javnosti od 1998. god., dok je najnovija verzija predstavljena 2016. god. Omogućava identifikaciju, procjenu i optimizaciju

tehničke i finansijske održivosti projekta. Za već postojeće projekte pomaže u pronalaženju dodatnih mogućnosti uštede energije.



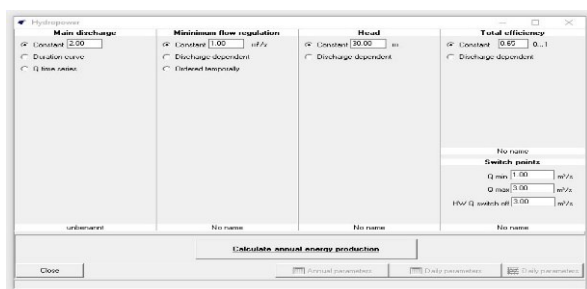
Slika 6. Početna strana programa RETScreen

RSMInerve je softver za simulaciju formiranja slobodnog površinskog protoka i njegovih karakteristika. Modeluje kompleksne hidrološke i hidrauličke mreže.



Slika 7. Početna strana programa RSMInerve

Prva verzija CASiMiR Hydropower programa je razvijena na Institutu za hidrauliku na Univerzitetu u Štutgartu. Nastao je u cilju proučavanja ekonomskih aspekata malih hidroelektrana.



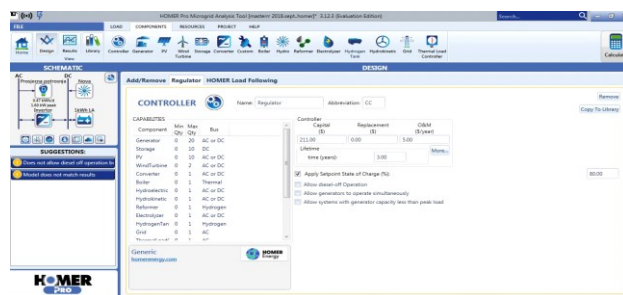
Slika 8. Početna strana programa CASiMiR Hydropower

Raznovrsne situacije rada sistema lako mogu biti izvedene i upoređivane upotrebom tabelarnog i grafičkog prikaza. Za pojedinačno modelovanje tehničkih elemenata sistema pogodan je program HOMER, a nedostatak se ogleda u nemogućnosti modelovanja različitih hidroloških uslova. Pouzdan je u radu, a izvještaji su dostupni u tabelarnom i grafičkom prikazu. Ekonomska analiza sistema biće najdetaljnija ako se izvodi preko modela u RETScreen programu i olakšava proces finansijske procjene projekta, ali izostaju detalji vezani za tehničku i hidrološku stranu sistema. Istaknuta osobina RSMInerve

programa jeste mogućnost procjene i praćenja hidroloških prilika, uzimajući u obzir operacije koje mogu izazvati poremećaje u hidrološkom i ekološkom sistemu, ali je za pravilan rad neophodno učitavanje podataka iz Geographical Information System baze podataka. Od navedenih programa, najskromnije performanse ima program CASiMiR Hydropower, jer proračunava samo godišnju proizvodnju električne energije i procjenjuje finansijsku dobit po fiksnoj cijeni koja nije podložna promjeni. Nepouzdan je u radu, često ne može da se pokrene izvršavanje proračuna i dolazi do prekida rada ako se brzo zadaju promjene opcija.

6. PRIMJER KORIŠĆENJA MODELA ZA PIKO HE

Pregledom mogućnosti četiri programa, za modelovanje piko HE najpogodniji je programski paket HOMER. Elemente sistema postavljamo uz opciju Components. Za hidrosistem biramo komponente: Controller, Storage, Converter i Hydro. Nakon unošenja osnovnih podataka o protoku vode, visini pada i efikasnosti turbine simulacijom uprogramskom paketu HOMER dobijeno je da će hidroturbina snage 1600 W (efikasnost od 53,8%) omogućiti proizvodnju od 1604 W pri visini pada od 38 m i protoku od 8 l/s.

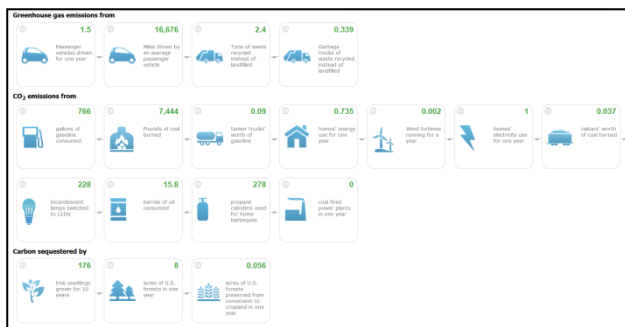


Slika 9. Biranje komponenta sistema

Ovaj program dao je podatak o neophodnim uslovima visine pada i protoka vode radi postizanja nominalne proizvodnje turbine. HOMER računa izlaznu električnu snagu upotrebom sledeće formule:

$$P_{hyd} = \frac{\eta_{hyd} \cdot \rho_{water} \cdot g \cdot h_{net} \cdot Q_{turbine}}{1000} \quad [kW] \quad (7)$$

Olakšao je uvid u raspodelu troškova sistem i procijenio koliko košta svaki proizvedeni kWh. Pri izradi investicionog plana, nezaobilazan korak jeste provjera da li će se projekat isplatiti i u kom vremenskom periodu. Pošto je mogućnost primjene piko hidrosistema ograničena samo na proizvodnju električne energije za sopstvenu potrošnju, takav sistem ne može predstavljati priliv novca. Izgradnji sistema obnovljivih izvora energije pristupilo se u cilju smanjenja upotrebe prirodnih ograničenih resursa (ugalj, nafta) i smanjena emisije zračenja ugljen-dioksida. Na internetu postoje razvijeni konvertori za preračunavanje električne energije u ekvivalentnu količinu uglja, nafte i prirodnog gasa. Upotrebom Unit Juggler, proračunato je da se ostvarila ušteda 1,23 tone mrkog uglja, a upotrebom Greenhouse Gas Equivalencies konvertora procijenjena je ušteda od 7,5 tona ugljen-dioksida.



Slika 10. Ekvivalentne vrijednosti za 7,5 tona CO₂ [7]

7. MALE HIDROELEKTRANE U NAŠEM SISTEMU

Izgradnja hidroenergetskih objekata, pored opštih pozitivnih energetske i vodno-bilansnih efekata, kao posljedicu ima određeni uticaj na životni prostor i okruženje samih objekata. Kvalitetnim projektovanjem, izgradnjom i upravljanjem, MHE se mogu uklopiti u okolinu tako da se nepovoljni uticaji na stanje životne sredine svedu na minimum. Analizama se sagledavaju mogući pozitivni i negativni uticaji planiranih objekata, sa ciljem da se pravovremeno predvide mjere kojima bi se eventualni nepovoljni uticaji sveli na minimum ili u potpunosti eliminisali.

Zapažanje u vezi sa izgradnjom MHE jeste da predstavlja unosan posao za investitore i prijetnju životnoj sredini. Dobijanjem statusa povlašćenog proizvođača električne energije investitor može prodati proizvedenu električnu energiju ovlašćenom distributeru po podsticajnoj cijeni. Prema Uredba o merama podsticaja za povlašćene proizvođače električne energije [6], nakon isteka podsticajnog perioda povlašćeni proizvođač ima pravo sklapanja ugovora sa javnim snabdijevачem o otkupu ukupno proizvedene električne energije po tržišnim uslovima.

Tabela 1. Podsticajne cijene za električnu energiju proizvedenu u HE [6]

Redni broj	Vrsta elektrane povlašćenog proizvođača električne energije	Instalisana snaga P (MW)	Podsticajna otkupna cena (c€/kWh)	Maksimalno efektivno vreme rada (h)
1	Hidroelektrana			5000 u godini podsticajnog perioda
1.1		Do 0,2	12,60	
1.2		0,2 – 0,5	13,933 – 6,667*P	
1.3		0,5 - 1	10,60	
1.4		1 - 10	10,944 – 0,344*P	
1.5		10 - 30	7,50	5000 u godini podsticajnog perioda
1.6	Na postojećoj infrastrukturi	Do 30	6,00	

8. ZAKLJUČAK

Mali hidroenergetski sistemi predstavljaju primjenu hidroenergije u snabdijevanju malih potrošača. Ovakvi sistemi se dijele na mini, mikro i piko hidroelektrane i prepoznati su kao ključne tehnologije u primjeni obnovljivih izvora električne energije za ruralne predjele. Mali hidroenergetski sistemi su principijski slični velikim, samo je izvršeno smanjivanje snage kako bi sistem bio kompatibilan za različite lokacije. U zemljama u razvoju, modeli mikro i piko hidroelektrana su se već pokazali kao praktična rješenja i rješenja potencijalno niske cijene za

proizvodnju električne energije na udaljenim lokacijama, naročito u malim selima u brdovitim predjelima. Mikro hidroenergija ponekad uključuje piko hidroenergiju zato što piko hidroenergija generiše električnu energiju na nivou domaćinstva, zato je ovaj model u mnogim zemljama prozvan kao "porodična hidroelektrana" (family hydro).

Programski paketi za analizu hidrosistema olakšavaju projektovanje, ali je neophodno koristiti različite programe, jer je svaki specijalizovan za određen aspekt sistema. Za procjenu tehničke izvodljivosti sistema najpogodniji je program HOMER jer omogućava procjenu uslova (protok i visina pada) za nominalnu iskoristivost. Detaljnu procjenu hidrološkog aspekta omogućava RSMInerve program, a finansijske tokove obrađuje RETScreen. Program CASiMir Hydropower pruža najmanje mogućnosti za modelovanje sistema, zbog slabo razvijenih funkcionalnosti.

9. LITERATURA

- [1] Privredna komora Srbije, "Privreda Srbije - Energetika i energetske rudarstvo", <http://www.pks.rs>
- [2] Srđan Maričević, "Određivanje osnovnih komponenti za izgradnju male hidroelektrane", Diplomski rad, Univerzitet u Osijeku, Osijek, 2017.
- [3] A.M.A. Haidar "Utilization of pico hydro generation in domestic and commercial loads", University of Southern Queensland, Australia, 2012.
- [4] <http://www.powerspout.com/>
- [5] A.Najdanović, "Proizvodnja električne energije putem mirko hidroelektrane", Diplomski rad, Univerziteti u Nišu, Elektronski fakultet, Niš 2013
- [6] "Uredba o uslovima i postupku sticanja statusa povlašćenog proizvođača električne energije, privremenog povlašćenog proizvođača i proizvođača električne energije iz obnovljivih izvora energije", "Službeni glasnik RS", br. 56/2016, 15.6.2016. god.
- [7] <https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator>

Kratka biografija:



Jelena Savković, rođena je u Bugojnu 1992. godine. Na Fakultetu tehničkih nauka u Novom Sadu diplomirala je 2016. godine na studijskom programu Elektroenergetika. Master studije na istom fakultetu na studijskom programu Elektroenergetika – energetska elektronika i električne mašine upisala je školske 2016/17. god. Master rad na temu iz oblasti obnovljivih izvora električne energije odbranila je 2018. godine.



dr Vladimir A. Katić, red.prof. rođen je 1954. godine u Novom Sadu. Doktorirao je na Univerzitetu u Beogradu 1991. godine. Oblasti interesovanja su mu energetska elektronika, obnovljivi izvori električne energije, kvalitet električne energije i električna vozila.

PROVERA IDENTITETA I AUTORIZACIJA KORISNIKA U MICROSOFT AZURE PLATFORMI**AUTHENTICATION AND ACCESS CONTROL WITHIN MICROSOFT AZURE PLATFORM**

Dražen Đanić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je na primeru softverskog sistema za rezervaciju u restoranima opisano korišćenje platforme Microsoft Azure za potrebe rešavanja problema provere identiteta i autorizacije korisnika. Opis sistema obuhvata opis njegovog modela i implementacije.

Ključne reči: Provera identiteta, Autorizacija, Microsoft Azure, Azure Active Directory, Softverski sistem

Abstract – This document describes features of Microsoft Azure for authentication and authorization of users. These features are demonstrated on a system for managing reservations in restaurants. The system is described through its model and implementation.

Keywords: Authentication, Authorization, Microsoft Azure, Azure Active Directory, Software system

1. UVOD

U svetu IT-a (Informacione tehnologije) je sve veći trend da se obezbeđivanje računarskih resursa, hardvera i softvera, ostvaruje putem takozvanog *cloud computing*-a (računarstvo u oblacima). *Cloud computing* se definiše kao model za isporuku računarskih resursa korisnicima u vidu servisa. Ta isporuka se obavlja preko odgovarajuće računarske mreže i to po prijemu zahteva. Sistem iz kojeg se pružaju traženi resursi nosi naziv *cloud*. Tako na primer, *cloud computing*-om se može dobiti pristup različitim aplikacijama, elementima za razvoj i isporuku aplikacija, skladištima podataka itd. Upotrebom *cloud computing*-a se potencijalno može povećati produktivnost, povećati pouzdanost, smanjiti finansijski troškovi itd. Među čitavim nizom *cloud* sistema koji postoje na tržištu i koji su dostupni širokim masama korisnika nalazi se i *Microsoft Azure, cloud* koji je u vlasništvu kompanije *Microsoft* [1, 2, 3].

Pojedine aplikacije mogu imati potrebu za izvršavanjem procesa provere identiteta (autentifikacije) i autorizacije. Autentifikacija predstavlja proces kojim se proverava identitet nekog entiteta, odnosno proverava se da li je entitet zaista onaj za koji tvrdi da jeste. Autorizacija predstavlja proces kojim se utvrđuje da li je nekom entitetu dozvoljen pristup traženom resursu. Autorizacija zahteva uspešno izvršenu autentifikaciju. Treba imati na umu da entitet može biti korisnik, aplikacija ili nešto slično.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Goran Sladić, vanr. prof.

Implementacija pomenutih procesa u pojedinim situacijama može biti prilično složena i može oduzeti puno vremena i energije koje bi se inače mogle iskoristiti za implementaciju ključnih elemenata aplikacije, odnosno elemenata koji predstavljaju poslovnu logiku same aplikacije. Zbog toga se sve više javlja potreba da se obavljanje autentifikacije i autorizacije ostvari upotrebom nekog servisa koji je za to specijalizovan. Jedan takav servis jeste *Azure AD (Active Directory)*, servis koji nudi *Microsoft Azure* [4, 5, 6].

U ovom radu je razmatrano obavljanje procesa autentifikacije i autorizacije korisnika upotrebom mogućnosti koje pruža *cloud Microsoft Azure*, odnosno upotrebom servisa *Azure AD*. Primena servisa *Azure AD* je pokazana na primeru jednog softverskog sistema. Ovaj softverski sistem, između ostalog, omogućava rezervaciju stolova u restoranima, a autentifikaciju i autorizaciju korisnika ostvaruje upotrebom *Azure AD*-a. Sistem u svoj sastav uključuje dve komponente, od kojih je jedna implementirana kao SPA (*Single Page Application*), a druga kao *web API (Application Programming Interface)*.

2. MICROSOFT AZURE

Microsoft Azure je javni *cloud* [1] koji nudi niz servisa koje klijent može iskoristiti za razvijanje, testiranje, isporučivanje i upravljanje aplikacijama na kojima radi. Pomoću tehnologija koje nudi *Microsoft*, klijent može kreirati hibridni *cloud* [1] u čiji sastav bi ušao i *Microsoft Azure* [3].

Microsoft Azure klijentima pruža servise preko međusobno povezanih datacenter-a (centar podataka, centar koji se sastoji od velikog broja povezanih računarskih resursa) smeštenih u različitim regionima sveta. Od klijenta traži da u slučaju određenih servisa navede region koji odgovara datacenter-u iz kojeg želi da mu se pruži željeni servis. Nije sve jedno iz kojeg regiona se obezbeđuje servis jer udaljenost utiče na latenciju mrežnog zahteva. Zbog toga je veoma bitno da region iz kojeg se pruža neki servis krajnjem entitetu bude što bliži tom entitetu.

Klijentu su data na raspolaganje, korišćenje, različita sredstva pomoću kojih može da upravlja servisima koje *Microsoft Azure* pruža. Tako servisima može da upravlja upotrebom *web* aplikacije *Azure Portal*, kojoj se pristupa preko *web browser*-a (aplikacija za pregledanje *web* sadržaja). Pored upravljanja servisima, *Azure Portal* omogućava i uvid u informacije vezane za plaćanje pruženih usluga. Većinu poslova klijent može da obavi i programskim putem upotrebom odgovarajućeg REST

API-a. Inače, i sam *Azure Portal* koristi skup REST API-a. Za pojedine platforme su obezbeđeni SDK-ovi (*Software development kit*) koje klijent može da upotrebi da programskim putem upravlja servisima. Među platformama za koje su obezbeđeni SDK-ovi nalaze se: .NET, Node.js, Java, PHP, Python i Ruby. Za upravljanje servisima se mogu upotrebiti i alatke *Azure Command-Line Interface* i *Azure PowerShell*.

2.1. Azure AD

Azure AD je jedan od mnogobrojnih servisa koje *Microsoft Azure* nudi svojim klijentima. On se definiše kao servis koji omogućava nekoj organizaciji da upravlja identitetima i pristupanjima. Njegovom upotrebom organizacija može povećati produktivnost svojih korisnika i može povećati sigurnost svojih resursa.

Klijent upotrebom *Azure AD*-a može da za svakog korisnika obezbedi po jedan jedinstveni identitet, a zatim i da upravlja tim identitetima. Korisniku se može omogućiti da identitet koji mu je dodeljen iskoristi kako bi ostvario SSO pristup određenim aplikacijama. Te aplikacije mogu biti one aplikacije koje obezbeđuje sama organizacija kojoj pripada ili one aplikacije koje obezbeđuju neke druge organizacije kroz servise SaaS [1] tipa. *Azure AD* omogućava da se korisniku dozvoli da sam poništi zaboravljenu lozinku kako bi postavio novu i time izbegne potrebu da se obrati administratoru.

Zahvaljujući svemu tome, korisnik može da poveća kako svoju produktivnost tako i produktivnost same organizacije. Klijentu je data mogućnost da korisnike grupiše shodno potrebama kako bi njima lakše upravljao. Tako se, na primer, korisniku može automatski omogućiti ili zabraniti pristup nekoj aplikaciji na osnovu grupe kojoj pripada. *Azure AD* ima podršku za RBAC (*Role Based Access Control*) pa se nadležnosti i mogućnosti korisnika mogu odrediti dodeljivanjem odgovarajućih uloga i dozvola.

Sigurnost identiteta korisnika, aplikacija i ostalih resursa se može podići na viši nivo upotrebom multi-faktorske autentifikacije koja je podržana od strane *Azure AD*-a. Treba napomenuti da se sigurnost može povećati i upotrebom drugih mehanizama koje *Azure AD* nudi. Među njima se nalazi onaj koji daje mogućnost da se kontroliše kako neki korisnik može pristupiti nekoj aplikaciji povezanoj sa *Azure AD* za koju mu je dato pravo korišćenja. To kontrolisanje klijent može ostvariti definisanjem niza *conditional access policy*-a, odnosno definisanjem politike kojom se određuje načina na koji *Azure AD* treba da reaguje na pokušaj pristupa koji zadovoljava određene uslove.

Za *Azure AD* se vezuju dva pojma: *Azure AD tenant* i *Azure AD directory*. *Azure AD tenant* ili kraće *tenant* (zakupac, stanar) označava instancu *Azure AD* servisa koja je kreirana za potrebe jedne organizacije. Svaki *tenant* je izolovan i odvojen od ostalih *tenant*-a. To znači i da između *tenant*-a ne postoji odnos roditelj-dete, tj. ne postoji hijerarhija *tenant*-a. Svaki *tenant* ima svoj *Azure AD directory*, odnosno direktorijum u koji se smeštaju informacije o samom *tenant*-u i organizaciji. Tako se u *Azure AD directory*-u između ostalog mogu naći informacije o korisnicima, grupama i aplikacijama. Obično se u literaturi i praksi pojmovi *tenant*, *Azure AD*

directory i *Azure AD* izjednačavaju, odnosno koriste se kao sinonimi.

2.2. Autentifikacija i autorizacija korišćenjem Azure AD-a

Klijenti servis *Azure AD* mogu iskoristiti da aplikacijama koje razvijaju obezbede mogućnost obavljanja autentifikacije i autorizacije. Zahvaljujući tome, klijenti ne moraju da implementiraju čitav mehanizam koji je neophodan za sigurno obavljanje takvih procesa. To, između ostalog, doprinosi značajnom pojednostavljenju implementacija aplikacija. Takođe, klijenti mogu veću pažnju posvetiti implementaciji poslovnih logika samih aplikacija.

Svaka aplikacija koja ostvaruje autentifikaciju i autorizaciju korišćenjem *Azure AD*-a mora biti registrovana u *Azure AD dictionary*-u odgovarajuće organizacije. Registrovanje je potrebno izvršiti kako bi *Azure AD* postao svestan te aplikacije i kako bi mu se obezbedile neophodne informacije za komunikaciju sa tom aplikacijom prilikom obavljanja autentifikacije i autorizacije ili prilikom razmene tokena. Takođe, registrovanje aplikacije se obavlja kako bi joj se obezbedile neophodne dozvole za pristupanje podacima u direktorijumu organizacije, drugim aplikacijama organizacije itd.

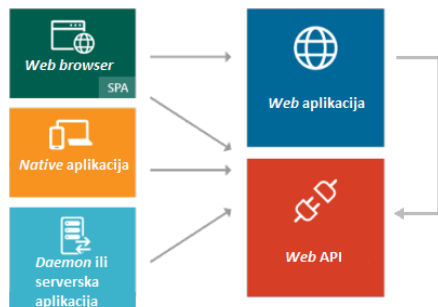
Aplikacije koje koriste *Azure AD* se mogu svrstati u jednu od dve kategorije: *single-tenant* aplikacije i *multi-tenant* aplikacije. *Single-tenant* aplikacije su one aplikacije koje su namenjene za korišćenje u samo jednoj organizaciji. Stoga one treba da budu registrovane u samo jednom *Azure AD dictionary*-u i to onom koji odgovara organizaciji kojoj je namenjen. *Multi-tenant* aplikacije su aplikacije koje su namenjene za korišćenje u više različitih organizacija, a ne samo u onoj u kojoj su inicijalno registrovane. Organizacija koja želi da koristi aplikaciju takve vrste mora dati saglasnost da bude registrovana i u njenom *Azure AD dictionary*-u.

Na slici 1 prikazani su različiti scenariji koje *Microsoft* navodi kao podržane scenarije od strane *Azure AD*-a, odnosno scenariji u kojima se *Azure AD* može koristiti za autentifikaciju i autorizaciju. Klijent nije ograničen na izbor programskih jezika i platformi koje će koristiti prilikom implementacije nekog od tih scenarija. Na slici se mogu uočiti pet osnovnih scenarija: *web browser* ka *web* aplikaciji, *SPA*, *native* aplikacija ka *web API*-u, *web* aplikacija ka *web API*-u, *daemon* ili serverska aplikacija ka *web API*-u. *Azure AD* omogućava klijentu i da veoma lako kombinuje više prostih pa da tako dobije složeniji scenarijo koji zadovoljava njegove potrebe.

Aplikacije mogu koristiti *Azure AD* radi obavljanja autentifikacije i autorizacije pomoću nekog od široko prihvaćenih i standardizovanih protokola za autentifikaciju i autorizaciju, a koje *Azure AD* podržava. Podržani protokoli od strane *Azure AD*-a su: OAuth 2.0, OpenID Connect, SAML 2.0 i WS-Federation (*Web Services Federation*). *Microsoft* za čitav niz programskih jezika i platformi nudi klijentima kroz odgovarajuće biblioteke otvorenog koda implementacije svakog od tih protokola.

U slučaju da je autentifikacija i autorizacija entiteta uspešna, *Azure AD* klijentskoj aplikaciji u odgovoru dostavlja i token. Format tokena i njegov sadržaj zavise od korišćenog protokola i izvesnih podešavanja. Tako na

primer ID tokeni i *access token*-i imaju format JWT-a čija je struktura JWS. U tokenu se nalaze različite tvrdnje koje se mogu iskoristiti u različite svrhe. Tako se, na primer, mogu koristiti za proveru ispravnosti samog tokena, utvrđivanje autorizacije entiteta, prikazivanje informacija o tom entitetu itd.

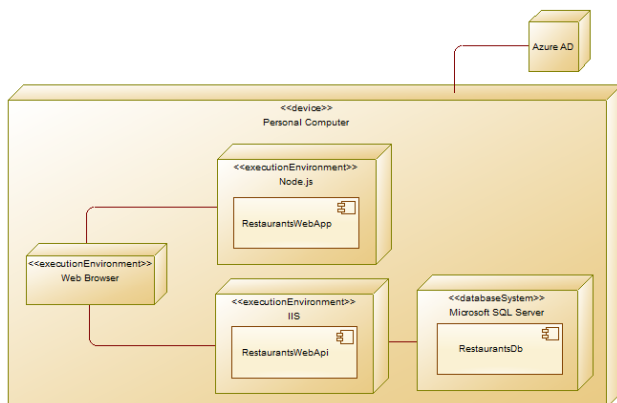


Slika 1. Podržani scenariji i tipovi aplikacija od strane Azure AD-a

3. MODEL SISTEMA ZA REZERVACIJU U RESTORANIMA

Sistem za rezervaciju u restoranima predstavlja softverski sistem koji kao osnovnu funkciju nudi onu koja omogućava gostima da kreiraju rezervacije stolova u restoranima. Sistem sprovodi autentifikaciju i autorizaciju korisnika koristeći servis *Azure AD*.

Na slici 2 prikazan je dijagram raspoređivanja koji se odnosi na sistem u slučaju njegovog razvoja i testiranja.



Slika 2. Dijagram raspoređivanja sistema

Kao što je već spomenuto u uvodu, sistem za rezervaciju u restoranima u svoj sastav uključuje dve komponente. Prva komponenta je nazvana *RestaurantsWebApp*. Ona je implementirana kao SPA i izvršava se na Node.js-u. Druga komponenta je nazvana *RestaurantsWebApi*. Ona je implementirana kao web API i izvršava se na IIS-u (*Internet Information Services*). *Microsoft SQL Server* predstavlja sistem za upravljanje bazama podataka na kojem je kreirana baza podataka koja je nazvana *RestaurantsDb*. Ona je formirana u skladu sa odgovarajućim konceptualnim modelom podataka. Između IIS-a i *Microsoft SQL Server*-a postoji komunikacija jer komponenta *RestaurantsWebApi* koristi bazu podataka *RestaurantsDb* u ostvarivanju svoje poslovne logike. Klijenti mogu pomoću nekog *Web Browser*-a da pristupe komponenti *RestaurantsWebApp*. *RestaurantsWebApp* koristi usluge *RestaurantsWebApi*-a preko poziva upućenih sa svog *frontend*-a (prezentacioni sloj) koji je učitao u *Web Browser*-u. *Web Browser*, *Node.js*, *IIS* i *Microsoft*

SQL Server se izvršavaju na uređaju označenom sa *Personal Computer* (lični računar), koji predstavlja računar na kome se sistem razvija i testira. U obavljanju autentifikacije i autorizacije se koristi servis *Azure AD* pomoću *OpenID Connect* i *OAuth 2.0*. U odgovarajućem *Azure AD tenant*-u su registrovani *RestaurantsWebApp*, *RestaurantsWebApi* i korisnici. To sve znači da između *Personal Computer*-a i *Azure AD*-a postoji komunikacija preko interneta. Komunikacija sa servisom *Azure AD*, komponentom *RestaurantsWebApp* i komponentom *RestaurantsWebApi* se vrši korišćenjem protokola *HTTPS*. Time su ispoštovane preporuke koje su date u dokumentacijama za *OAuth 2.0* i *OpenID Connect*.

4. IMPLEMENTACIJA SISTEMA ZA REZERVACIJU U RESTORANIMA

4.1. Registracija komponenti

Korišćenjem web aplikacije *Azure Portal* su obavljene sve neophodne aktivnosti nad servisom *Azure AD* kako bi ga sistem za rezervaciju u restoranima mogao koristiti radi obavljanja autentifikacije i autorizacije korisnika.

Za potrebe sistema je kreiran novi *Azure AD tenant*. U njemu su registrovane komponente *RestaurantsWebApi* i *RestaurantsWebApp*.

Obe komponente su registrovane kao *single-tenant* aplikacije. Njima je dozvoljeno da koriste protokol *OAuth 2.0* zasnovanog na implicitnom *authorization grant*-u. Za obe komponente je specificirano da one prepoznaju, tj. definišu, tri uloge: *administrator*, *manager* i *guest*. Komponenti *RestaurantsWebApp* je omogućeno da komponenti *RestaurantsWebApi* šalje zahteve u ime korisnika koji je trenutno na nju prijavljen.

4.2. Implementacija RestaurantsWebApi-a

Komponenta *RestaurantsWebApi* je implementirana kao poseban projekat korišćenjem programskog jezika C# i *framework*-a (radni okvir) *ASP.NET Web API 2*.

Da bi se mogle koristiti mogućnosti koje pružaju *OWIN* i *Katana*, u projekat je dodat *NuGet* paket *Microsoft.Owin.Host.SystemWeb* čija je verzija 3.1.0. Pošto *Katana* obezbeđuje *OWIN* komponentu koja u protočnoj obradi *HTTP* zahteva može vršiti validaciju pristiglog *access token*-a izdatog od strane servisa *Azure AD*, u projekat je dodat i *NuGet* paket *Microsoft.Owin.Security.ActiveDirectory* čija je verzija 3.1.0.

Za registraciju *OWIN* komponente koja je sposobna da u protočnoj obradi vrši validaciju *access token*-a izdatog od *Azure AD* je iskorišćena *extension* metoda *UseWindowsAzureActiveDirectoryBearerAuthentication* koja je interfejsu *IAppBuilder* obezbeđena pomoću klase čiji je naziv *WindowsAzureActiveDirectoryBearerAuthenticationExtensions*.

Zaštita od neovlašćenog pristupa pojedinim akcijama kontrolera je ostvarena korišćenjem atributa *Authorize*, atributa koji je definisan pomoću klase *AuthorizeAttribute*.

Informacije o korisniku u čije ime se izvršava neka metoda se mogu dobiti pristupanjem statičkom svojstvu *Current* klase *ClaimsPrincipal*. To je iskorišćeno da se dobije korisničko ime korisnika koji je pokrenuo akciju.

4.3. Implementacija RestaurantsWebApp-a

Implementacija *frontend* dela komponente *RestaurantsWebApp* je ostvarena korišćenjem JavaScript *framework*-a AngularJS. AngularJS je obezbeđen uz pomoć *Bower* paketa *angular 1.7.5*. Radi ostvarivanja autentifikacije i autorizacije korisnika pomoću *Azure AD*-a je korišćena *Microsoft*-ova biblioteka *ADAL JS (Active Directory Authentication Library for JavaScript)* koja je obezbeđena pomoću *Bower* paketa *adal-angular 1.0.16*.

Paketom *adal-angular* su obezbeđena dva JavaScript fajla: *adal.js* i *adal-angular.js*. Fajl *adal.js* sadrži implementaciju elemenata i mehanizma koji omogućavaju ostvarivanje autentifikacije i autorizacije korisnika pomoću *Azure AD*-a korišćenjem protokola *OAuth 2.0* zasnovanog na implicitnom *authorization grant*-u kroz protokol *OpenID Connect*. Ta implementacija ne zavisi ni od jedne JavaScript biblioteke, te se stoga može koristiti u *frontend*-u implementiranog (ne)korišćenjem nekog JavaScript *framework*-a. U fajlu *adal-angular.js* se nalazi implementacija koja se ponaša kao AngularJS omotač oko stvari obezbeđenih kroz *adal.js*. Ona u AngularJS aplikacijama olakšava korišćenje mogućnosti koje pruža *adal.js*. Zbog svega toga su iskorišćena oba fajla.

Prilikom razvoja sistema je uočeno da *ADAL JS* ima određeni nedostatak. Taj nedostatak se može ilustrovati na sledećem primeru. Neka je korisnik neprijavljen na sistem i neka želi da prikaže spisak restorana. Nakon pokretanja te akcije *frontend RestaurantsWebApp*-a će uputiti zahtev na putanju */api/restaurants RestaurantsWebApi*-a. Taj zahtev će uputiti koristeći metodu HTTP GET. Za tu putanju i tu HTTP metodu *RestaurantsWebApi* poseduje akciju koja omogućava obradu zahteva, tj. obezbeđuje dostavu spiska restorana. Pri tome, ta akcija ne zahteva da korisnik bude prijavljen. *ADAL JS* će presresti zahtev, a zatim će pokušati da dobavi odgovarajući token kako bi ga ubacio u zaglavlje zahteva jer je prilikom konfiguracije *ADAL JS*-a specificirano da se *RestaurantsWebApi* nalazi u spisku odredišta za koje je neophodan token, tj. u spisku odredišta zaštićenih pomoću *Azure AD*-a, i zato što se ta adresa ne nalazi u spisku izuzetaka, odnosno adresa za koje nije potreban token. Pošto u tome ne uspeva, *ADAL JS* će prekinuti isporuku HTTP zahteva. Ishod toga je da se neće uspeti dobiti spisak restorana i da ga korisnik neće videti.

Za tu istu adresu, ali za metodu HTTP POST, *RestaurantsWebApi* poseduje akciju koja zahteva da korisnik bude prijavljen na sistem. Ona omogućava dodavanje novog restorana.

Da je ova adresa navedena u spisku izuzetaka i da se za prijavljenog korisnika upućuje zahtev kojim se želi dodati novi restoran, *ADAL JS* u zahtev ne bi ubacio odgovarajući token, pa bi *RestaurantsWebApi* odbila taj zahtev. Da bi se izbegle takve i slične situacije, implementacija *ADAL JS*-a je izmenjena tako da *ADAL JS* propušta svaki HTTP zahtev upućen nekom *web API*-u iako ne uspe da dobavi i u njega ubaci odgovarajući token, jer *web API* sam treba da odluči da li će ga prihvatiti ili odbiti. Izmena je načinjena u *adal-angular.js*, a izmenjeni segment je prikazan na stranici [7]. Sadržaj fajla *adal-angular.js* sa primenjenom izmenom je prikazan na stranici [8].

5. ZAKLJUČAK

U radu je opisano rešavanje problema autentifikacije i autorizacije korisnika korišćenjem mogućnosti koje pruža *Microsoft Azure*. Date su teorijske osnove vezane za *Microsoft Azure* kao jedan od dostupnih sistema koji omogućavaju *cloud computing*. Opisan je servis *Azure AD*, servis koji nudi *Microsoft Azure* i koji omogućava obavljanje autentifikacije i autorizacije.

Primena *Microsoft Azure*-a, tj. servisa *Azure AD*, je pokazana na primeru softverskog sistema za rezervaciju u restoranima. Prilikom razvoja sistema su korišćene već dostupne biblioteke koje omogućavaju korišćenje *Azure AD*-a radi ostvarivanja autentifikacije i autorizacije. Jedna od njih je *ADAL JS*. U njoj je uočen i otklonjen nedostatak.

Prilikom prijavljivanja na sistem korisnik unosi korisničko ime i lozinku. Nivo sigurnosti bi se mogao povećati, na primer, primenom multi-faktorske autentifikacije za koju postoji podrška od strane *Azure AD*-a.

6. LITERATURA

- [1] "Cloud Computing Synopsis and Recommendations", <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-146.pdf>
- [2] Microsoft Azure, "What is cloud computing?", <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/>
- [3] Microsoft Azure, "What is Azure?", <https://azure.microsoft.com/en-us/overview/what-is-azure/>
- [4] OWASP, "Authentication Cheat Sheet", https://www.owasp.org/index.php/Authentication_Cheat_Sheet
- [5] OWASP, "Access Control Cheat Sheet", https://www.owasp.org/index.php/Access_Control_Cheat_Sheet
- [6] Microsoft Azure, "What is Azure Active Directory?", <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis>
- [7] <https://github.com/AzureAD/azure-activedirectory-library-for-js/pull/688/commits/db2dde58c6a344f446efffc7baadf04ffaed5af8>
- [8] <https://github.com/AzureAD/azure-activedirectory-library-for-js/blob/db2dde58c6a344f446efffc7baadf04ffaed5af8/lib/adal-angular.js>

Kratka biografija:



Dražen Đanić rođen je 1993. godine u Vrbasu. Srednju tehničku školu „Mihajlo Pupin“ u Kuli završio je 2012. god., pri tome dobio diplomu „Vuk Karadžić“. Iste godine upisao se na osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu, odsek Računarstvo i automatika. Osnovne akademske studije je uspešno završio 2016. odbranom diplomskog rada na temu „Podsistem za generisanje sertifikata u okviru informacionog sistema skupštine grada“. Iste godine je upisao master akademske studije na istom fakultetu i smeru.

**ПРОЈЕКТОВАЊЕ LED РАСВЈЕТЕ
СА ПРИМЈЕРОМ ЈАВНЕ РАСВЈЕТЕ У КОЗАРСКОЈ ДУБИЦИ
LED LIGHTING DESING
WITH CASE STUDY OF KOZARSKA DUBICA STREET LIGHTING**

Дејан Марин, Факултет техничких наука, Нови Сад

Област – Електротехника и рачунарство

Кратак садржај – Овај рад се бави употребом LED технологије расвјете у системима јавне расвјете и начинима спровођења пројеката реконструкције система јавне расвјете, на примјеру општине Козарска Дубица. При томе су објашњени основни принципи рада LED технологије и система јавне расвјете, употреба софтвера за пројектовање и симулацију расвјете, као и анализа оправданости пројекта реконструкције, посматрано са различитих аспеката.

Кључне речи: LED расвјета, јавна расвјета, расвјетни софтвери, енергетска ефикасност

Abstract – This thesis is about the use of LED lighting technology in street lighting systems and ways of implementing the street lighting reconstruction projects on the example of Kozarska Dubica. There are explained the basic principles of LED technology and street lighting systems, the use of lighting softwares for design and simulation of lighting systems, as well as the feasibility analysis of the reconstruction project, viewed from different aspects.

Keywords: LED lighting, street lighting, lighting softwares, energy efficiency

1. УВОД

Савремени трендови намећу високе захтјеве за све аспекте друштва и технологије, па тако и за системе јавне расвјете. Повећање безбједности свих учесника у саобраћају, омогућавање сталне интеракције између људи и истицање естетског аспекта освјетљења, представљају основне циљеве јавне расвјете.

Фокусирајући се на енергетски ефикасну расвјету, смањење потрошње електричне енергије и емисије CO₂, заштиту животне средине, смањење оперативних трошкова и потребе за високим квалитетом освјетљења, потребно је наћи адекватно рјешење за системе јавне расвјете.

LED технологија расвјете представља најраспрострањеније модерно рјешење што се тиче повећања енергетске ефикасности и квалитета освјетљења, у

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био проф. др Владимир Катић.

већини расвјетних система, па тако и у јавној расвјети. Стога, важан дио рада чине опис LED технологије расвјете и начина примјене у системима јавне расвјете, модерне методе управљања LED расвјетом, те преглед софтвера који су способни за пројектовање и симулацију система јавне расвјете.

Поред тога, један од главних циљева рада јесте анализа техничких, економских, еколошких, друштвених и правних аспеката реализације пројекта реконструкције јавне расвјете, користећи примјер система јавне расвјете општине Козарска Дубица.

Поред типичних анализа за материјале ове врсте, додатак чини анализа помоћу расвјетног софтвера, односно доказивање поставке расвјетних тијела на датом примјеру, док се истицање самих карактеристика LED технологије прожима кроз све анализе.

Према томе, рад представља својеврсни водич за пројекте реконструкције система јавне расвјете дат на примјеру реконструкције система јавне расвјете општине Козарска Дубица. При томе су обрађени различити аспекти и предности коришћења LED расвјете, као најповољнијег рјешења за модерне захтјеве који се стављају пред системе јавне расвјете. Поред објашњења типичних корака пројекта реконструкције, омогућено је и упознавање са самом LED технологијом расвјете, и са расвјетним софтверима.

2. ПРИМЈЕНА LED ТЕХНОЛОГИЈЕ У СИСТЕМИМА ЈАВНЕ РАСВЈЕТЕ

Свјетлећа диода (енг. Light Emitting Diode – LED) је полупроводнички извор свјетлости, тачније PN диода која емитује свјетлост када је пропусно поларисана. Када кроз њу тече струја одговарајућег смјера, електрони се рекомбинују са шупљинама, при чему се ослобађа енергија у облику фотона. Овај ефекат се назива електролуминисценција, а боја свјетлости (која зависи од енергије фотона) је одређена енергетском баријером полупроводника, односно коришћеним полупроводничким материјалима.

У пракси, диоде које се користе у системима јавне расвјете су снажне свјетлеће диоде, снаге углавном од 0,5 W па навише, радне струје типично око 350 mA, и јачине свјетлости и до пар стотина lm/W. Уобичајене свјетлеће диоде које се користе као индикатори имају знатно ниже наведене вриједности [1].

2.1. Карактеристике LED расвјетних тијела

LED расвјетна тијела се састоје од скупа пакованих снажних свјетлећих диода које емитују углавном бијелу свјетлост високог квалитета (сличну природној), са могућношћу подешавања јачине и нијанси свјетлости. За остварење предности LED расвјетних система, попут ефикасности, дуготрајности и исплативости битан је квалитет основних дијелова лампи. Сам принцип рада и састав свјетлећих диода омогућује предности LED лампи у односу на друге изворе свјетлости попут брзог достизања пуног сјаја, стално паљење/гашење, непостојање нити, отпорност на ударе, већа ефикасност у хладнијим временским условима, итд.

Конструкција самих LED лампи (слика 1), тј. основни дијелови су најчешће: носач лампе, тијело лампе, хладњак (за одвођење топлоте које производе свјетлеће диоде), драјвер (електрично коло за напајање), LED чипови, сочива (служе за правилно усмјеравање свјетлости).



Слика 1. Изглед LED лампе [2]

Што се тиче техничких карактеристика LED лампи, напредне технологије склапања лампи омогућују високи степен механичке заштите, док квалитет електронског склопа одређује могућност имплементације микрорачунарских система, и остваривање квалитетног освјетљења, комуникационих, информационих, визуелних ефеката расвјете, и сличних захтјева. У односу на друге најчешће коришћене технологије расвјете, као што су инкандесцентне сијалице, штедљиве сијалице (CFL), метал – халидне лампе (MH), и натријумске лампе (натријум под високим притиском – HPS), потребне снаге LED лампи, за исти ниво освјетљености су 2 – 4 пута мање, првенствено због високе ефикасности (lm/W) и могућности прецизног усмјеравања свјетлости [3]. Животни вијек LED лампи је предвидив, и дужи је за 3 – 25 пута од других врста лампи, па чак и више због рјеткости изненадних отказа LED лампи [3]. Због тога, трошкови одржавања LED система расвјете су константни, а исплативост инвестиције дугорочна.

Економски аспекти се могу приказати кроз почетна улагања и потрошњу електричне енергије. Почетна улагања у набавку LED технологије су већа од улагања у стандардне видове расвјете. То је највише због коришћења скупих материјала при изради свјетлећих диода и потребног квалитета осталих дијелова лампи. Међутим, већа ефикасност и дужи животни вијек од других технологија знатно смањују трошкове електричне енергије и одржавања, што током дужег периода коришћења доноси мање укупне трошкове функционисања система јавне расвјете, и

сигуран поврат инвестиције (трошкови рада LED расвјете су 5 пута мањи у односу на инкандесцентне, а 2 пута мањи у односу на штедљиве сијалице) [3].

Еколошке предности се огледају кроз смањену потрошњу електричне енергије и самим тим емисије штетних гасова, и смањено ослобађање топлоте у току рада. Највећа забринутост у вези коришћења LED расвјете је у вези утицаја плаве свјетлости на људско здравље. Правилним усмјеравањем снопа свјетлости и подешавањем интензитета свјетлости, свјетлосно загађење може бити сведено на минимум.

2.2 Методе управљања јавном расвјетом

Паметни системи јавне расвјете представљају прилагодљиве, интерактивне системе, који прилагођавају интензитет свјетлости према кретањима учесника у саобраћају (возила, бициклиста, пјешака) и према типовима расвјетних површина (саобраћајнице, пјешачки прелази, бицикличке и пјешачке стазе, паркинзи итд.). Ови системи омогућују да свака појединачна лампа буде укључена само када је то потребно, када је детектована активност, тј. кретање, а у осталим периодима буде димована (пригушена) или потпуно угашена, али у стању приправности. У односу на стационарне (паљење/гашење по утврђеном распореду) и полустационарне (димовање по утврђеном распореду), паметни системи јавне расвјете захтјевају већи ниво сложености, али и доносе бројне предности попут: повећање енергетске ефикасности, смањење оперативних трошкова, високи степен контроле и управљивости, повећање осјећаја комфора и безбједности, могућност имплементације других функција и система, итд.

Основни дијелови паметних система управљања расвјетом су (слика 2): 1. управљачки центар, 2. IoT протоколи, 3. локале управљачке станице, 4. локални комуникациони протоколи, 5. микроконтролери и сензори.



Слика 2: Управљање системом јавне расвјете [4]

3. СОФТВЕРИ ЗА ПРОЈЕКТОВАЊЕ И СИМУЛАЦИЈУ СИСТЕМА ЈАВНЕ РАСВЈЕТЕ

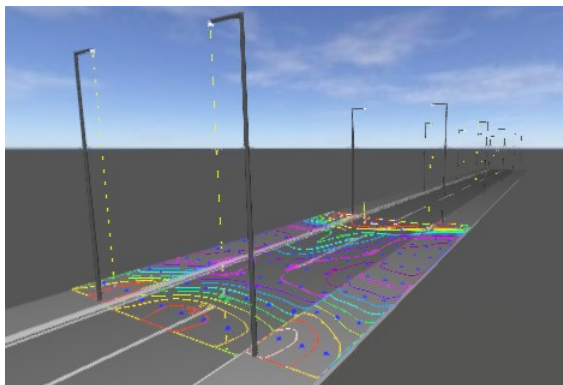
Расвјетни софтвери се, за системе јавне расвјете, користе за анализу фотометрије (начин дистрибуицања свјетлости у простору), 2D и/или 3D пројектовање система расвјете, симулацију различитих сцена-рија расвјете, итд.

3.1. Особине расвјетних софтвера

Ови софтвери су ефикасни, интуитивни, једноставни за коришћење и омогућују веома квалитетну симулацију и комплексне анализе различитих система расвјете у различитим окружењима и намјенама, без ограничења за величину, облик и особине просторних површина и објеката. Такође, нуде и бројне алате и

функције које омогућују анализу природног освјетљења, анализу различитих фотометријских величина, динамичко прилагођавање расвјете различитим сценаријима, симулације у реалном времену, прорачун у складу са важећим нормама и стандардима, широку базу података и производа, интерактивне методе оптимизације, различите методе прорачуна, флексибилно приказивање резултата (на више начина), компатибилност са другим програмима, увоз/извоз CAD, BIM, сликовних и фотометријских фајлова (садрже податке о свјетлосним карактеристикама расвјетних елемената), писаних докумената, итд.

У суштини, главна сврха расвјетних софтвера је, након завршеног процеса моделовања физичких и фотометријских карактеристика система јавне расвјете, добијање и представљање резултата комбинацијом графичких и нумеричких података. Због тога, најбитније карактеристике расвјетних софтвера представљају моћ програма за извршавање прорачуна и опсег могућности функције рендеринга, која се односи на конвертовање модела у 2D слике. За анализу система јавне расвјете посебно је битна способност за брзе и квалитетне прорачуне, који се извршавају за одређене тачке освјетљених површина, што омогућује једноставност симулација и флексибилно приказивање резултата (слика 3).



Слика 3: *Примјер симулације јавне расвјете*

3.2. Најпознатији расвјетни софтвери

Од расвјетних софтвера доступних на тржишту, свега четири софтвера су прилагођена у довољној мјери за комплексне анализе система јавне расвјете, без ограничења у вези сложености или величине система. То су AGI32, Visual, DIALux, Relux, те Radiance, који знатно заостаје за наведеним програмима, али је значајан као софтвер отвореног кода и због могућности квалитетног рендеринга. Остали доступни софтвери нису конкурентни наведенима углавном због: застарјелости (не постоје новије верзије и надоградње); могућности им заостају за наведеним расвјетним софтверима; нису прилагођени за анализу система јавне расвјете, и сл.

Програм DIALux је далеко најпопуларнији софтвер за пројектовање и симулацију јавне расвјете, нарочито међу инжењерима. Такође, DIALux је најбољи у категоријама брзине и квалитета добијених резултата за вјештачке изворе свјетлости. Једноставност, компатибилност, широка база произвођача и података,

бесплатна лиценца, подршка и основна обука, само доприносе популарности овог софтвера.

3.3 Кораци при пројектовању јавне расвјете

Основне улазне податке приликом пројектовања система јавне расвјете, у нпр. програму DIALux, чине подаци о путу (профил пута, карактеристике појединачних елемената (трака), класа освјетљености у складу са стандардом, итд.) и подаци о свјетилкама (електричне и фотометријске карактеристике, физички положај у односу на пут и друге свјетилке, итд.).

Након покретања оптимизације, односно прорачуна и евентуалног прилагођавања изабраних параметара система, добијају се резултати, од којих су примарни вредновање одговарајућих величина према изабраној класи освјетљености, и секундарни попут података о потрошњи и енергетској ефикасности, о свјетлосном загађењу и расипању свјетлости, и сл. Након тога се може приступити неком од начина израде резултата.

4. ПРОЈЕКАТ РЕКОНСТРУКЦИЈЕ ЈАВНЕ РАСВЈЕТЕ ОПШТИНЕ КОЗАРСКА ДУБИЦА

Рад тренутног система јавне расвјете општине Козарска Дубица је на незадовољавајућем нивоу, првенствено због мале ефикасности и застарјелости расвјетних тијела, што узрокује велике трошкове рада система, укључујући трошкове потрошње електричне енергије и трошкове одржавања, те смањени квалитет и нарушавање одговарајуће функције јавне расвјете на већини локација. Због тога је неопходна реконструкција целокупног система јавне расвјете, односно замјена расвјетних тијела на свих 3195 прикључних мјеста, тачније 3167 лампи и 28 рефлектора, уз максимално искоришћење осталих дијелова постојећег система расвјете (стубови, напојни водови, итд.).

4.1. Предлог новог рјешења јавне расвјете

Као основа за већину анализа користи се одговарајућа поставка која користи LED расвјетна тијела, чије су снаге и фотометријске карактеристике изабране у складу са функцијом расвјете и поставком стубова на одговарајућој локацији. Што се тиче функције расвјете, за освјетљавање саобраћајница главни критеријум је безбједност учесника у саобраћају; код освјетљења пјешачких површина додатно је укључен и естетски критеријум; док је код декоративног освјетљења главни акценат стављен на наглашавање детаља објекта који се освјетљава. С обзиром на важност функције освјетљавања, пројектовање се врши у складу са одговарајућим стандардима и препорукама за параметре освјетљености појединих категорија путева, а избор лампи треба да буде одговарајући, у смислу обезбјеђивања довољне освјетљености, правилног распрострањања свјетлости, изгледа лампи, итд.

У Табели 1 је могуће видјети разлику између снага тренутне поставке расвјетних тијела и предложене будуће поставке са LED расвјетним тијелима, изабраним према функцији расвјете и карактеристикама система на одговарајућим локацијама.

Табела 1. *Поређење тренутне и будуће поставке расвјетних тијела*

Тренутна поставка		Будућа поставка	
Снага (W)	Број лампи	Снага (W)	Број лампи
70	1802	15	93
100	305	20	53
125	512	30	903
150	156	40	1721
250	266	65	302
400	28	100	95
		150	28

Потребно је уочити да старе сијалице од, нпр. 70 W, нису замјењене само једним типом LED лампи, већ лампама од 15, 20, 30 и 40 W, зависно од потреба на датој локацији. Исто важи за лампе 100 и 125 W које су замјењене лампама од 30 и 40 W, лампе 150 W са 65 W, и лампе од 250 W са лампама снага 65 и 100 W. Такође, све лампе од 30 W или од 40 W нису међусобно једнаке, него подразумевају неколико типова лампи, зависно од критеријума, односно захтјева на одређеној локацији.

Доказивање тачности усвојене нове поставке у складу са важећим стандардима и функционалним захтјевима за одговарајуће дијелове система јавне расвјете, извршено је у програму Dialux. Према добијеним резултатима симулације рада система јавне расвјете општине Козарска Дубица са предложеном поставком LED лампи, дато рјешење задовољава све критеријуме и захтјеве с обзиром на снаге лампи, при чему је приликом избора лампи у пракси потребно водити рачуна о оптичким карактеристикама и углу поставки лампи.

4.2. Могуће уштеде коришћењем новог рјешења

Коришћењем нове поставке, смањена је укупна инсталисана снага за 62,61%, а годишња потрошња електричне енергије за 77,34%, и трошкови одржавања за 48,19%, што чини укупно смањење трошкова рада система јавне расвјете на подручју општине Козарска Дубица за 69,17%. Дате вриједности су веома блиске планирам процентима смањења (уобичајено 80% за смањење потрошње). При прорачунима је уважена одлука о употреби димовања као методе уштеде, што је еквивалентно смањењу броја радних сати са садашњих 11 на 8 сати дневно. Проценат смањења емисије штетних гасова се поклапа са процентом смањења потрошње (77,34%).

4.3. Извођење пројекта реконструкције

Динамика спровођења пројекта реконструкције обухвата следеће основне кораке:

- спровођење тендера за израду Пројекта реконструкције, и израда и техничка контрола Пројекта,
- припрема и спровођење тендера за набавку кредита,
- припрема и спровођење тендера за набавку лампи,
- демонтажа старих и монтажа нових лампи,

- финална верификација система.

Приликом пројектовања новог рјешења потребно је пратити важеће стандарде и норме за освјетљавање путева, док је приликом самог спровођења пројекта, потребно је задовољити све институционалне и правне аспекте реализације пројекта ове врсте.

5. ЗАКЉУЧАК

LED технологија у данашње вријеме нуди толико велике разлике у уштеди и побољшању квалитета освјетљења у односу на конвенционалне изворе свјетлости, да се намеће као најбоље рјешење за побољшање енергетске ефикасности и остваривање уштеда у раду система јавне расвјете. Уз наведене предности и особине поузданости, дуготрајности и могућности имплементације микрорачунарских система за остваривање паметних начина управљања, могуће је остварити све захтјеве који се стављају пред модерне системе јавне расвјете, у складу са важећим стандардима и критеријумима.

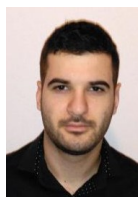
Иако су почетна улагања у LED системе расвјете знатно виша у односу на друге технологије, првенствено због неопходног квалитета механичких дијелова лампи (сочива, напајање, хладњак, заптивање и заштита, итд.), особине попут ниске потрошње, велике поузданости и дуготрајности, доносе вишеструке уштеде у укупним трошковима рада система расвјете током релативно дугог временског периода. Анализом датог примјера је утврђено да је процјењени животни вијек при усвојеним условима рада око 17,5 година, што је знатно дуже у односу на друге технологије расвјете.

У складу са тим, пројекат реконструкције јавне расвјете на подручју општине Козарска Дубице, коришћењем LED технологије расвјете, доноси бројне предности попут уштеда укупних трошкова рада система јавне расвјете (потрошња електричне енергије и трошкови одржавања), поврат инвестиције из планираних уштеда, повећање квалитета освјетљености и остваривање одговарајуће функције јавне расвјете, задовољавање трендова енергетске ефикасности и заштите животне средине, итд.

6. ЛИТЕРАТУРА

- [1] *Light Emitting Diode*, <https://en.wikipedia.org/>
- [2] Web page: <http://www.globalmarket.com/>
- [3] *Light Comparasion*, <https://learn.eartheasy.com/>
- [4] OSRAM AG, „*Street Light Control: Innovative Light Control*“, 2017

Кратка биографија:



Дејан Марин рођен је у Приједору 1991. год. Основне академске студије (ОАС) на Факултету техничких наука из области Електротехнике и рачунарства – Енергетска електроника и електричне машине завршио је 2016. год. Мастер студије (MAC) је уписао школске 2016/2017, а мастер рад одбранио 2018. год..

STATIČKA POBUDA SINHRONOG GENERATORA

STATIC EXCITATION OF THE SYNCHRONOUS GENERATOR

Bogdan Stajić, Zoran Ivanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su navedeni osnovni parametri sistema pobude sinhronog generatora, opisan je energetski dio, regulacija sistema pobude, paljenje tiristora, akvizicioni sistem, zaštite, upravljanje i merenje. Cilj sistema pobude realizovan je korišćenjem savremenih tehnologija redundancijom i u energetskom i u upravljačkom bloku. Sistem pobude realizovan je pomoću dva tiristorska mosta, jedan je uvijek u radu, dok drugi služi kao rezerva. Time je postignuta velika pouzdanost sistema pobude da ne može doći do prekida procesa proizvodnje električne energije.

Cljučne reči: Sistem pobude, tiristor, sinhrona mašina.

Abstract – This paper presents the basic parameters of the synchronous generator excitation system. It describes the energy part, the regulation of the ignition system, trigger thyristors, acquisition system, protection, control and measurement. The goal of the excitation system was realized using modern technologies in redundancy both in the energy and in the control block part. The system of the excitation was realized using two thyristor bridges, one of which is always in operation while the other serves as a reserve. This ensured the high reliability of the excitation system without interruptions in the electricity generation process.

Keywords: excitation system, thyristor, synchronous machine

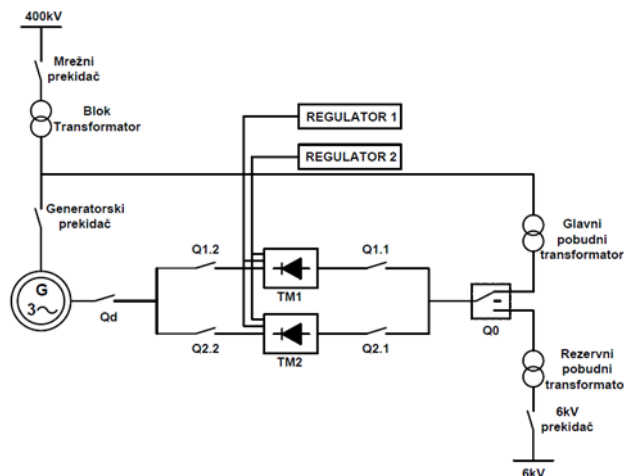
1. UVOD

Ovim radom obrađen je sistem pobude sinhronog generatora snage 353 MVA u TE „Ugljevik“.

Sistem pobude predstavlja regulisani izvor napajanja pobudnog namotaja sinhronog generatora. Sistem pobude u TE "Ugljevik" je statičkog tipa sa tiristorskim mostovima. Napajanje sistema pobude je moguće sa 20kV izvoda generatora preko glavnog pobudnog transformatora (ITU) ili iz 6kV postrojenja preko rezervnog pobudnog transformatora (TUR). Regulacija napona pobude se vrši u okviru regulatora pomoću fazne regulacije impulsa za paljenje tiristora. Blok šema pobude prikazana je na slici 1. Sistem pobude se može podeliti na dva bloka: blok energetike i blok upravljanja. Preko bloka energetike se napaja pobudni rotorski namotaj generatora. Takođe, u okviru ovog bloka je realizovano i brzo razbuđivanje u havarijskim režimima, početno pobuđivanje i prenaponska zaštita. U okviru bloka upravljanja su realizovane regulacione funkcije, zaštite, komandovanje, nadzor itd.

NAPOMENA:

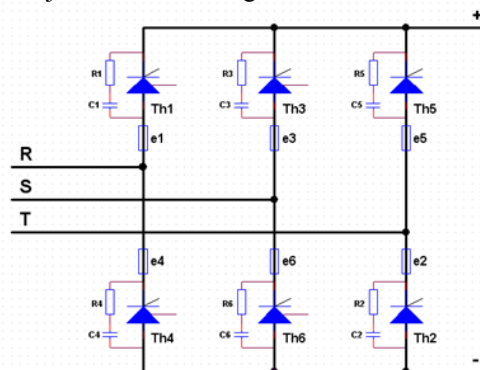
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Zoran Ivanović, docent.



Slika 1: Blok šema sistema pobude

2. ENERGETSKI DIO SISTEMA POBUDE

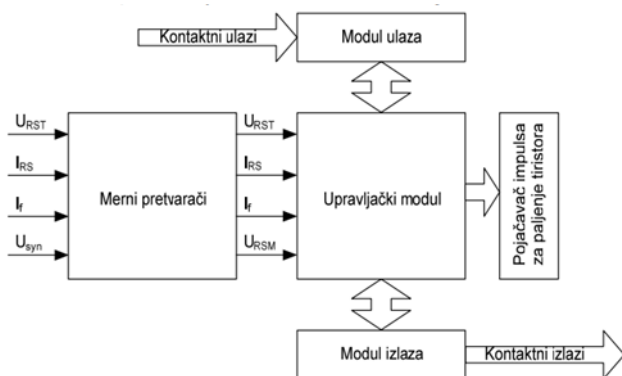
Blok energetike služi za napajanje pobudnog rotorskog namotaja generatora. Takođe, u okviru ovog bloka je realizovano i brzo razbuđivanje u havarijskim režimima, početno pobuđivanje i prenaponska zaštita. U okviru bloka upravljanja su realizovane regulacione funkcije, zaštite, komandovanje, nadzor, akvizicija itd. Primena digitalnih tehnologija i tiristorskih mostova omogućila je redundanciju i u upravljačkom i u energetskom delu sistema pobude. Tiristorski mostovi su trofazni, puno-upravljivi energetski pretvarači koji pretvaraju trofazni naizmenični napon u jednosmerni. Rastavljači koji su prikazani na šemi služe za vidno odvajanje postrojenja koji je pod napon od postrojenja koji nije pod naponom. Pobudni sistem se napaja preko jednog od dva pobudna transformatora. Pobudni transformatori snižavaju napon na naponski nivo potreban za rad sistema pobude, omogućavaju galvansku izolaciju, suzbijaju pojedine harmonike i ograničavaju struju pri kvarovima. Na slici 2 prikazana je šema tiristorskog mosta.



Slika 2: Šema tiristorskog mosta

3. REGULATORI POBUDE

U okviru regulatora pobude su realizovane regulacione funkcije, limiteri, zaštite, samonadzor, komandovanje i signalizacija. Regulatori pobude su urađeni u digitalnoj mikroprocesorskoj tehnologiji. U cilju povećanja pouzdanosti sistem pobude ima dva međusobno nezavisna digitalna regulatora REG1 i REG2 sa identičnim funkcijama pri čemu je jedan u radu, a drugi je topla rezerva. U bilo kom trenutku je moguće izvršiti prelazak sa jednog na drugi regulator, bez ometanja rada generatora, na nalog operatera ili zaštite. Paralelan rad je realizovan tako da u slučaju kvara na nekoj od komponenata regulatora koji je u radu, rezervni regulator automatski preuzima sve upravljačke funkcije, bez prekida procesa proizvodnje električne energije. Svaki regulator može da radi zajedno s drugim regulatorom i samostalno uz isključen i izvađen drugi regulator. Regulacija se vrši obradom ulaznih analognih i digitalnih signala, i generisanjem izlaznih upravljačkih signala za paljenje tiristora ispravljačkih mostova što može da se vidi na slici 3.



Slika 3: Blok upravljačke elektronike

Parametre regulatora podešavamo prilikom ispitivanja sistema pobude. Pre pokretanja programa neophodno je povezati serijski port računara sa regulatorom pobude povezivanjem specijalnog kablja sa konektorom na zadnjoj ploči regulatora pobude.

Treba imati u vidu da je istovremeno moguće pokrenuti više programa tako što će svaki od njih biti povezan sa drugim rekovima regulatora pobude (odnosno, moguće je istovremeno pratiti signale na oba regulatora pobude, u varijanti sa implementacijom dvostruke redundanse regulacije, korišćenjem dva reka regulatora u paralelnom radu).

4. ISPITIVANJE SISTEMA POBUDE SINHRONOG GENERATORA

Ispitivanje sistema pobude se vrši svake godine za vreme remonta. Prilikom ispitivanja pomoću indikatora faze „FLUKE“ koji je prikazan na slici 4, zadaju se željene vrednosti napona i mere se otpornosti izolacije. Takođe pomoću ispitnog uređaja „RETOM-61“ koji je prikazan na slici 5 proveravaju se zaštite.

Prilikom ispitivanja otpora izolacije indikatorom faze zadaje se napon od 1000V, da bi dobijene vrednosti otpora bile zadovoljavajuće moraju biti veće od 2MΩ.

Otopri izolacije tiristorskih mostova koje smo dobili prilikom ispitivanja kreću se u granicama (60 – 250)MΩ što zadovoljava granice otpora ispod koje ne smije biti.



Slika 4: Indikator faze

Ispitivanje zaštita vrši se pomoću ispitnog uređaja „RETOM-61“ koji je povezan USB kablom na laptop koji poseduje instaliran softver preko koga se zadaju potrebne vrednosti struja i napona u sve tri faze. U zavisnosti od prenosnog odnosa strujnog i naponskog transformatora zadaju se određene vrednosti napona i struja da bi se ispitala prorada zaštite i ostale potrebne vrednosti (naponi, struje, aktivna, reaktivna, prividna snaga, prenaponska zaštita itd).



Slika 5: Ispitni uređaj „RETOM-61“

Merenjem otpora između gejta i katode tiristora dobijene su vrednosti između (5,9 - 6,8)Ω, deklarisanе vrednosti proizvođača su 6,7Ω ±20% , što znači da dobijene vrednosti zadovoljavaju dozvoljene granice.

Prilikom merenja napona statora zadat je napon od 0 do 120V (100V odgovara nominalnom naponu statora 20kV) ispitnim uređajem koji je prikazan na slici 5. Merenjem preko naponskog transformatora određenog prenosnog odnosa vrednosti napona koje su dobijene kreću se u granicama (0-24)kV.

Prilikom merenja struja statora zadata je struja od 0 do 7A(4,25A odgovara nominalnoj struji statora 10,2kA) preko ispitnog uređaja. Vrijednosti koje su dobijene kreću se u granicama (0 – 16,8)kA.

Prilikom merenja struja tiristorskih mostova pomoću ispitnog uređaja zadata je struja od 0 do 6A(3,94A odgovara nominalnoj struji pobude 2,9kA, odnosno 2,73kA na naizmjeničnoj strani). Prilikom merenja preko mjernog strujnog transformatora određenog prenosnog odnosa, vrednosti koje su dobijene su u granicama (0-4,45) kA koje su zadovoljavajuće.

Prilikom merenja napona pobude doveden je napon u opsegu od -30 do 60V preko ispitnog uređaja.

Merenjem preko naponskog transformatora određenog prenosnog odnosa dobijene zadovoljavajuće vrednosti napona u granicama (-300 – 600)V.

Prilikom merenja struje pobude preko šanta otpornosti doveden je napon u opsegu od 0 do 90mV(43.5mV odgovara nominalnoj struji pobude 2900A) preko ispitnog uređaja. Merenjem preko strujnog transformatora određenog prenosnog odnosa dobijene su zadovoljavajuće vrednosti struja u granicama (0 - 6)kA.

Ispitivanje RC filtera sistema pobude: Filter se sastoji iz dva kondenzatora od 2.2 μ F i naponske klase 5 kVDC, dva otpornika od 1 Ω snage 200 W i dva brza osigurača za 25 A, 1500 VDC. Filter se koristi za filtriranje pikova napona pobude generatora. Prilikom ispitivanja dobijene su zadovoljavajuće vrijednosti otpornika i kondenzatora koje su date u tabeli.

Deklarisane vrednosti otpornika su 1 $\Omega \pm 10\%$, a kondenzatora 2,2 μ F $\pm 10\%$. Izmerene vrednosti su date u tabeli:

Tabela 1: Vrednosti RC filtera sistema pobude

Otpornik R1 (Ω)	1,02
Otpornik R2 (Ω)	1,02
Kondenzator C1 (μ F)	2,23

Prilikom provere prorade zaštita, na ispitnom uređaju se zadaje struje u sve tri faze i prati se vreme prorade. Ispitana je kratkospojna i prekostrujna zaštita.

Tabela 2: Vrednosti odrade kratkospojne zaštite

Faza	Struja odrade (A)	Vreme odrade (s)
R	9,5	0.07
S	9,5	0.07
T	9,5	0.07

Tabela 3: Vrednosti odrade prekostrujne zaštite

Struja (A)	Vreme odrade (s)
4,20	98
7,17	12

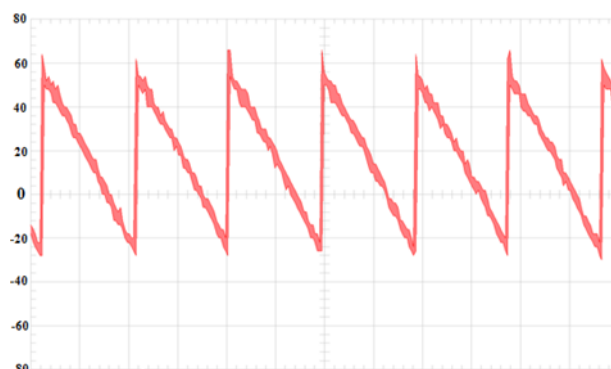
Prilikom ispitivanja zaštite od fazne nesimetrije, uređajem zadajemo različite vrijednosti struja po fazama i pratimo vreme prorade zaštite. Isto ispitivanje se vršilo i za TM2 i dobijeni su iste vrednosti. Navescemo primjer dobijenog rezultata sa zadatim vrednostima u sve tri faze: I-R(7,11A) I-S(3,59A) I-T(3,59) , pri ovim vrednostima dobijeno vrijeme zadrške je 2s.

Prilikom ispitivanja pobude u test režimu rada regulatora, osciloskopom je snimljen oblik jednosmjernog napona tiristorskog mosta TM1 koji je prikazan na slici 6. Na jednosmernoj strani tiristorskih mostova su povezane sijalice kao opterećenje.

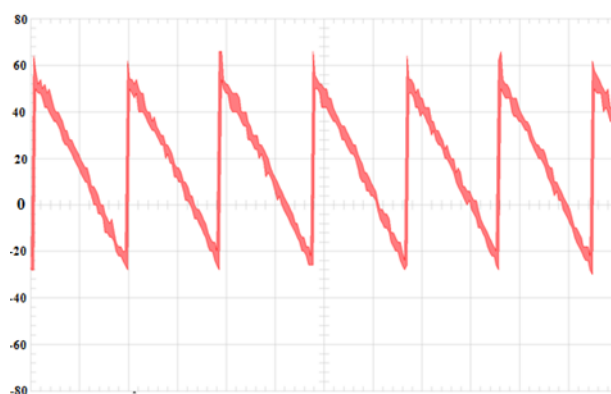
Na upravljačkom panelu je izabran TM1 i test režim. Komandom *Više* jednosmerni napon tiristorskog mosta TM1 je postepeno povećan od 0 do 300V, nakon čega je komandom *Niže* napon ponovo spušten na 0V.

Osciloskopom je snimljen oblik jednosmernog napona tiristorskog mosta TM1.

Na grafiku se vidi da su oblici signala simetrični(testera) kakvi trebaju da budu pri normalno režimu rada. Na isti način je ispitano i za drugi tiristorski (TM2) čiji je dijagram prikazan na slici 7.



Slika 6: Oblik jednosmernog napona TM1



Slika 7: Oblik jednosmernog napona TM2

5. ZAKLJUČAK

Sistem pobude odlikuju brz odziv, visoka pouzdanost, mali gubici, jednostavno upravljanje i održavanje. Primena digitalnih tehnologija i tiristorskih mostova omogućila je redundanciju i u upravljačkom i u energetsom delu sistema pobude. Ugradnjom dva bloka tiristorskih mostova i dva regulatora postiže se visoka pouzdanost sistema pobude. Akvizicioni sistem pruža mogućnost kontinualnog praćenja rada sistema pobude i dijagnostikovanja eventualnih poremećaja u radu.

6. LITERATURA

- [1] Z. Čirić, Đ. Stojić, D. Joksimović, S. Veinović, M. Milinković, *Sistem pobude bloka B1 u TE "Nikola Tesla B" - tehnička dokumentacija*, Beograd, 2013.
- [2] D. Arnautović, Z. Čirić, Đ. Stojić, D. Joksimović, N. Milojević, „Modernizacija, rekonstrukcija i razvoj statičkih sistema pobude“, *Elektroprivreda*, broj 4, str. 111-119, Beograd, 2008.
- [3] Application Manual - Power Semiconductor: *Application Notes for Thyristors and Rectifier Diodes*, Semikron, 2011.
- [4] Technical Information - Bipolar Semiconductors, Infineon, 2012.

- [5] Z. Ćirić, I. Stevanović, Đorđe Stojić, „*Dimenzionisanje energetske opreme za statičke sisteme pobude*“, 26. Savetovanje JUKO CIGRE, Banja Vrućica - Teslić, 2003.
- [6] D. Joksimović, N. Miložčić, „*Primena NI DAQ uređaja i Lab View-a u monitoringu sistema pobude sinhronih generatora*“, NI Days Srbija, Beograd, 2008.
- [7] <https://www.wikipedia.org/>

Kratka biografija:



Bogdan Stajić rođen je 1991. godine u Bijeljini, gdje je živio do početka studija. U Bijeljini stekao je osnovno i srednje obrazovanje. Fakultet tehničkih nauka u Novom Sadu upisao je školske 2010/2011, odsek energetika, elektronika i telekomunikacije. Nakon završenih osnovnih akademskih studija, nastavio je sa master studijama i odbranio master rad 2018. godine iz oblasti Elektrotehnike i računarstva.



Zoran Ivanović docent je na Fakultetu tehničkih nauka u Novom Sadu, na katedri za Energetsku elektroniku i pretvarače. Predaje i bavi se obnovljivim izvorima električne energije.

PRIMENA INTERNET STVARI U RAZLIČITIM ŽIVOTNIM OKRUŽENJIMA INTERNET OF THINGS USE CASES IN DIFFERENT ENVIRONMENTS

Sonja Živković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Cilj ovog rada jeste analiza upotrebe internet stvari (eng. Internet of Things – IoT) u različitim životnim okruženjima. Objasnjeno je šta pojam IoT označava. Pomenute su tehnologije koje su dobar izbor pri kreiranju IoT sistema. Predstavljena je upotreba IoT-a u rudama, gradovima i kućama.

Ključne reči: Internet stvari, pametni uređaji, pametni gradovi, pametne kuće

Abstract – The aim of this paper is to analyze the use of IoT in different environments. It explains what the term Internet of Things (IoT) means. It describes technologies are a good choice when creating an IoT system. The use of IoT in mines, cities and houses is presented.

Keywords: Internet of Things, smart devices, smart cities, smart houses

1. UVOD

IoT je mreža međusobno povezanih fizičkih uređaja koji razmenjuju podatke i informacije preko ugrađenih senzora i aktuatora.

Jednostavno rečeno, IoT predstavlja mrežu uređaja povezanih na internet kako bi ljudski život bio olakšan.

IoT sadrži određen broj protokola, domena i aplikacija. Što je bolja konekcija i što su bolje razvijeni servisi, to će povezanost između uređaja biti bolja i efikasnija. Uređaji će omogućiti automatizaciju u skoro svakom okruženju, polazeći od pametnih kuća, preko pametnih zgrada, do pametnih gradova. Pametni uređaji prikupljaju bitne informacije koristeći najnovije tehnologije i zatim šalju ove informacije drugim uređajima.

1.1. Referentni model IoT-a

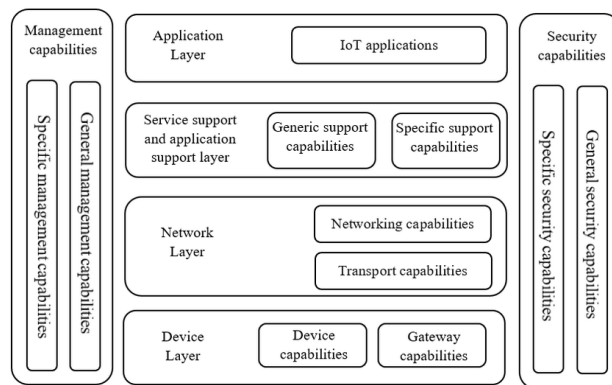
ITU-T definiše referentni model za IoT. Model je podeljen u četiri sloja [1]:

- Aplikativni sloj (Application Layer),
- Servisna podrška i sloj za aplikativnu podršku (Service support and application support layer),
- Mrežni sloj (Network Layer) i
- Fizički sloj (Device layer)

Svaki od ovih slojeva sadrži upravljačke sposobnosti (Management Capabilities) i bezbednosne sposobnosti (Security capabilities). Ove sposobnosti imaju mogućnosti da prekriju više slojeva, što je prikazano na slici 1.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinac, vanr. prof.



Slika 1. Referentni model IoT-a

Prednosti IoT-a su kvalitetna komunikacija, automatizacija i kontrola, informativnost, ušteda vremena i novca, efikasnost i kvalitetniji život.

Mane IoT-a su kompatibilnost, povećanje nezaposlenosti, kompleksnost, bezbednost, privatnost i tehnološka zavisnost.

U skladu sa napretkom tehnologije i razvojem pametnih uređaja, upotreba internet stvari se širi na sve više aspekata ljudskog života. Neki od polja na kojima se koriste, i koristiće se sve više, su: pametne kuće, nosivi uređaji, industrije, pametni gradovi, poljoprivreda, energija, zdravstvo i životna sredina.

2. PROBLEMI U RAZLIČITIM ŽIVOTNIM OKRUŽENJIMA

2.1. Problemi u rudama

Rudna nalazišta mogu da budu zatvorenog (tuneli) i otvorenog tipa (jame).

Ako dođe do odrona u tunelima ruda, vrlo brzo nestaje vazduha. Takođe, put do nekih delova tunela u tom slučaju može biti potpuno zakrčen, a ljudi unutra zarobljeni. Različite otrovne ili zapaljive supstance mogu da se nađu u vazduhu i to predstavlja opasnost za ljude koji se nalaze u tako ograničenom prostoru, sa malom količinom vazduha.

Ni operacije u jamama nisu bezazlene, ako se uzmu u obzir razmere ovakvih operacija. Takođe, zemljište ovih jama se konstantno menja usled rada ogromnih mašina, koje upotrebljavaju ogromnu silu, i time utiču na promenu tla. Takođe se tuda kreću i ljudi, koji treba da budu zaštićeni na takvom terenu. Opasnost nije vezana samo za ova ogromna vozila i prašinu u vazduhu, koja čini da vozači tih vozila ne vide jasno sve oko sebe, već mogu da ugroze bezbednost ljudi oko. Oko jama se takođe nalaze

brda sa strmim ivicama, i potrebno je nadgledati ovakve površine i detektovati i najmanji pomeraj ovog zemljišta, jer svaki može da izazove katastrofu i ugrozi živote ljudi koji se nalaze u jamama. Ovde se takođe koristi puno hemije i eksploziva, i potrebno je na adekvatan način upravljati istim da ne bi došlo do eksplozije.

U rudarstvu se koriste ogromna vozila koja prelaze jako velike rute kako bi se iskopale i utovarile razne mineralne sirovine. Jako je važno da ljudi, koji upravljaju vozilima i ostalim mašinama, budu bezbedni.

2.2. Problemi u gradovima

Upravljanje gradovima predstavlja veliki izazov jer ima za cilj da olakša život stanovništvu, ali i ujedno da smanji troškove.

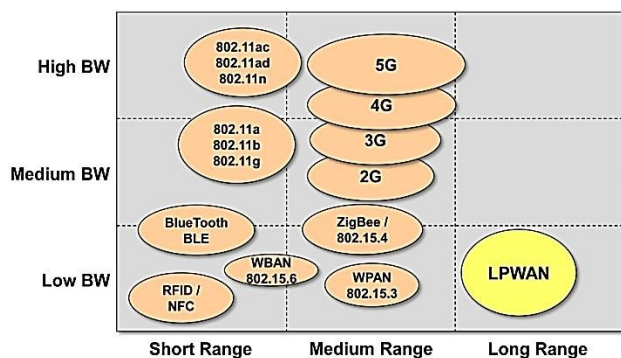
Gradovi koji ne koriste pametnu rasvetu imaju jako veliki svakodnevni utrošak električne energije. Svetla na banderama rade po čitavu noć istog intenziteta, čak i kada na ulicama nema nikoga. To je uzaludno trošenje električne energije. Ljudi koji žive u gradovima jako puno vremena provedu u saobraćaju, tražeći parking. Često je slučaj da su neke zone grada više posećene nego druge. Visoku posećenost u svim gradovima ima sam centar grada. To rezultuje manjkom parking mesta u pomenutom delu grada.

3. TEHNOLOGIJE

Da bi svi pametni uređaji mogli međusobno da komuniciraju, potrebno je koristiti različite tehnologije.

Pri projektovanju i implementaciji novog IoT sistema potrebno je razmotriti sledeće faktore: cena uređaja i mreže, trajanje baterija, brzina prenosa, kašnjenje, mobilnost, domet, pokrivenost i model razvoja.

Na slici 2 prikazan je odnos protoka podataka i opsega nekih od tehnologija koje se koriste u IoT-u. Za rešenje problema u ovom radu predlaže se korišćenje LPWAN (Low Power Wide Area Network), zbog velikog opsega koji pokriva i niskog protoka podataka, što je za slučaj pametnih kuća, gradova, parkinga, ruda i poljoprivrede odgovarajući izbor.



Slika 2. Odnos protoka podataka i opsega tehnologija

3.1. Low Power Wide Area Network (LPWAN)

Stvorene za machine-to-machine (M2M) i IoT mreže, LPWAN mreže funkcionišu po nižim cenama i uz veću energetska efikasnost, u odnosu na tradicionalne mobilne mreže. Takođe su u mogućnosti da podrže veći broj povezanih uređaja na većem području.

LPWAN mogu da koriste licencirane ili nelicencirane frekvencije. Neke od LPWAN tehnologija su: LoRa, SigFox, Weightless, Nwave, Symphony Link, LTE-M [2,3].

Nekoliko najvažnijih izazova LPWAN mreže su:

- Baterija uređaja treba da ima životni vek od najmanje 10 godina
- Rastojanje krajnjih uređaja od bazne stanice treba da bude minimalno 10 kilometara
- Cena uređaja treba da bude niska i pristupačna svima

Ako se posmatra merenje električne energije, vode ili gasa, LPWAN je odličan izbor. Takođe, u slučaju uličnog osvetljenja, gde u gradu može da postoji hiljade sijalica, LPWAN je odličan izbor. Sve te sijalice se konektuju na mrežu, a podaci se šalju onda kada je potrebno - što štedi život baterije. Takođe, u pitanju su velike razdaljine, a LPWAN je dizajniran upravo da omogući da bazne stanice pokrivaju bar 15 kilometara, čime se redukuje sam broj stanica. Senzori su, takođe, dizajnirani da se uklape u nisku cenu.

Bitnu ulogu igra baterija senzora. Ako se senzor nalazi ugrađen ispod parking mesta, baterija bi trebalo da traje dugo jer je nije zgodno menjati često. Ovakva baterija bi trebalo da traje bar 5, ako ne i 10 ili 20 godina. Ako se mnoštvo senzora nalazi u zidu, takođe je potrebno da baterija traje dugo jer ni njih nije zgodno menjati često. Senzori koji ne troše puno baterije jer šalju jednostavne podatke, i to samo onda kada je neophodno, su dobar izbor. Ovde je LoRa dobar izbor jer je brzina prenosa podataka niska, što omogućava manju potrošnju energije, i, u skladu sa tim, baterija će trajati duže.

3.2. LoRa

LoRa (Long Range) je tehnologija koju je razvio Cycleo iz Grenobla, Francuska, a Semtech je 2012. godine kupio. LoRa koristi frekvencije radio-frekvencijskog opsega od 868 MHz (Evropa) i 915 MHz (Severna Amerika) [4]. LoRa omogućava prenos na velike udaljenosti (više od 10km u ruralnim područjima) sa malom potrošnjom energije. Tehnologija je predstavljena u dva dela - LoRa, fizički sloj i LoRaWAN (Long Range Wide Area Network), koji predstavlja gornje slojeve.

LoRa i LoRaWAN omogućavaju jeftinu, dugogodišnju povezanost pametnih uređaja u ruralnim i udaljenim industrijama. Mogu da se koriste u rudarstvu, upravljanju prirodnim resursima, obnovljivoj energiji, kao i u mnogim drugim okruženjima.

Iako se LoRa modulacija može koristiti za prenošenje proizvoljnih okvira, u Semtech-ovim predajnicima i prijemnicima je specificiran i implementiran format fizičkog okvira. Propusni opseg i faktor širenja su konstantni za okvir.

Okvir LoRa počinje sa uvodnom sekvencom (Preamble). Posle uvodne sekvence postoji zaglavlje (Header), koje je opciono. Sama poruka (Payload) se šalje posle zaglavlja. Maksimalna veličina tog polja je 255 bajta. Na kraju je opciono polje Payload CRC. Dužina ovog polja je 1 bajt [5]. Šema koja sumira okvir formata može se videti na slici 3.



Slika 3. Format LoRa okvira [5]

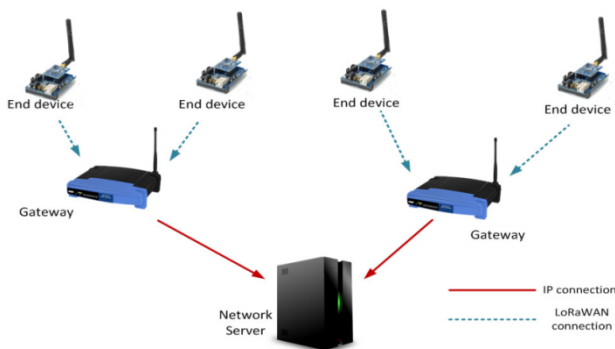
3.3. LoRaWAN

Da bi se implementirala LoRaWAN mreža, potrebne su sledeće komponente:

- Krajnji uređaj (eng. End device): Uglavnom senzori male potrošnje koji komuniciraju sa gateway-om koristeći LoRa modulaciju.
- Gateway: Posredni uređaji koji primaju pakete iz krajnjih uređaja i prosleđuju ih ka mrežnom serveru putem standardnih IP konekcija koje mogu da budu Ethernet ili neka druga mreža. U LoRa okruženju može da postoji više gateway-a i paketi podataka mogu da budu primljeni ili prosleđivani od strane više od jednog gateway-a.
- Mrežni server (eng. Network Server): server koji upravlja mrežom. Mrežni server eliminiše duple pakete, raspoređuje potvrde, i prilagođava brzine prenosa podataka.

Krajnji uređaji nisu povezani sa određenim gateway-om kako bi imali pristup mreži. Gateway služi jednostavno kao veza i prosleđuje pakete, koje dobija od krajnjih uređaja, do mrežnog servera. Prema tome, krajnji uređaj je povezan sa mrežnim serverom koji je odgovoran za otkrivanje duplih paketa, odabira odgovarajućeg gateway-a za slanje odgovora (ukoliko postoji), odnosno za slanje paketa ka krajnjim uređajima [5].

Na osnovu prethodnog opisa, moguće je steći sliku o arhitekturi LoRaWAN mreža, koja je prikazana na slici 4.



Slika 4. LoRaWAN arhitektura [5]

4. PRIMENA IOT-A U RAZLIČITIM ŽIVOTNIM OKRUŽENJIMA

4.1. Uticaj IoT-a na bezbednost u rudama praćenjem kretanja vozila

Jedno rešenje bi bilo da se snima vozačevo ponašanje i izraz lica. Ugradi se kamera koja snima vozačeve oči i položaj glave. Drugi način bi bio da se meri vozačev puls i brzina disanja. Treći način bi bio praćenje samog vozila i detektovanje neočekivanog ponašanja. Kada se detektuje da se vozaču spava, sistem će da zaustavi vozilo i da alarmom probudi vozača. Takođe, rešenje bi bilo i da se koriste potpuno automatska vozila, koja ne zahtevaju da ih vozi čovek.

Svi senzori koji mogu da se nalaze na vozilu prikupljaju različite informacije. Prikupljene informacije mogu se koristiti na više načina. Na primer, prašina je kritična tema u rudnicima otvorenih kopova u suvim vremenskim uslovima. Senzori mogu pomoći utovarivaču da dobije osećaj položaja kamiona za vuču, omogućavajući vozaču utovarivača da dobije pravu informaciju o poziciji i uglu koji treba da zauzme pre nego što počne da utovara sirovine u kamion. Senzori za kamione takođe mogu pružiti povratne informacije utovarivaču.

4.2. Uticaj IoT-a na bezbednost u rudama praćenjem okruženja

Mogu da se prate vremenski uslovi, tako da radnici mogu da znaju kada da očekuju kišu i, u skladu sa tim, budu oprezniji sa stvaranjem većih nagiba. Ove "vremenske stanice" mogu da se nalaze i nekoliko kilometara udaljene od mesta gde se vrše radovi, kako bi detektovale dolazak oluje. Isto tako, mogu da se nalaze i na mestu radova, kako bi merile količinu kiše ili snega koja pada u tom području.

Takođe, mogu se postaviti i radari za kontrolisanje nagiba koji će konstantno meriti položaj, teksturu i ostale osobine zemljišta. Ako se detektuje neka značajna promena, tim radnika može da se fokusira na problem vezan za nagib i tako spreči nastanak klizišta i ugrožavanje bezbednosti ljudi i radova.

Još jedna stvar koju IoT omogućava jesu servisi bazirani na lokaciji. Senzori mogu biti postavljeni na kacige radnika i ugrađeni u vozila. Ovo je veoma bitno i kod radova koji podrazumevaju planirane eksplozije. Senzori mogu sa sigurnošću da daju uvid u to da li su svi radnici van zone opasnosti pre početka operacije.

Još jedna korisna stvar bi bila postavljanje senzora unutar tunela, kod rudarskih radova koji se vrše ispod površine zemlje. Pošto su opasni gasovi uvek problem, ovi senzori bi detektovali prisustvo gasova, npr. ugljen-dioksida. Dávali bi informaciju o tačnoj lokaciji - gasnim džepovima unutar tunela. Praćenjem svakog radnika unutar tunela se tačno zna kada se koji radnik gde nalazi. To omogućava da radnicima bude pružena pomoć u kratkom vremenskom periodu.

4.3. Uticaj IoT-a na uštedu električne energije

Jedan od načina upravljanja uličnom rasvetom je korišćenje pametnih bandera, čiji senzori mogu da detektuju pokret i prisustvo ljudi. U slučaju da nema nikoga na ulicama, senzor će to detektovati i, u tom slučaju, sijalice će se ugasiti. Takođe, kada se neko približi ulici ili prolazi kroz ulicu, sijalice će da se upale. Isto tako, moguća je upotreba sijalica koje menjaju intenzitet svog osvetljenja u zavisnosti od okolne osvetljenosti i prisustva ljudi. Ako je noć, i svetla su ugašena, može se detektovati kada se neko približava ulici, i u skladu sa tim, pojačavati svetlo u toj ulici. Isto tako, ako je sumrak, intenzitet osvetljenja će biti slabiji od intenziteta kada je noć.

Takođe, ukoliko dođe do kvara na nekoj banderi ili sijalica jednostavno prestane da radi, u ovakvom sistemu je to lako uočljivo. Senzor će detektovati da je svetlo prestalo da sija, i poslati podatak o tome na cloud. Ljudi koji su zaduženi za menjanje sijalica i popravku bandera

će odmah biti obavješteni o tačnoj lokaciji kvara, pa mogu pravovremeno i da izvrše popravku, umesto da kvar ostane neprimijećen danima, nedeljama ili čak mesecima.

4.4. Uticaj IoT-a na regulisanje parkinga

Mogu da budu postavljene kamere na sva mesta gde je moguće parkiranje, koje bi detektovale kada postoji slobodno parking mesto. Takođe, može da bude ugrađen senzor ispod parking mesta. Takav senzor detektuje masivni metal. Ispred parkinga može da stoji semafor koji će da sija zeleno ili crveno, u skladu sa tim da li postoji slobodno parking mesto ili ne. Druga opcija bilo bi korišćenje aplikacija koje pružaju informaciju da li, i gde, u okolini, postoji ulica sa slobodnim parking mestom.

Isto tako, korisno je saznanje koja parking mesta su zauzeta više nego ostala, koliko dugo se ljudi zadržavaju na određenim parking mestima, a takođe i koja mesta nikad nisu zauzeta jer ih ljudi nikad ne pronađu. Na takvim mestima se može podići cena parkiranja, da bi ljudi bili podstaknuti da parkiraju na malo udaljenijem mestu. Ako postoji mesto na kom se niko ne parkira, cena tog parking mesta bi mogla da bude manja. Tada bi aplikacija mogla da pošalje obavještenje da se u toj zoni može pronaći slobodno parking mesto, kao i da je parkiranje u toj zoni jeftinije. Dakle, imajući uvid u celokupno stanje parking zona moguće je analizirati sve podatke i uvideti neki šablon, i, koristeći takva znanja, unaprediti sistem parkiranja i omogućiti ljudima da efikasnije pronađu i iskoriste parking mesta.

4.5. Upotreba IoT-a u kućama

Dobar primer je upotreba pametnih brava za vrata. Na bravi može da bude ugrađen senzor koji prepoznaje vlasnikov pametni telefon. Da bi se osiguralo da je do vrata došao baš vlasnik sa telefonom, a ne neko drugi sa njegovim telefonom, potrebno je, recimo, da telefon traži vlasnikov otisak prsta. Kada se vlasnik sa telefonom u ruci približi, vrata se otključavaju. Pametne stvari u kući mogu da uče šablone vlasnikovog ponašanja. Tako, ako vlasnik svaki dan zaključa vrata nakon što uđe u kuću, pametna brava će naučiti to ponašanje i ona će početi automatski da se zaključava svaki put kada vlasnik uđe u kuću. Takođe, automatski će se zaključati svaki put kada vlasnik izađe iz kuće.

Isto se odnosi i na garažna vrata, gde se može koristiti daljinski upravljač za otvaranje vrata, ali isto tako ona mogu i da, na veoma pametan način, detektuju automobil koji prilazi ulazu, i da, prepoznajući automobilsku registraciju ili oblik automobila, otvore garažu. Takodje, garažna vrata se mogu upariti sa pametnim telefonom, gde GPS na telefonu poziva program koji će obavestiti garažna vrata da vlasnik ulazi u ulicu u kojoj živi i vrata će se otvoriti automatski.

Ugrađene kamere na vratima su takođe od velike pomoći. Ugrađeni sistem će naučiti kako izgleda vlasnik, kakav mu je oblik tela i kako hoda. To omogućava da, ukoliko se vratima približi neko, sistem reaguje u skladu sa naučenim ponašanjem. Ako prepozna vlasnika, reagovaće u skladu sa tim – pustiće ga u kuću.

Ako pak prepozna neki drugi oblik tela ili način hodanja, a pritom vlasnik nije najavio da će neko drugi doći, može da uključi alarm, ili pozove vlasnika. Tada vlasnik može preko telefona da kaže svojoj pametnoj bravi šta da radi - da odobri ulazak te osobe u svoju kuću, ili da pozove policiju, ili da jednostavno ignoriše pojavu nepoznatog lica.

Korišćenje pametnih klima uređaja, takođe, sve više ulazi u upotrebu. Klima može da nauči vlasnikovu omiljenu temperaturu prostorije, i da, u skladu sa tim, reguliše temperaturu. Dok je vlasnik na poslu, prostorija može da postane previše topla ili hladna. Kada vlasnik krene kući, pametan telefon može da pošalje poruku da je u blizini kuće, i da se klima automatski upali i reguliše temperaturu, tako da vlasniku bude prijatno kad se vrati sa posla.

5. ZAKLJUČAK

Upotrebom IoT-a itekako može da se doprinese efikasnosti i bezbednosti na svakom polju ljudskog života.

Primena IoT-a će tek da postane deo svakodnevnice ljudi, čak u nekim stvarima u kojima još ljudi ne mogu ni da zamisle. Ono što je loša strana uplitanja IoT-a u život čoveka jeste to što će čovek u bliskoj budućnosti da postane potpuno zavisn od tehnologije – neće moći da zamisli nijednu aktivnost bez nje. U svakom polju svog života će se oslanjati na uređaje i tehnologiju. Već sada je u velikoj meri tako.

Još jedna loša strana toga jeste konstantno zračenje živog sveta, koliko god sada ti signali koji se emituju izgledali beznačajno – kada potpuno okruže čoveka i živi svet uopšte, sigurno će imati nekog lošeg uticaja po zdravlje.

6. LITERATURA

- [1] ITU Telecommunication Standardization Sector, "ITU-T Recommendation database," 2012.
- [2] Kais Mekki, „A comparative study of LPWAN technologies for large-scale IoT deployment“, 2018.
- [3] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara, „Low Power Wide Area Networks: An Overview“, 2017.
- [4] Saša Verić, „Pregled rješenja za realizaciju LoRa bežične mreže“, 2018
- [5] D. Dobrilović, M. Malić, D. Malić, Ž. Stojanov "Analiza performansi lora tehnologije kod mobilnih senzorskih stanica u urbanim okruženjima", 2017

Kratka biografija:



Sonja Živković rođena je u Beogradu 1992. god. Master akademske studije upisala je 2016. god. na Fakultetu tehničkih nauka.

kontakt: sonjazivkovic@hotmail.rs

AUTOMATSKA REINSTALACIJA DMS SCADA SISTEMA POMOĆU DSC-A**AUTOMATIC REINSTALLATION OF DMS SCADA SYSTEM USING DSC**

Jelena Spasojević, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu opisani su problemi upravljanja konfiguracijama koji nastaju usled sve veće potražnje za novijim softverom kao rastom kompleksnosti softvera. U radu je dat opis Desired State Configuration sistema za upravljanje konfiguracijama, njegova arhitektura, dat je pregled različitih verzija DSC-a kao i prednosti DSC-a u odnosu na ručno upravljanje konfiguracijama. Na kraju rada dato je poređenje upravljanja konfiguracijama uz pomoć DSC-a i ručno upravljanje kao i rezultati ovog poređenja.

Ključne reči: *Desired State Configuration, upravljanje konfiguracijama, DMS SCADA.*

Abstract – This paper describes issues in configuration management which occur due to high demand in new software and due to rise of its complexity. Desired State Configuration, a system for configuration management was described, its architecture, different versions of DSC were also described and also its advantages over manual configuration management. At the end of this paper there is a comparison between DSC and manual configuration management and the result of comparison is also described.

Ključne reči: *Desired State Configuration, configuration management, DMS SCADA.*

1. UVOD

Upravljanje konfiguracijom igra centralnu ulogu u integraciji i plasiranju softvera. Kompanije se danas vode logikom brzog i čestog plasiranja manjih softverskih proizvoda. S tim u vezi, proizvodi se prave sa jednom funkcijom u tom trenutku i to se sa vremenom sve više proširuje i nadograđuje. Da bi to bilo moguće proizvodi se moraju napraviti, testirati a zatim često plasirati na tržište.

Ovde je izazov kako omogućiti stalno plasiranje softvera. Plasiranje softvera mora biti dobro osmišljeno i automatizovano kako bi se eliminisale ljudske greške. Kompleksnost brzog plasiranja raste sa kompleksnošću softvera. Upravo zbog sve veće kompleksnosti softvera kao i težnji kompanija za što bržim plasiranjem softvera cilj je što više eliminisati ljudski faktor prilikom upravljanja konfiguracijama.

Rešenje svih nabrojanih problema jeste automatizacija u upravljanju konfiguracijama odnosno Desired State Configuration (DSC).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je dr Darko Čapko.

2. DESIRED STATE CONFIGURATION

Desired State Configuration predstavlja opciju upravljanja konfiguracijama koja je ugrađena u Windows operativni sistem. DSC je relativno novi učesnik u svetu upravljanja konfiguracijama. Prvi put se javlja u okviru Windows Powershell 4.0 verzije koja je izašla u oktobru 2013 godine [2]. DSC je veoma značajan jer je proizašao iz samog operativnog sistema i što se alati ugrađeni u sistem mogu koristiti da se izvršavaju konfiguracione promene. DSC zauzima veoma značajno mesto jer je proizašao iz samog operativnog sistema i što se alati ugrađeni u sistem mogu koristiti da se izvršavaju konfiguracione promene.

Windows operativni sistem nudi standardizovan način upravljanja konfiguracijama a upravo na tim standardima zasnovan je i DSC. DSC treba shvatiti kao platformu za upravljanje konfiguracijama pre nego kompletno rešenje za upravljanje konfiguracijama. Za razliku od drugih alata ili rešenja, DSC ne nudi alate za upravljanje i nadgledanje konfiguracija po modelu end to end, već nudi API (Application Programming Interface) kao i platformu koju čak drugi alati mogu da koriste [1].

2.1 Arhitektura DSC-a

Arhitektura DSC je prilično jednostavna. Na slici 1 prikazana je arhitektura DSC gde su jasno vidljive dve izražene faze.

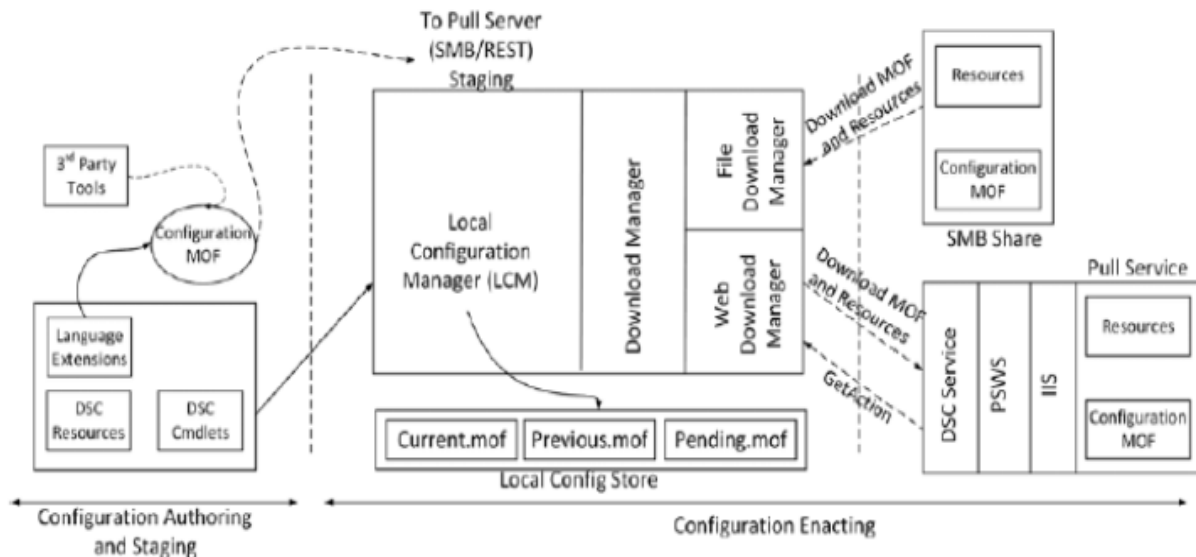
Svaka od ovih faza uključuje komponente DSC koje imaju specifičnu svrhu. Dve glavne faze su:

- faza izrade i testiranja konfiguracije
- faza puštanja u rad

U fazi izrade pišu se konfiguracione skripte koje definišu kako se resursi na sistemu moraju koristiti, kao i koji resursi su neophodni i kako se konfiguracija dalje priprema za sledeću fazu. Nakon faze izrade sledi faza gde se konfiguracija testira i nakon toga sledi faza puštanja u rad. U fazi puštanja u rad konfiguracija se primenjuje na ciljani sistem, a tu se izdvajaju dva različita načina na koje se konfiguracija može primeniti, a to su Push i Pull metode. Zapravo, u fazi izrade dolazi do prijema konfiguracionog fajla na ciljani sistem a zatim primenjivanje promena. Svaka od ovih faza ima specifičnu ulogu u arhitekturi DSC –a [1].

2.2 Faza izrade i testiranja

Ova faza je prva faza u DSC arhitekturi. U njoj se pišu konfiguracione skripte i testiraju i pripremaju za fazu puštanja u rad gde se izmenjena konfiguracija primenjuje. Kao što se iz samog imena prve faze vidi, ona se sastoji iz dva dela a to su izrada i testiranje.



Slika 1 – Arhitektura Desired State Configuration

U fazi izrade koriste se Windows PowerShell deklarativne skripte. DSC PowerShell modul je prva komponenta koja se posmatra pri pisanju skripti u DSC PowerShell-u. Sve jezičke ekstenzije, koje se koriste u skriptama, implementiraju se unutar DSC PowerShell modula.

Ovde je najvažnije izdvojiti funkciju Configuration. Ona se smatra ključnom komponentom DSC-a, pošto ova funkcija definiše željenu konfiguraciju sistema.

Ono što sledi nakon reči Configuration jeste ime ili identifikator koji je dodeljen konfiguraciji.

Konfiguracija željenog sistema definiše se koristeći PowerShell skript blok koji se identifikuje koristeći blok kod unutar zagrada {}.

```
Configuration IdentifierString {
}
```

Iako je funkcija Configuration ključni aspekt upravljanja konfiguracijama, ova funkcija sama ne definiše konfiguraciju koju želimo da kontrolišemo. Da bi se to uradilo mora se koristiti funkcija Node jer ona određuje na koji sistem se primenjuje konfiguracija. Nakon toga neophodno je navesti šta se menja u konfiguraciji odnosno navode se resursi koje menjamo. Postavlja se pitanje kako da znamo koji resursi su dostupni i koja su njihova imena, jer su nam neophodna da bismo ih koristili.

To se može videti korišćenjem DSC PowerShell modula odnosno naredbe Get-DscResource. Ovom naredbom mogu se videti svi DSC resursi koji su dostupni na sistemu. Unutar Node skript bloka može se navesti bilo koji broj resursa, bilo istih, bilo različitih sve dok se koristi drugačiji identifikator kao i ključne karakteristike poput imena.

DSC skripta bi izgledala ovako

```
Configuration WebsiteConfig {
    WindowsFeature WebServer1{
        Name = "Web-Server"
    }
}
```

Kada je DSC skripta napisana, to nije dovoljno da bi ona napravila promene na sistemu. Neophodno je da se skripta prevede u format koji DSC Local Configuration Manager razume. Taj format je Managed Object Format (MOF). MOF format se generiše jednostavnim učitavanjem konfiguracije u memoriju, a zatim njenim pozivanjem.

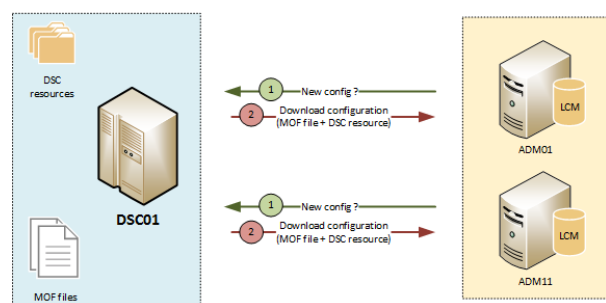
2.3 Faza puštanja u rad

Ova faza dolazi nakon faze izrade i testiranja. U fazi izrade generisan je MOF fajl na mrežnoj lokaciji, ali to nije dovoljno da ga sistem koji želimo da promenimo primi i da reaguje na promene. MOF se mora postaviti na željeni sistem. Da bi se to uradilo koriste se Push i Pull funkcije u DSC-u. [1]

2.4 Pull model

U Pull modelu nodovi se konfigurišu da dobiju konfiguraciju od strane pull servera. Ovaj način slanja konfiguracije je dobar kada se mašine moraju konfigurišati brzo.

Da bi se koristio Pull model neophodan je server sa instaliranim Windows DSC. Takođe, server mora da ima Windows Management Framework 4 (WMF) instaliran. Jedan jednostavan primer tehničkog okruženja prikazan je na slici ispod.



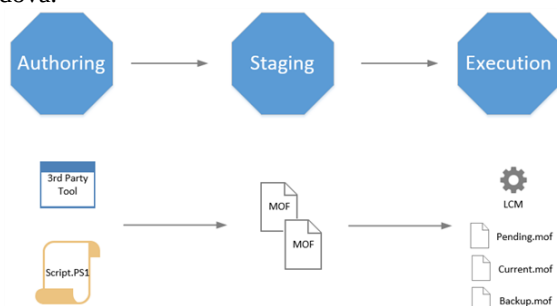
Slika 2 – Pull model – tehničko okruženje

Pull mode je najkompleksniji, ali najlakši za održavanje. Jedan primer nedostatka push modela u odnosu na pull model je ako u push modu nod bude offline promena konfiguracije neće uspeti. U tom slučaju Sistem administrator mora stalno da pokušava da pošalje novu konfiguraciju. U pull modu to nije tako, on je mnogo dinamičniji. Kakav god da je status noda konfiguracija će se primeniti čim on bude online.

Pull server je IIS Web server koji sadrži MOF fajlove za nodove. Ovaj server takođe odgovara na zahteve LCM (Local Configuration Manager) zahteve za distribucijom konfiguracija. To je upravo i prednost Pull moda jer se DSC resursi preuzimaju automatski sa servera od strane svakog noda, a zatim u regularnim intervalima nodovi šalju svoj status pull serveru da provere promene u konfiguraciji [4].

2.5 Push model

Push model je idealan u uslovima testiranja ili u okruženjima sa serverima gde se konfiguracija primenjuje jednom na duži period. Kod Push modela korisnik odlučuje da ručno primeni konfiguraciju na jedan ili više nodova.



Slika 3 – Prikaz Push modela

Kao što je već napomenuto u fazi izrade stvara se DSC konfiguracija. To se može uraditi sa jednostavnim tekstualnim editorom poput Notepad ili PowerShell ISE. Kada se konfiguracija napravi onda se ona mora izvršiti kako bi se napravili jedan ili više MOF fajlova koji sadrže informacije za konfigurisanje nodova. Po nodu može postojati samo jedan MOF fajl. To omogućava da nod ima celokupnu konfiguraciju odjednom i osigurava da je primenio odgovarajuću konfiguraciju.

Nakon sastavljanja MOF fajlova na jednom računaru koristi se push model kako bi se konfiguracija prenela na druge računare uz pomoć Start-DSCConfiguration komandleta. Ovde postoji opcioni parametar - ComputerName koji omogućava korisniku da naznači na koji nod želi da primeni konfiguraciju. Ako nije naznačeno ime noda LCM utvrđuje sve fajlove koji se nalaze u konfiguracionom folderu kako bi identifikovao na koje nodove se fajlovi odnose.

Konačni korak je konfiguracija nodova sa informacijom primljenom preko MOF fajla. LCM odmah izvršava MOF fajlove nakon njihove analize. Fajlovi se smeštaju u svakom nodu na sledećem direktorijumu:

\$env:systemRoot\system32\configuration

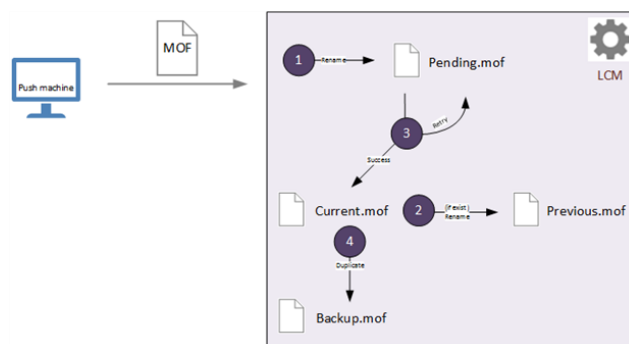
Kada se izvrši push jednog MOF fajla na nod pojaviće se nekoliko MOF fajlova. [4]

Ti fajlovi su kreirani prema specifičnoj svrsi. Koraci u njihovom nastajanju su sledeći:

- Prvi korak: LCM prima i izvršava MOF fajl. Preimenuje ga u **pending.mof**

- Drugi korak (opciono): ako postoji, preimenuje se aktuelni MOF fajl **current.mof** u **previous.mof**
- Treći korak: kada se nova konfiguracija primeni uspešno, preimenuje se **pending.mof** u **current.mof**
- Konačni korak: od **current.mof** se pravi duplikat **backup.mof**

Na sledećoj slici može se videti dijagram ovog procesa.



Slika 4 - Dijagram procesa MOF fajlova

3. PREDNOSTI DSC-a

Neke od osnovnih prednosti Desired State Configuration su sledeće:

- DSC je veoma lak za čitanje, čuvanje i promenu.
- Opisuje stanje u kom se treba naći ciljani uređaj umesto da sadrži instrukcije kako da uređaj dođe u željeno stanje.
- Manje košta da se nauči i implementira.
- Ponovne implementacije specifičnog seta mašina su mnogo manje podložne greškama.
- Mnogo brže se implementiraju promene konfiguracija i mnogo su pouzdanije implementacije tih promena.
- Konfiguracije se mogu deliti uz pomoć PS galerije, što znači da mogu postojati uobičajeni scenariji.
- Konfiguracije se može dodati u source kontrolu u verziji zajedno sa aplikacijom koja se razvija. [5]

4. IMPLEMENTACIJA DSC

Configuration funkcija je najvažnija komponenta DSC-a jer kao što je već napomenuto ona definiše željenu konfiguraciju sistema. Svaki Configuration fajl počinje na isti način.

```
Configuration Name
{
}
```

Nakon reči Configuration sledi ime ili identifikator, a potom sledi kod koji se nalazi unutar zagrada {}.

Konfiguracija se, u ovom radu, sastoji od sledećih koraka opisanih skriptama:

- LogOffUsers – odjavljivanje korisnika
- StopServices – zaustavljanje servisa
- Uninstall – uklanjanje instalacije
- Install – postavljanje nove instalacije
- ConfigSystem – konfigurisanje sistema
- DistribuSys – dobijanje instance CIM klase
- Database – podešavanje baze podataka

5. REZULTATI TESTIRANJA

Bilo kakva vrsta automatizacije, kao osnovni rezultat ima uštedu potrošenog vremena na realizaciji nekog poslovnog procesa.

Tabela 1. prikazuje ukupno utrošeno vreme za automatizaciju. Prikaz podataka je dat hronološkim redosledom realizacije, kao što je prethodnom poglavlju objašnjeno.

***Napomena:** Za potrebe testiranja u ovom radu, razmatraće se vreme koje je potrebno pri reinstalaciji softvera na jednom mikro sistemu.

Tabela 1: Prikaz utrošenog vremena kod kod automatske reinstalacije pomoću DSC-a

Automatizovani koraci	Vreme trajanja
Odjavljivanje svih korisnika sa sistema	7 sec
Stopiranje servisa	25 sec
Uklanjanje postojeće instalacije softvera	50 sec
Postavka nove instalacije softvera	40 sec
Distribusys	12 sec
Konfiguracija	5 min
Podešavanje baza	15 min
Ukupno utrošeno vreme:	22 min i 21 sec

Tabela 2: Prikaz utrošenog vremena kod manualnih koraka za reinstalaciju softvera

Manuelni koraci	Vreme trajanja
Odjavljivanje svih korisnika sa sistema	5 min
Stopiranje servisa	10 min
Uklanjanje postojeće instalacije softvera	15 min
Postavka nove instalacije softvera	10 min
Distribusys	15 min
Konfiguracija	30 min
Podešavanje baza	45 min
Ukupno utrošeno vreme:	2h i 5 min

Kada se ovakav rezultat utrošenog vremena kod ručne reinstalacije sistema uporedi sa utrošenim vremenom automatskereinstalacije, zaključuje se da je automatizacija ovakvog poslovnog procesa od velike važnosti.

6. ZAKLJUČAK

Cilj ovog rada bila je automatizacija reinstalacije softvera na distribuiranom sistemu. Predstavljena je automatizacija u upravljanju konfiguracijama, odnosno Desired State Configuration, njena arhitektura, prednosti kao i razlozi za uvođenje.

Dat je i pregled razvoja DSC-a od prve verzije pa do verzije koja se danas koristi.

Na samom kraju rada iznet je i praktični test sa rezultatima koji jasno govore prednost automatizacije u odnosu na klasične pristupe upravljanja konfiguracijama.

Sam proces automatizacije konfiguracionih koraka za reinstalaciju softvera, dosta je ubrzao rad na distribuiranom sistemu. Prilikom manualnog izvršavanja ovih konfiguracionih koraka, potrebno je u proseku oko 3 do 4 sata, u zavisnosti od veličine sistema, dok je za ceo proces sa automatizacijom pomenutih koraka potrebno oko 25 minuta, što je znatno ubrzalo sam proces.

Ostale bitne karakteristike i prednosti ovakve automatizacije ogledaju se i u:

- Ponovljivost koraka.
- Minimalizovan faktor ljudske greške.
- Poboľšane performanse koje daju bolje rezultate.

Jedan od daljih koraka razvoja jeste modifikacija i primena navedenih koraka automatskog konfigurisanja na većim distribuiranim sistemima, jer je u ovom radu ovaj proces automatskog konfigurisanja primenjen na mikro (malom) distribuiranom sistemu.

Jasno je da je automatizacija neophodna i da će se u budućnosti sigurno tražiti načini za njeno unapređenje jer se softveri razvijaju sve većom brzinom i postaju sve više i više kompleksniji.

7. LITERATURA

- [1] Windows PowerShell Desired State Configuration Revealed - Ravikanth Chaganti, izdata 23.09.2014
- [2] <https://www.itprotoday.com/management-mobility/windows-powershell-40-released-windows-7-windows-2008-r2-and-windows-server-2012>, datum pristupa 23.09.2018.
- [3] Learning PowerShell DSC Second edition – James Pogram, izdata 19.10.2015
- [4] <https://www.red-gate.com/simple-talk/sysadmin/powershell/powershell-desired-state-configuration-the-basics/>, datum pristupa 22.09.2018.
- [5] <https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/?view=powershell-6>, datum pristupa 22.09.2018.
- [6] Pro PowerShell Desired State Configuration – Ravikanth Chaganti, izdata 25.04.2018.

Kratka biografija:



Jelena Spasojević rođena je 1990. godine u Kragujevcu. Srednju školu je završila u Kragujevcu 2009. godine. Fakultet Tehničkih Nauka u Novom Sadu je upisala 2009. godine. Ispunila je sve obaveze i položila sve ispite predviđene studijskim programom na smeru Energetika, elektronika i telekomunikacije, usmerenju Telekomunikacioni sistemi. Odmah nakon toga je upisala master akademske studije, takodje na istom na smeru i ispunila sve svoje obaveze i položila sve ispite predviđene studijskim smerom.

REALIZACIJA MAŠINE PRAVILA U RUKOVANJU ALARMIMA U NADZORNO-UPRAVLJAČKOM SISTEMU**AN IMPLEMENTATION OF ALARMS RULES ENGINE IN SCADA SYSTEM**Vladimir Papuga, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu opisan je razvoj kao i implementacija mašine pravila (en. Rules engine) za alarmiranje u SCADA sistemima. Opisana je implementacija rešenja koje se zasniva na upotrebi expression trees i ukratko su pomenuti SCADA sistemi i značaj alarmiranja u tim sistemima.

Ključne reči: SCADA sistemi, Mašina pravila, Alarmni sistemi, Expression Trees

Abstract – This paper describes the development and implementation of Rules Engine for alarming in SCADA systems. Implementation of the solution based on expression trees is described. SCADA systems and significance of alarm in these systems are briefly described.

Ključne reči: SCADA systems. Rules engine, Alarming systems, Expression Trees

1. UVOD

Od pojave upravljačkih sistema, SCADA (*Supervisory Control and Data Acquisition*) je odigrala važnu ulogu u oblasti automatizacije. SCADA sistemi nude sredstvo za nadzor i upravljanje industrijskim procesom posredstvom udaljenih uređaja. Nadzor se realizuje prikupljanjem podataka, pri čemu se podaci dobijeni od uređaja, dalje obrađuju i prikazuju korisniku – operateru. U okruženju koje kontroliše operater, koristeći SCADA-u, neophodan je alarmni sistem. Alarmni sistem služi da obavesti operatera o abnormalnim procesnim uslovima i neispravnosti uređaja. Na primer, kada god izmerena procesna vrednost (signal) prekorači očekivane granice, to je abnormalnost koja se mora proveriti kako bi se sprečile moguće posledice. Alarm se mora generisati kada se uoči da se proces ne ponaša kako je predviđeno, pre nego što se desi neki kritičan događaj, ali samo onda kada je u pitanju kritičan događaj. Svako kritično stanje trebalo bi generisati jedan alarm koji bi precizno ukazivao na problem u sistemu i tako omogućio operateru pravilnu reakciju. Kompleksnost upravljanja alarmima raste uporedno sa kompleksnošću nadziranog sistema. Tu se prikuplja velika količina podataka koji mogu “preplaviti” operatera i zbog toga alarmni sistemi moraju biti dobro dizajnirani, implementirani i održavani.

Ovaj rad fokusira se na pojednostavljenju održavanja dela alarmnog sistema zaduženog za generisanje alarma, pogotovo kada se radi o čestim izmenama (dodavanje novih uslova alarma, menjanje logike algoritama, itd.).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Erdeljan, red. prof.

Jedno od mogućih rešenja za opisivanje stanja alarma zasnivalo bi se na upotrebi većeg broja uslovnih iskaza (if-else, switch), koji bi vremenom, sa proširivanjem sistema doveli do nepreglednog koda i težeg održavanja. Kako bi se izbegao ovaj problem osmišljena je mašina pravila zasnovana na expression trees, čiji je cilj da omogući jednostavnu modifikaciju alarmnih pravila, samim tim i jednostavno održavanje ovog dela sistema, a da se pri tome postignu zavidne performance.

2. SCADA ALARMNI SISTEM

SCADA je akviziciono-upravljački sistem, sa primarnim funkcijama objedinjavanje i obrada svih procesnih podataka, i komunikacije sa operaterima. Tipično obuhvata više računara: servera i radnih stanica, gde se serveri brinu o prikupljanju, obradi i skladištenju informacija o nadziranom sistemu, a radne stanice prikazuju te informacije operaterima i omogućavaju im blagovremene reakcije. Objedinjavanje procesnih podataka ostvaruje se komunikacijom sa mrežom instaliranih procesnih kontrolera.

Obrada prikupljenih podataka podrazumeva proveru ispravnosti i vrednosti merenja, uz eksplicitno izveštavanje operatera o otkrivenim neregularnostima [1]. Pored prikupljanja merenja, SCADA sistem može formirati izvedene (sračunate) vrednosti, pratiti ispravnost računarske i komunikacione opreme, pratiti aktivnosti korisnika i sl. Stoga je alarmni sistem naizostavna komponenta SCADA-e i svojim performansama bitno utiče na upotrebljivost sveukupnog rešenja.

Osnovni parametri kojima se mere real-time performanse SCADA softvera vezani su za ukupan broj procesnih veličina koje se mogu konfigurisati u SCADA bazi, i za broj promena koje se mogu obraditi i prikazati u nekom vremenskom periodu. Zato je dizajn SCADA softvera primarno orijentisan ka efikasnosti ukupnog sistema, ali i njegovoj adaptivnosti na raznovrsne aplikativne zahteve prisutne u realnim implementacijama.

Alarmni sistem je osnovni sistem za podršku operatera za rukovanje abnormalnim situacijama i ima dve funkcije:

1. Primarna funkcija alarmnog sistema jeste da upozori operatera o situaciji koja nije normalna. Ova funkcija pomaže operateru da kontroliše buduće ponašanje složenog procesa skrenuvši mu pažnju na neželjene procesne uslove. Sistem treba da obavesti operatera o uslovima procesa koji zahtevaju blagovremenu procenu i eventualno korektivne mere u cilju održavanja ciljeva procesa u smislu sigurnosti, produktivnosti, efikasnosti, itd. Svaki alarm treba da upozori, obavesti i usmerava operatera [2].

2. Sekundarna funkcija alarmnog sistema je da beleži događaje i omogući njihovu naknadnu analizu i optimizaciju postrojenja. Treba da bude fleksibilan i da sadrži događaje, potisnute alarme i druge delove informacija koje nisu predstavljene na glavnoj listi alarma, ali koje bi mogle biti korisne za *offline* istraživanje incidenata. [2] Cilj potisnutih alarma je da obezbedi da predstavljeni alarmi u bilo kom trenutku budu relevantni za rad operatera u trenutnim procesnim uslovima i da se izbegnu "poplave" alarma. Potiskivanjem alarma ne uklanjaju se informacije iz sistema, već se samo razdvajaju informacije o alarmu na jedan ili više detaljnih nivoa. Glavni alarmni prikaz treba da obezbedi praćenje i kontrolu budućeg ponašanja sistema privlačenjem pažnje operatera prema uslovima procesa koji zahtevaju procenu ili akciju. Trebalo bi da prikaže samo alarme koji su relevantni u trenutnim uslovima procesa. Alarmi trebaju biti relevantni za ulogu operatera, da nagoveste potrebnu akciju, da se prikazuju po stopi koju operater može da isprati i da su lako razumljivi, kako bi sistem ostao upotrebljiv za operatera u svim situacijama.

2.1 Rukovanje alarmima

Pod rukovanjem alarmima (*en. Alarm management*) podrazumeva se skup praksa i procesa koji osiguravaju efikasan alarmni sistem. ISA18.2 standard definiše procese i postupke potrebne za kreiranje efikasnog sistema rukovanja alarmima. Jedan od najčešće primenjivanih modela jeste životni ciklus upravljanja alarmima (*en. Alarm Management Lifecycle*). Ovaj model se može primeniti na novi ili postojeći alarmni sistem [3].

3. DOSADAŠNJA REŠENJA

Realizacija jednostavnog rešenja za generisanje alarma se može zasnovati na izvršavanju brojnih uslovnih iskaza gde se npr. izmerene vrednosti porede sa željenim granicama. Mana ovakvog rešenja nije uočljiva dok je sistem mali, ali sa porastom sistema i njegove kompleksnosti raste broj, kao i kompleksnost uslovnih iskaza potrebnih za njegovo opisivanje. Ukoliko je sistem podložan čestim modifikacijama, ovakvo rešenje će dovesti do otežanog, kao i vremenski zahtevnijeg održavanja.

Kako bi se izbegao gore opisan scenario, osmišljena je mašina pravila bazirana na *expression trees* čiji je cilj da omogući jednostavno održavanje sistema, razlaganjem složenih komponenti alarmiranja na manje delove. Na ovaj način, obrada alarma opisana preko mnoštva ugnježdenih petlji se pojednostavljuje i opisuje preko skupa jednostavnih, čitljivih pravila. Modifikacija sistema vrši se jednostavnim dodavanjem ili brisanjem pravila iz konfiguracionog fajla.

Mašina pravila je ekspertski sistem baziran na pravilima i sastoji se od skupa parova tipa "*uslov - akcija*". Ukoliko je uslov pravila zadovoljen, izvršava se akcija vezana za taj uslov. Glavna prednost ovakvog pristupa jeste da se pomoću pravila može znatno pojednostaviti opisivanje rešenja veoma složenih problema.

Ekspertski sistem je program koji je projektovan da modelira sposobnosti rešavanja problema ljudskog eksperta u nekoj oblasti. Kod ekspertskih sistema naglasak nije na obradi podataka (algoritmom,

programom) kao kod konvencionalnih rešenja, već na oblikovanju (baze) znanja.

U daljem tekstu biće detaljnije opisan ekspertski sistem i mašina pravila.

4. EKSPERTSKI SISTEMI

Ekspertski sistem je inteligentni računarski program koji koristi znanje i postupke zaključivanja u procesu rešavanja problema za koje je potreban visok stepen stručnosti i iskustva iz domena kome se ekspertski sistem obraća. Osnovu ekspertskih sistema karakterišu sledeća svojstva: mora da sadrže znanje (iz konkretne oblasti – domena), da vrše zaključivanje, prosuđivanje i odlučivanje na osnovu nepouzdatih i nepotpunih informacija i da omoguće tumačenje svog ponašanja. Ekspertski sistem čini vise delova: deo za izvršavanje problema (baza znanja, mehanizam zaključivanja i globalna baza podataka) i okruženja (korisnički interfejs). **Baza znanja (knowledge base)** – baza činjenica i heuristika u području za koje je namenjen ekspertski sistem.

Mehanizam zaključivanja – softver sposoban da sredi informacije iz baze znanja i da na osnovu toga izvuče zaključke.

Globalna baza podataka – radna memorija za beleženje trenutnih statusa sistema, ulaznih podataka za određeni problem.

Komunikacioni interfejs – deo koji omogućava dijalog između donosioca odluke (korisnika) i sistema.

4.1 Predstavljanje znanja u ekspertskim sistemima

Osnovna namena reprezentacije znanja (*en. Knowledge representation*) jeste predstavljanje znanja (eksperta sistema) u simboličkom obliku. Ne postoji idealan tip predstavljanja znanja koji može pokriti i okarakterisati sve delove nekog ekspertskog sistema. U mnogim formalnim sistemima pod činjenicom, kao oblikom deklarativnog znanja, obično se podrazumeva iskaz – rečenica koja ima dve moguće vrednosti, tačna ili netačna. Osnovni modeli za predstavljanje znanja u ekspertskim sistemima su: *pravila, mreže zaključivanja, semantičke mreže, okviri, O-A-V trojke, objekti, iskazna logika i logika prvog reda*.

5. MAŠINA PRAVILA - PRODUKCIONI SISTEM PRAVILA

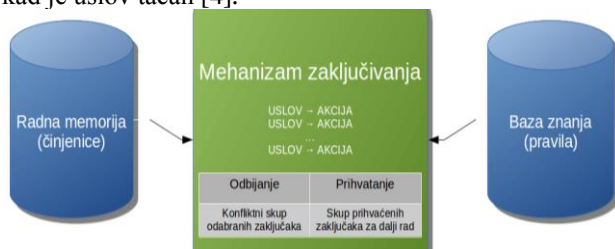
Mašina pravila (*en. Rules Engine*) je ekspertski sistem koji se sastoji od skupa parova tipa "*uslov - akcija*". Ovako definisana pravila nazivaju se produkciona pravila ili još pravila odlučivanja, pa se često u literaturi i praksi ovakvi sistemi nazivaju produkcionim ekspertskim sistemom ili *produkcionim sistemom pravila (en. Production Rule System)*. Osnovu sistema čine baza činjenica koje se smeštaju u radnu memoriju, baza znanja i procedura za izvršavanje produkcionih pravila i mehanizma zaključivanja. (slika 1).

Baza znanja - produkciona pravila u deklarativnom obliku (*uslov - akcija*) baziraju se na principima prediktivne logike, oblika "*AKO... TADA...*" (*IF... THEN...*). Svako produkciono pravilo sadrži uslov koji mora biti zadovoljen pre nego što se izvrši akcija.

Radna memorija (činjenice) - sadrži skup informacija koje se testiraju kako bi se zaključila istinitost nekog

uslova. Predstavlja činjenice iz globalne baze podataka koje se dovlače u radnu memoriju.

Mehanizam zaključivanja - jednostavna programska petlja koja testira istinitost uslova pravila i izvršava akcije kad je uslov tačan [4].



Slika 1: Mašina pravila - osnove sistema

5.1 Mehanizam zaključivanja

Osnovna struktura pravila u produkcionom sistemu pravila je prilično jednostavna. Svako pravilo ima svoj uslov i akciju. Uslov je *boolean* izraz dok akcija može biti bilo šta, ali i može biti ograničena u zavisnosti od konteksta produkcionog sistema pravila. Na primer, ukoliko sistem vrši samo validaciju, onda se akcije mogu odnositi samo na prijavljivanje grešaka. [5] Stoga je opravdana primena takvih mehanizama zaključivanja u alarmnom sistemu.

Mehanizam zaključivanja je osnovna komponenta u sistemu mašine pravila. Funkcioniše u nizu ciklusa zaključivanja. Sistem sa velikim brojem pravila i činjenica može dovesti do toga da su mnoga pravila istinita za istu činjeničnu tvrdnju. Za ova pravila se kaže da su u sukobu. Agenda upravlja redosledom izvršenja ovih konfliktnih pravila koristeći strategiju za rešavanje konflikata (*izbor pravila sa najvišim prioritetom, izbor prvog pravila, najskorije dodatog pravila, ...*). Konflikt se rešava primenom jedne ili kombinacijom više strategija za rešavanje konflikata pravila. Nakon izvršavanja svakog pravila, ažurira se baza činjenica i mehanizam počinje novi ciklus pretraživanja skupa pravila. Sistem prestaje sa radom kada mehanizam zaključivanja ne nađe pravilo koje može izvršiti, tj. uslovi svih pravila su neistiniti.

Mehanizam zaključivanja zasnovan je na metodama *ulančavanju napred* i *ulančavanju nazad*.

5.2 Prednosti sistema sa mašinom pravila

Prednosti sistema sa mašinom pravila su mnogobrojne i najbolje se mogu opisati preko sledećih karakteristika ovih sistema:

Deklarativni pristup: Prednost ovakvog pristupa jeste da se pomoću pravila može znatno pojednostaviti opisivanje rešenja veoma složenih problema. Pravila su mnogo čitljivija od klasičnog koda.

Razdvajanje logike od podataka: Podaci su u domenu objekta, dok je logika u pravilima. Ovo u osnovi krši principe objektno-orijentisanog programiranja, ali omogućava lakše održavanje i proširivanje sistema u budućnosti.

Centralizacija znanja: Korišćenjem pravila kreira se baza znanja koja je izvršna. U idealnom slučaju, pravila su dovoljno precizno napisana da mogu poslužiti kao dokumentacija.

Integracija: Laka integracija u već postojeće sisteme, kao i velika *brzina* i *skalabilnost*.

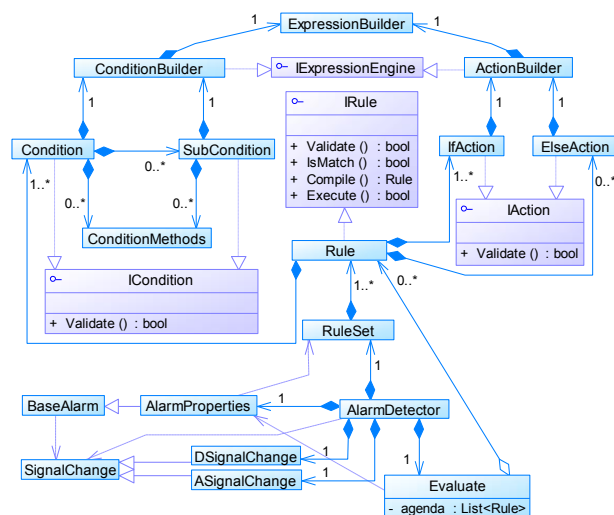
Razumna pravila: Kreiranjem objektnih modela i

opciono, *domen specifičnih jezika* (en. *Domain Specific Languages*) koji modeliraju domen problema može se omogućiti pisanje pravila koja su vrlo blizu prirodnom jeziku. Kao rezultat, opisana logika može biti razumna i ne tehničkim stručnjacima domena [6].

6. PREDLOŽENO REŠENJE

Svaka promena u nadziranom sistemu (npr. promena stanja prekidača, neko merenje, itd.) detektuje se u vidu promene signala koji predstavlja taj element. Da li će se alarm generisati i koji tip alarma će biti generisan zavisi od svojstava signala koja su promenjena, kao i od tipa samog signala. Na osnovu ovih podataka pozivaju se prethodno kreirana pravila definisana u konfiguracionom fajlu. Sva pravila čiji uslovi sadrže svojstva signala koja su promenjena evaluiraju se i u zavisnosti od rezultata evaluacije izvršavaju se odgovarajuće akcije koje mogu dovesti do generisanja alarma.

Na slici 2 prikazan je klasni dijagram koji predstavlja model na osnovu koga se implementira rešenje.



Slika 2: Klasni dijagram modela rešenja

Dato rešenje zasniva se na primeni *expression trees* u implementaciji mašine pravila. *Expression trees* predstavlja strukturu podataka koja sadrži izraze (en. *Expression*), tj. strukturu stabla gde je svaki čvor izraz. Ova struktura je blisko povezana sa lambda izrazima i koristi se za njihovo opisivanje, tj. daje informaciju šta lambda izraz treba da uradi. Ova osobina *expression tree-a* omogućava dinamičnije i fleksibilnije pisanje pravila. Gubi se potreba za korišćenjem složenih uslovnih iskaza jer se pravila, tj. uslovi i akcije opisuju preko ove strukture podataka. Kako bi se dobio izvršni kod koji će izvršiti zadatak opisan u *expression tree-u*, *expression tree* je potrebno kompajlirati. Kompajliranjem dobija se delegat koji predstavlja opisan lambda izraz.

Rešenje se može podeliti na tri celine, kreiranje *expression tree-a*, kompajliranje i izvršavanje pravila.

Kreiranje izraza: Pravila su opisana u konfigurabilnom fajlu *Rules.xml*. Klasa *RuleSet* parsira konfiguracioni fajl i kreira pravila (en. *Rule*). Svako pravilo sadrži kolekciju uslova i dve kolekcije akcija, *IfAction* i *ElseAction*. Svaki tip alarma poseduje svojstva (en. *Property*) koja ga opisuju. Ove osobine se nalaze u klasi *AlarmProperties* i predstavljaju činjenice mašine pravila, tj. skup

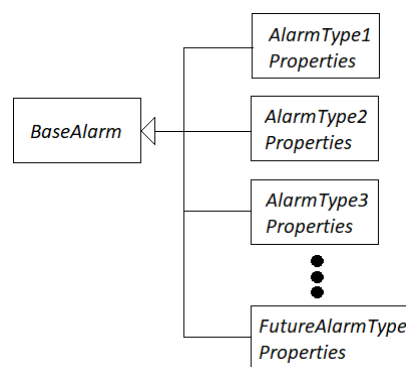
informacija koje se testiraju kako bi se zaključila istinitost nekog uslova. Ukoliko su svi uslovi u kolekciji zadovoljeni, smatra se da je uslov pravila zadovoljen i izvršavaju se akcije. Svako pravilo mora imati definisan *IfAction* - akcije koje se izvršavaju kada je uslov pravila zadovoljen. Svaki tip alarma generiše se na osnovu matematičkog izraza koji predstavlja jedan od uslova pravila, dok se drugi uslovi mogu odnositi na to da li je alarm prigušen, da li je dozvoljeno alarmiranje na konkretnoj tački, itd. Uslovi pravila kreću se od jednostavnih, gde se poredе svojstva alarma sa nekom vrednošću, do kompleksnih, gde se koristi složen matematički izraz. Važna osobina *expression tree-a* je mogućnost spajanja više manjih *expression tree-a* u jedan veći. Imajući u vidu pomenutu osobinu, složeni matematički izraz se razlaže na manje celine na osnovu kojih se prave *expression trees*. U modelu, klasa *SubCondition* služi za opisivanje pomenutih manjih celina koje se objedinjuju u uslovu (*en. Condition*) pravila. Klasa *ConditionMethods* je pomoćna klasa koja ima ulogu u opisivanju složenih izraza. U ovom radu, akcije pravila predstavljaju *expression tree* gde su opisani pozivi odgovarajućih metoda od interesa. U kreiranju *expression tree-a* učestvuju klase *ConditionBuilder*, *ActionBuilder* i *ExpressionBuilder*.

Kompajliranje *expression trees-a*: Kreirana pravila još uvek se ne mogu izvršiti. Uslovi i akcije predstavljeni su preko strukture stabla. Kako bi se dobio izvršni kod koji će neki podatak propustiti kroz stablo potrebno je izvršiti kompajliranje. Bitno je napomenuti da se u procesu kompajliranja vrši validacija *expression tree-a*. Proverava se sintaksa i tipovi korišćeni u *expression tree-u*. Pravila je potrebno kompajlirati samo jednom čime se dobija na efikasnosti.

Izršavanje pravila: Promene signala opisuje klasa *SignalChange* i one se mogu odnositi na diskretne ili analogne tipove signala. U zavisnosti o kojem tipu signala je reč proveravaju se različiti tipovi alarma. *AlarmDetector* detektuje promene signala, određuje tip signala i inicijalizuje svojstva određenih tipova alarma na osnovu pristiglih promena. Samo pravila čiji uslovi vrše neku proveru nad promenjenim svojstvima se evaluiraju. U procesu evaluacije ispituje se istinitost uslova pravila i u zavisnosti od rezultata izvršava se pravilo, tj. njegova akcija. Agenda sadrži sva pravila koja treba izvršiti i ukoliko ih ima više određuje redosled njihovog izvršavanja. Izvršeno pravilo se briše iz agende. U ovom rešenju mehanizam zaključivanja zasnovan je na metodi *ulančavanja unapred* što znači da se, nakon izvršavanja jednog pravila, analiziraju ostala pravila i ona čiji uslovi sadrže svojstvo afektovano predhodno izvršenim pravilom se dodaju u agendu. Proces prestaje kada je agenda prazna, tj. nema pravila koja čekaju na izvršenje.

6.1 Dodavanje novih tipova alarma u mašinu pravila

Svaki tip alarma opisan je svojstvima (*AlarmProperties*) koje predstavljaju činjenice mašine pravila. Dodavanje novog tipa alarma odvija se u dva koraka. Prvi korak je kreiranje činjenica, tj. dodavanje nove klase *AlarmProperties* koja sadrži svojstva koja opisuju novi tip alarma (Slika 3). Drugi korak je pisanje pravila u konfiguracionom fajlu na osnovu novo dodatih činjenica.



Slika 3: Dodavanje novog tipa alarma u mašinu pravila

7. ZAKLJUČAK

Osnovni cilj ovog rada bilo je pojednostavljenje održavanja alarmnog sistema u SCADA softveru uvođenjem rešenja zasnovanog na mašini pravila. Prednosti koje su dobijene implementacijom mašine pravila su pre svega razlaganje složenog algoritma alarmiranja na manje delove. Sa pojednostavljenjem logike olakšana je modifikacija alarma, u smislu dodavanja novih pravila alarma, ali i njihovog brisanja. Poboljšana je čitljivost koda i smanjen nivo kompleksnosti.

Dalji razvoj mašine pravila bio bi fokusiran na kreiranju korisničkog interfejsa (*en. User Interface*) preko kojeg bi se unosila pravila. Korisnički interfejs bi olakšao pisanje pravila i omogućio ne ekspertima sistema da na jednostavniji način unose pravila.

8. LITERATURA

- [1] Branislav Atlagić: "Softver sa kritičnim odzivom u elektroenergetskim sistemima, Univerzitet u Novom Sadu, Fakultet tehničkih nauka", 2014
- [2] Norwegian Petroleum Directorate: "Principles for alarm system design", 2001
http://www.ptil.no/getfile.php/135975/Regelverket/Alarm_system_design_e.pdf
- [3] IEC 62682: "Management of Alarm Systems for the Process Industries", 2010
- [4] Miloš Radosavljević: "'Mašina pravila" ekspertski sistem dokazivanja zasnovan na pravilima", master rad, Matematički fakultet Beograd, 2016
- [5] Martin Fowler: "Domain Specific Languages", Addison Wesley, 2011
- [6] "The Rule Engine" <https://docs.jboss.org/drools/release/5.3.0.Final/drools-expert-docs/html/ch01.html>

Kratka biografija:



Vladimir Papuga je rođen je 1992. godine u Vrbasu. Osnovne akademske studije na Fakultetu tehničkih nauka, smer "Računarstvo i automatika", završio je 2015. godine. Na istom fakultetu upisuje Master akademske studije 2015. godine.

PRIMENA SOFTVERA INSEL ZA REŠENJA FN SISTEMA U SRBIJI**APPLICATION OF INSEL SOFTWARE FOR PV SYSTEM SOLUTIONS IN SERBIA**Vanja Gojnić, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je obrađena solarna energija kao generalni pojam, kao i solarni potencijal za Republiku Srbiju. Pomoću INSEL softvera napravljen je virtuelni FN sistem za jedno domaćinstvo u Beogradu.

Ključne reči: Fotonaponski sistemi, obnovljivi izvori energije, solarna energija, INSEL softver

Abstract – This paper presents solar energy as a general term, as well as solar potential for the Republic of Serbia. With INSEL software, a virtual PV system was created for household in Belgrade.

Keywords: Photovoltaic systems, renewable energy sources, solar energy, INSEL software

1. UVOD

Ubrzani rast svetske populacije ima veliki uticaj na planetu Zemlju, a jedan od njih je definitivno povećana potražnja za električnom energijom, koja se smatra vitalnom za život u 21. veku. Zahtev za nove izvore energije takođe je podstakao ograničenje drugih resursa, poput masivnih fosilnih goriva. Govoreći o fosilnim gorivima, koji su danas glavni izvori proizvodnje energije, fosilna goriva igraju ogromnu ulogu za stvaranje efekta staklene bašte, što izaziva mnoge probleme ne samo na lokalnom nivou, već i na regionalnom i globalnom nivou. S obzirom na to da je postojanje budućih generacija dovedeno pod znak pitanja, uvode se nove tehnologije, kao što su solarne, hidro, vetroelektrane i tako dalje. Neobnovljiva energija dolazi iz izvora koji će nestati ili neće biti dopunjeni tokom života. Zbog svih prethodno navedenih činjenica i na predstojeći period energetskeg deficita, svetska zajednica je podstaknuta na razvoj održivih energetskeg sistema. Tokom poslednje decenije zemlje i vlade širom sveta su izgradile stroge okvire energetske politike za proširenje korišćenja i ulaganja u alternative energije i sektor OIE. U ovoj oblasti svakako Evropska unija (EU) ima vodeću poziciju u globalnom društvu. EU trenutno razvija energetske politiku čiji je cilj pretvaranje celog energetskeg sistema na sistem korišćenja tehnologije sa niskim emisijama ugljenika i OIE kroz Evropski strateški plan za energetske tehnologije.

Među ostalim opcijama OIE, fotonaponske (FN) elektrane postale su stvarnost. Fotonaponski sistemi su prihvaćeni širom sveta za proizvodnju čiste, beščujne i skoro

potpuno ekološki prihvatljive energije kao oblik direktne konverzije sunčeve svetlosti u električnu. U tom cilju razvijeno je niz softverskih alata, koji treba da omoguće jednostavno određivanje potencijala solarne energije na pojedinim lokacijama, kao i projektovanje kompletnih FN elektrana, odnosno FN sistema. Jedan od tih softvera je i INSEL, koji će biti primenjen u ovom radu.

Poznato je da je Srbija dobro geografski locirana, ali uprkos tome ne koristi dovoljno i ovu mogućnost proizvodnje električne energije. U ovom radu će se dalje razmatrati solarni potencijal Republike Srbije i mogućnosti primene INSEL softvera za projektovanje jedne krovne FN elektrane u Beogradu snage 10 kW. Takođe, proučiće se i ekonomski aspekti ovako instaliranog FN sistema.

2. SOLARNA ENERGIJA

Poznato je da je Sunce neiscrpan izvor sunčeve energije, što se naravno smatra besplatnom energijom. Sunce je otvoreni reaktor, koji pretvara oko 600 miliona tona vodonika u helijumu svake sekunde, gde oslobađa ogromnu količinu energije i ta energije se šalje u svemir.

U poređenju sa drugim vrstama obnovljivih izvora energije kao što su geotermalna energija i biomasa, koja imaju svoja ograničenja vezana za lokaciju ili snabdevanje, sunčeva energija se stavlja na stvarno visoku poziciju kada se govori o potencijalu. Kada se faktori kao što su geografska širina, klima i druge geografske varijacije uzmu u obzir, može se odrediti sunčevo zračenje koje prolazi kroz atmosferu Zemlje. Zemlja dobija više energije od Sunca tokom jednog sata nego što ljudska populacija koristi za celu godinu. Količina solarne energije koja dolazi na površinu planete je toliko ogromna da godišnje doprinosi dvostruko veću energiju nego svi neobnovljivi izvori energije zajedno.

3. FOTONAPONSKI SISTEMI

FN sistem predstavlja međusobno povezane nizove koji su sačinjeni od FN modula. Postoje različite vrste FN sistema, ali zajednički elementi za sve njih su FN moduli, regulator punjenja, inverter i baterijski sistem. Naravno, energija protiče kroz razne uređaje, kao što je žičana mreža. Pod dodatnom opremom može se uključiti ožičenje, kontroleri i tako dalje. U zavisnosti od načina rada, postoje tri vrste FN sistema: off-grid (samostalni) sistemi, on-grid (mrežni) sistemi, hibridni FN sistemi. On-grid sistem, koji se simulira i u ovom radu, je poznat kao i mrežni sistem, s obzirom na to da je solarna instalacija priključena na distribuiranu mrežu, od koje uzima potrebnu energiju ili višak proizvedene energije šalje nazad u mrežu. Ovaj princip je dobro poznat kao

NAPOMENA:

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

princip negativnog brojača: kada se višak proizvedene energije šalje nazad u mrežu, brojač oduzima iskorišćenu energiju i smanji račun. To je sjajna prilika za kupce koji žele da se instaliraju solarni sistem na svojim kućama i smanje period otplate.

4. INSEL – INTEGRATED SIMULATION ENVIRONMENT LANGUAGE

INSEL označava INtegrated Simulation Environment Language. INSEL nije simulacioni program, već obezbeđuje integrisano okruženje i grafički programski jezik za kreiranje simulacionih aplikacija. Osnovna ideja INSEL-a je povezivanje blokova za blok dijagrame koji predstavljaju rešenje za određeni simulacijski zadatak. Grafički programski jezici poput INSEL-a koriste potpuno različit pristup gde podaci igraju ključnu ulogu.

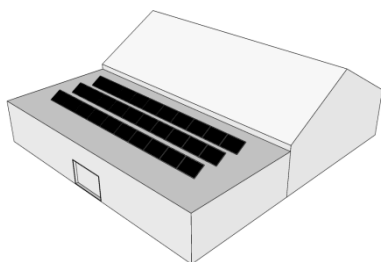
Umesto naredbi ovi jezici pružaju grafičke simbole koji se mogu međusobno povezati fizički putem miša kako bi izgradili veće modele. Grafički simboli mogu da predstavljaju matematičke funkcije, stvarne komponente kao što su solarno - termički kolektori, fotonaponski moduli, vretogeneratori i baterije ili čak na primer i kompletne tehničke sisteme bilo koje vrste. Grafički elementi INSEL-a su blokovi.

Pored najvećeg belog prostora na ekranu koji predstavlja radnu površinu, s leve strane se mogu videti 13. kategorija. Svaka kategorija ima padajući meni, koji se može otvoriti levim klikom miša na mali + simbol. Neke od kategorija su: Vreme, meteorologija, električna energija, zgrade, matematika i statistika itd.

5. STUDIJA SLUČAJA

5.1. Model zgrade u Beogradu

Model objekta koji je korišćen u ovom radu se nalazi u ulici Bežanijskih ilegalaca 32, Beograd na Bežanijskoj kosi. Totalna površina krova iznosi oko 439 m² i sastoji se iz jednog ravnog dela (koji će u ovom radu biti korišćen za postavljanje solarnih panela) i dela krova pod nagibom. Površina koja će biti korišćena za postavljanje solarnih panela iznosi 170 m². (Slika 1.)



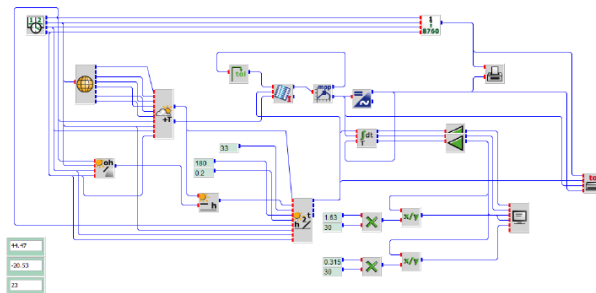
Slika 1: Vizuelni prikaz objekta sa postavljenim panelama

5.2. Model 1

Napravljena su dva modela u INSEL-u kako bi se uporedili rezultati. Prvi model koji će biti opisan jeste model za dobijanje globalnog, direktnog, difuznog zračenja i refleksije tla za svaki sat u godini. Ovim modelom će se postepeno doći do rezultata DC, AC električne energije, specifičnog prinosa, dok je drugi model kreiran tako da direktno daje ove rezultate.

5.3. Model 2

Drugi model, kao što je pre napomenuto, predstavlja kompletan PV sistem, pomoću čijih simulacija će dobijeni rezultati biti upoređeni sa rezultatima modela broj 1



Slika 2: Prikaz modela 2

5.4. FN konfiguracija

S obzirom na kravnu površinu s kojom se raspolože gde će biti postavljene solarne panele, broj panela koji je potrebno da pokrije instalisanu snagu sistema od 10.000W:

$$N_{mod} = \frac{10\,000\,W}{P_{max}} \quad (1)$$

N_{mod} – predstavlja broj panela

P_{max} – maksimalna snaga panela

Izračunato je 30 panela, koje će zbog prostora s kojim se raspolože biti instalirane deset panela paralelno po tri reda, što se može videti s Slike 1.

Od velikog značaja kod panela je parametar U_{oc} otvorenog kola. Kod panela Sunpower 315, koji je izabran u INSEL-u za simulaciju, napon otvorenog kola je 64.4 V. V_{mpp} je 54.7 V i predstavlja napon pri kojem će panel biti najefikasniji. Pomoću naponom otvorenog kola, možemo izračunati koliki je moguć najveći napon u jednom nizu sa odgovarajućim invertorom i to je predstavljeno sledećom jednačinom:

$$U_{maxinput} = N_{mod} \cdot U_{oc} \quad (2)$$

$U_{maxinput}$ - maksimalan napon za jedan niz panela

N_{mod} - broj panela (maksimalan koji je izračunat za 10kW)

U_{oc} - napon otvorenog kola

Izračunato je da je maksimalna DC snaga na invertoru 1938V, što je mnogo više od snage invertora koji je korišćen (850V).

$$N_{inverter} = \frac{U_{maxDC}}{U_{oc}} \quad (3)$$

Tako je izračunato da je maksimalan broj panela povezanih u red za jedan invertor 13, ali je broj panela raspoređen 10 panela redno u tri paralelna reda.

5.5. Poređenje dobijeni rezultata modela 1 i 2

Tabela 1: Rezultati modela 1

	Satno
Globalno zračenje [kWh/m ² god]	1.576,8
DC snaga [kWh/god]	14.744,4
AC snaga [kWh/god]	14.198,9
Specifični prinos [kWh/kWp]	1.419,8
Specifični prinos [Wh/m ²]	290,2

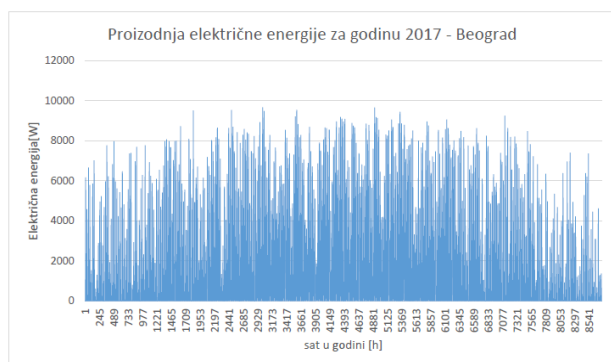
Tabela 2: Rezultati modela 2

	Mesečno
Globalno zračenje [kWh/m ² god]	1.525,6
DC snaga [kWh/god]	13.246,5
AC snaga [kWh/god]	12.705,7
Specifični prinos [kWh/kWp]	1.344,5
Specifični prinos [Wh/m ²]	1.344,5

Za razliku od prvog modela, gde se satne vrednosti konvertuju, drugi model konvertuje mesečne vrednosti u godišnju. Ove vrednosti se nalaze u bloku MTM weather data basa. Varijacije rezultata se kreću od 3 – 10 %. Koji je model tačniji može se diskutovati. Prvi model ima prednost u odnosu na drugi, iz razloga što su vremenski podaci satni, što znači da postoji veći broj uzoraka, tačnije 8760 sati u godini i time je napravljen tačniji prosek vrednosti. S druge strane u modelu 2 su korišćene mesečne vrednosti, što se smatra manje preciznim.

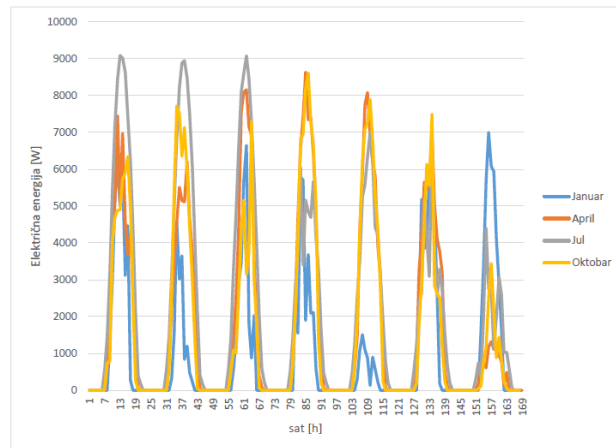
Druga stvar o kojoj se može diskutovati jesu vrste panela. U prvom modelu je uzet u obzir manji broj podataka, kao što je broj panela, njihove dimenzije, efikasnost panela prilikom standardnih uslova i efikasnost invertora. Što se tiče drugog modela, u njemu se bira tačno proizvođač i model proizvođača i model automatski uzima sve parametre u obzir.

Na slici 3 je prikazana proizvodnja električne energije na godišnjem nivou za grad Beograd. Na x-osi su predstavljeni sati u godini, kojih ima 8760, a na y-osi proizvedena električna energija. Kao što se može primetiti, na slici 3 postoje određeni usponi i padovi, ali nažalost određeni padovi izgledaju toliko mali, da ispada da panela ne proizvodi električnu energiju samo sat-dva dnevno, što je naravno daleko od tačnog.



Slika 3: Proizvodnja el. energije za 2017

Na Grafikonu 1 su predstavljene proizvodnje el. energije za vremenski period od 7 dana. Takođe zbog boljeg poređenja uzeti su vremenski periodi iz četiri godišnja doba (januar, april, jul, oktobar). Na x-osi su prikazani sati u nedelji, dok y-osa predstavlja proizvodnju električne energije. Na legendi sa desne strane Grafikona se mogu videti četiri boje koje reprezentuju ova četiri perioda.



Grafikon 1: Predstava proizvodnje el. Energije četiri perioda

6. TEHNO-EKONOMSKA ANALIZA

Prilikom izračunavanja ekonomskog aspekta FN sistema, potrebno je istražiti cene svih potrebnih elemenata. U sledećoj tabeli su predstavljeni troškovi sistema od 30 komada panela za 10W sistem.

Tabela 3: Troškovi sistema 10kW

Naziv	Cena bez PDV [€]	Cena sa PDV [€]
SunPower 315	5.400	6.480
Invertor Sunway NT 10000	1.800	2.160
Noseća konstrukcija	895	1.074
Drugi troškovi	700	840
Troškovi održavanja	172	206,4
Ukupni troškovi	8.967	10.760,4

6.1. Period otplate

Period otplate je vremenski period potreban za povraćaj troškova investicije, tj. Intuitivno meri koliko dugo treba nešto da „plati za sebe”.

Metoda I

Odluka o regulisanoj ceni električne energije za garantovano snabedanje na teritorije Republike Srbije se objavljuje u „Službenom glasniku R. Srbije”. Godišnja cena ovog sistema na godišnjem nivou je godišnja proizvodnja električne energije pomnožena sa cenom električne energije za datu godinu. Period otplate je količnik ukupnih troškova investicije i godišnje cene sistema.

6.2. „Feed in“ Tarifa

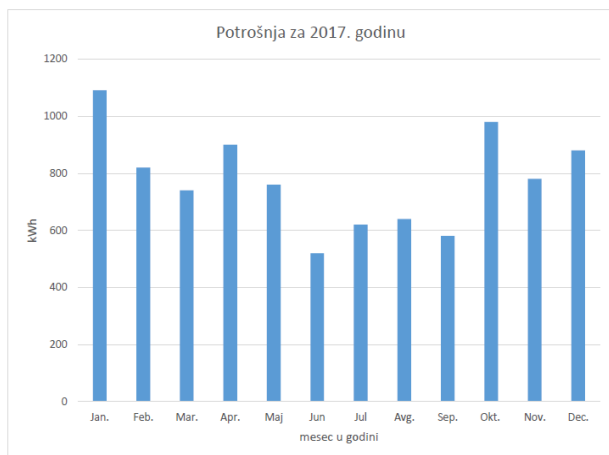
Feed-in tarifa je državni mehanizam podsticanja projektovan za ubrzanje ulaganja u tehnologije vezane za OIE.

Metoda II

S obzirom da sistem koji se izučava u ovom radu ima instaliranu snagu 10 kW, on spada u prvu kategoriju otkupne cene, što znači da vlasnik elektrane prodaje 1 kWh za 0,266€. Kada se ova cena pomnoži sa proizvedenom strujom godišnje koju smo dobili simulacijom sistema, dobija se koliko vlasnik elektrane zarađuje godišnje.

Metoda III

Na Grafikonu 2 je predstavljena potrošnja električne energije za objekat koji je pomenut ranije. Ukupna potrošnja električne energije za celu godinu se može videti u mesečnom dostavljenom računu za električnu energiju dostavljeno od strane EPS-a. Ako bi bilo uzeto u obzir da godišnje FN sistem proizvede 14.198,8 kWh ili tj. 12.705,7kWh, kad bi se oduzela godišnja potrošnja objekta, dobijaju se vrednosti od 4.888,8 kWh odnosno 3.395,7 kWh. Ovim je uzeto u obzir da proizvođač električne energije ne samo da troši koliko njemu treba, već da električna energije prodaje.



Grafikon 2: Godišnja potrošnja EE za 2017 god.

6.3. Poređenje isplativosti tri modela

Tabela 4: Poređenje vrednosti Metode I, II, III

Metoda I [god]		Metoda II [god]		Metoda III[god]	
13,06	14,61	5,19	5,08	15,07	21,7
13,83		5,13		18,38	

Kao što se može primetiti u Tabeli 4 svaka metoda ima po dve kolone. Te dve kolone ustvari predstavljaju rezultate dobijene simulacijom modela br. 1 modela br. 2. (broj godina otplate instalisanog sistema). Treća vrsta ustvari, predstavlja proseku godina instalisanog za model 1. i model 2. radi lakše observacije.

7. ZAKLJUČAK

Smanjenje energetske nezavisnosti Republike Srbije je jedan od glavnih ciljeva energetskog sektora. Prema podacima sa sajta EPS distribucije, do 2020. godine Republika Srbija ima plan da instalira ukupno 10 MW solarnih elektrana.

Takođe, jedan od ciljeva jeste povećanje registra povlašćenih proizvođača električne energije, a i samo edukovanje stanovništva o svesti globalnog zagrevanja i beneficija OIE.

Izračunato je da je godišnja proizvodnja električne energije u kWh za instalirani sistem od 10 kW dostiže i do 14.198,9 kWh, a da je specifična proizvodnja u Wh po metru kvadratnom 290,2 Wh/m² za model 1, a da model dva odstupa svega od 3 - 10%. Grad Beograd, kao i većina Srbije ima veoma dobar potencijal solarne energije, što pokazuje i činjenica da period otplate za instalaciju od 10kW može da bude od 5, 13 do 18 godina, u zavisnosti od načina otplate i da li proizvedenu energiju prodaje EPS ili ne.

8. LITERATURA

- [1] M. Burgess, D. Pimentel, „World Human Population Problems“, *Earth Systems and Environ. Scien.*, 2018
- [2] M. Akbari, A. Shahsavari, „Potential of solar energy in Developing countries for reducing energy related emissions“, *Renew. and Sust. Energy Rev.*, 90, 2018, pp. 275-291.
- [3] European Commision, „Strategic Energy Technology (SET) Plan“.
- [4] D. Djurdjevic, „Perspectives and assessments of solar PV power engineering in the Republic of Serbia“, *Renew. and Sust. Energy Rev.*, 15, 2011, pp. 2431-2446.
- [5] Y. Goswami, „Solar Energy“, *Inter. Energy Society*, 2017, <https://www.journals.elsevier.com/solar-energy>
- [6] E. Kabir, P. Kumar „Solar energy: Potential and future prospects“, *Renew. and Sust. Energy Rev.*, 82, 2018, pp.894–900
- [7] W.H. Terjung, „A global classification of solar radiation“, *Solar Energy*, 2003.
- [8] M. Milošević J. Brankov, „Geografija Srbije“, BIGZ Školstvo, 2016.
- [9] D. Milosavljevic. T. Pavlovic, „Possibility of electricity generation using PV solar plants in Serbia“ *Renew. and Sust. Energy Rev.*, 20, 2013, pp.201–218.
- [10] A. P. Kirk, „Solar Photovoltaic Cells“, Elsevier, USA, 2015.
- [11] K. Mertens, „Photovoltaic: fundamentals, technology and practice“, J. Wiley & Sons, 2014.
- [12] V. Katic, I. Kapetanovic, N. Sarajlić, „Obnovljivi izvori električne energije“, CEFES, Novi Sad, 2007.
- [13] P. Manimekalai, R. Harikumar „An Overview of Batteries for Photovoltaic (PV) Systems“, *Intern. Journal of Computer Applic.*, Vol. 82, No. 12, 2013.

Kratka biografija:



Vanja Gojnić rođena je u Novom Sadu 1993. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi obranila 2017. Iste godine upisala se na master studije.



Dr Vladimir A. Katić, red.prof. rođen je 1954. godine u Novom Sadu. Doktorirao je na Univerzitetu u Beogradu 1991. godine. Oblasti interesovanja su mu energetska elektronika, obnovljivi izvori električne energije, kvalitet električne energije i električna vozila.

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2018. godine učestvovali su sledeći recenzenti:

Aco Antić	Dušan Uzelac	Matija Stipić	Slobodan Dudić
Aleksandar Erdeljan	Duško Bekut	Milan Rapajić	Slobodan Krnjetin
Aleksandar Ristić	Đorđe Ćosić	Milan Simeunović	Slobodan Morača
Bato Kamberović	Đorđe Lađinović	Milan Trifković	Sonja Ristić
Biljana Njegovan	Đorđe Obradović	Milan Trivunić	Srđan Kolaković
Bogdan	Đorđe Vukelić	Milan Vidaković	Srđan Popov
Kuzmanović	Đula Fabian	Milena Krklješ	Srđan Vukmirović
Bojan Batinić	Đura Oros	Milica Kostreš	Staniša Dautović
Bojan Lalić	Đurđica Stojanović	Milica Miličić	Stevan Gostojić
Bojan Tepavčević	Filip Kulić	Mijodrag Milošević	Stevan Milisavljević
Bojana Beronja	Goran Sladić	Milovan Lazarević	Stevan Stankovski
Branislav Atlagić	Goran Švenda	Miodrag	Strahil Gušavac
Branislav Nerandžić	Gordana	Hadžistević	Svetlana Nikoličić
Branislav Veselinov	Milosavljević	Miodrag Zuković	Tanja Kočetov
Branislava Kostić	Gordana Ostojić	Mirjana	Tatjana Lončar -
Branislava	Igor Budak	Damnjanović	Turukalo
Novaković	Igor Dejanović	Mirjana Malešev	Toša Ninkov
Branka Nakomčić	Igor Karlović	Mirjana Radeka	Uroš Nedeljковиć
Branko Milosavljević	Ivan Beker	Mirko Borisov	Valentina Basarić
Branko Škorić	Ivana Katić	Miro Govedarica	Velimir Čongradec
Damir Đaković	Ivana Kovačić	Miroslav	Veran Vasić
Danijela Lalić	Ivana Miškeljin	Hajduković	Veselin Perović
Darko Čapko	Jasmina Dražić	Miroslav Popović	Vladimir Katić
Darko Marčetić	Jelena Atanacković	Mitar Jocanović	Vladimir Strezoski
Darko Reba	Jeličić	Mladen Kovačević	Vlado Delić
Dejan Ubavin	Jelena Borocki	Mladen Radišić	Vlastimir Radonjanin
Dejana Nedučin	Jelena Kiurski	Momčilo Kujačić	Vuk Bogdanović
Dragan Ivanović	Jelena Radonić	Nemanja	Zdravko Tešić
Dragan Ivetić	Jovan Petrović	Stanisavljević	Zoran Anišić
Dragan Jovanović	Jovanka Pantović	Nemanja Sremčev	Zoran Brujić
Dragan Kukolj	Ksenija Hiel	Nikola Đurić	Zoran Jeličić
Dragan Mrkšić	Laslo Nađ	Nikola	Zoran Mitrović
Dragan Pejić	Lazar Kovačević	Jorgovanović	Zoran Papić
Dragan Šešlija	Leposava Grubić	Nikola Radaković	Željen Trpovski
Dragana Bajić	Nešić	Ninoslav Zuber	Željko Jakšić
Dragana	Livija Cvetičanin	Ognjen Lužanin	
Konstantinović	Ljiljana Vukajlov	Pavel Kovač	
Dragana Šarac	Ljiljana Cvetković	Peđa Atanasković	
Dragana Štrbac	Ljubica Duđak	Petar Malešev	
Dragoljub Šević	Maja Turk Sekulić	Predrag Šiđanin	
Dubravka Bojanić	Marko Todorov	Radivoje Dinulović	
Dušan Dobromirov	Marko Vekić	Radovan Štulić	
Dušan Gvozdenac	Maša Bukurov	Slavica Mitrović	
Dušan Kovačević		Slavko Đurić	