



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXXV

Број: 6/2020

Нови Сад

Едиција: „Техничке науке – Зборници“

Година: XXXV

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

В.проф. др Дарко Стефановић

В.проф. др Себастиан Балош

В.проф. др Драган Ружић

В.проф. др Мирослав Кљајић

В.проф. др Бојан Лалић

В.проф. др Дејан Убавин

В.Проф. др Мирослав Ђукић

В.проф. др Борис Думнић

Проф. др Јелена Атанацковић Јеличић

Проф. др Властимир Радоњанин

Проф. др Драган Јовановић

Проф. др Мила Стојаковић

Проф. др Ливија Цветићанин

Проф. др Драгољуб Новаковић

Проф. др Теодор Атанацковић

Редакција:

Проф. др Владимир Катић, главни
уредник

В.проф. др Жељен Трповски, технички
уредник

В.проф. др Дарко Стефановић

Проф. др Драгољуб Новаковић

Доц. др Иван Пинђер

Бисерка Милетић

Језичка редакција:

Бисерка Милетић, лектор

Софија Рацков, коректор

Мр Марина Катић, преводилац

Савет за библиотечку и издавачку делатност ФТН,
проф. др Милан Мартинов, 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 25.10.2019. до 31.01.2020. год., а који се промовишу 18.05.2020. 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 и
- архитектуре.

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

- саобраћаја,
- графичког инжењерства и дизајна,
- инжењерског менаџмента,
- инжењерства заштите на раду и заштите животне средине,
- мехатронике,
- геодезије и геоматике,
- управљања ризиком од катастрофалних догађаја и пожара,
- инжењерства информационих система и
- биомедицинског инжењерства.

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. Aleksandar Saša Lukić,
PROJEKTOVANJE TEHNOLOGIJE POSTUPKA I UREĐAJA TERMOHEMIJSKE OBRAD
ZUPČANIKA | 1007-1009 |
| 2. Damir Kovčić,
OPTIMIZACIJA RASHLADNOG POSTROJENJA PRIMENOM REGULACIJE POTISNOG
PRITISKA RASHLADNOG KOMPRESORA | 1010-1013 |

Radovi iz oblasti: Elektrotehnika i računarstvo

- | | |
|--|-----------|
| 1. Nikša Bogovac,
PREDLOG REŠENJA POSLOVNO-INFORMACIONOG SISTEMA ZA PRODAJU KARATA ZA
DOGAĐAJE U DVORANAMA | 1014-1017 |
| 2. Борис Шаргач, Владимир Катић, Золтан Чорба,
СОЛАРНИ ПУЊАЧ ЗА ЕЛЕКТРИЧНИ БИЦИКЛ | 1018-1021 |
| 3. Немања Стевановић,
ПРИМЕНА SERVERLESS АРХИТЕКТУРЕ НА АПЛИКАЦИЈУ ЗА УПРАВЉАЊЕ
РАДОВИМА У ЕЕ СИСТЕМУ | 1022-1025 |
| 4. Stevan Kosijer,
PODRŠKA ZA UPRAVLJANJE SPRING CLOUD GATEWAY KOMPONENTOM U
MIKROSERVISNOM EKOSISTEMU | 1026-1029 |
| 5. Nemanja Čabrilo,
PACIJENT SIMULATOR – MAJČIN I FETALNI SIGNAL GENERATOR NEINVAZIVNIH
ABDOMINALNO SNIMLJENIH EKG I EHG BIOPOTENCIJALA | 1030-1033 |
| 6. Jovan Janjušević,
SISTEM ZA AUTOMATSKO OTVARANJE I ZATVARANJE AUTOMOBILSKIH VRATA SA
DETEKCIJOM PREPREKE I MERENJEM UDALJENOSTI DO PREPREKE | 1034-1037 |
| 7. Danilo Bujiša,
KOMPARATIVNA ANALIZA METODA RAZVOJA MOBILNOG SOFTVERA SA STANOVIŠTA
UPOTREBLJIVOSTI | 1038-1041 |

	STRANA
8. Milica Samardžić, Karolina Kasaš-Lažetić, UTICAJ NESIMETRIJE TROFAZNOG SISTEMA NA MAGNETSKO POLJE “U” SABIRNICA	1042-1045
9. Ljiljana Popović, Željien Trpovski, KABLOVSKI DISTRIBUTIVNI SISTEMI	1046-1049
10. Nemanja Juzbaša, Željien Trpovski, SIMULACIJA PRENOSA PODATAKA U 5G SISTEMU	1050-1053
11. Dragan Raca, REALIZACIJA ELEKTRONSKOG SISTEMA ZA OČITAVANJE JEDNODIMENZIONALNOG BAR-KODA	1054-1057
12. Jadranka Ivković, Karolina Kasaš-Lažetić, MAGNETSKO POLJE SIMETRIČNOG TROFAZNOG VODA, SASTAVLJENOG OD “U” SABIRNICA	1058-1061
13. Jovana Antić, KLASIFIKACIJA SLIKA PRIMENOM FASTAI BIBLIOTEKE	1062-1065
14. Danilo Dimitrijević, ANDROID APLIKACIJA ZA NADGLEĐANJE SISTEMA BAZIRANIH NA AMAZON AWS PLATFORMI	1066-1069
15. Данило Аћимовић, ANDROID АПЛИКАЦИЈА ЗА НАДГЛЕДАЊЕ СИСТЕМА БАЗИРАНИХ НА AZURE ПЛАТФОРМИ	1070-1073
16. Nataša Rašeta, Vladimir Katić, OPTIMIZACIJA RADA VETROTURBINE UPOTREBOM MAŠINSKOG UČENJA	1074-1077
17. Никола Плавшић, ИНВЕРТОРИ СА СИЛИЦИЈУМ – КАРБИДНИМ ТРАНЗИСТОРИМА	1078-1081
18. Дарко Тачић, NATIVESCRIPT ОКВИР ЗА РАЗВОЈ МОБИЛНИХ АПЛИКАЦИЈА	1082-1085
19. Бранко Јелић, РАЗВОЈ SDK БИБЛИОТЕКЕ ЗА ИМПЛЕМЕНТАЦИЈУ ИНДУСТРИЈСКИХ ПРОТОКОЛА	1086-1089
20. Rade Ilić, KONVERZIJA PODATAKA ELEKTROENERGETSKE MREŽE DEFINISANE U CIM-u U ESRI SHAPEFILE FORMAT	1090-1093
21. Марко Гајић, МОДЕЛИ ПОДАТАКА ЕЛЕКТРОЕНЕРГЕТСКОГ СИСТЕМА ДЕФИНИСАНИ У ГРАФ ОРИЈЕНТИСАНИМ БАЗАМА ПОДАТАКА	1094-1097
22. Вукашин Иличић, ПРЕПОЗНАВАЊЕ ГРАФИЧКИХ СИМБОЛА НА ШЕМАТСКИМ ДИЈАГРАМИМА	1098-1101
23. Ivan Stanojević, Savo Đukić, UTICAJ NAČINA UZEMLJENJA DISTRIBUTIVNE MREŽE NA PRORAČUN LOKACIJE KVARA ...	1102-1105
24. Aleksa Ristić, ZAŠTITA VODOVA U DISTRIBUTIVNIM I PRENOSNIM MREŽAMA	1106-1109
25. Luka Maletin, GENERISANJE SLIKA SA VIŠE OZNAČENIH MNIST CIFARA UPOTREBOM GENERATIVNIH NEURONSKIH MREŽA	1110-1113
26. Doroteja Anđelić, OTKRIVANJE SIGURNOSNIH PROPUSTA U WEB APLIKACIJAMA METODOM PENETRACIONOG TESTIRANJA	1114-1117
27. Ђорђе Радовановић, Борис Думнић, КОНЦЕПТ НАПАЈАЊА СЛОЖЕНИХ ТЕХНОЛОШКИХ ПРОЦЕСА НА ПРИМЕРУ ПОГОНА ЗА ОДСУМПОВАЊЕ ДИМНИХ ГАСОВА	1118-1121
28. Ivana Ćirić, Dejan Jerkan, MODELOVANJE TROFAZNE KAVEZNE ASINHRONE MAŠINE UZ UVAŽAVANJE KONSTRUKCIONIH OSOBENOSTI	1122-1125

	STRANA
29. Jovan Topolić, OPTIMIZACIJA RADA WPF APLIKACIJA UPOTREBOM VIRTUALIZACIJE	1126-1129
30. Марина Бобар, ДИНАМИЧКА ПРОМЈЕНА ПРОФИЛНЕ КОНФИГУРАЦИЈЕ У СИСТЕМУ ЗА УПРАВЉАЊЕ ПАМЕТНИМ БРОЈИЛИМА	1130-1133
31. Ivona Jovanović, IDENTIFIKACIJA ELEMENTARNIH NEPOGODA U SISTEMU ZA UPRAVLJANJE PAMETNIM BROJILIMA	1134-1137
32. Marko Bubnjević, MASTER/SLAVE KOMUNIKACIONI PROTOKOLI ZA SCADA SISTEME	1138-1141
33. Aleksandar Nikolić, SISTEM ZA PRIKUPLJANJE I ANALIZU FOTOGRAFIJA SA ULIČNIH KAMERA UPOTREBOM MAŠINSKOG UČENJA	1142-1145
34. Aleksandar Maričić, INTEROPERABILNI ADAPTER ZA RAD SA LDAP SISTEMIMA	1146-1149
35. Jelena Ećimović, ARHITEKTURA MJERNO-INFORMACIONOG SISTEMA U TELEMEDICINI	1150-1153
36. David Ivanović, STUDIJA PODEŠENJA ELEKTRIČNIH ZAŠTITA TRANSFORMATORSKE STANICE 33/220kV VETROELEKTRANE KOVAČICA	1154-1157
37. Miloš Ostojić, PREDIKCIJA ZARADE FILMOVA TOKOM VIKENDA	1158-1161
38. Goran Maksimović, POREĐENJE NIRS UREĐAJA PRIMENJENIH U SPORTU	1162-1165
39. Mladen Vuković, Dejan Jerkan, TEHNIČKI ASPEKTI INSTALACIJA INDUSTRIJSKIH POGONA	1166-1169
40. Filip Stojanović, IDENTIFIKACIJA I UPRAVLJANJE CNG SISTEMOM OPISANOG IRACIONALNOM FUNKCIJOM PRENOSA	1170-1173

PROJEKTOVANJE TEHNOLOGIJE POSTUPKA I UREĐAJA TERMOHEMIJSKE OBRADJE ZUPČANIK**DESIGN OF PROCESS TECHNOLOGY AND GEAR THERMOCHEMICAL PROCESSING DEVICES**

Aleksandar Saša Lukić, Branko Škorić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Tema rada jeste projektovanje tehnološkog postupka i uređaja termohemijske obrade zupčanika. Odabran je tip termohemijske obrade, konstruisan uređaj i pogon za termičku obradu. Cilj rada jeste da se analiziraju postupci termohemijske obrade, da se odabere proces i oprema za pogon termičke obrade.

Ključne reči: Termohemijska obrada, uređaji za termičku obradu, cementacija, zupčanik, pogon

Abstract – The topic of the paper is the design of technological process and devices of thermochemical gearing. The type of thermochemical treatment, the controlled device and the heat treatment plant were selected. The aim of this paper is to analyze the processes of thermochemical treatment, to choose the process and equipment for the operation of thermal treatments.

Keywords: Thermochemical treatment, heat treatment devices, cementation, gear, treatment plant

1. UVOD

Projektovanje uređaja i pogona za termohemijsku obradu počinje detaljnom analizom postupaka i proizvodnog pogona. Određuju se parametri procesa, koji se sastoje od: vremena zagrevanja, zadržavanja na određenim temperaturama, hlađenja i određivanja radnih temperatura.

Izabrana termohemijska obrada jeste cementacija u gasovitom sredstvu. Konstruisana je peć za termičku obradu koja je komornog tipa i u kojoj se nalazi zaštitna atmosfera. Za navedenu peć urađena je detaljna analiza gubitaka energije i snage.

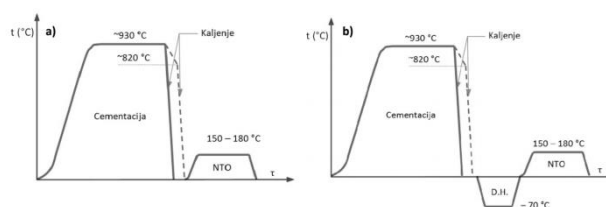
2. TERMOHEMIJSKA OBRADA ČELIKA

Termohemijskom obradom čelika naziva se termička obrada koja se izvodi kao kombinacija termičkog i hemijskog dejstva sa ciljem da se izmeni sastav, struktura i svojstva površinskog sloja. Ovom obradom nastaju strukturne promene, ali i promene hemijskog sastava površinskog sloja putem apsorpcije i difuzije elemenata. Termohemijske obrade se izvode zagrevanjem delova do određenih temperatura u čvrstoj, tečnoj ili gasovitoj sredini, pri čemu dolazi do obogaćivanja površinskog

sloja elementima kao što su C, N, Al, Si, Cr, B i drugi, putem difuzije njihovih atoma iz spoljne sredine [1].

2.1. Cementacija

Cementacija je postupak termohemijske obrade kojim se naugljeničava površina slojeva čeličnih radnih predmeta. Cilj cementacije je dobijanje žilavog jezgra otpornog na udare i tvrde površine otporne na habanje. Visoka tvrdoća se postiže kada se nakon cementacije izvrši i kaljenje, ne postiže se samo cemenacijom. Tvrdoća najviše zavisi od sadržaja ugljenika, pa se potrebna količina ugljenika u površinskom sloju određuje na osnovu propisane tvrdoće. Tvrdoća površinskog sloja nakon cementacije iznosi od 55 do 65 HRC. Posle procesa cementacije je potrebno izvršiti dodatnu termičku obradu, a vrsta te termičke obrade zavisi od željenih osobina delova ili njihove primene [1].



Slika 1. Dijagram obrade cementacijom [2]

Nosilac ugljenika u procesu cementacije može biti tečan, čvrst i gasovit. Prema nosiocu ugljenika razlikuju se tri vrste cementacije:

1. Cementacija u tečnom sredstvu
2. Cementacija u čvrstom sredstvu
3. Cementacija u gasovitom sredstvu

Atmofere koje se koriste za naugljeničenje mogu se podijeliti na: zemni gas, endogeni generatorski gas (endogas), egzogeni generatorski gas (egzogas), gas aktiviran nekim tečnim sredstvom [1].

2.2. Nitriranje

Nitriranje je proces termohemijske obrade kojim se u površinski sloj čelika ili odlivaka od sivog liva uvodi azot. Obrada se sastoji iz zagrevanja delova na temperaturu od 500 do 600°C u sredstvu koje otpušta azot. Nitrirani delovi koji su izrađeni od odgovarajućeg čelika, odlikuju se veoma visokom tvrdoćom (značajno

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Škorić, red.prof.

veća nego kod najveća kod martenzita) i velikom otpornošću na habanje. Tvrdća nitriranog sloja je postojana i na višim temperaturama (500 do 550°C). Pre nitriranja je potrebno definitivno odraditi delove jer kasnija mehanička obrada je veoma teška.

U nekim slučajevima nije moguće ni brušenje nitriranih površina. Nakon nitriranja dolazi do porasta debljine komada za 0,01 mm do 0,07 mm [1][3].

2.3. Karbonitriranje

Karbonitriranje je hemijsko-termički proces kojim se u površinski sloj istovremeno uvode ugljenik i azot. Sam naziv karbonitriranje ne govori o kom procesu je reč, pa mu se mora dodati i još jedan podatak o tome da li je [1]:

- pretežno obavljena cementacija ili nitriranje
- komad kaljen posle hemijsko-termičke obrade ili nije
- postupak obavljen na čeliku za cementaciju, nitriranje ili nekom drugom

2.4 Sulfidiranje i sulfocijanje

Sulfidiranje i sulfocijanje su procesi obrade kojima se u površinski sloj čelika unosi sumpor. Kod sulfidiranja se obavlja uvođenje samo sumpora u površinski sloj čelika, a kod sulfocijanja se sloj obogaćuje sumporom, ali i azotom i ugljenikom.

Sulfidiranje povećava klizne osobine površinskog sloja, naročito u slučaju slabog podmazivanja, ali ne menja mehaničke osobine površinskog sloja. Dubina sloja je 0,1 mm, ali može biti i 0,2 do 0,3 mm. Sulfocijanje sloj ima kombinaciju povoljnih osobina, kao što su dobra otpornost na habanje, uvećanu kontaktnu čvrstoću, uvećanu dinamičku čvrstoću, dobru prilagodljivost površina u trenju [1].

2.5. Boriranje

Boriranje je termohemijski proces obogaćivanja površine delova borom. Proces se vrši hemijskom depozicijom isparenja pri visokim temperaturama na kojima atomi difunduju u osnovu metala.

Uvođenjem bora u površinu čelika nastaju boridi gvožđa visoke tvrdoće. Borirani delovi se odlikuju veoma visokom tvrdoćom, otpornošću na habanje, abrazivnom i korozionom postojanošću, kao i vatrostalnošću [4].

2.6. Hromiranje

Hromiranje je postupak termohemijske obrade kojim se u površinu delova unosi hrom. Postupak se još naziva inhromiranje ili difuziono hromiranje. Ovaj postupak se primenjuje u parnoj tehnici za rad na povišenim temperaturama u dodiru sa vodom, parom ili produktima sagorevanja.

Razlog primene u ovim uslovima je što hromirani delovi poseduju osobine hemijske postojanosti u određenim sredinama i pod određenim uslovima, povećana tvrdoća koja daje i otpornost na habanje [1].

2.7. Alitriranje

Alitriranje je termohemijski proces difuzionog obogaćivanja površinskog sloja aluminijumom. Ovaj proces se

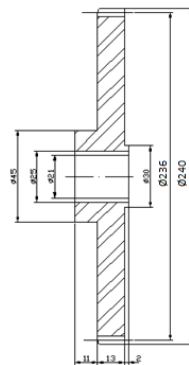
najviše koristi kod niskougljeničnih čelika i livenog gvožđa (najčešće sivi liv). Površina koja je podvrgnuta procesu alitriranja se odlikuje hemijskom postojanošću i vatrostalnošću.

Ove osobine su posledica pasiviziranja površine komada čvrstim i hemijski postojanim oksidom aluminijuma (Al_2O_3). Takođe, alitirani sloj ima dobru otpornost na koroziju u morskoj vodi [3].

2.8. Siliciranje

Siliciranje je termohemijski proces difuzionog obogaćivanja površinskog sloja čelika silicijumom zagrevanjem u odgovarajućoj sredini. Primenjuje se za poboljšavanje hemijske postojanosti delova od konstrukcionih čelika. Osnovna karakteristika siliciranog sloja je hemijska postojanost u morskoj vodi (tj. slanoj vodi) i blažim rastvorima sone, fosforne, sirćetne i sumporne kiseline [1].

3. PRORAČUN ZUPČANIK ZA CEMENTACIJU



Slika 2. Zupčanik za cementaciju

Grube dimenzije komada su Ø240x25 mm. Usvojen je materijal Č4320 koji je pogodan za cementaciju i ujedno odgovara za izradu zupčanika.

3.1. Određivanje vremena predgrevanja (500°C)

Vreme zagrevanja

$$\tau_z = F_o^p \cdot \frac{x^2}{a} = 65 \cdot \frac{0,0125^2}{8,38 \cdot 10^{-6}} = 4196 \text{ s} = 69 \text{ min}$$

Gde je:

$$F_o^p = \frac{a \cdot \tau_z}{x^2} = 225 - \text{Furijev kriterijum}$$

Vreme zadržavanja na temperaturi predgrevanja je 30 min.

3.2. Određivanje vremena do temperature cementacije (930°C)

Vreme zagrevanja

$$\tau_z = F_o^p \cdot \frac{x^2}{a} = 80 \cdot \frac{0,0125^2}{5,73 \cdot 10^{-6}} = 2181,5 \text{ s} = 37 \text{ min}$$

Gde je:

$$F_o^p = \frac{a \cdot \tau_z}{x^2} = 80 - \text{Furijev kriterijum}$$

3.3. Vreme zadržavanja na temperaturi cementacije

Prema preporukama, koje se zasnivaju na iskustvima iz prakse, vreme zagrevanja na temperaturi cementacije, koja iznosi 930°C, treba da bude oko 130 minuta. Zagrevanjem i zadržavanjem u ovim uslovima, dobijaju se dubine cementiranog sloja od 0,4 do 0,6 milimetara.

3.4. Kaljenje na temperaturi od 860°C

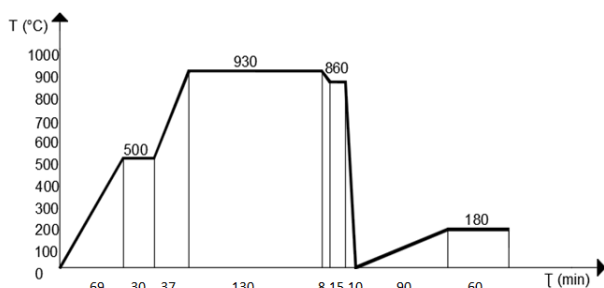
Nakon obrade od 130 minuta na temperaturi od 930°C, vrši se hlađenje zupčanika u peći. Zupčanici se hlade do temperature od 860°C, kada sledi postupak kaljenja.

Predviđeno vreme za hlađenje sa 930°C na 860°C je oko 8 minuta. Vreme zadržavanja na temperaturi od 860°C je 15 minuta, nakon čega sledi hlađenje zupčanika u ulju. Za hlađenje u ulju se predviđa vreme od 10 minuta.

3.5. Određivanje vremena zagrevanja do temperature otpuštanja

Otpuštanjem treba da se dobije tvrdoća zupčanika od 60 HRC. Da bi se tražena tvrdoća dobila, radni predmet je potrebno otpuštati na temperaturu od 180°C.

Na osnovu iskustva ćemo usvojiti vreme zadržavanja na temperaturi otpuštanja od 60 minuta, a vreme zagrevanja do te temperature 90 minuta.



Slika 3. Dijagram toka termičke obrade

4. IZBOR OPREME ZA POGON TERMIČKE OBRADU

Tabela 1. Pregled potrebne opreme

Redni broj	Naziv opreme	Proizvođač	Broj komada
1	Komorna peć		3
2	Uređaj za pranje i odmašćivanje WP-21-E	„IPSEN“	1
3	Endogenerator G-1500-G	„IPSEN“	2
4	Kran nosivosti 600kg	„VETTER“	1
5	Ručni hidraulični viljuškar	„Total Lifter“	3
6	Ručna presa za ispravljanje	„Pobeda“ Novi Sad	1

5. ANALIZA IZABRANOG IDEJNOG REŠENJA

Tabela 2. Ukupan potreban prostor za pogon termičke obrade

Redni broj	Prostor	Površina (m ²)
1	Za smeštaj opreme	50,332
2	Za laboratorije	39
3	Za skladišta	70
4	Za poslovni prostor	40

Ukupan prostor za pogon termičke obrade je potrebno uvećati za veličinu prostora potrebnog za transport i manevrisanje materijala u pogonu. Usvojena je hala dimenzija 32 x 17.

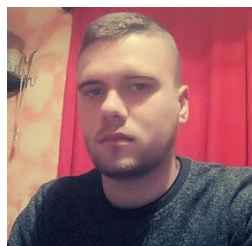
6. ZAKLJUČAK

Poznavajući termohemijske obrade i uticaje atmosfera, pristupili smo analiziranju obrade zupčanika. Zupčanici su jedni od najznačajnijih mašinskih elemenata. U eksploataciji, zupčanici su specifično opterećeni, zato se javlja potreba za preciznom termohemijskom obradom. Da bi se dobile osobine koje su najpogodnije za zupčanike, odabran je proces cementacije. Cementacijom se naugljeničava površinski sloj zupčanika, čime se dobijaju površine otporne na habanje, a jezgro ostaje žilavo i otporno na udare.

7. LITERATURA

- [1] Ilija Pantelić, „Tehnologija termičke obrade čelika 2“, Novi Sad, 1974.
- [2] Doc.Dr. Aleksandar Miletić: Dijagrami i tablice
- [3] Dipl.Inž.Maš.Domagoj Augusta: „Dupleks alumiziranja i nitriranja čelika“, Fakultet strojarstva i brodogradnje, Zagreb, 2011
- [4] Željko Stojanović, Milan Topličević: „Razvoj i primena postupaka boriranja u funkciji produženja radnog veka mašinskih delova“, Tehnički fakultet „Mihailo Pupin“, Zrenjanin, 2011

Kratka biografija:



Aleksandar Saša Lukić rođen je u Somboru 1994. god. Diplomirao na Fakultetu tehničkih nauka u Novom Sadu iz oblasti tehnologije termičke obrade 2018. god. Master studije upisao iste godine na usmerenju Savremene tehnologije oblikovanja materijala.

OPTIMIZACIJA RASHLADNOG POSTROJENJA PRIMENOM REGULACIJE POTISNOG PRITISKA RASHLADNOG KOMPRESORA

OPTIMIZATION OF COOLING PLANT BY APPLYING HEAD PRESSURE CONTROL ON COOLING COMPRESSOR

Damir Kovčić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Rashladna postrojenja su veliki potrošači električne energije i njihova optimizacija je ključna u pogledu smanjenja potrošnje električne energije. Ovaj rad se bazira na primeru optimizacije postrojenja u okviru pivare. Na osnovu izračunatih rashladnih potreba i stvarne potrošnje energije u jednoj kalendarskoj godini izračunata je potencijalna ušteda ukoliko se postrojenje vodi pomoću vlažnog termometra umesto dosadašnjim ručno podešenim zadatim potisnim pritiskom. Ovim načinom upravljanja radom postrojenja moguće je ostvariti uštede od približno 7 %.

Ključne reči: rashladna postrojenja, regulacija potisnog pritiska; potrošnja energije

Abstract – Cooling plants are huge consumers of electrical energy and their optimization is critical in terms of energy usage. This paper is focusing on one example of cooling plant optimization. Based on calculated cooling requirements and actual energy consumption for one calendar year, optimization of cooling plant by using wet bulb temperature as set point for head pressure was researched and compared. Calculation displays that it is possible to achieve electricity consumption reduction by applying this method and that savings are around 7 %.

Key words: Cooling plants, head pressure regulation, energy usage

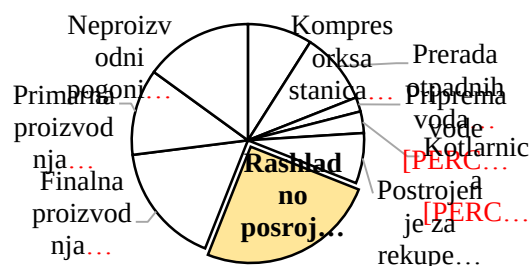
1. UVOD

Rashladna odeljenja su glavni potrošači električne energije u pivari. Njihova optimizacija u pogledu smanjenja vršne potrošnje ili promene načina rada postrojenja (kao npr. snižavanje potisnog pritiska kompresora promenom temperature kondenzacije) može da ima veliki uticaj na ukupnu potrošnju električne energije pivare.

U poslednjih nekoliko godina mnoge pivare su optimizovale svoja postrojenja u smislu ugranje frekventnih regulatora, podizanjem temperatura isparavanja rashladnih fluida itd. Sva ova unapređenja kao rezultat daju smanjenje potrošnje električne energija ali za ove vrste optimizacije neophodan je zajednički rad tehnologa i ostalih inženjera kako ne bi došlo do narušavanja kvaliteta proizvoda.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Miroslav Kljajić.



Slika 1. Raspodela potrošnje energije

2. PROCES PROIZVODNJE PIVA

Proizvodnja piva je šaržni proces, u toku jednog proizvodnog dana u jednom sistemu za kuvanje moguće je proizvesti šest do osam šarži u zavisnosti od recepture. Osnovna sirovina za proizvodnju piva je ječmeni slad, koji se dobija sladovanjem ječma u sladarama. Slad se doprema u pivare gde se priprema za mlevenje i ukomljavanje, osim slada u zavisnosti od podnevlja moguće je dodati određene surogate u kominu kao što su nesladovani ječam, kukuruz, riža, itd.

Tokom ukomljavanja se šećeri u ranije pomenutim sirovinama razlažu na prostije šećere, ovaj proces je neophodan kako bi fermentacija bila dobra. Sledeći korak u proizvodnji piva je bistrenje komine i nakon toga bistra sladovina je spremna za kuvanje. Tokom procesa kuvanja u sladovinu se dodaje hmelj koji pivu obezbeđuje gorčinu. Nakon kuvanja na temperaturi od 100°C sladovinu je u kratkom vremenskom intervalu potrebno ohladiti na temperaturu od +20°C kako bi je doveli na temperaturu u kojoj se obavlja dodavanje kvasca.

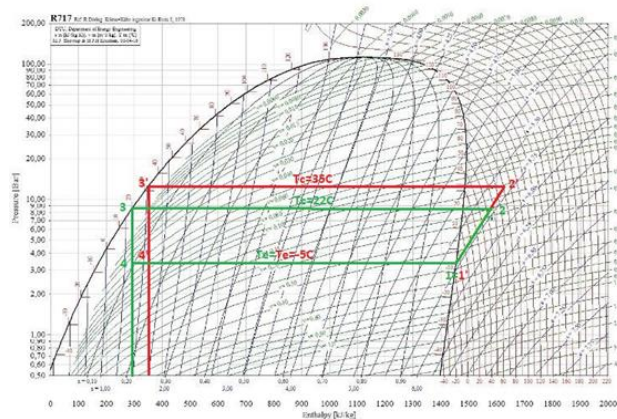
Nakon hlađenja sladovina se cevovodima transportuje prema fermentorima. Na putu do fermentora sladovina se aeriše i dodaje joj se kvasac. Fermentori su sudovi u kojima se obavlja proces fermentacije, tokom fermentacije kvasac konzumira šećer i pretvara ga u alkohol i CO₂, a kao nusproizvod nastaje toplota. Toplota podiže temperaturu i pritisak u fermentoru što u nekim slučajevima može ubrzati proces fermentacije. S obzirom da je proces fermentacije potrebno zadržati u okviru recepture koja podrazumeva određene temperature i pritiske neophodno je zadate parametre održavati pomoću rasteretnih ventila i sistema za hlađenje. Ugljen dioksid koji nastaje kao produkt fermentacije se cevovodima transportuje do postrojenja za rekuperaciju ugljen dioksida gde se prečišćava, stečnjava i u tečnom obliku skladišti na temperaturi -23°C. Za potrebe proizvodnje CO₂ se isparava i u gasnom stanju se transportuje do potrošača.

Tokom procesa fermentacije, koji traje četiri do osam dana, ćelije kvasca se talože u konusu fermentora i ispuštaju kako ne bi uticale na ukus piva. Nakon završetka primarne fermentacije mlado pivo se hladi na temperaturu koja je u rasponu od -1.5°C do 0°C kako bi se ubrzalo taloženje preostalog kvasca i bistrenje. Ovaj proces se naziva odležavanje piva. Pre punjenja u ambalažu u procesu proizvodnje piva potrebno je filtrirati pivo do receptom definisane mutnoće nakon čega se pivo pasterizuje na temperaturi od približno $+74^{\circ}\text{C}$ nakon čega se ponovo hladi i pakuje u ambalažu.

Iz kratkog opisa procesa proizvodnje piva jasno je da hlađenje igra veoma važnu ulogu u kvalitetu i ukusu gotovog proizvoda. U potrošnji električne energije rashladna odeljenja utiču sa 20-25% ukupne potrošnje energije cele pivare. S obzirom na uticaj u ukupnoj potrošnji neophodno je baviti se optimizacijom potrošnje energije ovih postrojenja kako bi se snizili troškovi proizvodnje i uticaj pivara na životnu sredinu.

3. REGULACIJA POTISNOG PRITISKA

U polju optimizacije rada kompresora, regulacija potisnog pritiska je veoma interesantna oblast za razmatranje. Ideja je zasnovana na smanjenju potrošnje električne energije rashladnog kompresora koja je ostvarena na uštrb veće potrošnje električne energije na evaporativnim kondenzatorima. Zbog razlike u nominalnim snagama elektro motora kompresora i kondenzatora moguće je ostvariti uštede u potrošnji električne energije.

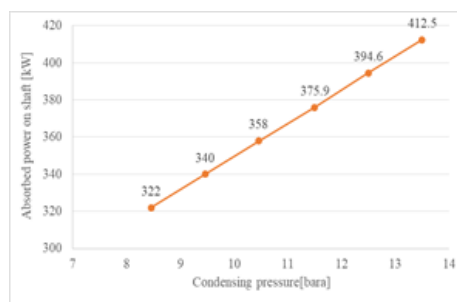


Slika 2. Dijagram pritisak-entalpija

Na slici 2 izvršeno je poređenje dva amonijačna ciklusa koji započinju na istom usisnom pritisku ($T_e = T_e' = -5^{\circ}\text{C}$, $p = 3.5$ bara), ali imaju različite pritiske kondenzacije ($T_c = 22^{\circ}\text{C}$, $T_c' = 35^{\circ}\text{C}$, $p = 13.52$ bara). Dve vrednosti pritiska kondenzacije predstavljaju najbolji i najgori scenario za posmatrani rashladni kompresor (minimalni i maksimalni dozvoljen pritisak kondenzacije).

U slučaju nižeg pritiska kondenzacije, razmenjena količina toplote u isparivaču iznosi $Q_e = 1152.7$ kJ/kg dok u drugom slučaju razmenjena količina toplote iznosi $Q_e' = 1090.3$ kJ/kg. Takođe, važno je napomenuti da kompresor apsorbuje veću količinu energije kada radi na većim pritiscima. Iz podataka dobijenih od strane proizvođača opreme na nižoj vrednosti pritiska kompresor absorbuje 340kW snage, dok na pritisku od 13.52 bara absorbovana snaga iznosi 412.5kW. Drugi parametar koji se ne sme zanemariti jeste pad volumetrijskom protoka

kroz kompresor sa porastom pritiska. Za ove dve granične vrednosti pad protoka iznosi 1.4%. Na slici 3 grafički su prikazane vrednosti apsorbovane snage kompresora.



Slika 3 promena apsorbovane snage u zavisnosti od potisnog pritiska kompresora

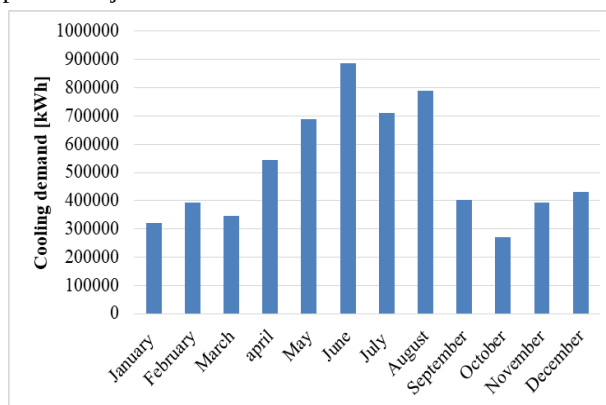
4. PRORAČUN RASHLADNIH POTREBA PIVARE

Kako bi se odredile mesečne potrebe za rashladnom energijom proračunate su dnevne potrebe za rashladnom energijom u zavisnosti od nivoa proizvodnje. Tabela 1 daje pregled rashladnih potreba u zavisnosti od obima proizvodnje.

Tabela 1. Dnevne potrebe za rashladnom energijom u zavisnosti od proizvodnje

Broj radni u toku dana	#	7	6	5	4
Zapremina uzorka	hl	500			
Zapremina skuvanog piva	hl	3500	3000	2500	2000
Hladnjak za vodu	kWh	4171	3575	2979	2383
Fermentacija i odležavanje	kWh	14589	12505	10421	8337
Postrojenje za rekuperaciju CO ₂	kWh	5760	4937	4114	3291
Stanica za zagrijavanje kvasca	kWh	3600	3086	2572	2058
Klimatizacija objekta	kWh	10800	10800	10800	10800
Potrošnja u letnjem režimu	kWh	38920	34903	30886	26869
Potrošnja u zimskom režimu	kWh	28120	24103	20086	16069

Broj radnih dana i nivo proizvodnje uzeti su iz istorijskih podataka kako bi se odredila dnevna i mesečna potreba za rashladnom energijom. Slika 4 predstavlja rashladne potrebe u jednoj proizvodnoj godini u zavisnosti od nivoa proizvodnje.

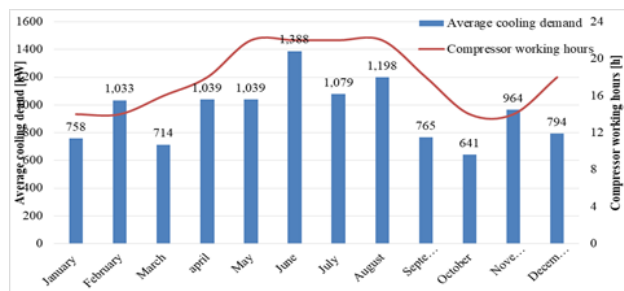


Slika 4. Mesečne rashladne potrebe

Sa prethodno prikazane slike jasno je da je u pitanju pivara koja ima visoku sezonalnost i da rashladne potrebe pivare diktira nivo proizvodnje kao i godišnje doba (leto) koje zahteva rad klimatizacije objekta.

Prosečna potreba za rashladnom energijom izračunata je kao odnos potrebne rashladne snage i realnog broja radnih sati rashladnih kompresora za određeni mesec. Na slici 5

je grafički prikazan broj radnih sati kompresora i prosečni potrebni rashladni kapacitet postrojenja.



Slika 5. Prosečan potrebni rashladni kapacitet i radni sati kompresora

Radni režim kompresora određen je na osnovu rashladnih potreba, usisnog i potisnog pritiska. Uisni pritisak je izračunat iz jednačine za prenos toplote. S obzirom da su koeficijent prelaza toplote i površina izmenjivača poznate veličine moguće je izračunati srednju logaritamsku temperaturu kao razliku prethodno dve pomenute veličine i potrebne rashladne energije.

$$Q = k \cdot A \cdot \Delta t_{log}$$

S obzirom da je temperatura amonijaka jednaka unutar isparivača moguće je izračunati temperaturu isparavanja iz sledeće jednačine za srednju logaritamsku temperaturu:

$$\Delta t_{log} = \frac{(t_{i(NH3)} - t_{iz(PG)}) - (t_{i(NH3)} + t_{ul(PG)})}{\ln \frac{(t_{i(NH3)} - t_{iz(PG)})}{(t_{i(NH3)} - t_{ul(PG)})}}$$

Vrednosti usisnog pritiska koje se mogu pojaviti u predmetnom rashladnom sistemu date su u tabeli 2.

Tabela 2. Temperature isparavanja amonijaka

Δt_{log}	$t_{out(NH3)}$	$t_{in(PG)}$	$t_{out(PG)}$
-3.41	-5.50	1	-4
-2.79	-5.00	1	-4
-2.09	-4.50	1	-4
-1.53	-4.20	1	-4
-1.41	-4.15	1	-4
-1.27	-4.10	1	-4
-1.08	-4.05	1	-4

4.1. Proračun sa fiksnim potisnim pritiskom

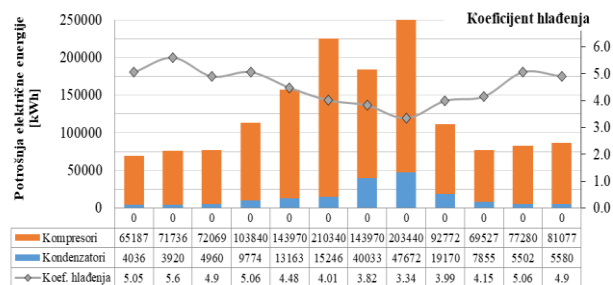
U situaciji kada rukovalac potrojenjem podešava parametre kondenzacije, postoje predefinisane vrednosti za određene mesece. Pregled parametara potrojenja je dat u tabeli 3.

Tabela 3. Parametri postrojenja sa fiksnim potisnim pritiskom

		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Parametar potisnog pritiska	bar[g]	9	9	9	10	12	12	12	12	11	11	10	9
Temperatura kondenzacije	C	25	25	25	28.1	33.7	33.7	33.7	33.7	31	31	28.1	25
Srednja logaritamska temperatura		-1.43	-1.95	-1.35	-1.96	-1.96	-2.62	-2.04	-2.26	-1.45	-1.21	-1.82	-1.50
Temperatura isparavanja	C	-4.2	-4.5	-4.2	-4.5	-4.5	-5	-4.5	-5	-4.2	-4.1	-4.5	-4.2

Pomoću softvera koji je obezbedila kompanija MYCOM dobijeni su podaci o apsorbovanoj snazi i rashladnom kapacitetu svakog kompresora u zavisnosti od usisnog i potisnog pritiska. Vodeći kompresor je odabran tako da

obezbeđuje maksimalnu efikasnost u zavisnosti od rashladnih potreba pivare. Na slici 6 prikazana je potrošnja električne energije rashladnog postrojenja po mesecima u zavisnosti od obima proizvodnje predstavljenim na slici 4.



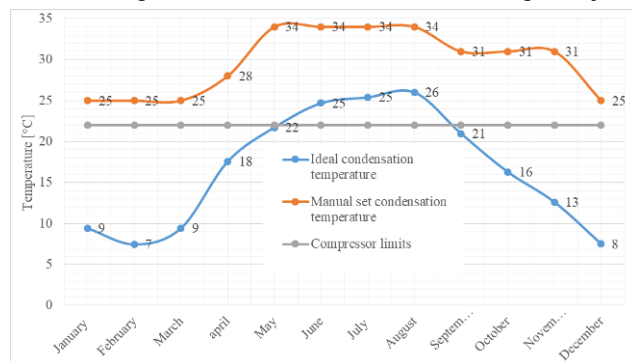
Slika 6. Potrošnja sa fiksnim parametrom potisnog pritiska

4.2. Proračun sa promenljivim potisnim pritiskom

Regulacija potisnog pritiska se vrši u zavisnosti od atmosferskih uslova. Idealna temperatura kondenzacije se računa pomoću softvera, a u idealnim uslovima temperatura kondenzacije se nalazi 8°C iznad izmerene temperature vlažnog termometra. Za postrojenje koje je predmet rada postoje određena ograničenja u pogledu minimalnog potisnog pritiska kako bi se ostvarilo efikasno razdvajanje ulja za podmazivanje kompresora i rashladnog fluida. U slučaju kompresora MYCOM VLD250-N i VLD200-N, koji su instalirani na postrojenju, minimalni dozvoljeni potisni pritisak je 8.1bar[g] što odgovara temperaturi kondenzacije od 22°C.

Osnovna ideja je da se smanji potrošnja električne energije na elektromotoru kompresora, na račun povećanja potrošnje električne energije kondenzatora. Ukupni bilans je smanjenje ukupne potrošnje rashladnog postrojenja što povećava COP postrojenja.

Slika 7 predstavlja razlikz između predefinisanih parametara kondenzacije i proračunatih parametara na osnovu prosečne vrednosti tačke rose po mesecima. Temperatura od 22°C predstavlja graničnu vrednost i sve temperature kondenzacije koje se nalaze ispod nisu u opticaju kako bi se zaštitila oprema od eventualne štete tokom eksploatacije.



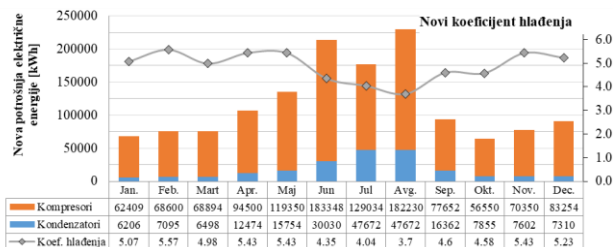
Slika 7. Razlika između kalkuliranih i predefinisanih vrednosti

Vodeći kompresor je izabran po istim kriterijumima kao i u slučaju fiksnog parametra kondenzacija, razlika u apsorbovanoj snazi je rezultat manjeg rada kompresora usled nižeg potisnog pritiska.

Kapacitet evaporativnih kondenzatora je podesen tako da zadovoljava dvostruke kapacitete rashladnih kompresora u zimskom periodu, dok je u letnjem periodu kapacitet

utrostručen kako bi se osigurali da se parametri zadrže u zadatim granicama. Regulacija kondenzatora je kaskadna i to takva da se prvo svi kondenzatori pokreću redno na nižim brzinama ventilatora u kombinaciji sa vodenom pumpom, a nakon toga u slučaju potrebe pale se druge brzine na ventilatorima.

Kada se Slika 6, uporedi sa Slikom 8 se vidi da je potrošnja električne energije na kondenzatorima porasla, dok se potrošnja energije na kompresorima smanjila. Takođe može se uočiti da je COP potrojenja porastao kako na mesečnom tako i na godišnjem nivou.



Slika 8. Potrošnja energije sa promenljivim potisnim pritiskom

5. ZAKLJUČAK

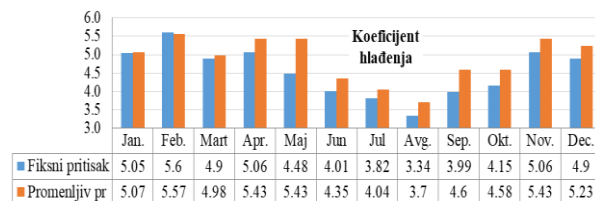
Tema ovog rada je rashladno postrojenje na kom je predstavljena kalkulacija koja prikazuje potencijal za uštedu energije. Cilj kalkulacije je bio poređenje fiksnog parametra potisnog pritiska sa parametrom potisnog pritiska koji je vođen vlažnim termometrom.

Rashladni kapaciteti su proračunati na osnovu istorijskih podataka za jednu proizvodnu godinu, za potrebe proračuna korišćeni su i podaci o rashladnim kompresorima i evarporativnim kondenzatorima. Na osnovu ulaznih podataka izračunat je COP za opciju vođenja postrojenja pomoću fiksnog parametra potisnog pritiska.

Za isto potrojenje kreiran je model u kome se parameter potisnog pritiska menjao u zavisnosti od temperature vlažnog termometra. Za potrebe proračuna korišćene su prosečne vrednosti atmosferskih uslova po mesecima.

Proračunom je prikazano da je moguće ostvariti uštede u potrošnji električne energije primenom regulacije potisnog pritiska na osnovu atmosferskih uslova i da uštede iznose približno 7 % što u za pivaru koja je uzeta za primer iznosi uštedu od 100.000 kWh na godišnjem nivou. Da bi se ovakva regulacija ostvarila neophodne su određene investicije kako bi se povećao rashladni kapacitet kondenzatora. U slučaju pivare koja je razmatrana dodatni rashladni kapaciteti postoje.

Najveći potencijal za uštede jeste u periodu proleće-jesen gde su primećene najveće oscilacije u dnevnim temperaturama, kao i razlike dnevnih temperatura u odnosu na noćne. Slika 9 prikazuje COP rashladnog postrojenja u zavisnosti od zadatih parametara na postrojenju.



Slika 9. COP rashladnog postrojenja

6. LITERATURA

- [1] I. Vanjur, V. Sovilj, Mogućnosti smanjenja potrošnje električne energije amonijačnih rashladnih instalacija u velikim pivarama, Međunarodni kongres i izložba o KGH
- [2] K. A. Manske, D. T. Reindl, S. A. Klein, "Evaporative Condenser Control in Industrial Refrigeration Systems", International Journal of Refrigeration, Vol. 24, No. 7, pp 676-691, 2001
- [3] R. J. Love, D. J. Cleland, I. Merts, B Eaton, "What is the optimum compressor discharge pressure set point for condensers", IRHACE technical conference, 2005
- [4] T. W. Davies, "A universal floating head system", international Congress of Refrigeration, ICR0079, Washington D.C, 2003
- [5] D. Demma, "Understanding the fundamentals of head pressure control", 2004
- [6] MYCOM SCREW COMPRESSOR design software

Kratka biografija:



Damir Kovčić rođen je u Novom Sadu 1990. god, gde je završio osnovnu i srednju školu. Diplomirao je 2013.god na Fakultetu tehničkih nauka, studijski program Mašinstvo, smer Energetika i procesna tehnika. Zaposlen je u kompaniji "Heineken".

PREDLOG REŠENJA POSLOVNO-INFORMACIONOG SISTEMA ZA PRODAJU KARATA ZA DOGAĐAJE U DVORANAMA

A PROPOSAL FOR BUSINESS INFORMATION SYSTEM FOR TICKET SALES FOR HALL EVENTS

Nikša Bogovac, *Fakultet tehničkih nauka, Novi Sad*

Oblast– ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U okviru ovog rada izvršen je razvoj poslovno-informacionog sistema za prodaju karata za događaje u dvoranama čija je namena da olakša i ubrza proces poslovanja kompanije koja se bavi pružanjem usluga prodaje karata. Prikazan je postupak implementacije jednog takovog sistema sa integracijom plaćanja putem kreditnih kartica koristeći Microsoft .NET tehnologiju, MVC, AngularJS i Web API.

Ključne reči: *Automatizacija, Elektronsko poslovanje, prodaja ulaznica, Microsoft .NET, AngularJS, MVC.*

Abstract – This paper gives and overview of the developing business and information system for selling tickets for hall events. Purpose of such solution is to simplify and speed up process for companies that are selling tickets for different events in halls. Implementation of system including payment gateway was done using Microsoft .NET framework, MVC, AngularJS and WebAPI.

Keywords: Automation, e-business, tickets event sales, AngularJS, MVC

1. UVOD

Tehnologija je promenila način prodaje ulaznica i kontrolu ulaska posetioca u koncertne dvorane, sportske i druge objekte. Učinila ih je efikasnijim i omogućila bolju kontrolu i praćenje broja posetilaca.

Pojavila se potreba za programskom podrškom koja će na savremen. elektronski način vršiti prodaju karata za neki događaj, a da pri tome korisnik u realnom vremenu ima sve potrebne podatke o broju karata, njihovoj poziciji u dvorani i ostale korisne informacije, kako grafički, tako i numerički prikazane.

Elektronska programska podrška treba da olakša rezervaciju i prodaju karata, izbor mesta, njihovo grupisanje, statističke izveštaje o ukupnom broju prodatih karata, itd. Zahvaljujući grafičkom prikazu svih mesta u koncertnoj dvorani potrebno je npr. prodana mesta označiti crvenom, rezervisana žutom, a slobodna mesta sivom bojom.

Ovakav sistem ima široku primenu u ustanovama kulture kao i u sportskim objektima (hale, stadioni, bazeni).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Milosavljević, red. prof.

Glavni cilj ovog rada jeste da se razvije veb platforma koja se koristi za jednostavno upravljanje koncertnim aktivnostima i administracijom koncertnih događaja, kao i rezervacijom i prodajom karata. Platformu treba prilagoditi za rad svim korisnicima, uključujući prodavce, administratore i klijente. Platforma treba svim korisnicima da omogući brz, lak i jednostavan rad.

Takođe, potrebno je modernizovati prethodno rešenje i proširiti ga sa novim opcijama, kao što je prijava na sistem koristeći Google i Facebook Sign in, integrisati platformu za plaćanje platnim karticama, uvesti elektronske karte i razne načine dostave karata kupcima. Opis postojećeg rešenja, kao i ograničenja opisana su u tački 2.

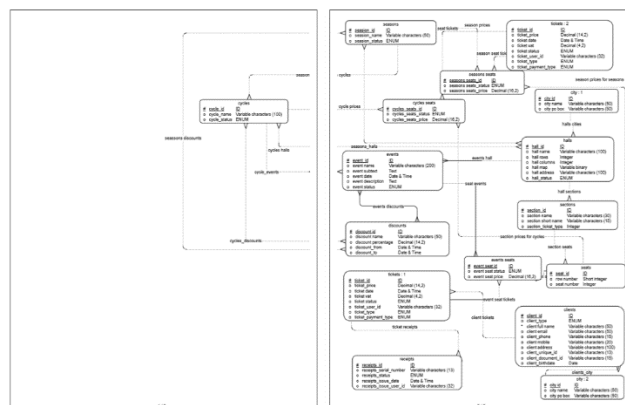
2. PRETHODNO REŠENJE

Sistem za online prodaju karata postojao je kao deo sistema za rezervaciju, prodaju i kontrolu ulaznica za koncerte i druge društvene događaje.

Preduslov za korišćenje online aplikacije bio je da korisnik ima instaliran program Microsoft Silverlight [1] što je predstavljalo problem određenim korisnicima koji nisu mogli da instaliraju Microsoft Silverlight aplikaciju na svoj računar.

Takođe, nije bilo moguće korišćenje veb aplikacije za korisnike mobilnih uređaja koji su danas znatno više zastupljeni, nego korisnici koji koriste računare.

Prethodni sistem je imao dve baze podataka. Prva baza je služila za autentifikaciju korisnika, rola, prava pristupa i deo je ASP.NET aplikacionih servisa. Njen naziv je aspnetdb.mdf. Druga baza je predstavljala šemu baze podataka same aplikacije TicketService čija je šema prikazana u nastavku rada (slika 1.).



Slika 1 – Postojeća šema baze podataka

2.1. Postojeća rešenja na tržištu

Trenutno na tržištu Srbije postoji nekoliko sličnih sistema za prodaju karata:

- Ticket Vision doo. - <https://www.tickets.rs/>
- Gigstix - <http://www.gigstix.com/>
- DDTickets - <https://www.ddtickets.rs/>

Na primeru DDTickets može se videti aplikacija koja nije intuitivna i laka za korišćenje. Prilikom odabira događaja, sekcije i samog sedišta potrebno je proći kroz nekoliko koraka kako bi se kupila ulaznica što značajno usporava proces. Za razliku od DDTickets, rešenje firme Ticket Vision doo je značajno unapređeno po pitanju iskustva korisnika i omogućava jednostavnu i brzu kupovinu, kao i velik izbor pretraga. Nedostatak je da omoguću prodaju sezonskih i ciklusnih karata gde jedna karta važi za grupu događaja. Takođe, prilikom kupovine moguće je i kreirati elektronsku kartu. Kompletno kretanje kroz celu veb aplikaciju je moguće bez prijave i kad se karte dodaju u korpu i pokuša uraditi plaćanje, aplikacija traži prijavu na sistem. Ovo je vrlo pozitivno korisničko iskustvo i intuitivno za korišćenje.

3. OPŠTE INFORMACIJE

Aplikacija se sastoji iz četiri dela:

- **TicketServiceWebAPI** predstavlja jezgro aplikacije i obezbeđuje komunikaciju sa bazom podataka.
- **Web Frontend** omogućava korisnicima da koriste aplikaciju koristeći internet pretraživač (WEB Browsers)
- **TicketServiceClientApp** omogućava korisnicima da pristupe aplikaciji instaliranoj na računaru sa Microsoft Windows operativnim sistemom.
- **TicketServiceAdministrator** omogućava administratorima da pristupe aplikaciji instaliranoj na računaru sa Microsoft Windows operativnim sistemom.

3.1. Definicije

TicketServiceWebAPI i TicketServiceServer su servisi koji služi za razmenu podataka između baze podataka i klijentske aplikacije, veb i administratorske aplikacije. Pruža otvorenu pristupnu tačku za:

- registraciju novih korisnika i pristup postojećim
- kreiranje događaja, sezone, ciklusa i sala za događaje
- rezervaciju i kupovinu karata za događaje, cikluse i sezone
- uvid u statistiku i broj prodatih karata za svaki pojedinačni događaj, seriju i sezonu, itd

Web Frontend omogućava:

- onlajn registraciju i prijavu na sistem
- uvid u korisnički profil i unošenje informacija
- pregled trenutno aktivne sezone, ciklusa i koncerata, kao i kupovinu i rezervaciju karata
- kupovinu karata koristeći mogućnost onlajn plaćanja i slanje karte putem elektronske pošte

TicketServiceClientApp se obično instalira na računaru čiji je korisnik osoba koja radi kao blagajnik koji organizuje prodaju karata.

Aplikacija pruža određene mogućnosti poput gore pomenute veb aplikacije u smislu prodaje i rezervacije karata za koncerte, cikluse i sezone, uz manjak funkcionalnosti da se upravlja korisničkim profilom. Dodatne karakteristike su:

- kreiranje i štampa statističkih izveštaja o broju prodatih i rezervisanih karata koji uključuju i detaljnije informacije na dnevnom, mesečnom i godišnjem nivou.
- Pretragu prodatih karata i uvid u informacije o korisnicima, salama, cenama, događajima, ciklusima i sezonama prodatih karata - moguća pretraga prema korisničkom imenu, lozinci, bar-kodu, broju narudžbenice itd.
- Mogućnost skeniranja karata uz korišćenje čitača bar-kodova
- Lako dodavanje novih klijenata
- Prodaja karata izabranim klijentima

Većina ovih mogućnosti sa klijentove strane su dostupne samo onim korisnicima sa naprednim privilegijama (administratorima i prodavcima), koji su jedini korisnici kojima je dozvoljeno da koriste ovu desktop aplikaciju za klijente.

TicketServiceAdministrator predstavlja deo aplikacije koji koriste samo administratori sistema kako bi kreirali nove događaje/koncerte, cikluse, sezone i koncertne sale gde će se događaji odvijati, uključujući sve potrebne detalje kao što su broj sedišta, cenu karata i opise događaja.

3.2. Tehnologije

Tehnologije koje su korišćene za razvoj ovog sistema se uglavnom baziraju na Microsoft .NET [2] razvojnom okviru:

- Backend aplikacija - .NET Web API[5], REST, JSON, Entity Frameworks and MSSQL Server 2017
- Klijent i Administrator desktop aplikacija - .NET WPF
- Frontend veb aplikacija – AngularJS [4].

Alati koji su korišćeni u razvoju:

- Visual Studio Community Edition 2017
- Microsoft SQL Server Management Studio 2017

4. NEFUNKCIONALNI ZAHTEVI

Glavni cilj je bio da se ceo poslovno informacioni sistem unapredi tehnološki, prelaskom na novije tehnologije kao što su AngularJS i ASP.NET WebAPI.

Sistem treba da bude postavljen na server koji će omogućiti brzu reakciju na korisničke akcije, a treba da sadrži sledeće softverske pakete:

- Microsoft Windows Server 2012 i novije verzije,
- MSSQL Server 2014 i noviji,
- .NET Framework 4.5 i više verzije kao i
- RIA Services,
- Svi vezani paketi potrebni da gore naveden softver radi

5. FUNKCIONALNI ZAHTEVI

Izradom veb aplikacije bazirane na .NET WebAPI na „bckendu“ i AngularJS na „frontendu“ zamenjen je stari sistem, koji je radio na Microsoft Silverlight, a koji više nije podržan od većine internet pretraživača. Time je omogućena online kupovina karata.

5.1. Role i ovlašćenja

U okviru Ticket Service aplikacije, postoje tri vrste korisnika koji imaju permisije: administrator, klijent (kupac) i prodavac (blagajnik). Takođe, postoje i neki delovi sistema kojima može pristupiti samo jedan tip korisnika.

5.1.1 Administrator

TABELA 1 – KORISNIK ADMINISTRATOR

Zahtev	Opis
Prijava	Sistem treba da obezbedi autentifikaciju korisnika
Ažuriranje događaja	Sistem treba da omogući administratoru da ažurira sve događaje u sistemu
Ažuriranje ciklusa	Sistem treba da omogući administratoru da ažurira sve cikluse u sistemu
Ažuriranje sezona	Sistem treba da omogući administratoru da ažurira listu sezona
Ažuriranje sala	Sistem treba da omogući administratoru da ažurira sve sale u sistemu
Aktivacija stanja događaja, ciklusa i sezone	Sistem treba da omogući administratoru da ažurira status svakog događaja, ciklusa i sezone kako bi isti bili vidljivi ili ne kupcima
Kreiranje korisnika	Sistem treba da omogući administratoru da kreira nove korisnike i role
Odjava	Sistem treba da omogući korisniku odjavu

5.1.2 Prodavac

TABELA 2 – KORISNIK PRODAVAC

Zahtev	Opis
Prijava	Sistem treba da obezbedi autentifikaciju korisnika
Unos kupaca	Sistem treba da omogući prodavcu ručni unos novih klijenata kao i ažuriranje postojećih
Provera karata bar-kod čitačem	Sistem treba da omogući prodavcu da koristeći bar-kod čitač uradi proveru validnosti izdatih ulaznica za događaj
Prikaz pojedinačnog događaja	Sistem treba da omogući prodavcu prikaz svakog pojedinačnog događaja kao i slobodnih i zauzetih mesta u sali
Prikaz pojedinačnog ciklusa	Sistem treba da omogući prodavcu prikaz svakog pojedinačnog ciklusa kao i slobodnih i zauzetih mesta u sali
Prikaz pojedinačne sezone	Sistem treba da omogući prodavcu prikaz svake pojedinačne sezone kao i slobodnih i zauzetih mesta u sali
Rezervaciju ulaznica za sve događaje, cikluse i sezone	Sistem treba da omogući prodavcu da doda u korpu, odabere tip ulaznice, način preuzimanja kao i rezervišće istu za svakog kupca

Kupovina ulaznica za sve događaje, cikluse i sezone	Sistem treba da omogući kupcu da doda u korpu, odabere tip ulaznice, način preuzimanja kao i plaćanja za karte za svakog kupca
Pretraga karata	Sistem treba da omogući prodavcu pretragu i ažuriranje karata po tipu karte, korisničkom imenu i događaju/ciklusu/sezoni kao i način pretrage (po bar-kodu, događaju/redu/sedištu, korisničkom imenu, broju porudžbine i imenu klijenta)
Statistika	Sistem treba da omogući prodavcu osnovnu i detaljnu dnevnu statistiku za sve prodane i rezervisane karte za svaku sezonu ponaosob
Odjava	Sistem treba da omogući korisniku odjavu

5.1.3 Klijent

Potrebno je implementirati Google Sign-in [6] i Facebook Sign-in [7].

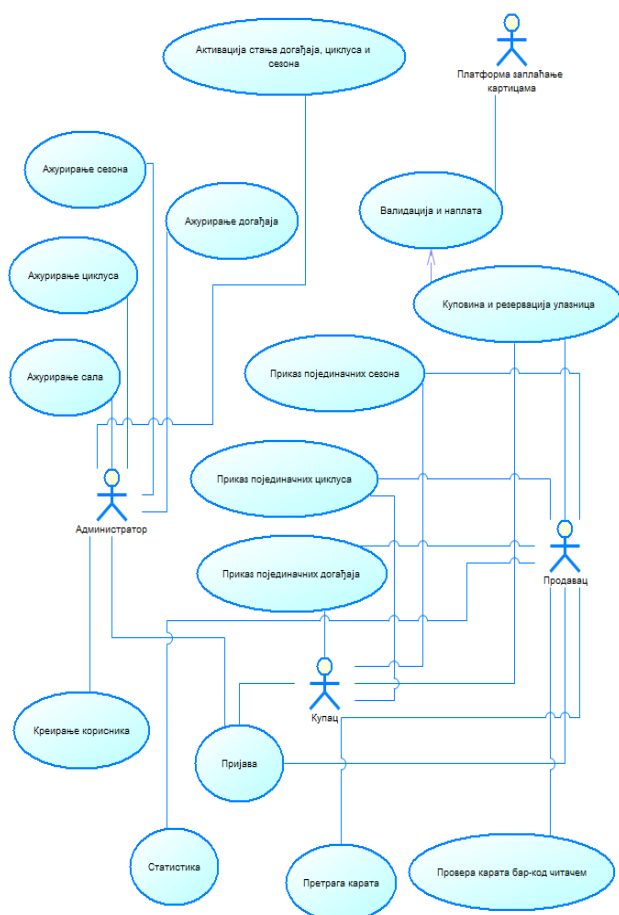
TABELA 3 - KORISNIK KLIJENT

Zahtev	Opis
Prijava	Sistem treba da obezbedi autentifikaciju korisnika
Kreiranje naloga	Sistem treba da omogući kreiranje naloga kao i korišćenje
Reset zaboravljene lozinke	Sistem treba da omogući klijentu da resetuje lozinku
Prikaz pojedinačnog događaja	Sistem treba da omogući korisniku prikaz svakog pojedinačnog događaja kao i slobodnih i zauzetih mesta u sali
Prikaz pojedinačnog ciklusa	Sistem treba da omogući korisniku prikaz svakog pojedinačnog ciklusa kao i slobodnih i zauzetih mesta u sali
Prikaz pojedinačne sezone	Sistem treba da omogući korisniku prikaz svake pojedinačne sezone kao i slobodnih i zauzetih mesta u sali
Rezervaciju ulaznica za sve događaje, cikluse i sezone	Sistem treba da omogući kupcu da doda u korpu, odabere tip ulaznice, način preuzimanja kao i rezervišće istu a potom dobije obaveštenje o rezervaciji putem elektronske pošte
Kupovina ulaznica za sve događaje, cikluse i sezone	Sistem treba da omogući kupcu da doda u korpu, odabere tip ulaznice, način preuzimanja kao i plaćanja za karte a potom dobije obaveštenje o rezervaciji na email
Izmene ličnih podataka	Sistem treba da omogući korisniku da izmeni sve lične podatke
Odjava	Sistem treba da omogući korisniku odjavu

6. SPECIFIKACIJA DIZAJNA

Struktura rešenja je predstavljena kao MVC [3] arhitekturni obrazac koji predstavlja troslojnu arhitekturu. Svi konfigurabilni parametri aplikacije nalaze se u Web.config datoteci u krenom direktorijumu rešenja.

6.1 Dijagram slučaja korišćenja



Slika 2- Dijagram slučaja korišćenja

6.2 Autentifikacija

Prijava na sistem je moguća na dva načina:

- Pomoću korisničkog imena i lozinke koji se čuvaju u bazi podataka
- Eksterni provajder kao što su Google i Facebook

Za oba načina se koriste OAuth [8] otvoreni standard za autentifikaciju i autorizaciju baziran na tokenima. OAuth omogućuje da informacije krajnjeg korisnika budu korišćene od drugih servisa, bez potrebe da deli lozinku. Trenutno sistem koristi verziju OAuth v2.

7. ZAKLJUČAK

Izradom veb aplikacije bazirane na .NET WebAPI na „bekendu“ i AngularJS na „frontendu“ zamenjen je stari sistem, koji je radio na Microsoft Silverlight, a koji više nije podržan od većine internet pretraživača. Time je omogućena online kupovina karata. Takođe, uvođenjem elektronskih karata olakšana je kupovina karata i nema potrebe za dolazak kupaca na blagajnu i ličnog preuzimanja karte. Povezivanjem sa platformom za plaćanje platnim karticama u Srbiji omogućen je još jedan dodatni način plaćanja, pored standardnog plaćanja putem opšte uplatnice, kao i onlajn bankarstva.

Sa druge strane procesom prijave na sistem, kao i evidencije o kupovini karata za određene koncerte i sedišta, prodavci imaju uvid u detaljnu statistiku kupovine i ostvarenog prometa za određeni događaj.

Prodavcima je omogućeno da te informacije koriste u marketinške svrhe, kao i u svrhu poboljšanja prodaje. Na ovaj način prodavac može bolje da organizuje prodaju, s obzirom da ima uvid u stanje prodatih karata u realnom vremenu.

Korišćenjem REST API i JSON formata za razmenu podataka omogućeno je i drugim servisima da se povežu sa ovim sistemom za prodaju karata i time razmenjuju neke od podataka. Npr. na taj način je omogućeno na veb sajtu određene ustanove koja se bavi organizacijom događaja prikazati sve koncertne događaje i broj slobodnih sedišta, kao i ponuditi drugim veb platformama prodaju karata preko ovog sistema. Takođe, ovim unapređenjem sistema se stvorila mogućnost za povezivanje ovog sistema za prodaju karata sa mobilnim aplikacijama na pametnim telefonima, a sve u svrhu povećanja prodaje i boljoj dostupnosti za kupce.

Iako uvek ima prostora za napredak, a najviše kad je u pitanju korisnički interfejs ovog sistema za prodaju karata, aplikacija se pokazala kao upotrebljiva za rad, omogućila je veću dostupnost kupcima, mogućnost prodaje preko eksternih prodavaca-agencija, a samim tim i značajno unapredila poslovanje prodavaca. Aplikacija je ponudila rešenje za prodaju sezonskih i ciklusnih karata, što je trenutno velika prednost u odnosu na postojeće sisteme na tržištu, a ustanovama koje organizuju koncerte klasične muzike je ovo neophodno.

8. LITERATURA

- [1] Microsoft - Silverlight https://en.wikipedia.org/wiki/Microsoft_Silverlight
- [2] Wikipedia - .NET Framework https://sr.wikipedia.org/wiki/.NET_Framework
- [3] MVC design pattern <https://www.geeksforgeeks.org/mvc-design-pattern/>
- [4] AngularJS <https://angularjs.org/>
- [5] Microsoft -ASP.NET Web API <https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api>
- [6] Integrate Google signin <https://developers.google.com/identity/sign-in/web/sign-in>
- [7] Facebook login <https://developers.facebook.com/docs/facebook-login/>
- [8] Wikipedia – OAuth <https://en.wikipedia.org/wiki/OAuth>

Kratka biografija:

Nikša Bogovac rođen je u Somboru 1981. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarske nauke i informatika odbranio je 2019.god.

kontakt: niksabogovac@yahoo.com

СОЛАРНИ ПУЊАЧ ЗА ЕЛЕКТРИЧНИ БИЦИКЛ**SOLAR CHARGER FOR ELECTRIC BICYCLE**

Борис Шаргач, Владимир Катић, Золтан Чорба, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду су размотрене могућности израде система за пуњење батерије електричног бицикла енергијом добијеном из соларних панела. Пуњачи који су употребљени за израду система су испитани и тако добијени резултати су приказани, уз објашњења појединих резултата. Размотрена је и могућност примене ових пуњача у јавном систему за изнајмљивање електричних бицикала.

Кључне речи: Соларни панел, Батерија, Пуњач, Пројектовање, Испитивање

Abstract – In this paper, the possibilities of creating a system for charging the battery of electric bicycle using energy generated by solar panels are discussed. Chargers that were used for creating the system are tested and the results of this test are displayed, along with explanation of certain results. A possibility of using this chargers in public system for renting electric bicycles is also discussed.

Keywords: Solar panel, Battery, Charger, Designing, Testing

1. ОПИС ФОТОНАПОНСКОГ СИСТЕМА

Основу система чине фотонапонски (соларни) панели који врше претварање светлосне енергије у електричну. Могуће је користити један или више панела, а у том случају је најбоље да они буду истог типа. Панеле је могуће везивати редно (ради добијања већег излазног напона) или паралелно (ради добијања веће излазне струје). При њиховом постављању је потребно водити рачуна о њиховој усмерености и нагибу.

Како би се омогућило пуњење батерије бицикла независно од количине расположивог Сунчевог зрачења (ноћу или током облачних дана), користи се додатна (стационарна) батерија у којој се произведена енергија складишти до тренутка када је потребна. У ову сврху се користе оловни желатинасти (гел) акумулатори који се одликују добрим односом цене и капацитета, затвореним кућиштем које омогућава постављање у произвољан положај и малим захтевима за одржавањем.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Владимир Катић, ред. проф.

Како би се спречило да дође до препуњавања акумулатора и да би се производња енергије из соларних панела што више повећала, панели се не везују непосредно на акумулаторе, већ се повезивање врши преко DC/DC претварача који има могућност праћења тачке највеће снаге (енг. Maximum Power Point Tracking – MPPT).

У систему се користи још један DC/DC претварач који се налази између акумулатора и батерије бицикла. Његова улога је да обезбеди пуњење батерије бицикла на одговарајући начин, као и да спречи прекомерно пражњење акумулатора.

2. ОПИС СКЛОПОВА БИЦИКЛА

Као основа при изради пуњача је искоришћен бицикл са ознаком Monesada Li36 (слика 1). Он се може покретати на три начина: преко педала (коришћењем физичке снаге корисника), помоћу мотора (коришћењем енергије ускладиштене у батерији) и комбинацијом та два начина (користе се педале за покретање бицикла, док мотор обезбеђује додатну снагу).



Слика 1. Електрични бицикл Monesada Li36

Основу електричног система овог бицикла чине мотор, контролер и батерија.

Мотор спада у групу DC мотора без четкица код кога се стални магнети налазе на ротору који се обрће око статора на коме се налазе намотаји. У овом случају, статор се налази на осовини задњег точка, док се ротор налази на кућишту мотора које се обрће око статора и истовремено игра улогу главе точка.

Радам мотора управља контролер који такође, помоћу одговарајућих сензора, читава његову брзину и брзину окретања педала. На основу улаза које му корисник задаје преко ручице за гас и кочнице које се налазе на корману бицикла, контролер управља радом целог система, а такође помоћу приказивача кориснику пружа неопходне податке.

Енергију за рад система обезбеђује батерија (слика 2) која се налази између седишта и задњег точка бицикла. Она је смештена у херметички затвореном кућишту на којем се налазе улазни прикључак за пуњење, излазни прикључак на који се повезује остатак система и брава која служи за осигуравање батерије у њеном лежишту, као и за одабир начина рада бицикла (са коришћењем мотора или без њега).



Слика 2. Батерија бицикла

Уз бицикл је испоручен и фабрички пуњач, а на основу резултата добијених његовим испитивањем [1] извршено је пројектовање одговарајућег пуњача.

3. ПРОЈЕКТОВАЊЕ ПУЊАЧА

3.1. Батерије

Батерија бицикла се састоји од 10 редно везаних литијум-гвожђе-фосфатних (LiFePO_4) ћелија. Напон потпуно напуњене батерије износи 42 V, а контролер онемогућава њен рад када напон на њој опадне на 32,4 V. Номинални капацитет батерије је 10 Ah, док препоручена струја пуњења има вредност од 1 C (струја једнака капацитету батерије у јединици времена), односно, 10 A. Пошто напон на батерији, због хемијских реакција у њој, опада и када се она не користи, усвојено је да доња вредност напона исправне батерије износи 30 V, док највећа струја пуњења износи 5 A.

Номинални напон оловних акумулатора је 12 V, али се његова вредност мења у зависности од нивоа напуњености. У фази пуњења сталним напонем, његова вредност износи 14,2 V, док у фази одржавања (енг. float charging) напон акумулатора треба да буде између 13,6 V и 13,8 V. Вредност напона при којој се сматра да је акумулатор испразњен зависи од његове карактеристике, а овде је усвојено да је то напон од 10,5 V.

Најједноставнији случај је када се као стационарна батерија користи само један акумулатор. Међутим, да би се смањила вредност струје која се повлачи из акумулатора, користи се редна веза два акумулатора. Тиме се олакшава израда DC/DC претварача, а омогућено је и постављање пуњача на већу удаљеност од стационарне батерије без знатног повећања губитака на водовима.

3.2. Пуњач батерије бицикла

Пошто је напон на стационарној батерији увек нижи од напона на батерији бицикла која се сматра исправном, потребно је користити подизач напона

(енг. boost converter) како би се омогућило њено пуњење. Струја пуњења I_{BAT} има приближну вредност

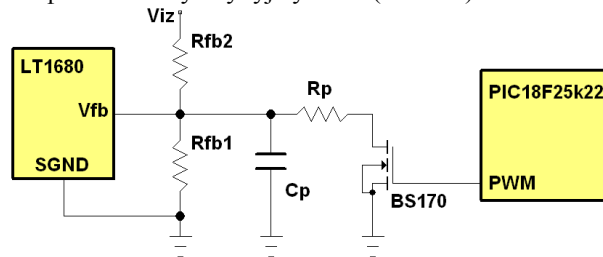
$$I_{BAT} = \frac{V_{boost} - V_{BAT}}{R_{BAT}}, \quad (1)$$

при чему је V_{boost} напон на излазу претварача, V_{BAT} је напон отвореног кола батерије, а R_{BAT} је њена унутрашња отпорност. Вредност V_{BAT} се мења током пуњења, те је потребно мењати и вредност V_{boost} како би струја пуњења остала приближно иста.

Да би се то постигло, користи се DC/DC претварач заснован на контролеру LT1680 [2] чији излазни напон V_{IZ} је одређен напонским разделником који чине отпорници R_{FB1} и R_{FB2} , и он износи

$$V_{IZ} = 1,25 \cdot \left(1 + \frac{R_{FB2}}{R_{FB1}}\right) [V]. \quad (2)$$

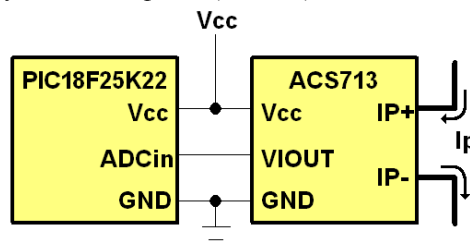
Вредности $R_{FB1} = 3 \text{ k}\Omega$ и $R_{FB2} = 69 \text{ k}\Omega$ су тако одабране да излазни напон има вредност од 30 V. Како би се добио излазни напон од 42 V, паралелно са R_{FB1} је везан отпорник $R_P = 7,2 \text{ k}\Omega$ који се помоћу мосфета BS170 укључује у коло (слика 3).



Слика 3. Коло за управљање излазним напонем

Укључивањем и искључивањем мосфета управља микроконтролер PIC18F25K22 [3] који користи импулсно-ширинску модулацију (PWM) како би управљао излазним напонем, а самим тим и излазном струјом. Подизач напона је пројектован у складу са препорукама произвођача контролера [2] како би обезбедио жељену излазну струју од 5 A, а потребно је напоменути да је коришћен калем индуктивности 47 μH , а као прекидачки мосфет је употребљен IPR029N06N.

Податке о вредности излазне струје микроконтролер добија преко чипа ACS713 [4] који мери струју помоћу Холовог ефекта (слика 4).



Слика 4. Мерење излазне струје пуњача

Напон батерије која се пуни микроконтролер мери помоћу напонског разделника који је везан на излаз пуњача. Овај излаз је одвојен од излаза подизача напона помоћу релеја чијим укључивањем, помоћу транзистора, такође управља микроконтролер. Он управља и радом вентилатора, такође помоћу транзистора, као и стандардним приказивачем са 2x16

карактера преко кога се корисник обавештава о раду уређаја.

Софтвер микроконтролера је осмишљен тако да омогућава подешавање рада пуњача у складу са потребама корисника, што се пре свега односи на могућност избора вредности струје пуњења. За унос команди предвиђена су три тастера.

3.3. Соларни пуњач акумулатора

Као основа за израду овог пуњача одабран је чип LT8490 [5] који представља контролно коло спуштача-подизача напона (енг. buck-boost) са уграђеном могућношћу управљања поступком пуњења батерије и MPPT. Избор ове компоненте значајно олакшава израду претварачког склопа јер се рад контролера подешава избором компоненти са којима се повезује, без потребе за додатним програмирањем. Првобитна замисао је била да се овај чип искористи као основа за израду потпуно новог пуњача који је потпуно прилагођен захтевима пројектног задатка. Међутим, утврђено је да већ постоји велики број доступних пуњача заснованих на овом чипу који испуњавају неке од постављених захтева. Због тога је одлучено да се изврши испитивање једног од њих (слика 5) чиме би се утврдило са коликом успешношћу такав пуњач обавља постављени задатак.



Слика 5. Пуњач заснован на чипу LT8490

Одабрани пуњач има могућност избора врсте извора напајања (соларни панел или стабилисани извор једносмерне струје) и избора врсте батерије која се пуни (оловни акумулатор или литијумска батерија). Улазни напон може имати вредност до 75 V, а највећа вредност излазног напона у фази пуњења сталним напонам износи 28,4 V. Највећа вредност улазне струје је ограничена на 20 A, док се вредност излазне струје може подешавати, помоћу потенциометра на плочи, у распону од 0,5 – 20 A.

4. ИСПИТИВАЊЕ РАДА ПУЊАЧА

Одлучено је да се испитивање пуњача обави мерењем снага на њиховим улазима и излазима у циљу одређивања степена искоришћења енергије који они постижу. Такође, на овај начин се може извршити упоређивање пројектованог пуњача батерије бицикла са комерцијално доступним пуњачем у смислу степена искоришћења који они постижу.

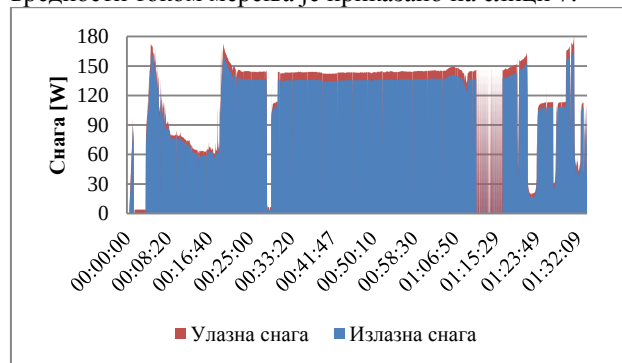
Код оба пуњача, снаге су мерене посредно, мерењем напона и струја на њиховим улазима и излазима помоћу уређаја Sanwa PC720M који су били повезани

на рачунар и чија читавања су бележена помоћу програма PC Link 7. Читавања су вршена са временом одабирања од 1 секунде. Мерење струја је вршено посредно, мерењем падова напона на шант отпорницима како би се избегло оштећивање уређаја услед превелике струје, пре свега током прикључивања батерија и извора напајања на пуњаче. Током испитивања соларног пуњача вршена су и повремена читавања вредности Сунчевог зрачења помоћу уређаја Seaward Solar Survey 200R.

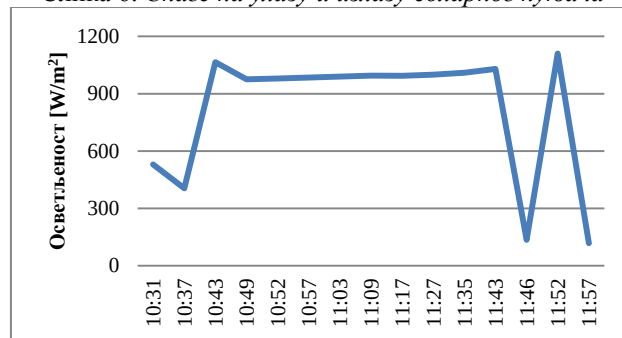
4.1. Испитивање соларног пуњача

Као извор напајања соларног пуњача коришћен је панел Schott EFG 165 [6] номиналне снаге 165 W. Пошто код употребљеног пуњача не постоји ознака која би указала на који начин је потребно подесити потенциометар како би се постигла жељена излазна струја, он је подешен на позицију која приближно одговара половини његовог опсега. На тај начин се постиже да жељена излазна струја буде око 10 A, чиме је обезбеђено да чип LT8490 током свог рада сигурно користи свој MPPT алгоритам.

Измерена улазна и излазна снага су приказане на слици 6. Види се да излазна снага прати промену улазне снаге, а њихова разлика представља снагу губитака који се јављају у самом пуњачу. Такође, улазна снага није непроменљива, већ зависи од јачине Сунчевог зрачења, а приближно кретање његове вредности током мерења је приказано на слици 7.

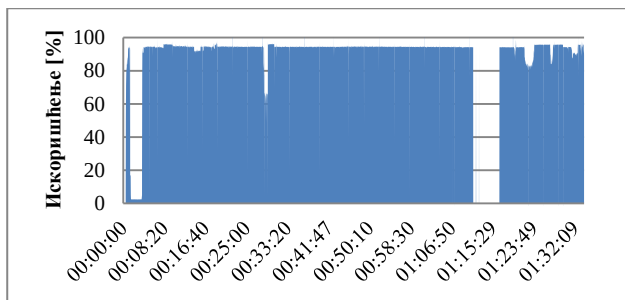


Слика 6. Снаге на улазу и излазу соларног пуњача



Слика 7. Снага Сунчевог зрачења током мерења

Иако је у току мерења било одређених грешака које су се јавиле као последица рада чипа LT8490 (повремено врши снимање карактеристике соларног панела) или због случајног одласка рачунара у стање мировања, оне чине мали део добијених резултата и могу се занемарити. Коришћењем добијених података, израчунат је степен искоришћења улазне снаге који пуњач постиже и то је приказано на слици 8.



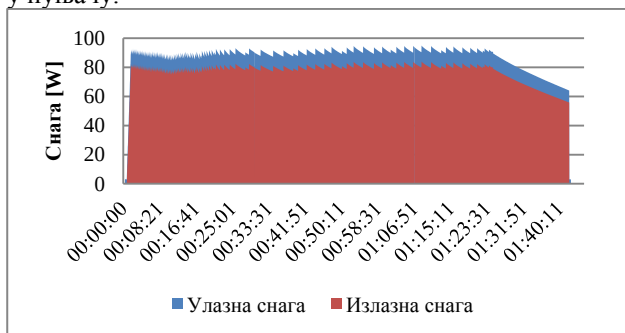
Слика 8. Степен искоришћења који постиже пуњач

Види се да пуњач постиже висок степен искоришћења улазне снаге који готово увек прелази 90%. Падови се јављају само у случајевима знатног смањења улазне снаге до којих долази услед појаве наоблачења или намерно изазваног засенчења (време од 00:28:27 до 00:29:18 од почетка мерења).

4.2. Испитивање пуњача батерије бицикла

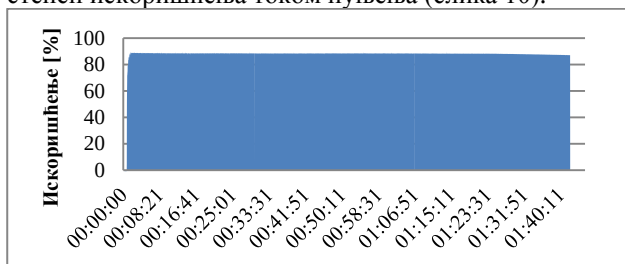
Испитивање пуњача батерије бицикла је изведено на сличан начин као и у претходном случају, с тим да је овде коришћен стабилисани извор једносмерног напона од 24 V који је подешен да на излазу даје напон од 25,7 V. Пуњач је подешен тако да вредност струје пуњења батерије буде 2 A уз коришћење софтверске компензације струје пуњења (неопходна је због уочене неправилности у раду чипа ACS713).

Измерене снаге на улазу и излазу пуњача су приказане на слици 9. Као и у претходном случају, њихова разлика настаје због губитака који се јављају у пуњачу.



Слика 9. Улазна и изразна снага пуњача батерије бицикла

Изразна снага прати улазну снагу, без обзира на режим пуњења (стална струја или стални напон), док је снага губитака у претварачу готово непромењена. Ово се најбоље може видети уколико се погледа степен искоришћења током пуњења (слика 10).



Слика 10. Степен искоришћења пуњача бицикла

Степен искоришћења је увек изнад 80%, а благи пад се јавља само по преласку из режима сталне струје у режим сталног напона.

Потребно је нагласити да поступак пуњења није потпуно завршен што је последица деловања система за надзор који је уграђен у батерију бицикла. Наиме, коришћена батерија није у оригиналном стању, већ су над њом претходно извршене преправке које су за последицу имале смањење њеног капацитета.

5. ЗАКЉУЧАК

Оба испитивана пуњача су током свог рада постигла високе степене искоришћења улазне снаге. Начин израде ових пуњача омогућава њихово коришћење у јавном систему за изнајмљивање електричних бицикала јер се они релативно лако могу прилагодити потребама и захтевима корисника.

6. ЛИТЕРАТУРА

- [1] Владимир Буха, „Експериментално испитивање елемената система за соларно допуњавање батерија електричног бицикла“, Србија, Нови Сад, 2013.
- [2] Linear Technology, „LT1680 – High Power DC/DC Step-Up Controller“
- [3] Microchip Technology, „PIC18(L)F2X/4XK22 Data Sheet – 28/40/44-Pin, Low-Power, High-Performance Microcontrollers with XLP Technology“
- [4] Allegro MicroSystems, „ACS713 – Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kV RMS Voltage Isolation and a Low-Resistance Current Conductor“
- [5] Linear Technology, „LT8490 – High Voltage, High Current Buck-Boost Battery Charge Controller with Maximum Power Point Tracking (MPPT)“
- [6] Schott Solar AG, „SCHOTT Solar polycrystalline solar modules with EFG™ cells“

Кратка биографија:



Борис Шаргач рођен је у Оџацима 1984. године. Гимназију је завршио у Оџацима 2003. године. Факултет техничких наука је уписао 2004. године.



Владимир Катих рођен је у Новом Саду 1954. године. Дипломирао је на Факултету техничких наука у Новом Саду 1978. године. Магистрирао је на Универзитету у Београду 1981. године, а докторирао је, такође на Универзитету у Београду, 1991. године. Од 2002. године је редовни професор Универзитета у Новом Саду. Тренутно је професор Факултета техничких наука. Области интересовања су енергетска електроника, обновљиви извори електричне енергије, квалитет електричне енергије и електрична возила.



Золтан Чорба рођен је у Новом Саду 1962. године. Од 1992. године је запослен на Факултету техничких наука. Тренутно је руководилац лабораторије за обновљиве и дистрибуиране изворе електричне енергије. Области интересовања су обновљиви извори електричне енергије, енергетска ефикасност и квалитет електричне енергије.

ПРИМЕНА SERVERLESS АРХИТЕКТУРЕ НА АПЛИКАЦИЈУ ЗА УПРАВЉАЊЕ РАДОВИМА У ЕЕ СИСТЕМУ**THE APPLICATION OF SERVERLESS ARCHITECTURE TO THE APPLICATION FOR MANAGEMENT OF WORKS IN EE SYSTEM**

Немања Стевановић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду представљена је примена Serverless архитектуре на апликацију која симулира управљање радовима у електроенергетском систему. Приказан је начин на који се може смањити оптерећење једног од сервиса апликације употребом Azure Functions које представљају Serverless архитектуру креиране од стране Microsoft Azure-a.

Abstract – This paper presents the application of Serverless architecture to the application which simulates management of works in electric power system. It shows way that can reduce load of one application service using Azure Functions which represents Serverless architecture made by Microsoft Azure.

Кључне речи: Cloud, Serverless, Azure Functions

1. УВОД

Електричне мреже представљају један од најзначајнијих делова електроенергетског система. Помоћу њих се обезбеђује непрекидно напајање потрошача електричном енергијом. Колико год да је електрична мрежа отпорна, увек постоји могућност њеног кvara. Због тога, оператери који врше надгледање мреже морају имати могућност извршавања планираних или непланираних радова на мрежи помоћу својих екипа. Како би се олакшао посао оператерима креирана је апликација која врши управљање радовима. Апликација је базирана на примени информационих технологија, Smart Grid система и Cloud технологија.

Идеја овог рада је да се постигне смањење оптерећења сервиса апликације који је задужен за креирање и управља планираним радовима. Циљ је да се упозна апликација и окружење у којем ради, као и да се упозна Azure Functions, начин њиховог рада, начин њихове интерграције са апликацијом, као и предности и мане коришћења ових функција као решење. Како се читав рад базира на употреби Cloud окружења, укратко ће бити описани нивои услуга које нам Cloud пружа као и Service Fabric окружење у којем сама апликација и ради.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Срђан Вукмировић, ванр. проф.

2. ТЕОРИЈСКЕ ОСНОВЕ**2.1 Smart Grid системи**

Smart Grid уводи значајне промене у начину на који се производи, преноси, дистрибуира и троши електрична енергија. Тачна дефиниција Smart Grid-a не постоји, већ се он посматра као интегрисани скуп технологија који смањује трошкове, повећава социјалну корист и смањује загађење животне средине. Под интегрисаним скупом технологија, Smart Grid подразумева повезивање електричне и комуникационе инфраструктуре са аутоматизованим постројењима и информационим технологијама у једну целину како би та целина пружила максималну корист. Циљеви којима Smart Grid тежи су следећи:

- *Интелигентна мрежа* која има могућност идентификације кварова и аутоматску минимизацију њихових утицаја
- *Ефикаснија мрежа* која је у стању да прихвати све веће захтеве за енергијом, а истовремено се фокусира на квалитет енергије
- *Прилагодљива мрежа* која прима енергију из свих извора и дозвољава интеграцију нових технологија
- *Отпорна мрежа* је отпорна на климатске катастрофе и терористичке нападе
- *Мотивишућа мрежа* нуди могућност реалтима комуникације кроз цео систем
- *Зелена мрежа* омогућава значајно унапређење у процесу очувања животне средине
- *Мрежа нових шанси* дозвољава креирање нових типова послова, фирми, клијената, опреме итд

2.2 CLOUD

Cloud се односи на дељење ресурса, софтвера и информација путем мреже. Информације и подаци се чувају на физичким или виртуалним серверима које одржава и контролише Cloud провајдер. Коришћење Cloud-a се ослања на исказ "pay as you go". Ово значи да, ако су нам у неком тренутку потребни додатни ресурси, ми их нећемо купити већ ћемо их изнајмити од провајдера, такође, ако нам ти ресурси више не буду били потребни, отказаћемо њихову употребу и више нећемо плаћати за њих. Приликом коришћења Cloud-a, обавезе провајдера и клијента се описују преко нивоа услуга које клијент изнајмљује. Три главна нивоа услуга су:

- *Инфраструктура као сервис (IaaS)* - састоји се од високо аутоматизованих и скалабилних рачунарских ресурса који су употпуњени могућностима складиштења и повезивања на мрежу. *IaaS* омогућава предузећима да изнајмљују хардвер на захтев и по потреби уместо да га директно купују.
- *Платформа као сервис (PaaS)* - обезбеђује платформу која омогућава клијентима да развијају, покрећу и управљају апликацијама без сложених проблема изградње и одржавања инфраструктуре која је повезана са развојем апликација.
- *Софтвер као сервис (SaaS)* - представља најчешће коришћен ниво услуга где нуди готов производ крајњим клијентима. Клијенти уопште не воде рачуна о развоју и управљању апликација. Приступна тачка овим апликацијама се махом нуди преко web претраживача.

Поред ових нивоа услуга, због саме теме рада, ми ћемо навести и *функцију као сервис (FaaS)* која представља *Serverless* начин за изградњу дела или читаве апликације. Клијент пише и ажурира парче кода које се покреће на одређени догађај. *FaaS* се као ниво услуга сврстава на врху *PaaS* нивоа.

2.3 Service Fabric

Service Fabric представља платформу дистрибуираних система који олакшавају инжењерима рад јер они могу остати усредсређени на саме захтеве апликација. *Service Fabric* нуди могућност управљања скалабилности и поузданости микросервиса који ће се користити у апликацији. На овај начин се тежи ка преласку са монолитних апликација на микросервисне. Микросервисну апликацију описују сервиси где се сваки сервис бави једном бизнис логиком. На овај начин добијамо разне предности, као на пример: сваки сервис се може посматрати као независна компонента, лакше се врши тестирање над сервисима који се баве само једном логиком, као што је овде случај, над сервисима могу радити мањи тимови инжењера итд. Треба споменути да пребацивањем са монолитне апликације на микросервисну постоје и могуће мане као што је случај са повећаном комуникацијом како би сервиси могли да размењују податке међусобно, а те комуникације можда нису ни постојале у монолитној апликацији. *Service Fabric* нам даје могућност креирања *stateless* и *stateful* сервиса. Разлика је у томе да ли у тим сервисима постоји неко стање које треба да се сачува, ако дође до пада сервиса, или не. Предност *stateful* сервиса је у томе што перзистирају своја стања кроз *reliable* колекције [1] помоћу којих ће имати брз приступ подацима, док је предност *stateless* сервиса у томе што су погодни за скалирање у случају већег оптерећења.

2.4 Serverless архитектура

Serverless представља архитектуру извршавања где *Cloud* провајдер обезбеђује подизање сервиса и руковање ресурсима. Иако носи назив *serverless* (без

сервиса) у позадини се подиже сервис који ће извршавати код. Назив је добио по томе што клијенти не плаћају провајдеру сервис већ само ресурсе који су утрошени за време извршавања његовог кода. Микросервисна архитектура се развила како би се логика велике апликације разводијала и олакшао развој јер се ради на мањим целинама. Таквом архитектуром смо имали предност јер смо могли да скалирамо само онај сервис који трпи веће оптерећење док друге сервисе нисмо морали да дирамо. *Serverless* архитектура је отишла и корак напред где ми добијамо могућност скалирања само једног дела једног сервиса. *Serverless* апликације су *event-driven* апликације. То значи да се њихово покретање ослања на неки догађај.

Провајдери који нуде коришћење *Serverless* оркужења цену одређују преко изабраних планова под којим ће се *Serverless* користити. Сваки од тих планова описује шта нуди, која су му ограничења и на који начин се формира цена. Неке од предности *Serverless*-а су цена где клијент плаћа само за потрошене ресурсе, еластичност и скалабилност где ће провајдер да води рачуна о овим концептима и продуктивност где су јединице кода спољашњем свету изложене као једноставне функције, па тако нпр. клијенти не морају да воде рачуна о *multithreading*-у.

Мане које *Serverless* носи са собом су те што клијент нема могућност избора машине на којој ће се његова функција покренути и ограничења која се могу појавити употребом коришћеног плана.

3. СИСТЕМ УПРАВЉАЊА РАДОВИМА

Као што је у уводу већ наведено, систем управљања радовима представља софтверски производ који се заснива на примени *Smart Grid* система и *Cloud* технологија. Апликација је направљена помоћу програмског језика *C#* и извршава се у *Service Fabric Cloud* окружењу. Оператери помоћу ове апликације имају могућност да преко графичког приказа мреже посматрају тренутно стање, као и могућност измена стања мреже. Под изменама стања мреже се подразумева ручно мењање стања прекидача као и креирање планираних радова над жељеним секцијама мреже у одређеном периоду.

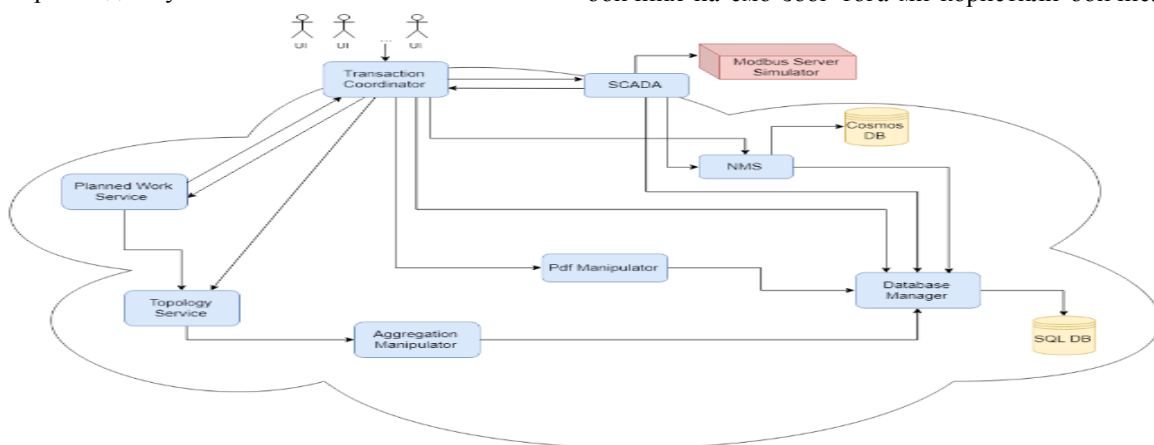
Приликом промене стања мреже мења се и њена енергизација, што се рефлектује и на графичком приказу за кориснике. Систем води рачуна о томе да приликом креирања и извршења планираних радова не дође до преклапања рада над истим секцијама као и да не дође до инцидентних ситуација. Такође, приликом креирања планираног рада систем алгоритамски одређује и генерише оптималне инструкције за његову реализацију. Циљ приликом креирања планираних радова јесте да што мање корисника остане без напајања. Планирани радови се могу извршити и пре заказаног времена уколико су испуњени безбедносни услови. У супротном се планирани рад извршава аутоматски у заказно време. Разлози за вршење планираних радова су потреба за ремонтом одређених делова мреже приликом квара или одржавање мреже, као и координација са осталим радовима на мрежи.

Предности система за управљање планираним радовима су следеће:

- Већа сигурност радника на терену, као и опреме
- Брз увид у стање мреже
- Могућност управљања на даљину из диспечерског центра
- Смањење трошкова управљања
- Лакше прикупљање статистичких података о испадима корисника
- Генерисање извештаја наспрам унетих параметара и на основу прикупљених података

3.1 Архитектура система

Систем се састоји од осам сервиса и две базе података које они користе. Уз то, SCADA сервис користи *Modbus Server Simulator* за симулацију RTU уређаја. Архитектура решења је приказана на слици 3.1.1. Како је апликација подигнута у *Service Fabric*-у, NMS, Topology и Planned Work сервис су изграђени као *stateful* сервиси док су остали *stateless*.



Слика 3.1.1 Архитектура система

За даље разумевање рада можемо још рећи да ће се план рада који клијент креира проследити до Transaction Coordinator сервиса који имплементира *API Gateway* образац [2] и да ће он након тога проследити план Planned Work сервису који ће у даљем раду бити повезан са *Serverless* архитектуром.

4. ОПИС ПРОБЛЕМА

Како ова апликација и носи назив, систем за управљање радовима, очекивано је да њен фокус буде на креирању и извршавању тих радова. За ово ће бити задужен Planned Work сервис који поред руковања акција које долазе са клијентске стране, има имплементирану и аутоматику у себи која ће, такође, радити над планираним радовима. Због овога ће Planned Work сервис трпети највише оптерећења или ће макар бити један од сервиса који трпе највише оптерећења. Пошто је овај сервис изграђен као *stateful* сервис, код њега скалирање није лако као код *stateless* сервиса. *Service Fabric* нуди решење и за скалирање *stateful* сервиса, али оно подразумева његово партиционисање. На овај начин, сервис би имао више партиција где би свака партиција била одговорна за део података. Мана овог приступа је што сервис који позива партиције мора да имплементира логику партиционисања да би могао да погоди жељену

партицију. На овај начин се ствара зависност између сервиса, а у случају лошијег плана партиционисања, гађане партиције неће имати равномерно оптерећење и опет постоји могућност да нека партиција трпи превише оптерећења.

5. РЕШЕЊЕ ПРОБЛЕМА

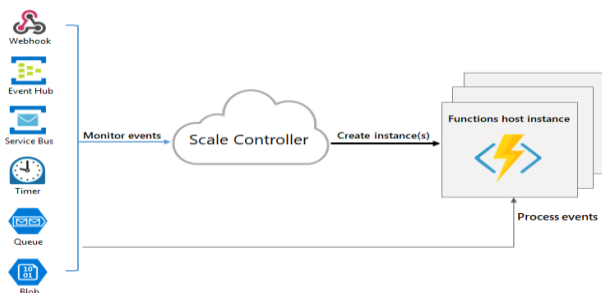
За решење претходно описаног проблема је изабрана *Serverless* архитектура коју је креирао *Microsoft Azure* преко својих *Azure Functions*. Циљ је да се креирање планова обавља преко *Azure Functions* како би се смањило процесорско оптерећење машине на којој ће бити покренут Planned Work сервис у замену за нешто дуже извршавање клијентског захтева.

5.1 Опис Azure Functions

Azure Functions се јавља у своја два облика. Један је у виду *Durable* функција, док је други обичне *Azure* функције. Разлика је што *Durable* представљају *stateful* функције што их чине нешто споријим од обичних па смо због тога ми користили обичне. Као

догађај на који ће се функција позивати изабрали смо *HTTP* окидач. Функција се састоји из два дела где је први део њено заглавље док је други тело функције које представља њену имплементацију. У заглављу је постављена рута *HTTP* захтева, тип захтева који је постављен на „post“ и ауторизациони ниво. Због ауторизационог нивоа, свако ко жели да погоди функцију мораће да поседује кључ. Поред функције, креира се и њена конфигурација где се постављају параметри за њен рад. Овим параметрима смо поставили ниво њеног логовања као и лимит за њено конкурентно извршавање. Свака функција се покреће из два могућа стања. Прво стање је *cold*, а друго је *warm*. Функција се покреће из *cold* стања ако се не користи одређено време или ако дође до оптерећења што ће изазвати њено скалирање, где ће се нове инстанце, такође, покренути из *cold* стања. Овим се повећава латенција што утиче на перформансе. Са друге стране, *warm* стање подразумева да је функција у могућности да се одмах покрене и изврши свој посао. О скалирању функција води рачуна *Scale Controller* који је приказан на слици 5.1.1. Скалирање може бити у размаку између 0 и 200 инстанци, где се може додавати највише једна инстанца по секунди. Као план над којим ће се покренути функција изабран је *Consumption* план. По овом плану на цену утичу

број извршавања, време извршавања и искоришћена меморија. Ограничења које он поставља су да извршавање функције не може да буде дуже од 10 минута, постоји *cold* стање, највише једно процесорско језгро се користи у току рада и максимална искоришћена меморија је до 1.5GB.



Слика 5.1.1. *Scale Controller*

5.2 Имплементација решења

Имплементација решења је обухватала имплементацију саме функције и њено повезивање на Planned Work сервис. Тело функције се састоји из два корака где први корак обухвата десеријализацију пристиглих података, а други корак обухвата коришћење алгорита за креирање планова. Функција ће вратити грешку у случају невалидних података или у случају покушаја неауторизованог приступа.

Planned Work сервис је морао да имплементира логику којом ће се одлучивати када ће се звати функција, а када ће извршити локално креирање плана. Коришћена су два параметара за ово и то лимит позива који се може мењати у конфигурацији и број извршених позива у сату који ће се чувати у *reliable* колекцији. У случају да лимит није достигнут, позиваће се функција, ако је број позива једнак лимиту, сервис ће послати *email* администратору и остале захтеве за креирањем планова у току тренутног сата ће извршавати локално. Planned Work сервис ће у случају позивања функције одрадити то у три корака. Први корак је припремање садржаја за слање што обухвата серијализацију података. Други корак представља слање *HTTP* захтева који у заглављу садржу кључ ауторизације и у телу садржи припремљене податке. Трећи корак обухвата логику након позива функције која зависи од пристиглог одговора. Последњи корак води рачуна и о могућим грешкама које могу настати у току слања захтева.

За слање захтева преко *HTTP*-а, изабрана је његова верзија 2. У односу на претходну верзију 1.1 која је била заступљена 15 година, а и даље је у употреби, ова верзија има могућност знатно бржег слања. Нова верзија је увела доста новина, али две најзначајније за наш пројекат су: мултиплексирање захтева и компресија заглавља. Мултиплексирање захтева представља начин слања више захтева кроз једну *TCP* конекцију. Ово нарочито може да утиче на перформансе апликације. Како је *HTTP* протокол *stateless*, то значи да сваки захтев мора да носи онолико података колико је потребно серверу да би он могао да изврши жељену акцију. Овим механизмом се проузрокује слање оквира информација који се понављају у захтевима. *HTTP/2* користи *HPACK* спецификацију за једноставну и сигурну компресију заглавља којом се избегава наведени проблем.

6. ЗАКЉУЧАК

У овом раду је представљено једно од могућих решења за постизање смањења оптерећења неког сервиса. Описана је *Serverless* архитектура као и њена употреба преко *Azure Functions*. Одрађени тестови су показали да употребом *Azure Functions* заиста долази до смањења оптерећења на Planned Work сервису и да се латенција код клијентског позива није много повећала.

7. ЛИТЕРАТУРА

- [1] *Programming Microsoft Azure Service Fabric, Second Edition*, Haishi Bai
- [2] *Microservices Patterns*, Chris Richardson

Кратка биографија

Немања Стевановић рођен је 09.06.1995. год. у Смедереву. Завршио је економску школу у Смедереву 2014. год. Исте године је уписао основне академске студије на Факултету техничких наука у Новом Саду. Дипломски рад из области Електротехника и рачунарства је одбранио 2018. године, након чега је исте године уписао мастер академске студије. Испунио је све обавезе и положио све испите предвиђене студијским програмом.

PODRŠKA ZA UPRAVLJANJE SPRING CLOUD GATEWAY KOMPONENTOM U MIKROSERVISNOM EKOSISTEMU**SUPPORT FOR MANAGING SPRING CLOUD GATEWAY COMPONENT IN THE MICROSERVICE ECOSYSTEM**

Stevan Kosijer, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su predstavljene osobine, osnovne komponente, prednosti i mane mikroservisne arhitekture. Obrazložena je funkcija komponente koja služi za kontrolu ulaznog saobraćaja u mikroservisnom ekosistemu. Opisana je implementacija proširenja koje omogućava direktnu manipulaciju rutama i filterima ove komponente. Nad realnim sistemom koji podržava dato proširenje izvršena su testiranja performansi pri obavljanju operacija manipulacije rutama. Predstavljene su rezultati testiranja kao i mogući pravci daljeg razvoja ovog rešenja.

Ključne reči: Mikroservisna arhitektura, Gateway komponenta, Spring Boot Admin, upravljanje mikroservisnim ekosistemom.

Abstract – Features, main components, advantages and disadvantages of microservice architecture are presented in this paper. The function of the component used to control the inbound traffic in the microservice ecosystem is explained. An extension implementation that allows direct manipulation of routes and filters of this component is described. A real system that supports this extension has been subjected to performance tests when performing route manipulation operations. Test results are presented as well as possible directions for further development of this solution.

Keywords: Microservice achitecture, Gateway component, Spring Boot Admin, manipulating of microservice ecosystem.

1. UVOD

Trenutno postoji mnoštvo tehnologija, arhitektura i praksi koje služe za rešavanje problema vezanih za određivanje i postavku infrastrukture softverskih rešenja. Mikroservisna arhitektura se izdvaja kao moguće rešenje za transformaciju i skaliranje velikih softverskih sistema. Razvoj mikroservisne arhitekture potstakao je izradu mnogobrojnih alata čija je namena olakšanje rada i razvoja mikroservisnih sistema.

Njihova primena se javlja u svim fazama razvoja mikroservisne arhitekture. Većinu ovih alata i projekata razvila je zajednica otvorenog koda. Jedan od popularnijih je *Spring Boot Admin* projekat.

Spring Boot Admin projekat je kreiran sa idejom da ponudi jednostavan način registrovanja i upravljanja svim komponentama u mikroservisnoj arhitekturi koje su napravljene upotrebom *SpringBoot* okvira za razvoj veb baziranih aplikacija. Projekat *Spring Boot Admin* nudi centralizovano mesto sa uređenim korisničkim interfejsom za uvid u trenutno stanje sistema i mogućnost manipulacije podešavanja određenih komponenti sistema. Ono što nedostaje u ovom projektu je podrška za *Spring Cloud Gateway* komponentu.

Podrška za *Spring Cloud Gateway* komponentu podrazumeva proširenje korisničkog interfejsa *Spring Boot Admin* projekta, koje bi bilo u stanju da prepozna prisustvo *Gateway* komponente u mikroservisnom ekosistemu i ponudi dodatnu specijalizovanu stranicu putem koje je moguće vršiti pregled trenutnog stanja definisanih ruta i filtera i manipulaciju njihovih konfiguracija. Manipulacija rutama i filterima bi omogućila kontrolu ulaznog toka *HTTP* saobraćaja u mikroservisnom ekosistemu, jer se na osnovu njihovih definicija formiraju pravila koja služe za upravljanje saobraćajem. *Spring Cloud Gateway* komponenta predstavlja posrednika u komunikaciji mikroservisnog ekosistema sa spoljnim svetom. Iz ovog razloga ona ima bitnu ulogu u aspektima sigurnosti samog sistema.

Podložna je čestom nadgledanju i sitnim izmenama kako bi se omogućio ili zabranio ulaz u sistem sa spoljne strane. Smatra se da bi ovo proširenje doprinelo u aspektima brzine izvršenja akcija manipulacije konfiguracijama i smanjenju broja koraka potrebnih za izvršavanje ovih akcija. Bez ovakvog rešenja manipulacija konfiguracionim podacima na *Gateway* komponenti svodi se na direktne izmene podataka u izvornom programskom kodu komponente i njeno potpuno restartovanje, što nije efikasno. Pored *Uvoda* i *Zaključka*, ovaj rad sadrži još tri poglavlja. U poglavlju *Mikroservisna arhitektura* predstavljene su teorijske osnove evolucije mikroservisne arhitekture, prednosti i mane mikroservisa i neke od najbitnijih komponenti neophodnih za rad i funkciju mikroservisa. U poglavlju *Spring Boot Admin projekat* predstavljen je *Spring Boot Admin* projekat sa osnovnim svojstvima i metodama koje obezbeđuju njegovu funkciju za nadgledanje i upravljanje servisima u mikroservisnoj arhitekturi. U poglavlju *Podrška za Spring Cloud Gateway komponentu* predstavljena je implementacija podrške za *Gateway* komponentu, osnovni entiteti i njihova funkcija, kao i analiza razlike u performansama pre i posle implementacije rešenja. Ovaj rad obuhvata deo rezultata koji su prezentovani u master radu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Slavica Kordić, vanredni prof.

2. MIKROSERVISNA ARHITEKTURA

Mikroservisi predstavljaju tehniku razvoja softvera koja služi za konstrukciju aplikacije kao kolekcije slabo povezanih servisa [1]. Svaki servis je samostalan i trebalo bi da implementira jednu funkciju poslovne logike, odnosno skup bliskih funkcionalnosti. Osnovne karakteristike mikroservisa su: postojanje komponenti koje su zadužene za pojedinačnu funkcionalnost, organizovanje i podela komponenti po poslovnim potrebama, decentralizovano upravljanje i manipulisanje podacima, fleksibilnost i robustnost. Mikroservisna arhitektura je dizajnirana da prevaziđe izazove, neuspehe i probleme većih aplikacija. Njenom upotrebom postiže se veća otpornost sistema na greške i nezavisnost komponenti u sistemu.

Zbog nezavisnosti komponenti različite grupe ljudi koje rade na razvoju projekta mogu da se fokusiraju isključivo na specifičan deo aplikacije. Mogu da upotrebe svojstven stil programiranja, bez brige da će ugroziti bilo koga drugog. Umesto deljenja jedne baze podataka, kao što je slučaj kod monolitne arhitekture, svaki mikroservis poseduje zasebnu bazu podataka. Postojanje posebne baze podataka za svaki mikroservis je ključno i doprinosi smanjenju zavisnosti među servisima. Svaki servis može da poseduje i različit tip baze podataka, baš onaj koji najbolje odgovara njegovim potrebama. Svaki mikroservis se može smatrati zasebnom komponentom. Za osnovni rad i funkciju implementacije mikroservisne arhitekture, izdvojene su *discovery*, *configuration* i *gateway* komponente i predstavljene u narednim potpoglavljima.

2.1. Discovery komponenta

Mikroservisima je neophodna međusobna komunikacija da bi delili informacije ili sinhronizovano izvršavali složene funkcionalnosti. Klijent za otkrivanje servisa (*discovery client*) u osnovi treba da omogući registraciju i rezoluciju instanci servisa. Pokrenuta instanca servisa, postupkom registracije servisa signalizira svoju dostupnost *discovery* servisu. Kada je instanca servisa dostupna, druge instance servisa koriste postupak rezolucije servisa da bi pronašli datu instancu servisa na mreži. Rezolucija predstavlja proces vraćanja fizičke mrežne lokacije pojedinačne instance servisa.

Ove dve osnovne operacije, međutim, obuhvataju širi spektar sofisticiranog ponašanja neophodnog za funkciju mikroservisa. Rezultat ovog ponašanja je mogućnost međusobne komunikacije svih instanci u sistemu.

2.2. Configuration komponenta

U mikroservisnoj arhitekturi se uglavnom svi servisi razmeštaju po različitim lokacijama i različitim serverima. Svaki servis ima svoju datoteku svojstava u koju se beleže karakteristike kao što su konfiguracija baze podataka, *URL*-ovi vezani za samo okruženje i svojstva vezana za taj mikroservis. Ako se svaka od ovih datoteka svojstava čuva zajedno sa samim servisom, na istom okruženju, može doći do različitih problema.

Na primer, ako treba da se izvrši čak i najmanja promena, kao što je izmena korisničkog imena za bazu podataka ili neka sitna izmena u nekom od *URL*-ova, prvo se mora pronaći lokacija instance ovog mikroservisa, zatim se mora pristupiti njegovom serveru i okruženju, napraviti

konkretna izmena i na kraju ponovo izgraditi i restartovati cela instanca servisa. Ako se ovakav pristup ponovi za svaku od instanci servisa, može se pokazati kao vrlo spor i zahtevan. Iz ovog razloga u mikroservisima je uvedena nova komponenta, pod imenom centralni konfiguracioni servis ili *configuration* komponenta. Centralni konfiguracioni servis je odgovoran za obezbeđivanje datoteka svojstava za svaki instancu servisa koja je registrovana na klijentu za otkrivanje mikroservisa.

U trenutku pokretanja, on preuzima sve datoteke svojstava sa putanje navedene u svojoj konfiguraciji i skladišti ih u memoriji. Ova putanja može biti fizička putanja na sistemu ili putanja do nekog repozitorijuma kao što je *Git*. *Configuration* komponenta postiže veću dinamičnost i brzinu sistema. Kontrola samih konfiguracija postaje preglednija i centralizovana.

2.3. Gateway komponenta

Klijenti aplikacije zasnovane na mikroservisima moraju da pristupaju i lociraju pojedinačne instance servisa. Zbog dinamičnosti koja je prisutna u implementaciji mikroservisne arhitekture ovo može biti izazov. Pored pristupa, potrebno je i kontrolisati saobraćaj upućen od strane klijenta. *Gateway* služi kao jedinstvena ulazna tačka za mikroservise i sve vrste klijenata. On opslužuje sve zahteve i preusmerava, rutira ili balansira do relevantne instance mikroservisa, na osnovu jedinstvene uloge i specifičnih svojstava svakog tipa mikroservisa.

Da bi obezbedio javni *API*, *Gateway* mora na pouzdan način da odredi lokacije i proveriti dostupnost mikroservisa. Zbog toga u mikroservisnom ekosistemu uvek postoji implementiran neki mehanizam pronalaženja i logičkog imenovanja mikroservisa. *Gateway* ima i sigurnosnu ulogu jer pruža mogućnosti za validaciju i autorizaciju sadržaja *HTTP* zahteva koje obrađuje. Pruža i mehanizme za otklanjanje grešaka u slučajevima kada se prosledeni *HTTP* zahtev ispostavi nevalidan.

Uz postojanje *Gateway* komponente mikroservisi interno mogu koristiti jedan optimizovan binarni protokol ili mehanizme za sinhronu i asinhronu komunikaciju i na taj način garantovati konstantnu isporuku.

3. SPRING BOOT ADMIN PROJEKAT

Spring Boot Admin predstavlja veb aplikaciju koja je nastala kao proizvod potreba *open source* zajednice. Njena glavna svrha jeste da obezbedi način za registrovanje i nadgledanje komponenti u arhitekturama kao što je mikroservisna arhitektura. Ona takođe rešava i problem modifikacije konfiguracija samih komponenti bez zaustavljanja njihovog rada. Pruža interfejs putem koga je moguće izvršavati akcije nadgledanja i upravljanja. *Spring Boot Admin* je prilagođen za rad sa ostalim *SpringBoot* aplikacijama. *SpringBoot* aplikacije su aplikacije nastale primenom *SpringBoot* okvira za razvoj *Java* baziranih aplikacija. *SpringBoot* okvir sadrži sveobuhvatnu infrastrukturnu podršku za razvoj mikroservisa [2]. *SpringBoot* aplikacije putem aktuator krajnjih tačaka pružaju informacije koje su neophodne za rad *Spring Boot Admin* projekta. Aktuator krajnje tačke proširuju funkciju aplikacija izlaganjem određenih informacija o aplikaciji, koje je potom moguće upotrebiti za svrhe upravljanja aplikacijom. Informacije koje

aktuator krajnje tačke pružaju mogu se iskoristiti u nadgledanju aplikacije, prikupljanju metrika, razumevanju *HTTP* prometa ili stanja baze podataka. Metrike se na primer mogu odnositi na iskorišćenost memorije u *Java* virtuelnoj mašini, iskorišćenost procesora, broj trenutno otvorenih datoteka u aplikaciji i slično.

4. PODRŠKA ZA *SPRING CLOUD GATEWAY* KOMPONENTU

Gateway komponenta kontroliše tok spoljnog *HTTP* saobraćaja na ulasku u sistem. U narednim potpoglavljima predstavljeni su entiteti kojima se *gateway* konfiguriše, njihova uloga i načini za njihovo podešavanje. Na osnovu postavljenih vrednosti ovih entiteta *gateway* komponenta bazira svoje ponašanje. Predstavljena su i proširenja dodata u trenutnu implementaciju *Spring Boot Admin* projekta koja omogućavaju manipulaciju nad *gateway* komponentom.

Ova proširenja podrazumevaju stranicu u projektu *Spring Boot Admin*, koja nudi načine za pregled i manipulaciju konfiguracija *gateway* komponente. Ovo je prva implementacija ovog tipa u oblasti mikroservisa i *gateway* komponente, tako da u određenoj meri olakšava dosadašnji način rada.

4.1. Rute i filteri

Svaka *gateway* komponenta u svojoj konfiguraciji sadrži listu ruta. Lista ruta je presudna za donošenje odluke koji zahtev treba proslediti u sistem, a koji ne. Ruta se sastoji od imena za identifikaciju, predikata rute, filtera i redosleda izvršavanja. Predikat rute predstavlja vrednost u vidu regularnog izraza. Filteri rute predstavljaju validacije i transformacije *HTTP* zahteva koje je potrebno izvršiti prilikom obrade zahteva koji pristiže na *gateway* komponentu. Filteri mogu biti lokalni ili globalni. Lokalni filteri su vezani za definiciju rute. Globalni filteri su nezavisni od odabira rute i izvršavaju se nad svakim pristiglim zahtevom.

Kada zahtev od klijenta stigne do *gateway* komponente, u prvom koraku se nad zahtevom izvršava set definisanih globalnih filtera. Oni na primer mogu vršiti sigurnosne provere, kao što je provera da li zahtev u zaglavlju sadrži sve neophodne informacije ili token sa pravima.

Filteri na osnovu ovih i sličnih drugih validacija u određenim situacijama mogu transformisati zahtev, proslediti zahtev dalje na obradu ili vratiti odgovor sa upozorenjem ili greškom. U drugom koraku *gateway* komponenta nad prosleđenim zahtevom izvršava poređenje URL-a iz zahteva sa predikatima svake od ruta u definisanoj listi ruta.

Odabira se prva ruta čiji jedan predikat ispunjava uslov. Poredi se regularni izraz iz predikata rute sa vrednošću URL-a iz zaglavlja zahteva. Iz razloga što postoji ovakav vid poređenja, postoji i podatak o redosledu izvršavanja u rutu. Na osnovu ove vrednosti se utiče na prioritete među rutama.

Nakon što je ruta odabrana nad zahtevom se izvršava niz njenih definisanih lokalnih filtera. Lokalni filteri vrše transformacije nad vrednostima u zahtevu i pripremaju ga za dalje prosleđivanje i finalnu realizaciju. Nakon ovog koraka, ukoliko je sve prošlo uspešno, dati zahtev zaista stiže na odredište i izvršava željenu poslovnu logiku u nekoj od instanci servisa.

4.2. Dodavanje i brisanje definicije rute

S obzirom da definicije ruta predstavljaju vrlo dinamičan podatak, koji je sklon čestim izmenama, javila se potreba za implementacijom rešenja koje bi omogućilo da se ovaj resurs menja dok je instanca servisa aktivna. Da bi ovako nešto bilo moguće, *gateway* komponenta mora imati podršku za direktnu izmenu konfiguracija. Ova podrška je omogućena dodavanjem aktuator krajnjih tačaka.

U ovom slučaju one pružaju mogućnost dodavanja i brisanja trenutnih definicija ruta na instanci *gateway* servisa. Na ovaj način se izbegava potreba za eksternom izmenom konfiguracije i restartom instance servisa radi osvežavanja verzije. Dodavanje definicije rute je omogućeno unosom podataka u *JSON* formatu koji predstavlja definiciju jedne rute sa svim neophodnim poljima.

Nakon uspešno izvršene akcije dodavanja, lista sa rutama koja se prikazuje na stranici se automatski osvežava i prikazuje novododatu rutu. Brisanje se može izvršiti takođe u listi ruta. Rezultati obe akcije se gotovo istovremeno primenjuju i na samu komponentu i menjaju način njenog rada. Iz ovog razloga se eliminiše potreba za direktnim izmenama u programskom kodu servisa, kreiranjem nove verzije servisa i zaustavljanjem rada instance servisa radi osvežavanja trenutne verzije novom verzijom.

4.3. Poređenje performansi

Razlika koja se postiže uvođenjem podrške za *gateway* komponentu ogleda se u izmeni koraka koje je neophodno izvršiti da bi se određena promena konfiguracije primenila na instanci *gateway* servisa. Uz mikroservisnu arhitekturu obično se primenjuje neka od modernih metodologija za razvoj i isporuku softverskih rešenja. Ovakve metodologije nalažu da postoji jasno definisana sekvenca procesa koje je neophodno izvršiti da bi se neka promena propagirala na radno okruženje. Pod radnim okruženjem podrazumeva se funkcionalan mikroservisni ekosistem koji je pokrenut na serverima.

Sekvenca procesa koju je potrebno izvršiti zavisi od projekta do projekta. U idealnom slučaju bilo bi dobro u potpunosti zaobići izvršavanje ovih procesa, što implementacija podrške za *gateway* komponentu u vidu stranice u *Spring Boot Admin* projektu i omogućava. U minimalnom skupu ovi procesi podrazumevaju kompajliranje i izgradnju datoteka sa programskim kodom nad kojim se vrše izmene, zatim pokretanje skupa testova nad uspešno izgrađenim programskim kodom, postavljanje nove verzije servisa na repozitorijum sa verzijama i finalno proces zaustavljanja rada trenutne verzije instance servisa na radnom okruženju i postavljanje i pokretanje nove. Svaki od ovih procesa ima određeno trajanje koje zavisi od obimnosti servisa, jačine hardvera na kom se izvršavaju procesi i veličine i kompleksnosti skupa testova. Skup testova može podrazumevati *unit* testove, integracione testove, *end-to-end* testove i druge [3].

Za izvršavanje testova neophodno je imati osposobljen i pokrenut čitav kontekst aplikacije. Zbog toga se česte izmene u konfiguraciji ne isplate s obzirom na vreme koje je potrebno da se primene i vreme koje prođe, a da blokiraju i stvaraju zavisnost u radu sistema.

Zavisnost se stvara tako što se privremeno zaustavlja rad trenutne instance *gateway* servisa, radi osvežavanja

novom. Testiranje u slučaju ovakvih promena nema smisla, jer se u testovima pokriva funkcionalni deo aplikacije, a konfiguracije se svakako izostavljaju ili se koristi specifičan set testnih konfiguracija. Za potrebe izvršavanja velikog dela procesa uključenih u proceduru izmene konfiguracija na gateway komponenti pre implementacije podrške, korišten je alat za automatizaciju *Jenkins*. *Jenkins* je alat koji pomaže u automatizaciji procesa koji se izvršavaju tokom razvoja softvera.

Ovi procesi podrazumevaju izgradnju projekta, testiranje, *deploy* i *release* proces. *Deploy* proces podrazumeva zaustavljanje rada trenutne instance servisa na radnom okruženju i pokretanje nove instance sa novom verzijom. *Release* proces podrazumeva kreiranje nove izvršive verzije servisa i njeno postavljanje na repozitorijum sa verzijama. U nekim slučajevima ovi procesi se izvršavaju manuelno, ali za potrebe mikroservisne arhitekture u kojoj je izvršavanje ovih procesa dosta učestalo, neophodna je primena nekog od alata za automatizaciju. *Jenkins* je pokrenut na zakupljenoj AWS (*Amazon Web Services*) instanci servera. Tip AWS instance servera je *c4.2xlarge*, tip procesora *2.9 GHz Intel Xeon E5-2666 v3*, sadrži *8 CPU-a* i *15 GB RAM* memorije. Gateway servis zauzima *68.9 MB*, sadrži *5* integracionih i *158* unit testova.

Za izmenu konfiguracije rute i propagiranje promene na radno okruženje pre implementacije podrške za gateway komponentu, izvršeni su sledeći koraci: izmena datoteke sa konfiguracijama i postavljanje na git repozitorijum (*2* minute i *15* sekundi), izgradnja programskog koda projekta (*21* sekunda), vreme izvršavanja testova (*2* minute i *7* sekundi), kreiranje izvršive *jar* datoteke i postavljanje na repozitorijum sa verzijama (*40* sekundi), *deploy* (*1* minut i *5* sekundi), pokretanje servisa i registracija instance na *discovery* servis (*41* sekunda).

Ukupno trajanje procesa izmene konfiguracije rute je *7* minuta i *9* sekundi. Za izmenu konfiguracije rute i propagiranje promene na radno okruženje nakon implementacije podrške za gateway komponentu, izvršeni su sledeći koraci: otvaranje stranice sa podrškom za gateway komponentu (*10* sekundi), dodavanje nove rute kroz formu na stranici (*42* sekunde). Vreme potrebno da se izmeni konfiguracija rute i da se ta izmena zaista primeni na radnom okruženju je *52* sekunde. Postignuta je optimizacija što se tiče broja neophodnih koraka i vremena koje je potrebno da se izvrše zadaci prilikom izmene definicije rute na gateway servisu.

Nakon implementacije podrške zadatak je moguće izvršiti približno sedam puta brže nego ranije. Pored ove razlike takođe treba napomenuti da je u mikroservisnom ekosistemu svaki trenutak prestanka rada neke od bitnih instanci servisa zapravo blokirajući za čitavo radno okruženje. S obzirom da je u datom primeru reč o servisu koji u potpunosti kontroliše dolazni saobraćaj u sistemu, to znači da prestanak rada ove instance stvara i privremeni prestanak rada čitavog sistema. Iz tog razloga je korisna implementacija rešenja koje pruža način da se isti zadatak izvrši bez obustavljanja rada bilo koje instance.

5. ZAKLJUČAK

Na osnovu rezultata dobijenih prilikom testiranja performansi rešenja dolazi se do zaključka da stranica za upravljanje gateway komponentom može imati široku primenu u sistemima sa mikroservisnom arhitekturom.

Pružna načine za brži rad prilikom nadgledanja podataka i smanjuje broj koraka za izvršavanje određenih akcija. Olakšava način za sticanje uvida u trenutno stanje i definiciju globalnih filtera i omogućava pregled i modifikaciju definicija ruta.

Pored toga, modifikacije primenjene na definicijama ruta se propagiraju na instancu servisa istog trenutka, dok je instanca servisa i dalje pokrenuta. U potpunosti se gubi potreba za izmenom konfiguracija u programskom kodu, pravljenjem nove verzije servisa i zaustavljanjem rada trenutne instance radi osvežavanja novom verzijom. Proces modifikacije definicije rute je ubrzan za sedam puta, a da se pri tome nikad ne zaustavlja rad instance gateway servisa. Isto se odnosi i na akciju brisanja definicije rute, koja takođe ne zahteva osvežavanje verzije instance servisa i samim tim ne blokira izvršavanje čitavog mikroservisnog ekosistema. Novi pravci razvoja podrazumevaju unapređenje dela gateway stranice zaduženog za unos nove definicije rute. Način unosa definicije rute može ponuditi formu sa boljom strukturom, koja je robustna i predlaže korisnicima moguće podatke za unos.

Deo stranice sa prikazom globalnih filtera je moguće proširiti tako da omogućava i detaljniji pregled svake od definicija globalnih filtera. Na ovaj način bi se sprečila potreba za pretraživanjem programskog koda da bi se izvršio pregled određenog globalnog filtera. S obzirom da je upotrebom aktuator krajnjih tačaka moguće saznati informacije o trenutno dostupnim krajnjim tačkama na bilo kom od servisa, ovaj podatak bi se mogao iskoristiti u implementaciji novog proširenja koje bi omogućilo korisnicima da na još brži način dodaju novu definiciju rute. Sistem bi mogao na osnovu postojećih definicija ruta, trenutnog stanja sistema i podataka o krajnjim tačkama servisa koje prikupi upotrebom aktuatora da predvidi neke od mogućih novih definicija ruta i da ih predloži korisniku. Ovo proširenje bi moglo biti od velike koristi jer je sam proces unosa novih definicija ruta obično redundantan i podložan korisničkim greškama.

6. LITERATURA

- [1] Chris Richardson, *Microservices Patterns*, 1st edition, Manning Publications, 2018.
- [2] John Carnell, *Spring Microservices in Action*, 1st edition, Manning Publications, 2017
- [3] Dinesh Rajput, *Hands On Microservices – Monitoring and Testing*, Packt Publishing, 2018

Kratka biografija:



kontakt:
stkosijer@gmail.com

Stevan Kosijer rođen je 24.9.1994. godine u Vojniću, Hrvatska. Živi u Novom Sadu, završio je Osnovnu školu „Đorđe Natošević“ u Novom Sadu. Društveni smer gimnazije „Isidora Sekulić“ završio takođe u Novom Sadu. Godine 2013. upisuje Fakultet tehničkih nauka u Novom Sadu, na studijskom programu Računarstvo i automatika. Godine 2017. uspešno diplomira i nastavlja sa master studijama na istom fakultetu. Položio je sve ispite propisane planom i programom.

PACIJENT SIMULATOR – MAJČIN I FETALNI SIGNAL GENERATOR NEINVAZIVNIH ABDOMINALNO SNIMLJENIH EKG I EHG BIOPOTENCIJALA **PATIENT SIMULATOR – MATERNAL & FETAL SIGNAL GENERATOR OF NON- INVASIVE ABDOMINALLY RECORDED ECG & EHG BIOPOTENTIALS**

Nemanja Čabrilo, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad opisuje proces dizajniranja i razvoja višekanalnog generatora elektrofizioloških signala zasnovanog na LabVIEW programskom okruženju i merne kartice za akviziciju podataka (DAQmx). Posebna pažnja posvećena je rekonstrukciji realističnih neinvazivnih signala u smislu električne aktivnosti majčinog i fetalnog srca i mišića materice. Ideja je bila da se predstavi funkcionalni prototip koji može da generiše sintetički signal i da se prema zahtevima korisnika koriguje u realnom vremenu bez gubitaka u kvalitetu signala, a sve to u cilju da bi se pridao značaj razvoju novih dijagnostičkih metoda.

Ključne reči: DAQ, EKG, EMG, EHG, obrada signala, višekanalni simulator, Matlab, LabVIEW, analogna elektronika, merno-akvizicioni sistem

Abstract – This paper describes the design and development process of a multi-channel electrophysiological signals generator based on LabVIEW programming environment and its associated data acquisition measurement cards (DAQmx). Special attention is paid to the reconstruction of realistic non-invasive signals in terms of maternal and fetal cardiac and uterine electrical activity. Main concern was to build a functional prototype that is able to generate and module synthetical signal in real-time without any losses of the signal's quality, which ultimately can produce a contribution to the development of new diagnostic methods.

Keywords: DAQ module, signal processing, EMG, EHG, Artefacts, multichannel simulator, non-invasive ECG, LabVIEW, Matlab, real-time, data acquisition, mixed signals, systems engineering, analog electronics,

1. UVOD

Visoko rizične trudnoće su sve više preovladavajuće zbog progresivno veće starosne dobi u kojoj žene započinju trudnoću. Danas oko dvadeset posto svih trudnoća se vode kao komplikovane u određenoj meri. Npr, u vidu prevremenog porođaja, nedostatka kiseonika fetusa, ograničenosti rasta, hipertenzije, itd. Rana detekcija ovih komplikacija je od vitalnog značaja za blagovremene lekarske intervencije, ali je sputana od strane trenutnih tehnoloških ograničenja u monitoringu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Platon Sovilj, red. prof.

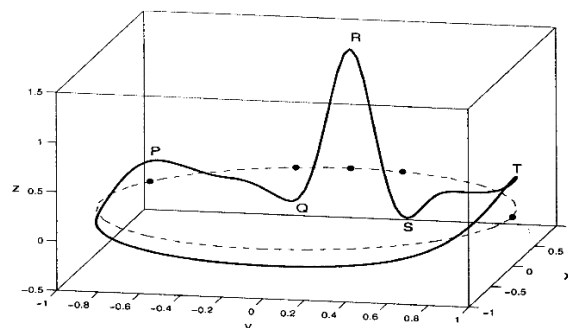
Najistaknutija metoda za praćenje zdravstvenog stanja fetusa je monitoring *heart rate variability*-a (HRV), u odnosu na odziv mišićnih aktivnosti uterusa (cardiotocography – CTG).

Ipak praktična primena fetalnog ECG-a nailazi na brojne nedostatke u akušerskoj dijagnostici, i trenutno teško može da parira još uvek pouzdanijim invazivnim metodama, pa se češće koriste kao njihov dodatak (CTG i ECG). Da bi se prevazišla ograničena primenljivost, različite metode obrade signala su testirane kako bi se pouzdano analizirao neinvazivni fECG sa abdomena trudnice, ali ovi pokušaji još uvek nisu doveli do ustanovljenih pristupa koji omogućavaju široku kliničku primenu.

Tokom eksperimentisanja sa ovakvim algoritmima, neophodno je imati odgovarajući signal pogodan za testiranje. Za ovu svrhu razvijen je prototip simulatora biofizioloških signala koji se mogu meriti najsavremenijim neinvazivnim metodama, i na taj način doprineti razvoju novih dijagnostičkih metoda.

Ovaj model se može lako prilagoditi tehničkim zahtevima mernih uređaja. Moguće je i dodavanje smetnji ECG-a (biološke i tehničke artefakte), kao i određenih osobina fECG signala. Treba naglasiti da je krajnji uređaj demonstrator, odnosno dokaz koncepta, koji je razvijen za interne potrebe kompanije *Nemo Healthcare*, stoga nije namenjen za tržišnu proizvodnju i neće biti komercijalno dostupan.

2. MORFOLOGIJA ECG-a



Slika 1. Tipična trajektorija koju generiše dinamički model u 3-D ravni, data sa (x, y, z) .

Dužina trajanja jednog PQRST kompleksa ~ 200 ms
Visina naponskog raspona (V_{pp}) ~ 1 mV

Svaki otkucaj srca može se posmatrati kao niz izvijanja signala dalje od osnovne linije na ECG-u. Ovi otkloni odražavaju vremensku evoluciju električne aktivnosti u

srcu koja inicira mišićne kontrakcije srca. Jedan sinusni (normalni) ciklus ECG-a, predstavlja sukcesivnu atrijalnu depolarizaciju/repolarizaciju i ventrikularnu depolarizaciju /repolarizaciju koja se javlja sa svakim otkucajem srca. Tradicionalno je označen sa slovima P, Q, R, S i T (*Willem Einthoven 1893, Slika. 1*).

3. MODEL

Realni snimci ECG-a uvek su kontaminirani artefaktima; stoga osim modeliranja srca kao izvora i međujima kojim se propagira signal, vrlo je važno imati i realne modele za različite izvore šuma. Ovi šumovi su tipično veoma nestacionarni u vremenu i spektralno prošarani (imaju dugoročne korelacije). Najčešći oblici šuma su: udaljavanje od osnovne linije signala, mišićni artefakti, pomeranje elektroda. Za fetalni ECG sa majčinog abdomena takođe se dotiraju i: majčin ECG, kretanje fetusa tokom snimanja, kontrakcije uterusa, promenljiva otpornost majčinskog volumena provodljivosti zbog *vernix caseosa* sloja oko fetusa.

Već tokom ranih faza razvoja ovog projekta napisan je model simulatora u Matlab programskom okruženju, na osnovu koje je koncipiran dalji razvoj „Pacijent simulatora”. Ovu Matlab skriptu je sastavio profesor sa TU/e-a i jedan od osnivača gore pomenute firme iz Eindhovena *dr. Rik Vullings*.

Rešenje se oslanja na kombinaciju najsavremenijih dostignuća u metodama modelovanja biosignala zajedno sa već prikupljenim snimljenim podacima stvarnih pacijenata iz bolnice *Máxima Medisch Centrum*. Njegov istraživački radovi umnogome su doprineli su razvoju novih dijagnostičkih metoda merenja fECG-a, uz pomoć modelovanja vektora jediničnog električnog polja stacionarnog porekla, opisanih kao *vectorcardiogram* (VCG).

Abdominalni ECG predstavlja projekciju VCG-a na vektor koji opisuje poziciju merne elektrode u odnosu na referentnu elektrodu, što omogućava da se bilo koji ECG signal izračunava poznavajući VCG. Slično važi i obrnuto kada su barem tri nezavisna ECG-a poznata, čime se postupkom može izbeći pojava viška šuma u signalu. Osim opisivanja električne aktivnosti u tri dimenzije, fVCG predstavlja i okvir za računanje fetalne orijentacije u materici. Izvorni kod u Matlabu predstavlja intelektualnu svojinu, i u interesu firme je da se sadržaj iste ne objavljuje u javnosti, zbog čega neće biti detaljno opisan.

4. IMPLEMENTACIJA

4.1. Opis sistema

Predviđen je razvoj sistema koji se sastoji od nekoliko nezavisnih celina: Definisanje tehničkih i sistemskih specifikacija, odabir tehnologije za generisanje i reprodukciju analognog signala uzimajući u obzir budžetska, vremenska i druga ograničenja (prototipovanje); razvoj algoritma, softverska implementacija, dizajn grafičko-korisničkog interfejsa (GUI) na front panelu, skaliranje analognog signala na opseg realnih (očekivanih) vrednosti napona, optimizacija, integracija, testiranje i verifikacija. Zbog specifičnosti okvira rada, validacija simulatora ne ulazi u obim ovog rada, ali će se naknadno izvršiti.

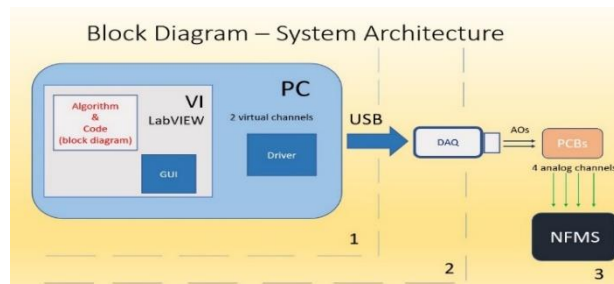
4.2. Tehničke i sistemske specifikacije

Specifikacije su bazirane na prikupljenim informacijama krajnjih korisnika inženjera i istraživačkog kadra Nemo Healthcare firme, praćenjem publikacija od interesa, kao i prostom analizom nedostataka postojećih simulatora na trenutnom tržištu. Za neke od osnovnih tehničkih zahteva ustanovljeno je da simulator treba da poseduje unapređene sledeće funkcije: simulacija aktivnosti uterus mišića, ECG signal različitih oblika, prilagođavanje vrednosti napona prema NFMS-u (*Nemo Fetal Monitoring System*), dinamična promena srčanog ritma majčinog i fetalnog ECG-a, širi opseg srčanog ritma, multikanalno generisanje signala, itd.

4.3. Izbor tehnologije

Prvobitno je bilo u planu da se simulator implementira uz pomoć eksterne zvučne kartice (*Focusrite Scarlett 2i4*), što se ubrzo pokazalo kao komplikovanije rešenje u tom momentu. Usled ograničenosti resursa, izbor je vrlo brzo spao na hardverskom rešenju *USB DAQ 6002* modula razvijen od strane kompanije *National Instruments*, sa kojim je prvo ozbiljnije prototipovanje započeto. Zbog visoke kompatibilnosti ovog uređaja sa programom za razvoj virtuelnog instrumenta *LabVIEW*, proizveden iz iste firme kao i *DAQmx*, odlučili smo se da faza implementacije bude zasnovana na ovom programskom okruženju.

4.4. Sistem arhitekture



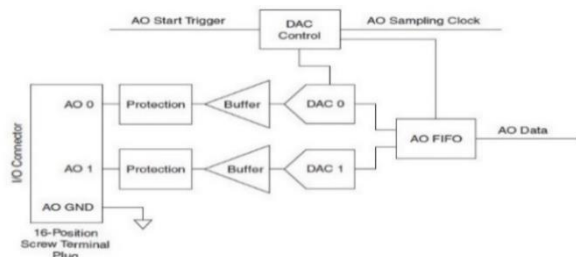
Slika 2. Blok šema sistem arhitekture

4.5. Hardver

Hardverski deo čini prvobitno personalni računar, idealno u baterijskom režimu rada, preko kojeg se upravlja simulacijom. Zatim se signal šalje u digitalnom formatu preko USB protokola ka DAQ uređaju gde se ujedno vrši digitalno-analogna konverzija. Potom se generiše signal na analognim izlazima uređaja. Praktičnost ovog uređaja ogleda se u tome što može da vrši akviziciju u simultanom procesu, bez potrebe za dodatnim uređajima. Jednostavno se prespoje izlazni sa ulaznim kanalima, gde se vrši A/D konverzija i prosleđuju dalje podaci nazad na računar posredstvom FIFO integrisanog baferskog bloka. Na taj način može se vršiti samo-provera generisanog signala i u okviru iste aplikacije (*LabVIEW*, *Python*, *C++*).

Jedan od glavnih nedostataka ovog DAQ modula je nemogućnost reprodukcije više od dva analogna kanala, pa je samim tim i generisanje biosignala bilo ograničeno na simulaciju samo dva fizička kanala preko ovog uređaja. Za uspešnu reprodukciju četiri kanala, koristio

sam dodatnu analognu elektroniku, kao i muzičku karticu računara.



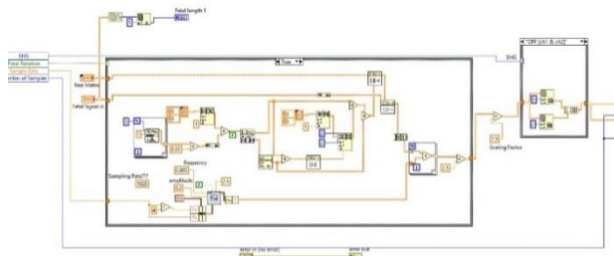
Slika 3. Analogno izlazno kolo - NI USB DAQ 6002

Smanjivanje napona do realnih vrednosti amplituda napona postignuto je povezivanjem atenuatorskog bloka na izlazne kanale DAQ uređaja. Atenuator je sačinjen od aktivnih i pasivnih komponenti analogne elektronike, koristeći prosti naponski razdelnik sa odgovarajućim odnosom vrednosti otpornika, kao i operacionim pojačavačem (op270G) sa odgovarajućim karakteristikama (npr. *Input offset voltage* tipično od oko $75 \mu\text{V}$). Ovaj opAmp zajedno sa otpornicima u kolu neinvertujućeg pojačavača uveden je zbog pojačavanja vrednosti napona (nakon smanjivanja) na dovoljno visoki naponski opseg kako bi merni instrument poput osciloskopa mogli to da izmere i prikažu. Na taj način sam uspeo da verifikujem prethodni korak smanjivanja napona na red μV .

4.6. Softverski razvoj

Za razvoj ovog simulatora korišćeni su razni programski alati. Od posebnog značaja je i svakako najčešće korišćen LabVIEW u kojem je razvijan algoritam modela simulatora i implementiran izvršni kod za upravljanje signalima preko korisničkog interfejsa. U samoj aplikaciji su sprovedena i dodatna, ali ne i neophodna svojstva, poput monitoring dial box-a, error handling, merna akvizicija podataka, mogućnost reprodukovanja i slušanja ECG-a u audio formatu preko integrisane zvučne kartice računara, itd. U LabVIEW programskom paketu korišćeni su još i posebni dodaci (biblioteke) kao npr. blokovi za akviziciju i kontrolu podataka i za obradu audio signala.

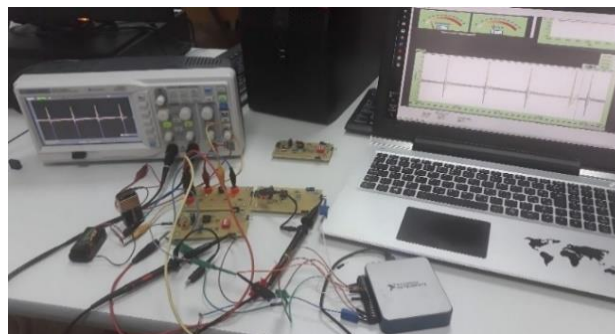
Hijerarhijska organizacija izvršnog koda je implementirana koristeći neke od standardnih postavki za organizaciju blok dijagrama. Master/Slave (producer/consumer) postavka omogućila je, pored efektivnije strukture koda, jasnu sledivost izvršavanja različitih sekvenci koda, tako što glavna petlja (*state machine*) komunicira sa ostalim sekvencama i upravlja redosled njihovih izvršavanja, koristeći operacije redova čekanja (*queue operations*).



Slika 4. Primer jedne petlje iz Blok Dijagrama

U programu sam koristio funkcionalne globalne varijable (FGV) za komuniciranje između dve ili više paralelnih procesnih petlji sadržane unutar različitih virtualnih

instrumenata (VI-a). "Funkcionalna" priroda FGV znači da možete stvoriti dodatnu funkcionalnost izvan one osnovne globalne varijable, što mi je omogućilo upisivanje u, i čitanje iz iste globalne promenljive na različitim mestima u toku izvršavanja jednog ili više VI-a.



Slika 5. Laboratorijska postavka za testiranje Sistema (Osciloskop, atenuator, DAQmx, Front panel VI-a, PC)

Na taj način sam mogao da definišem mECG i fECG signale kao posebne funkcionalne globalne promenljive koje se izvršavaju u paralelnim procesima, ne narušavajući kondicijale ostalih zahteva izvršnog programa i rada samog računara.

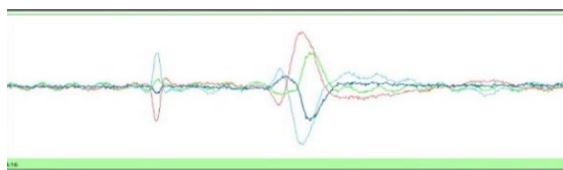
Za kraj je kreirana .exe datoteka koja otvara i pokreće izvršnu aplikaciju na svakoj platformi koristeći samo LabVIEW Run-Time Engine, bez dalje potrebe za instaliranjem čitavog LabVIEW softvera sa paketima.

4.7. majčin i fetalni ECG (mECG i fECG)

Podaci realnih snimaka pacijenta sačuvanih u MAT fajlu su prvo konvertovani u odgovarajući format, a zatim se kao takvi učitavaju u program i koriste za dalju manipulaciju. Podaci sadrže informacije o majčinom i fetalnom VCG-u, koje predstavljaju diskretizovane vrednosti amplituda napona u vremenskom domenu, koje su odabrane frekvencijom od 1000 Hz. Da bi se abdominalni signal prilagodio postavci elektroda sa NFMS mernim instrumentom, nad semplovanim odbircima primenjuje se odgovarajuća linearna transformacija (trokanalno merenje sa pravog pacijenta). Kada se vrednosti signala postave na odgovarajuće pozicije u prostoru manipulacijom matricama, primenjuju se prozorske funkcije (*Hanning i Tukey*) kako bi se ograničila dužina osnovne arkade PQRS kompleksa, ali i pogodno „zašiljili” krajevi segmenta signala koji će se reprodukovati tokom generacije.

U zavisnosti od zadatog srčanog ritma, automatski se dodaje jednoznačno određen broj odbiraka (*zero padding*) između susednih segmenata PQRS kompleksa. Na dodatim odbircima primenjuje se linearna interpolacija kako bi tranzicija između dva otkucaja srca imala verodostojniju reprezentaciju realnog signala. Ovde je implementiran algoritam za HRV koristeći raspodele Gausovog belog šuma, kojim varira dužinu trajanja između susednih otkucaja srca prema specifičnim zakonitostima. Odstupanje od osnovne trajektorije (*baseline wander*) po kojoj se signal pomiče uz pomoć cikličnog bafera, je simuliran kao artefakt niske učestanosti, koji se povećava usled disanja, pojave naelektrisanja elektroda, ili pokreta pacijenta tokom snimanja. Amplitudska modulacija je

uvedena sa uniformnom raspodelom kako oblici signala PQRST kompleksa ne bi bili statične prirode. Fetalna pomeranja u materici tokom merenja su simulirani preko faktorizacija singularnih vrednosti dekompozicije, gde rotiranje matrica poseduje određenu nivo inercije.



Slika 6. Četvorokanalni mECG i fECG signal

4.8. EHG

Kontrakcije uterus mišića su prouzrokovane akcionim potencijalima koji se propagiraju duž *myometrium* ćelija. Elektrohisterografija je mera električne aktivnosti uterusu. EHG se mogu simulirati kao naleti električnih impulsa u rasponu od 0,3 Hz do 0,8 Hz koji se pojavljuju i nestaju sa kontrakcijama.



Slika 7. Dvokanalno merenje ECG i EHG signala

Tranzicija između kontrakcija traje u napred zadatim vremenskim intervalim. Uobičajeno je da trajanje kontrakcije pre porođaja bude oko 20 sekundi, a pauza oko 40 sekundi. Takav signal je dobijen slaganjem fazno pomerenih 500 sinusnih funkcija različitih frekvencijskih vrednosti u navedenom opsegu, sa inkrementom od po 0,01 Hz, na koje je primenjena prozorska funkcija.

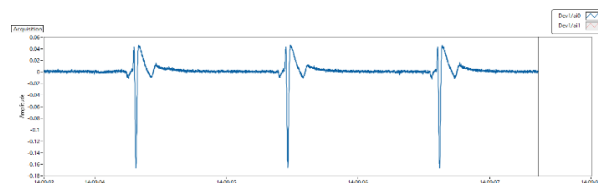
4.9. Artefakti

Da bi se modelirani signal učinio realnijim, moguće je dodati beli šum u rezultujući signal simulirajući šum instrumentacije. Takođe, može se uljučiti i signal koji simulira šum iz mreže, kao male vrednosti sinusoide frekvencija 50 Hz ili 60 Hz. Disanje prouzrokuje da osnovna izoelektrična linija oko koje se kreće ECG signal počne relativno sporo-periodično da menja svoj položaj. Jednostavan način modeliranja ovog efekta je dodati sinusni oblik talasa signala respiracije, koji može varirati od oko 0,2 Hz do 0,5 Hz (otprilike 12 do 30 udisaja u minuti).

5. VERIFIKACIJA

Za potrebe testiranja signala, koristio sam postavku merenja sa diferencijalnim povezivanje kanala. Ovakva postavka smanjuje prikupljanje šuma i povećava potiskivanje zajedničkog napona šuma kao i razlike potencijala između izvora signala i uzemljenja ako se personalni računar priključuje preko napajanja na kućnu utičnicu koja je uzemljena odgovarajućim provodnikom.

Korišćena test instrumentacija: Osciloskop; DAQ kartica; Texas Instruments ADS1299 modul za evaluaciju; Generator funkcije; Sistem zvučnih kartica sa DAW softver alatima; Ostala laboratorijska oprema.



Slika 8. Jednokanalno merenje mECG-a. Vpp smanjen na 20 μ V zatim pojačan na 200 mV i snimljen kao takav.

6. ZAKLJUČAK

Predloženi pacijent simulator modelovan je pomoću LabView softvera koji reprodukuje istaknute karakteristike fetalnog ECG-a. Ovakva rekonstrukcija realističnog ECG-a pruža standard za testiranje različitih tehnika obrade biomedicinskih signala. Učinkovitost prototipa verifikovana je i dokumentovana. S obzirom na opštost predloženog simulatora, moguće su brojne primene i jednostavna proširenja ovog simulatora. Neka od hardverskih unapređenja: Proširivanje broja simuliranih elektroda (kanala); Integracija razvijenog LabVIEW VI-a na *stand-alone* uređaj. Softverska unapređenja: Morfološka varijacija PQRST kompleksa ECG-a; simulacija abnormalnih srčanih otkucaja; Dodatne varijabilnosti izoelektrične linije.

7. LITERATURA

- [1] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith, "A dynamical model for generating synthetic electrocardiogram signals," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 3, pp. 289–294, 2003.
- [2] Vullings, R. (2010). "Non-invasive fetal electrocardiogram: analysis and interpretation" Eindhoven: Technische Universiteit Eindhoven DOI: 10.6100/IR692881

Kratka biografija:



Nemanja Čabrilo rođen je u Zrenjaninu 1991. godine. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu, na Katedri za električna merenja. 2016. godine. Interesovanje – biomedicinsko inženjerstvo. kontakt: nemanjac@outlook.com

SISTEM ZA AUTOMATSKO OTVARANJE I ZATVARANJE AUTOMOBILSKIH VRATA SA DETEKCIJOM PREPREKE I MERENJEM UDALJENOSTI DO PREPREKE**AUTOMATIC DOOR OPENING AND CLOSING SYSTEM WITH OBSTACLE DETECTION AND DISTANCE MEASUREMENT**Jovan Janjušević, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je prikazan sistem za automatsko otvaranje i zatvaranje automobilskih vrata sa detekcijom prepreke i merenjem udaljenosti do prepreke, opisane su komponente i CAN komunikacija koja se koristi.

Ključne reči: CAN komunikacija, ESP32, MCP2515, HC-SR04, ULN2003, Step motor, Arduino.

Abstract - The paper presents system for automatic opening and closing of car doors with obstacle detection and measuring distance to obstacle prevents the possibility of an car accident. The components and CAN communication used are described.

Key words: CAN communication, ESP32, MCP2515, HC-SR04, ULN2003, Step motor, Arduino.

1. UVOD

Zbog povećanog broja vozila na putevima, kako u svetu tako i kod nas javlja se problem nedostatka parking mesta i iz tog razloga sama parking mesta su sve uža, a vozila zbog sigurnosti sve šira.

Pomoću ovog sistema moguće je otvoriti vrata maksimalno kako bi prostor za izlaz i ulaz u vozilo bio što veći, a da pri tome ne dođe do oštećenja jednog ili drugog vozila.

U radu će biti detaljno opisana realizacija ovog sistema i moguća buduća unapređenja. Detaljno su opisane hardverske komponente i softverski deo projekta.

2. STRUKTURA I KOMPONENTE**2.1 Struktura i komponente**

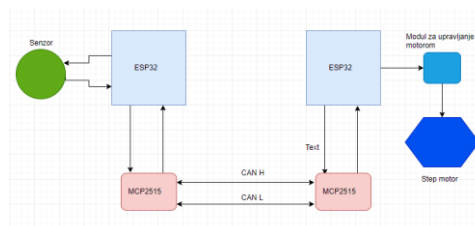
Senzor šalje podatke o udaljenosti prepreke na ESP koji ih obrađuje i prosleđuje poruku MCP-u (CAN modulu) koji tu poruku stavlja na CAN magistralu i započinje komunikaciju sa drugim CAN modulom.

Drugi ESP prima tu poruku i određuje maksimalni ugao pod kojim se vrata mogu bezbedno otvoriti.

Taj podatak dalje prosleđuje modulu za upravljanje motorom koji šalje signal step motoru i otvara vrata do zadatog ugla.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Platon Sovilj, red. prof.



Slika 2.1 –Blok šema celog sistema

2.2 ESP32

Za realizaciju ovog rada korišćena su dva mikrokontrolera ESP32 sa ugrađenom WiFi i dual Bluetooth konekcijom. Sam mikrokontroler ima dva jezgra i velike performanse pa se samim tim može koristiti za razne aplikacije, svako jezgro se može posebno kontrolisati.

Srce mikrokontrolera je čip ESP-D0WDQ6, klock se može podešavati na frekvenciju od 80Hz do 240Hz. Sam čip se napaja sa 3,3V DC, a na celu pločicu se može dovesti napon od 5V DC i sama potrošnja struje u je vrlo mala i u stanju mirovanja iznosi 5μA. Radna struja oko 80mA. SRAM memorija iznosi 512KB a flash memorija 16MB.



Slika 2.2 – ESP32

2.3 MCP2515-Arduino CAN Bus modul

Mikrokontroleri ESP32, korišćeni u izradi ovog sistema nemaju u sebi ugrađen modul za CAN Bus komunikaciju. Da bi se realizovala CAN komunikacija između mikrokontrolera bilo je potrebno ubaciti eksterne CAN module.

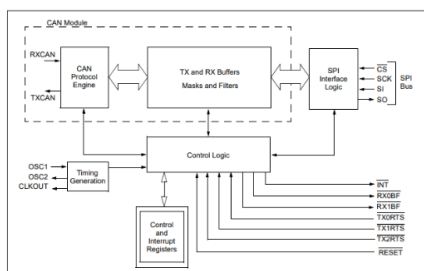
Arduino CAN Bus modul baziran je na MCP2515 CAN kontroleru i TJA1050 transiveru. Komunikacija između ESP-a i CAN modula je SPI. Verzija sa kojom radi ova pločica je CAN V2.0B, maksimalna brzina prenosa podataka iznosi 1Mb/s u ovom sistemu brzina je podešena na 500kb/s. Preporučena temperature na kojoj moduo normalno funkcioniše iznosi od -40 °C do +80 °C.



Slika 2.3-MCP 2515 CAN modul

2.3.1 MCP2515 čip

Na slici 2.5 vidimo unutrašnje blokove čipa 2515. Sam čip se sastoji od tri glavna bloka: 1. CAN modula (CAN protokoli, baferi za primanje i slanje poruka, filteri i maske); 2. Kontrolno-logički blok sa registrima; 3. Blok za SPI protokol.

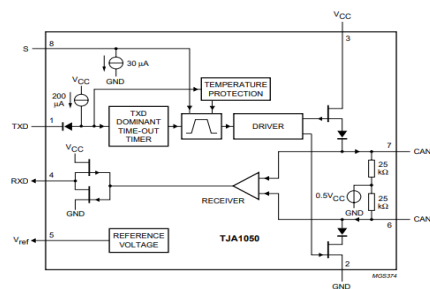


Slika 2.5-Blok šema MCP2515 čipa

1. CAN modul- blok izvršava funkcije vezane za primanje i slanje poruka preko CAN magistrale. Početak prenošenja poruke počinje iščitavanjem odgovarajućeg bafera i kontrolnog registra. Preko SPI-a se zadaje komanda za čitanje ili slanje poruke preko CAN magistrale, svaka poruka se proverava i propušta kroz filter da bi nakon toga bila svrstana u jedan od prijemnih bafera. MCP2515 sadrži dva prijemna bafera i tri bafera za slanje.
2. Kontrolno – logički blok kontroliše rad čipa i izvršava odgovarajuća podešavanja, INT pin se koristi za indicaciju kada je primljena validna poruka i upisana u bafer, proverava se može još i izvršiti preko statusnih registara kojima se pristupa preko SPI interfejsa.
3. Komunikacija sa mikokontrolerom je realizovana preko SPI interfejsa

2.3.2 TJA1050 čip

TJA1050 je interfejs između CAN-a i fizičke magistrale (Bus). Omogućava diferencijalne mogućnosti za slanje na magistralu i diferencijalne sposobnosti za primanje CAN modula. Namenjen je za aplikacije velike brzine u automotive industriji. TJA1050 je treći čip velike brzine za CAN komunikaciju fabrike Philips, prethodna dva modela su bila: PCA82C250 i PCA82C251. U odnosu na svoje prethodnike ima mnogo manju elektromagnetnu emisiju što doprinosi boljoj usklađenosti na CAN-H i CAN-L, bolje ponašanje u slučaju prekida napajanja i ne postoji režim čekanja. Napon napajanja čipa je od 4,75V do 5,25V.



Slika 2.6-Unutrašnjost TJA1050 čipa

2.3 Ultrazvučni sensor HC-SR04

Na slici 2.7 prikazan je fizički izgled senzora HC-SR04.



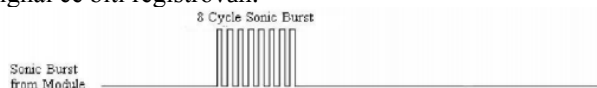
Slika 2.7-Izgled ultrazvučnog senzora HC-SR04

To je senzor za bezkontaktno merenje rastojanja. Odlikuje ga velika preciznost do $\pm 3\text{mm}$ uz vrlo malu cenu, mogućnost merenja rastojanja od 2cm do 4m. Senzor se napaja sa 5V DC. Sa mikrokontrolera dovodimo povorku pravugaonih signala minimalnog trajanja $10\mu\text{s}$. Na mikrokontroleru softverski generišemo taj signal tako što izlazni pin stavimo u logičko stanje „1“ u trajanju od $10\mu\text{s}$ i nakon toga ga vraćamo u logičko stanje „0“. Signal je prikazan na slici 2.8.



Slika 2.8- Izgled pobudnog signala koji se dovodi na pin „Trig“

Nakon primanja pobudnog signala sa mikrokontrolera, čip koji se nalazi na samoj pločici senzora generiše i emituje povorku od 8 četvrtki frekvencije 40kHz, signal prikazan na slici 2.9. Nakon toga na prijemniku očekuje signal koji se odbio od prepreke i vratio se prema senzoru. Ukoliko je rastojanje između senzora i prepreke manje od 4m signal će biti registrovan.

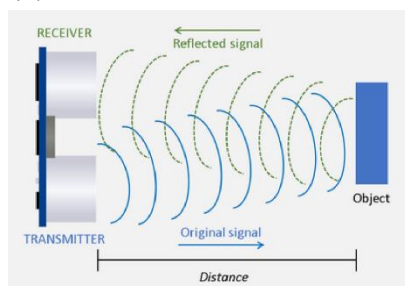


Slika 2.9-Izgled signala koji se generiše na samom senzorskom modulu

Kada se signal registruje, na pinu „Echo“ se javlja četvrtka čije je vreme trajanja proporcionalno udaljenosti prepreke od senzora. Na slici 2.10 prikazan je izgled signala.

Slika 2.10- Izgled signala koji se javlja na pinu „Echo”

Pročitani signal, sa ulaznog pina mikrokontrolera koji je povezan sa izlaznim pinom „Echo” sa senzora, koristimo da izračunamo udaljenost prepreke, što je prikazano u jednačini 1.1.



$$\text{udaljenost} = \text{trajanje} * 0.17$$

Jednačina 1.1- Izračunavanje udaljenosti na osnovu trajanja signala

2.5.1 ULN2003 modul za upravljanje step motorom

Struja koju mikrokontroler može da obezbedi na svom izlazu nije dovoljna kako bi se napajali potrošači sa većom potrošnjom struje. Iz tog razloga se koriste moduli za upravljanje kao što je ULN2003 (prikazan na slici 2.13) koji u sebi sadrži sedam NPN tranzistorskih Darlington parova (drugi tranzistor povećava izlaznu struju prvog tranzistora), koji mu omogućavaju da na izlazu ostvari veći napon i struju u odnosu na ulaz.



Slika 2.13-Fizički izgled modula ULN2003

Ukoliko imamo potrebu da nam je izlazna struja veća od 80mA i napon veći od 5V moramo kao spregu između mikrokontrolera ubaciti ULN2003. Izlazna struja po svakom pinu ULN-a iznosi 500mA, a u piku 600mA, paralelnim vezivanjem izlaza možemo dobiti veću struju. Sam moduo se povezuje preko ulaza: IN1; IN2; IN3; IN4 za sam mikrokontroler, Preporučljivo je dovesti napajanje na Vcc od 5V sa eksternog napajanja, a ne sa pina mikrokontrolera kako se on ne bi oštetio usled povlačenja veće struje za pogonjenje step motora. Na samo pločici se nalaze četiri indikatorske diode pomoću kojih možemo videti koji namotaj motora je u datom trenutku uključen.

2.5.2 Step motor 28BYJ-48

Step motor 28BYJ-48(slika 2.14) jedan od najčešće korišćenih step motora, koristi se od DVD plejera do podešavanja položaja ogledala u automobilu. Motor se sastoji od četiri namotaja, ugao svakog koraka iznosi 5,625 °, što znači da mu je potrebno 64 koraka da napravi pun krug, zbog prenosa od 64:1, kada se upravlja polu-

koračnom tehnikom maksimalan broj koraka će biti oko 4076, što znači da je izlaznom vratilu iz kućišta motora potrebno 4076 koraka kako bi se načinio pun krug. Kada motorom upravljamo sa puno-koračnom tehnikom kroz dva namotaja od ukupno četiri su u istom trenutku teče struja, kada se upravlja polu-koračnom tehnikom, prvo se pali prvi namotaj pa zatim drugi i tako redom. Motor se napaja sa 5V DC. Prednost step motora u odnosu na klasične DC motore je u tome što su: mnogo precizniji u pozicioniranju, imaju veći obrtni moment na malim obrtajima, lako se upravlja njima i jeftini su. Obrtni moment ovog motora iznosi 34 mN/m.



Slika 2.14-Step motor 28BYJ-48

3. CAN BUS KOMUNIKACIJA

CAN (Controll Area Network) sistem je proizvod kompanije Robert Bosch GmbH, Sa razvojem sistema započeto je 1983-će godine, dok je kao gotov proizvod plasiran na tržište 1986-te godine. CAN je serijski komunikacijski sistem osnovan za potrebe automotiv industrije. Zbog sve većeg broja elektronskih komponenti u vozilu, sistema komfora, sistema aktivne i pasivne zaštite količina same instalacije u vozilu dostigla je svoj maksimum i javila se potreba za uvođenjem magistrale na koju će biti povezani svi sistemi i preko koje će moći međusobno da komuniciraju (slika 3.1).

CAN je dvožični komunikacioni sistem sa priključcima za otpornike na svakom kraju komunikacione linije. Obezbeđuje centralizovanu kontrolu svih sistema povezanih na magistralu, brzina prenosa podataka je ograničena na 1Mb/s. Kada neki sistem želi da pristupi magistrali, izvršava proveru da li je magistrala slobodna i tek tada šalje paket, ako se desi da više sistema pristupa magistrali u isto vreme, pristup će biti prvo dozvoljen sistemu sa najvišim prioritetom. CAN sistem ima ugrađen sistem za detekciju greške i mehanizam za oporavak. Prijemni čvorovi proveravaju integritet poruka tako što pregledaju CRC polja. Ako se detektuje greška, ostali čvorovi na liniji se obaveštavaju porukama o grešci.

4. SISTEM

4.1 Hardver

Na slici 4.1 vidimo fizičku implementaciju samog sistema koji je tema rada. Ceo sistem je fiksiran na plastičnu ploču, komponente su povezane pomoću žica koje su fiksirane postupkom lemljenja na pinove. Sistem se može napajati na dva načina preko mikro USB portova na mikrokontrolerima ili preko USB A kabla koji je zalemljen odole na naponski čvor i masu celog sistema. Pritiskom na crveni taster šalje komandu za otvaranje vrata, a pritiskom na crni taster šalje komandu za zatvaranje vrata. Da bi oscilacije opruge u tasteru bile otklonjene na svaki taster vezano je kolo za debouncing kao na slici 4.2. Iz razloga što step motor nema svoj početni položaj, odnosno za početni položaj uzima onaj

položaj u kom se našao pre isključenja napajanja, dodat je mikro taster u sistem koji kada je pritisnut, motor se nalazi u početnom položaju. Kada se dovede napon napajanja, motor se automatski rotira dok ne naiđe na mikro taster.



Slika 4.1- Sistem za automatsko otvaranje i zatvaranje vrata

4.2 Softver

Kod koji je spušten na prvi ESP zadužen je za primanje signala sa senzora HC-SR04, na osnovu primljenog signala izračunava udaljenost i stavlja je u paket zajedno sa podacima sa tastera koji definišu da li je data komanda za otvaranje ili za zatvaranje vrata.

Funkcija koda, koji je spušten na drugi ESP, jeste da prihvati poruku sa CAN magistrale, raspakuje paket i na osnovu podataka odredi ugao za otvaranje vrata i prosledi zadati smer.

5. ARDUINO

Arduino je zajednički naziv za familiju mikrokontrolerskih platformi, namenjenih razvoju aplikacija. Razvoj softvera je omogućen korišćenjem Open-source integrisanog razvojnog okruženja. Pisan je u programskom jeziku Java i postoji u varijantama za operativni sistem Windows, Linux i MAC OS X.

Programski jezik Arduino aplikacija koristi osnovne elemente jezika C i C++.

6. ZAKLJUČAK

U ovom radu detaljno je objašnjen sistem za automatsko otvaranje i zatvaranje automobilskih vrata sa detekcijom prepreke i merenjem udaljenosti do prepreke, sve komponente koje su korišćene za izradu rada, kao i programsko okruženje. Detaljno je ispisan i objašnjen kod koji je napravljen za potrebe ovog sistema, U daljem radu moguće je unapređenje sistema, tako što bi se umesto ultrazvučnog senzora koristile kamere sa senzorom dubine.

7. LITERATURA

- 1.ESP32- <http://www.vitaelko.com/novosti/espressif-esp32-i-esp-wroom-32/>
- 2.https://www.elecrow.com/download/esp_wroom_32_da_tasheet_en.pdf
- 3.<http://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf>
- 4.HC-SR04- <https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>
- 5.UIN2003- <https://www.st.com/resource/en/datasheet/ulin2001.pdf>
- 6.28BYJ-48<https://components101.com/motors/28byj-48-stepper-motor>
- 7.Dogan Ibrahim: Controller Area Network
- 8.Arduino programsko okruženje, [http://www.elektronika.ftn.uns.ac.rs/images/Elektronika\(E2\)/Arduino%20uputstvo.pdf](http://www.elektronika.ftn.uns.ac.rs/images/Elektronika(E2)/Arduino%20uputstvo.pdf)
- 9.Setup petlja <https://www.arduino.cc/en/Reference/Setup>
- 10.Loop petlja <https://www.arduino.cc/en/Reference/Loop>
11. <https://www.allaboutcircuits.com/technical-articles/introduction-to-can-controller-area-network/>
- 12.Interrupt petlja <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>

Kratka biografija

Jovan Janjušević rođen je u Novom Sadu 1994. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu, na smeru Biomedicinsko inženjerstvo 2018. godine.

KOMPARATIVNA ANALIZA METODA RAZVOJA MOBILNOG SOFTVERA SA STANOVIŠTA UPOTREBLJIVOSTI**COMPARATIVE ANALYSIS OF MOBILE SOFTWARE DEVELOPMENT METHODS FROM A USABILITY STANDPOINT**

Danilo Buiša, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Rad se bavi problemom odabira optimalne metode razvoja mobilnog softvera sa stanovišta naučne discipline Interakcija čovek računar. Izvršena je komparativna analiza različitih strategija u razvoju softvera za mobilne uređaje sa stanovišta upotrebljivosti. Prvo je definisana upotrebljivost, principi dizajna korisničkog interfejsa i korisničkog doživljaja u mobilnim sistemima, a zatim je izvršena evaluacija pojedinačnih strategija, na osnovu ovih elemenata. Zaključak je da je za postizanje interfejsa visoke upotrebljivosti, najpogodnija izvorna programska paradigma, zatim hibridne i na kraju progresivne veb tehnologije.

Ključne reči: Upotrebljivost, mobilni softver, metode razvoja, korisnički interfejs

Abstract – The paper deals with the problem of choosing an optimal mobile software development method from the standpoint of the scientific discipline Human Computer Interaction. A comparative analysis of various strategies in the development of mobile software from a usability standpoint was performed. Firstly, the usability, principles of user interface design and user experience in mobile systems were defined, and then individual strategies were evaluated based on these elements. The conclusion is that in order to achieve a high usability interface, the native programming paradigm is optimal, followed by hybrid and finally progressive web technologies.

Keywords: Usability, mobile software, development methods, user interface

1. UVOD

Broj pametnih telefona u 2019. godini iznosi 2.71 milijardi, odnosno svaka treća osoba na svetu poseduje pametni mobilni telefon [1]. Uporedo s ovim trendom, internet saobraćaj koji dolazi sa mobilnih uređaja postojano je rastao od 2009. do danas, dok se udeo desktop-a postojano smanjivao.

Istraživanja pokazuju da je mobilni internet saobraćaj po prvi put prestigao saobraćaj sa desktop računara 2016. godine, pa je danas udeo mobilnog internet saobraćaja 62 % [2].

NAPOMENA:

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

Ako se navedena dva trenda, porast broja mobilnih uređaja, kao i njihov udeo u globalnom internet saobraćaju, uzmu kao glavni pokazatelji zastupljenosti različitih tipova računara, nesumnjivo je da su danas mobilni uređaji najzastupljenija vrsta računara, a da je razvoj mobilnih aplikacija, jedna od najbrže rastućih grana softvera.

Problem odabira konkretne strategije i podrške za različite operativne sisteme, a pritom zadržavajući standarde definisane platformom, predstavlja najveći izazov u razvoju mobilnog softvera. Kako trenutno postoje tri različite strategije razvoja mobilnih aplikacija, izvorne, hibridne i veb, neophodno je utvrditi najbolji način realizovanja ovih ciljeva, odnosno pronaći optimalan način razvoja softvera za mobilne uređaje. U ovom radu izvršena je evaluacija pomenutih pristupa, pre svega sa stanovišta upotrebljivosti, principa interakcije čoveka i računara i HCD-a (Human-centered Design).

2. IZAZOVI U RAZVOJU MOBILNIH APLIKACIJA

Uspeh mobilnih platformi direktno se meri brojem dostupnih aplikacija, a uspeh pojedinačnih programa brojem instalacija i aktivnih korisnika. Zbog toga se kompanije i pojedinci koji proizvode aplikacije trude da razviju verzije najmanje za Android i iOS, a neretko i za Windows Phone i druge operativne sisteme.

Ovde se uviđa osnovni problem u razvoju mobilnih aplikacija. On u mnogome zavisi od platforme za koju se program razvija. Aplikacije se moraju implementirati i dizajnirati za specifični operativni sistem tako da zadovolje principe i smernice koje definisane standardima određene platforme.

Koristeći izvornu (eng. *native*) strategiju, da bi se razvile aplikacije za više platformi potrebno je ekspertske znanje različitih profila softverskih inženjera, jer izvorni kod ne može da se deli između platformi. Veliki problem je što projektanti moraju da razmišljaju o UI (User Interface) standardima i paternima na konkretnim platformama i da u skladu sa njima prilagođavaju svoja dizajnerska rešenja. Korisnički doživljaj na mobilnim uređajima je vrlo specifičan: veličina ekrana, sadržaj, prethodno iskustvo i ponašanje korisnika koji koriste mobilne uređaje drugačiji su od onih na drugim uređajima i računarima. To je osnovni razlog zbog čega ne postoji neko opšte uputstvo za razvoj mobilnih aplikacija. Sve to podiže cenu softvera i vreme potrebno za realizaciju projekta, pa kupci softvera neretko odustaju od podrške za više platformi. Time su na gubitku ne samo oni, već i proizvođači softvera čija

rešenja veliki procenat potencijalnih korisnika nije u mogućnosti da koristi. Ovi problemi prouzrokovali su nastanak novih pristupa u programiranju mobilnih aplikacija, hibridnih i progresivnih veb tehnologija koji pokušavaju da reše problem **višeplatformske** podrške, zadržavajući deo baze koda.

3. UPOTREBLJIVOST U MOBILNOM OKRUŽENJU

Upotrebljivost (eng. *usability*) ocenjuje koliko je neki korisnički interfejs jednostavan za korišćenje. Neke od osnovnih komponenti upotrebljivosti su lakoća učenja, memorabilnost i satisfakcija. Zašto je upotrebljivost toliko važna? Na internetu, na primer, upotrebljivost je odlučujući faktor u zadržavanju korisnika. Ako je veb stranica preteška za korišćenje, ne nudi dobar uvid u to šta kompanija pruža ili ne pruža odgovore na korisnikova pitanja, korisnici će otići u potragu za drugim sajtom.

Dizajn usredsređen na korisnike (HCD - Human-centered Design) najvažniji je princip za kreiranje dizajna sa dobrim korisničkim interfejsom i doživljajem. Human Centered Design (HCD) u osnovi znači usredsrediti se na to kako korisnici zapravo koriste proizvod, umesto da se očekuje da će korisnici prilagoditi svoje ponašanje načinu na koji proizvod radi.

Jedna od najvažnijih heuristika dizajna korisničkog interfejsa u razvoju mobilnih aplikacija jeste eksterna **konzistentnost**, tj. konzistentnost u odnosu na ceo ekosistem u kom se aplikacija koristi. Interna konzistentnost, poput konzistentnosti sadržaja, fontova, boja itd. može da se postigne bez obzira na odabir platforme. Eksterna konzistentnost, sa druge strane, veoma zavisi od konkretnog operativnog sistema za koji se aplikacija razvija.

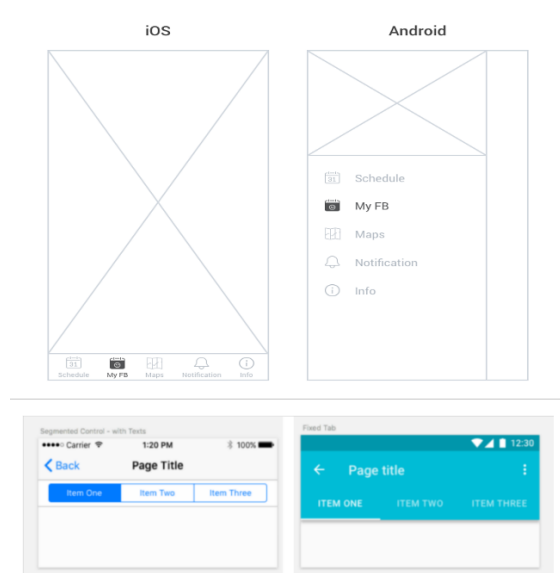
4. EVALUACIJA METODA

4.1. Konzistentnost i standardi

Jedna od osnovnih komponenti upotrebljivosti je lakoća učenja, odnosno koliko lako korisnik može da ostvari željene zadatke kada se prvi put susretne sa aplikacijom. U ovome, korisniku pomaže iskustvo koje je sticao koristeći druge aplikacije platforme koju koristi. Sa stanovišta dizajna, ono sto karakteriše različite platforme jeste postojanje različitih sistema dizajna. Na primer, za Android to je Material Design [3], a za iOS, Human Interface Guidelines [4]. Prema tome, da bi se korisniku olakšala upotreba aplikacija, neophodno je da ona bude u skladu sa UI i UX (User Experience) principima i standardima platforme na koju je navikao.

Zbog ovih standarda i principa, korisnički interfejs i kontrole mogu znatno da se razlikuju na različitim platformama, slika 1. Te kontrole nije moguće simulirati putem hibridnih ili veb metoda, toliko da bi pružile sličan doživljaj i familijarnost kao na izornoj platformi, jer svaka platforma ima svoje jedinstvene načine interakcije.

Dakle, jedan od glavnih benefita izvornih aplikacija jeste familijarnost. Developeri znaju kako da razvijaju softver za konkretnu platformu koristeći izvorne SDK-ove (Software Development Kit). Sa druge strane, korisnici poznaju osećaj, tok akcija i navigaciju aplikacija koje već koriste na svojim uređajima.



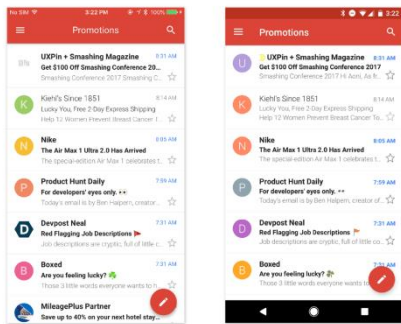
Slika 1. Neke od razlika iOS (levo) i Android dizajna (desno)

Stoga, razviti identičan korisnički interfejs i korisnički doživljaj na različitim platformama koristeći hibridne tehnologije nije uvek najbolja ideja. Naime, kako hibridne aplikacije ciljaju više platformi, one se često čine previše generično ili podsećaju na aplikacije konkurentskog operativnog sistema.

U idealnom slučaju, određena mobilna aplikacija treba da pruža iste funkcije i ponašanje bez obzira na ciljanu platformu na kojoj se pokreće. Međutim, zbog unutrašnjih razlika u različitim mobilnim uređajima i operativnim sistemima, generički dizajn za sve platforme ne postoji [5]. Razlog za to je činjenica da su HCI smernice na različitim platformama prilično različite, jer za mobilni svet ne postoje jedinstveni standardi, kao što je to slučaj, na primer, na vebu. Stoga se developeri stalno suočavaju sa dva konkurentna zahteva: familijarnost i specifičnost dizajna za svaku platformu, sa jedne, i postizanje konzistentnosti na različitim platformama, sa druge strane [45].

Prema tome, ne može se jednostavno klonirati iOS aplikacija na Android i obrnuto. Svaki operativni sistem prati potpuno drugačija pitanja programiranja, dizajna i interfejsa. Dobar dizajn je dizajn koji poštuje navike korisnika u svakom operativnom sistemu. Stoga, kreiranje dizajna koji će se lako prevesti na sve platforme uz očuvanje doslednosti interfejsa i ponašanja, je veoma izazovan posao.

Postoje izuzeci gde se aplikacije jedne platforme pridržavaju HCI smernica druge platforme. Na primer, neke iOS aplikacije slede smernice Android-a, poput Gmail-a, a neke Android aplikacije slede smernice ljudskog interfejsa (Human Interface Guidelines), poput Instagram-a. Na iOS verziji se vidi plutajuće dugme za kreiranje novog mejla, kontrole za navigaciju i drugo što je specifično za Android platformu, Slika 2. Ono što je specifično za ove aplikacije jeste to što su u pitanju hibridne aplikacije. U ovim slučajevima nije bilo potrebno zadovoljiti UI i UX standarde i smernice obe platforme, već samo jedne, pa je hibridna tehnologija bila pogodna opcija.



Slika 2. Gmail aplikacija na iOS-u (levo) i Android-u (desno)

4.2 Klasične hibridne tehnologije

Hibridna okruženja poput Apache Cordova i Ionic-a služe za razvoj generičnih interfejsa, koji pokušavaju da pruže kompromis između različitih standarda. Međutim, konzistentnost same platforme, često se gubi u ovakvim aplikacijama. Hibridna aplikacija i dalje može funkcionisati ne pridržavajući se standarda platforme, ali će korisnicima trebati više vremena da nauče da koriste aplikaciju i da se naviknu na njeno ponašanje. Takođe, zadovoljiti obe platforme istovremeno je znatno teže za implementaciju, jer se koriste neizvorne tehnologije poput HTML5, JavaScript-a, CSS-a i slično. U poređenju sa razvojem izvornih mobilnih aplikacija, korisničko iskustvo je teže održavati jer interfejs mora zadovoljiti korisnike različitih platformi. Ako se aplikacija previše nakloni iOS-u, na primer, iskustvo bi se pogoršalo za korisnike Android-a, i obrnuto. Zadovoljavanjem obe platforme istovremeno, dobija se interfejs drugačiji od izvornog, na kojem nisu navikli ni korisnici prve ni korisnici druge platforme.

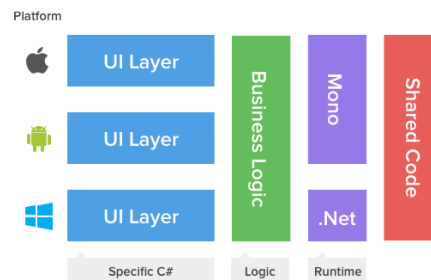
4.3 Izvorno-hibridne tehnologije

Glavna prednost React Native i sličnih izvorno-hibridnih tehnologija, poput Flutter-a i Xamarin-a, jeste mogućnost da se prilikom razvoja softvera koriste dva različita korisnička interfejsa. Na taj način, na primer, jedan može da prati Material Design principe Android-a, a drugi može da prati smernice ljudskog interfejsa iOS-a.

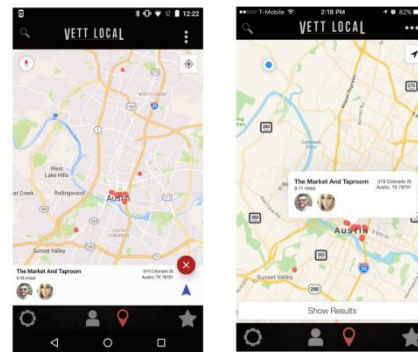
Ovo je moguće zahvaljujući odvojenim UI modulima aplikacije koji dele isti kod biznis logike, slika 3. Slika 4 ističe interfejse specifične za platformu, koji se mogu postići pomoću ovog programskog okvira.

Kao što se vidi, svaka aplikacija koristi svoju izvornu mapu i sadrži markere sa deskripcijom koji se pridržavaju smernica dizajna svake platforme. Na Android-u se kartice sa opisom podižu s dna mape. Na iOS-u se opis povezuje sa odabranim elementom na mapi. Iste radnje mogu se izvesti u obe aplikacije, većina koda se deli, ali ove dodatne specifičnosti pojedinačnih platformi pomažu u dizajnu specifičnom za platformu i celokupnoj upotrebljivosti.

Dakle, nova generacija hibridnih aplikacija poput React Native-a i Flutter-a je značajno poboljšana sa stanovišta upotrebljivosti, u odnosu na inicijalne hibridne tehnologije poput Ionic-a, Cordove i drugih. Razlog za to je upravo zbog mogućnosti kreiranja posebnog interfejsa za različite platforme, dok ostatak koda, tj kod biznis logike, ostaje isti.



Slika 3. Različiti UI moduli Xamarin aplikacije



Slika 4. Primer hibridne aplikacije sa različitim interfejsom u zavisnosti od platforme na kojoj se izvršava

4.4 Brzina odziva i performanse

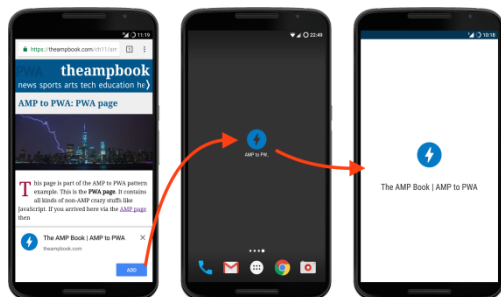
U studiji „Upotrebljivost i UX u izvornim i hibridnim aplikacijama“ [6] iz 2018. godine, pokušale su da se utvrde preferencije korisnika između izvorne i hibridne aplikacije. Utvrđeno je da koliko brzo aplikacije reaguju na akcije korisnika je takođe važan deo korisničkog iskustva, kao i da je ono bolje u izvornoj verziji, jer između ostalog, postoji ugrađeno kašnjenje na veb i hibridnim aplikacijama. Takođe, hibridne aplikacije su sporije u odnosu na izvorne aplikacije zato što hibridne aplikacije dodaju još jedan sloj između korisnika i aplikacije - ne izvršavaju se u izvornom okruženju već u veb pogledu (eng. *web-view*). Jedan od ključnih prednosti izvorne metode razvoja aplikacija je velika brzina i performanse koje pruža, jer je aplikacija optimizovana za iOS ili Android.

Hibridne aplikacije su i dalje sporije od svojih konkurenata, ali ne toliko kritično. Situacija je danas drastično poboljšana, delom zbog napretka u optimizaciji mobilnih operativnih sistema i značajnoj procesorskoj snazi modernih uređaja, a delom i zbog alata i biblioteka koje se koriste za razvoj, kao i iskustva projekatana i programera. Budući da se progresivne veb aplikacije izvršavaju unutar veb pregledača, one su sporije od prethodne dve vrste.

4.5. Namena softvera i zahtevi korisničkog interfejsa

Potrebe za performansama zavise od namene softvera i zahteva korisničkog interfejsa. Studije [7] i [8] potvrđuju da ako je potreban visok nivo detaljnosti korisničkog interfejsa i korisničkog doživljaja, izvorni pristup je i dalje najpovoljniji, jer izvorne tehnologije omogućavaju brzu grafiku, fluidnu animaciju i tečne tranzicije. Stoga, u slučaju razvoja aplikacija sa zahtevnim animacijama, kao i 3D igrice, izvorna strategija je najpogodnija, dok za

manje resursno zahtevne aplikacije, hibridne i progresivne veb tehnologije mogu biti dobra alternativa. Kako se progresivnim veb aplikacijama pristupa kao normalnim veb stranicama, one mogu da pomognu do dolaska novih korisnika. Nakon posete stranici, korisnik ima mogućnost kreiranja ikonice koja služi kao prečica do sajta uz uvodni ekran, simulirajući na taj način korisnički doživljaj izvornih i hibridnih aplikacija, slika 5.



Slika 5. PWA aplikacija nalikuje izvornim i hibridnim

4.6 Izvorne funkcionalnosti

Izvorne funkcionalnosti svake platforme obogaćuju korisničko iskustvo aplikacije. Izvorne aplikacije imaju prednost ugrađenih funkcija koje mogu pružiti poznatiji izgled i korisnički doživljaj [5]. Hibridne i progresivne veb aplikacije, zahvaljujući raznim *plugin*-ovima, imaju pristup ugrađenim funkcionalnostima i senzorima uređaja. Međutim, izvorni SDK omogućava pristup ovim funkcionalnostima bez potrebe za izvornim *plugin*-ovima, a nove funkcionalnosti će biti dostupne odmah pri njihovom objavljivanju. Zbog ovoga, izvorne aplikacije pružaju bolje performanse, boljeg odziva, i tečnijeg doživljaja pošto korisnik ne primećuje nikakva kašnjenja zbog učitavanja, *wrapper*-a, *plugin*-ova itd, za razliku od neizvornih aplikacija.

5. ZAKLJUČAK

Korisnički interfejs i korisnički doživljaj je presudan za uspeh aplikacije. Odluka o tome šta izabrati se zasniva na ciljevima i prioritetima softvera. Slučajevi korišćenja, namena, okruženje i sama priroda aplikacije takođe utiču na odluku. Kod izvorne strategije, zbog mogućnosti korišćenja izvornih komponenti, smernica i principa definisane platformom i slično, postiže se **konzistentnost** ne samo unutar jedne aplikacije, vec u celom **ekosistemu** jedne platforme. Pored toga, standarde koje definiše Android, iOS i druge platforme, jednostavnije je postići koristeći upravo njihove jezike, okruženja i pravila. Držeći se ovih standarda, aplikacije liče jedne na druge, pružajući korisnicima **familijaran** dizajn, olakšavajući im na taj način snalaženje i upotrebu aplikacije. Hibridne aplikacije mogu da imaju sličan i konzistentni doživljaj na različitim platformama. Međutim, dosta je jednostavnije pružiti dobro korisničko iskustvo i korisnički interfejs velike **odzivnosti** na izvornim aplikacijama, jer koriste izvorne komponente i **UI i UX standarde** definisane platformom.

Za optimalnu upotrebljivost mobilne aplikacije, aplikacija mora da sadrži jasan i dosledan način interakcije, da sledi prirodni tok platforme omogućavajući korisniku da intuitivno komunicira sa njom, koristeći prethodno iskustvo iz sa platforme.

Postizanje ovih zahteva moguće je korišćenjem izvorne metode ili eventualno izvorno-hibridne uz razvoj dva različita interfejsa, a zadržavajući biznis logiku. Ako je odluka da se na više platformi koriste HCI principi, UI i UX smernice samo jedne platforme, tada su hibridne aplikacije pogodna metoda.

6. LITERATURA

- [1] Statista, The Statistics Portal for Market Data, Market Research and Market Studies, <https://www.statista.com>, Uvidi i statistike.
- [2] StatCounter, Mobile and tablet internet usage, <https://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide> (pristupljeno u septembru 2019).
- [3] Android, Material Design for Android, <https://developer.android.com/guide/topics/ui/look-and-feel>, HCI principi za Android.
- [4] Apple, Human Interface Guidelines, <https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/>, iOS HCI principi.
- [5] M. E. Joorabchi, A. Mesbah, P. Kruchten. "Real challenges in mobile app development". In: Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on. IEEE. 2013, pp. 15–24.
- [6] L. Andersson, „Usability and User Experience in Mobile App Frameworks Subjective, but not Objective, Differences between a Hybrid and a Native Mobile Application”, 2018.
- [7] W. Jobe, „Native vs Mobile web apps“, 2013, https://www.academia.edu/4967196/Native_Apps_vs._Mobile_Web_Apps (pristupljeno u septembru 2019).
- [8] N. P. Huy and D. van Thanh, „Selecting the right mobile app paradigms”, in 2012 5th IEEE International Conference on Service-Oriented Computing and Applications (SOCA), 2012, pp. 1–6.

Kratka biografija:



Danilo Bujiša rođen je u Novom Sadu 1995. god. Diplomirao je 2018. godine na smeru Računarstvo i automatika – Primeunjene računarske nauke i informatika.

2019. godine odbranio je master rad na Fakultetu tehničkih nauka na smeru Elektronsko poslovanje.
kontakt: danilob@uns.ac.rs

UTICAJ NESIMETRIJE TROFAZNOG SISTEMA NA MAGNETSKO POLJE “U” SABIRNICA

THE INFLUENCE OF UNBALANCED THREE PHASE SYSTEM ON THE MAGNETIC FIELD OF “U” SHAPED BUSBARS

Milica Samardžić, Karolina Kasaš-Lažetić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ENERGETIKA, ELEKTRONIKA I TELEKOMUNIKACIJE

Kratak sadržaj – U ovom radu je ispitan uticaj nesimetrije faznih struja i različitih faznih stavova struja na raspodjelu magnetskog polja u okolini trofazne “U” sabirnice sa neutralnim provodnikom. Ispitan je i uticaj oklopljavanja sabirnice na raspodjelu magnetskog polja. Proračun je izvršen u programskom paketu “COMSOL Multiphysics”, koji radi na bazi metode konačnih elemenata. Rezultati proračuna su prikazani grafički, kao raspodjela intenziteta vektora magnetske indukcije po poprečnom presjeku sistema i preko najčešće korišćenih grafika i detaljno su protumačeni.

Ključne reči: Trofazne „U” sabirnice sa neutralnim provodnikom, Oklopljene sabirnice, Magnetsko polje, Metod konačnih elemenata (MKE)

Abstract – This paper researches the influence of current unbalance and different phase shifts of currents on the distribution of the magnetic field around three-phase „U” shaped busbars with neutral conductor. The effect of busbar shielding on the magnetic field distribution is also investigated. The calculation was performed in COMSOL MULTIPHYSICS software program package based on the Finite Element Method. The results of calculation are presented graphically, as the magnetic flux density distribution over the system’s cross-section and as the most common diagrams.

Keywords: Unbalanced three phase system, unshielded and shielded “U” shaped busbars with neutral conductor, magnetic field, finite element method (FEM)

1. UVOD

Svaki dio energetskeg sistema ima svoju ulogu i mora da funkcioniše besprekorno. Jedan od takvih dijelova su sabirnice, koje povezuju vodove koji dovode i odvođe električnu energiju. Za postrojenja nominalnih napona $U_n \leq 35$ kV, sabirnice se izvođe od krutih provodnika, od bakra ili aluminijuma, pravougaonog poprečnog presjeka ili su profilisani u takozvane sabirnice „U” profila [1]. U okolini opterećenih sabirnica postoji elektromagnetsko (EM) polje, čiji nivo može da se menja usled raznih faktora. Jedan od zahteva za zaštitu životne sredine i elektromagnetsku kompatibilnost jeste praćenje i reagovanje na eventualne promene EM polja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Karolina Kasaš-Lažetić, docent.

Zbog svega napomenutog, u ovom radu je izvršena analiza uticaja promene faznog stava struja trofaznog sistema, kao i strujne nesimetrije na raspodjelu magnetskog polja u okolini neoklopljenih i oklopljenih „U” sabirnica.

Numeričkim postupkom, primenom programskog paketa COMSOL Multiphysics [2] izvršena je analiza uprošćenog, dvodimenzionalnog (2D) modela sistema. Na osnovu izvršenih analiza i dobijenih rezultata je zaključeno, da do promjene raspodjele magnetskog polja, može da dođe usled promjene redosleda faznih struja u sabirnicama i usled strujne nesimetrije, a da se oklopljavanjem može postići značajno smanjenje magnetskog polja van oklopa.

2. SIMETRIJA TROFAZNIH SISTEMA

2.1. Pojam simetrije

Naponi i struje u trofaznim sistemima su prostoperiodični. Ovi sistemi su simetrični (uravnoteženi) ako su im amplitude napona i jačina struja međusobno jednake i ako su njihove trenutne vrednosti pomerene u vremenu za trećinu periode, tj. ako su međusobno fazno pomereni za ugao od $2\pi/3$.

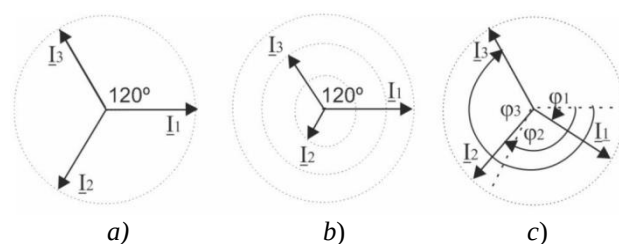
U linearnim sistemima, sa prostoperiodičnim strujama i naponima, analiza i rješavanje problema može da se izvrši u kompleksnom domenu. Kompleksni predstavnici jačina struja u trofaznom simetričnom sistemu su oblika

$$\underline{I}_1 = I e^{j0^\circ}, \quad \underline{I}_2 = I e^{-j2\pi/3^\circ}, \quad \underline{I}_3 = I e^{-j4\pi/3^\circ}. \quad (1)$$

Zbir tih struja u simetričnom trofaznom sistemu je uvijek jednak nuli [3].

2.2 Pojam nesimetrije

Nesimetrični režim rada nastaje kao posljedica različitog opterećenja faza, nesimetričnih kratkih spojeva, prekida neke od faza i dr., (slika 1).



Slika 1. a) simetrične struje, b) nesimetrija po amplitudi, c) nesimetrija po faznom stavu

2.3 Uzroci i posljedice nesimetrije

Glavni uzrok neravnoteže struje i napona jeste nejednaka raspodjela jednofaznih opterećenja između trofaznih

vodova [1]. Strujna nesimetrija ima negativne uticaje na sve sistemske dijelove: proizvodnju, prenos, distribuciju i opterećenja. Sa strane generatora, strujna asimetrija negativno utiče na efikasnost. Na strani prenosa i distribucije, strujna nesimetrija umanjuje prenosni kapacitet i efikasnost. Na strani opterećenja smanjuje efikasnost opreme. Kapacitet i efikasnost transformatora su takođe umanjani. Sve gore navedene činjenice govore u prilog tome, da nesimetrija negativno utiče na rad cjelokupnog elektroenergetskog sistema i da je ispitivanje uticaja nesimetrije uvek potrebno i korisno.

3. TEORIJSKA OSNOVA

Sabirnice stvaraju oko sebe EM polje niske frekvencije, predstavljajući time vremenski sporopromjenljiva polja, čija vrijednost naglo opada sa povećanjem udaljenosti od izvora. Analiza magnetske i električne komponente datog EM polja može da se vrši odvojeno, nazavisnim posmatranjem vremenski sporopromjenljivog magnetskog polja i vremenski sporopromjenljivog električnog polja.

3.1 Osnove proračuna

Magnetsko polje se opisuje vektorom magnetske indukcije, $\mathbf{B}$, a električno polje, vektorom jačine električnog polja, $\mathbf{E}$. Oba ova vektora, mogu da se predstavje preko električnog skalar potencijala V i magnetskog vektor potencijala, $\mathbf{A}$,

$$\begin{aligned}\vec{E} &= -\nabla V - \frac{\partial \vec{A}}{\partial t} \\ \vec{B} &= \nabla \vec{A}\end{aligned}\quad (2)$$

Analiza cijelokupnog sistema sabirnica može da se izvrši u kompleksnom domenu, pošto se cijeli sistem može smatrati linearnim, a struje u sabirnicama prostoperiodičnim. Radi pojednostavljenja proračuna, sabirnice se postavljaju u Dekartov pravougli koordinatni sistem, tako, da je z osa sistema normalna na poprečni presjek sabirnica (Slika 1).

Vektor gustine struje i magnetski vektor potencijal u tako izabranom koordinatnom sistemu, imaju samo z komponentu, dok im se intenziteti mijenjaju samo po koordinatama x i y. Da bi se odredila raspodjela magnetskog vektor potencijala, radi određivanja raspodjele magnetskog polja, treba da se reši parcijalna diferencijalna jednačina [3], [4]

$$\nabla^2 \vec{A} - j\omega\mu\sigma \vec{A} = -\mu \vec{J} \quad (3)$$

U Dekartovom koordinatnom sistemu, data jednačina prelazi u skalarni oblik,

$$\frac{\partial^2 \underline{A}_z}{\partial x^2} + \frac{\partial^2 \underline{A}_z}{\partial y^2} - j\omega\mu\sigma \underline{A}_z = -\mu \underline{J}_z \quad (4)$$

Usvajajući, da magnetski vektor potencijal iščezava van zamišljenog, dovoljno velikog prostora u 2D modelu sistema, dolazi se do partikularnog rešenja diferencijalne jednačine (4). Indukovano električno polje, preko indukovanih struja utiče na pojavu površinskog efekta i efekta blizine, a time i na raspodjelu magnetskog polja unutar sistema.

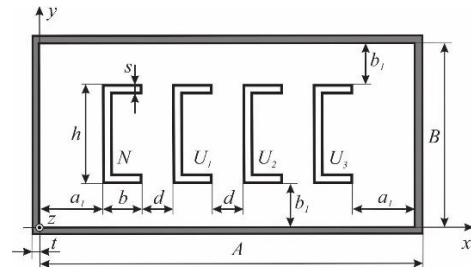
3. MODEL

Da bi se izvršila analiza uticaja promene faznog stava struja, strujne nesimetrije i/ili oklopljavanja trofaznih

sabirnica na raspodjelu intenziteta vektora magnetske indukcije, posmatran je simetričan trofazni sistem „U” sabirnica sa neutralnim vodom (bez i sa oklopom), čije su standardizovane dimenzije prikazane u tabeli 1, prema oznakama datih na slici 1 [5]. Neutralan vod i sabirnice pojedinih faza obeleženi su kao N , U_1 , U_2 i U_3 .

Tabela 1. Dimenzije standardizovanih „U6” sabirnica

Tip sabirnice	h [mm]	b [mm]	s [mm]	d [mm]	S [mm ²]
U6	60	30,0	4	25	448



Slika 1. Raspored i dimenzije oklopljenih „U6” sabirnica sa neutralnim vodom

Neutralna sabirnica je na rastojanju $a_1 = 35$ mm od y ose, dok su sve sabirnice na istom rastojanju $b_1 = 45$ mm, od x ose. Debljina oklopa iznosi $t = 3$ mm.

Za analizu raspodjele intenziteta vektora magnetske indukcije, uticaja nesimetrije, kao i oklopljavanja sabirnica, korišćenjen je programski paket COMSOL Multiphysics i pripadajući AC/DC modul. Nakon izbora podmodula i tipa simulacije, prvo je definisana geometrija sistema, a zatim su zadati potrebni ulazni parametri (jačina struje, električne i magnetske karakteristike sabirnica, oklopa i okolnog prostora). U sledećem koraku su usvojeni granični uslovi, a nakon formiranja mreže konačnih elemenata izvršen je proračun. Rezultat proračuna predstavljaju vrednosti magnetskog vektor potencijala u svim tačkama sistema, pomoću kojih se određuju ostale veličine od interesa; u ovom primeru, to je raspodjela intenziteta vektora magnetske indukcije.

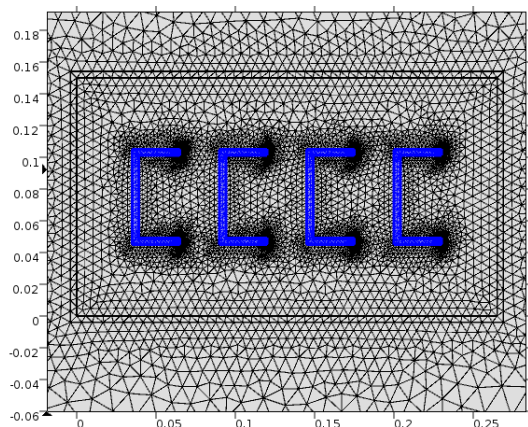
U radu su posmatrane bakarne sabirnice, specifične provodnosti $\sigma_{Cu} = 5,998 \times 10^7$ S/m, koje se nalaze u vazduhu. Električne i magnetske karakteristike vazduha su iznosile ($\epsilon_{r0} = 1$, $\mu_{r0} = 1$, $\sigma_0 = 0$). Ceo sistem je bio ograničen kružnom ljuskom poluprečnika $R_{gr} = 2$ m, debljine $d = 50$ mm, koja je simulirala domen „beskonačnih elemenata” u konačnim granicama. Oklop oko sabirnica je bio od aluminijuma, specifične provodnosti $\sigma_{Al} = 3,771 \times 10^7$ S/m.

U posmatranom modelu, prvo je izvršena analiza simetričnog trofaznog sistema, kada kompleksna vrijednost jačine struje u neutralnom provodniku sistema iznosi $\underline{I}_N = \underline{I}_1 + \underline{I}_2 + \underline{I}_3 = 0$. Zatim su posmatrana tri odvojena slučaja nesimetrije sistema, dobijena množenjem struja definisanih u (1), koeficijentima, preuzetih iz [6],

$$\begin{aligned}1ns &= 1, & 2ns &= 0,733, & 3ns &= 0,733 \\ 1ns &= 1, & 2ns &= 1, & 3ns &= 1 \\ 1ns &= 1, & 2ns &= 1, & 3ns &= 0,733.\end{aligned}\quad (5)$$

Mreža konačnih elemenata je adaptivna; gušća u oblastima sistema u kojima se očekuju izrazitije promene

veliĉine od interesa (u provodniku, u oklopu i duŹ pravaca proraĉuna), a rjeĉa unutar ostatka sistema. Izgled formirane mreŹe konaĉnih elemenata je prikazana na slici 3.



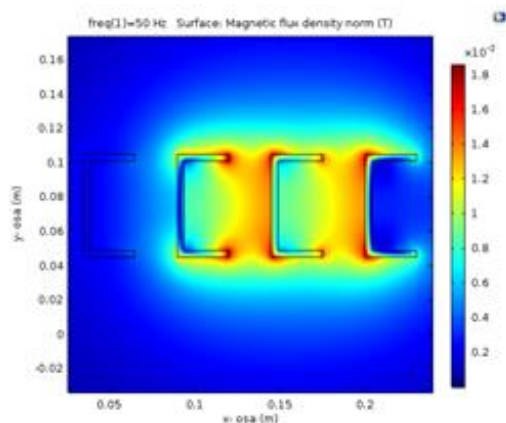
Slika 3. Dio modela sa formiranom mreŹom

3. REZULTATI

U radu je izvrŹena analiza raspodjele magnetskog polja u okolini simetriĉnog i nesimetriĉnog sistema, neoklopljenih i oklopljenih, „U6” sabirnica. Sabirnice su od neobojenog bakra, nominalne jaĉine struje 1000 A.

Prvo je izvrŹen proraĉun neoklopljenog simetriĉnog trofaznog sistema „U6” sabirnica, sa ciljem da se ispita uticaj promjene faza na raspodjelu intenziteta vektora magnetske indukcije. Analize su pokazale, da po tri fazna rasporeda daju istu raspodjelu intenziteta vektora magnetske indukcije: 123=231=312 i 132=213=321.

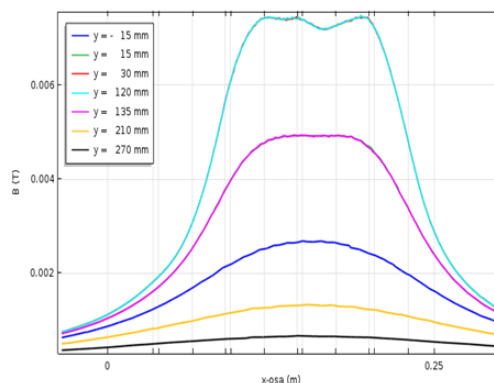
Na slici 4 prikazana je prostorna raspodjela intenziteta vektora magnetske indukcije kroz popreĉni presjek neoklopljenih simetriĉnih sabirnica, za fazne stavove trofaznih struja u provodnicima sistema, 123=231=312.



Slika 4. Prostorna raspodjela intenziteta vektora magnetske indukcije za raspored faza 123, 231, 312

Sa slike 4 moŹe da se uoĉi, da je magnetsko polje najjaĉeg intenziteta izmeĉu faznih sabirnica, usled dejstva povrŹinskog efekta i efekta blizine.

Na slici 5 prikazana je raspodjela intenziteta vektora magnetske indukcije u okolini neoklopljenih trofaznih sabirnica sa neutralnim provodnikom za simetriĉan redoslijed faza (123=231=312), duŹ pravaca koji prolaze kroz vazduh, ispod i iznad sabirnica, paralelno sa x osom. Sa slike 5 se vidi, da je raspodjela intenziteta vektora magnetske indukcije u vazduhu najveĉa ispod i iznad sabirnica, a smanjuje se udalŹavanjem od njih. Najveĉi intenzitet je duŹ pravaca $y = 30 \text{ mm}$ i $y = 120 \text{ mm}$.

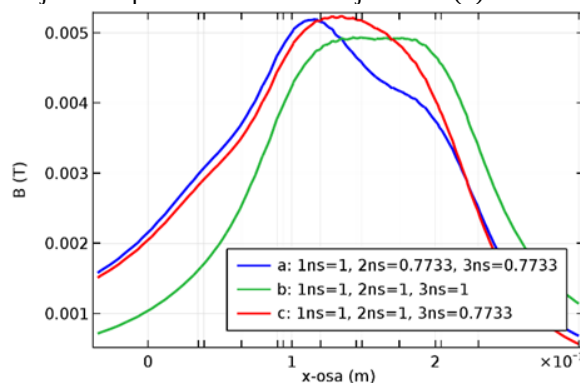


Slika 5. Raspodjela intenziteta vektora magnetske indukcije simetriĉnih neoklopljenih za raspored faza 132=213=321

koji su najbliŹi horizontalnim ivicama sabirnica (dati grafici se poklapaju). Isto takvo poklapanje se javlja i za vrijednosti intenziteta vektora magnetske indukcije na visinama 15 mm i 135 mm. MoŹe da se primjeti, da je pri vertikalnom poloŹaju sabirnica, raspodjela polja relativno ravnomjerna u oblasti ispod i iznad sabirnica.

Posle analize simetriĉnih neoklopljenih sabirnica, uraĉena je analiza nesimetriĉnih sabirnica, pri faznim rasporedima 123=231=312. Rezultati su prikazani na slici 6.

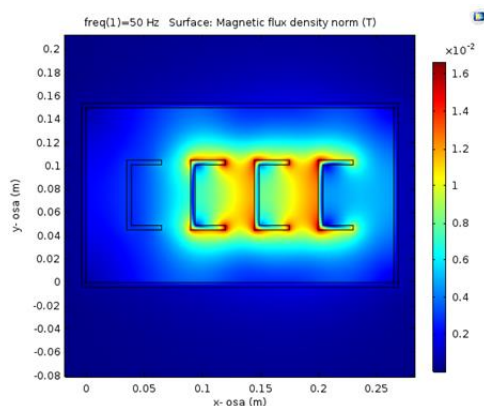
Na tim slikama je prikazana raspodjela intenziteta vektora magnetske indukcije duŹ pravca na visini $y = 135 \text{ mm}$ (30 mm iznad gornje horizontalne stranice sabirnica), za vrijednosti parametara nesimetrije date u (5).



Slika 6. Raspodjela intenziteta vektora magnetske indukcije nesimetriĉnih neoklopljenih sabirnica za raspored faza 123, 231, 312

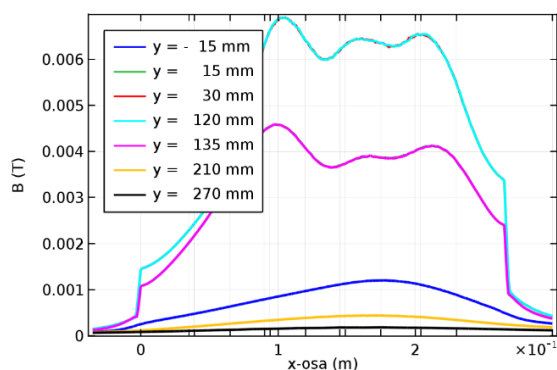
Na slici 6 se vidi uticaj nesimetrije na vrijednosti intenziteta vektora magnetske indukcije, pri razliĉitim kombinacijama parametara, pri ĉemu parametri pod *b* predstavljaju simetriĉan trofazni sistem. Nesimetrija blago poveĉava intenzitet vektora magnetske indukcije i unosi promjenu u raspodjelu magnetskog polja u okolini sabirnica.

Posle analize raspodjele intenziteta vektora magnetske indukcije u okolini neoklopljenih trofaznih „U” sabirnica sa neutralnim vodom, izvrŹena je analiza trofaznih „U” sabirnica sa neutralnim vodom i oklopom oko njih, da bi se utvrdilo kako utiĉe oklop na raspodjelu magnetskog polja. Analizom rezultata je utvrĉeno, da raspored faza utiĉe na raspodjelu magnetskog polja i u sluĉaju oklopljenih sabirnica i da je magnetsko polje smanjenog intenziteta izvan oklopa. Sa slike 7 se vidi, da se najveĉi intenziteti vektora magnetske indukcije javljaju u okolini vrhova sabirnica, izmeĉu faznih sabirnica, kao i ispod i iznad njih.



Slika 7. Prostorna raspodjela intenziteta vektora magnetske indukcije simetričnih oklopljenih sabirnica za raspored faza 123, 231, 312

Na slici 8 prikazana je raspodjela intenziteta vektora magnetske indukcije u slučaju simetričnih trofaznih sabirnica, u prisustvu oklopa.

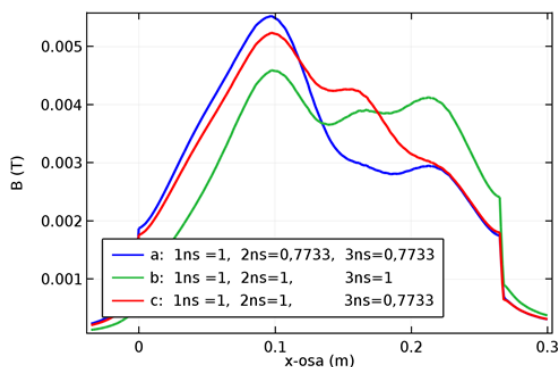


Slika 8. Raspodjela intenziteta vektora magnetske indukcije simetričnih oklopljenih sabirnica za raspored faza 123, 231, 312

Sa slike 8 se jasno vidi, da je prisustvo oklopa naglo smanjilo intenzitet vektora magnetske indukcije uz bočne ivice oklopa.

Ovim je pokazano, da prisustvo oklopa smanjuje intenzitet vektora magnetske indukcije u oblasti van oklopa i da, po potrebi, oklopljavanje može da se primjeni u cilju zaštite okoline od neželjenog dejstva magnetskog polja sabirnica.

Slika 9 prikazuje raspodjelu intenziteta vektora magnetske indukcije za nesimetrične, oklopljene sabirnice, duž prave $y = 135$ mm, pri istim koeficijentima nesimetrije kao za neoklopljen slučaj.



Slika 9. Raspodjela intenziteta vektora magnetske indukcije nesimetričnih, oklopljenih sabirnica za raspored faza 123, 231, 312

Analizirajući rezultate prikazane na slici 9, može da se vidi, da je usled nesimetrije, raspodjela intenziteta vektora magnetske indukcije neravnomernija između sabirnica i da je raspodjela veće oko neutralnog provodnika, što nije bio slučaj sa simetričnim sistemom.

Nesimetričnost faznih struja dovodi do veće neujednačenosti magnetskog polja. Oklop, i u ovom slučaju, naglo smanjuje intenzitet vektora magnetske indukcije izvan oklopa.

4. ZAKLJUČAK

Neravnoteža je ozbiljan problem kvaliteta električne energije, koji uglavnom pogađa distributivne sisteme niskog napona. U ovom radu je objašnjena neravnoteža struje, njihovi uzroci i negativni efekti do kojih dolazi usljed nesimetrije.

U radu je ispitan uticaj promene faznog stava struja u trofaznim, vertikalno postavljenim „U” sabirnicama, uticaj strujne nesimetrije na raspodjelu intenziteta vektora magnetske indukcije, kao i efekti do kojih dovodi oklopljavanje sabirnica.

Pokazano je da promena faznog stava struja menja raspodjelu magnetskog polja u okolini sabirnica, da se pomoću oklopa može izvršiti zaštita prostora oko sabirnica od neželjenog uticaja magnetskog polja sabirnica i da strujna nesimetrija takođe dovodi do promene raspodele intenzitet vektora magnetske indukcije.

Formiran model sabirica, kao i proračuni izvršeni u ovom radu, mogu da posluže za analizu sabirničkih sistema drugih oblika, osobina i pod različitim radnim uslovima.

5. LITERATURA

- [1] J. Nahman, V. Mijailović, „Razvodna postrojenja”, Akademska misao, Elektrotehnički Fakultet, Beograd, 2005.
- [2] COMSOL Multiphysics, documentation, 2016.
- [3] B. D. Popović, „Elektromagnetika”, Građevinska knjiga, Beograd, 2000.
- [4] M. Stojaković, „Matematička analiza 2”- drugi deo, 2. Izdanje, FTN Novi Sad, Novi Sad, 2007.
- [5] LJ. Gerić, „Razvodna postrojenja-zbirka zadataka, drugo izdanje”, Novi Sad, FTN, 2006.
- [6] K. Kasaš-Lažetić, G. Mijatović, D. Herceg, M. Prša, „Electromagnetic field distribution in vicinity of unbalanced mixed overhead power lines”, IEEE 2019, Niš, Srbija, 2019.

Kratka biografija:



Milica Samardžić rođena je 1993. godine u Trebinju. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo – Energetska elektronika i električne mašine je odbranila 2019.god.

kontakt: milicasamardzic97@yahoo.com



Karolina Kasaš-Lažetić, rođena je u Novom Sadu 1964. Doktorirala je na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva. Oblast interesovanja je primenjena elektromagnetika.

KABLOVSKI DISTRIBUTIVNI SISTEMI**CABLE DISTRIBUTION SYSTEMS**Ljiljana Popović, Željen Trpovski, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U današnjem svetu kada je Internet postao svakodnevica, podjednako važan kao voda ili vazduh, zahvaljujući DOCSIS tehnologijama, kablovske kompanije moraju da ispunjavaju zahteve za širinom opsega i brzinom koje su potrebne da podrže nove aplikacije i da budu u mogućnosti da isporučuju više megabajtni, brzi, širokopojasni Internet, stižući downstream brzine od 10 Gbps i upstream do 1 Gbps. Kako prihodi ostvareni putem kablovske distributivne mreže zauzimaju značajni udeo u strukturi prihoda ostvarenih od fiksnog širokopojasnog pristupa Internetu (41,3%) kao i u strukturi prihoda ostvarenih od distribucije medijskih sadržaja (59%), kablovski distributivni sistemi su veoma značajni za tržište telekomunikacija Republike Srbije.

Gljučne reči: KDS, DOCSIS standard, analiza tržišta

Abstract – In today's world, when the Internet has become a daily routine, equally important as water or air, thanks to DOCSIS technologies, cable companies must meet the bandwidth and speed requirements, they need to support new applications and be able to deliver more megabytes, faster, broadband Internet, reaching downstream speeds of 10 Gbps and upstream up to 1 Gbps. As the revenues generated through the cable distribution network take a significant share in the structure of revenues generated from fixed broadband Internet access (41.3%) as well as in the structure of revenues generated from the distribution of media content (59%), cable distribution systems are very important for the telecommunications market of the Republic of Serbia.

Keywords: KDS, DOCSIS standard, market analysis

1. UVOD

Iz dana u dan se tehnologija sve više razvija. XXI vek je definitivno vek digitalizacije, s obzirom da se svakodnevno koriste digitalni fotoaparati, digitalne kamere, digitalni satovi, digitalne novine i slično.

U svetu u kome se sve digitalizuje, bila je očekivana i digitalizacija televizije, koja je unapredila prijem TV programa sa tehničke strane, ali i pojačala doživljaj kod gledalaca. Zahvaljujući napretku i ubrzanom razvoju, desila se tranzicija iz analogne u digitalnu eru u kojoj danas živimo.

Digitalna era, kojoj je doprinela digitalna televizija (*engl. Digital television - DTV*), donosi mnoge tehničke

NAPOMENA:

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

inovacije i pogodnosti svom auditorijumu kao i celoj radio - televizijskoj produkciji.

Ona omogućava, pored gledanja TV-a, da gledaoci budu u prilici da primaju podatke i televizijske usluge kao što su dodatni titlovi, audio kanali, više informacija o programu (elektronski programski vodič) i drugo. Očekuje se da će DTV u budućnosti omogućiti servise pristupa Internetu, poput e-Uprave, e-Trgovine itd. Jednom kada se uvede DTV, korisnik može da izabere šta, gde i kada želi da gleda.

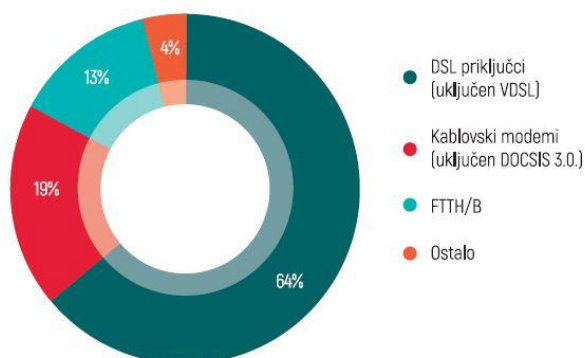
Pojavom novih tehnologija u prenosu signala, kao što su HFC-mreže i novije WDM-mreže, polako se prešlo na implementaciju optičkih kablova za prenos signala kablovske televizije. Tako su nastale HFC-mreže za distribuciju kablovskog signala. Takođe, u tim mrežama se mogao prenositi i analogni i digitalni signal, zavisno od zahteva korisnika kao i od mogućnosti samih provajdera.

Ovaj hiperbrz razvitak tehnologije, donosi nam stvari koje su donedavno bile skoro nezamislive. Danas je društvo i život stanovnika širom planete sasvim drugačiji od onog od pre deset ili više godina, a za promene je „odgovoran” razvoj Interneta. Zapravo, ne živimo više u svetu mehanike, već živimo u svetu mrežnih tehnologija, koje dopiru do svih pora svakodnevnog života. Provoditi vreme u svom domu uz mogućnost kontrolisanja sadržaja koji želimo da gledamo na televizoru ili nekom drugom uređaju, nekada je izgledalo kao stvar naučne fantastike, ali danas je tržište kablovske televizije i usluga koje ona pruža doživelo veliku ekspanziju. Kablovski Internet predstavlja jednu od najbrže rastućih tehnoloških struktura u današnjem svetu. Provajderi kablovskog Interneta moraju da pažljivo prate sve tehnološke inovacije koje im omogućuju kompetitivnu prednost, a njihovim korisnicima veći kvalitet usluge i niže cene.

2. ANALIZA TRŽIŠTA TELEKOMUNIKACIJA REPUBLIKE SRBIJE U POGLEDU KDS-A

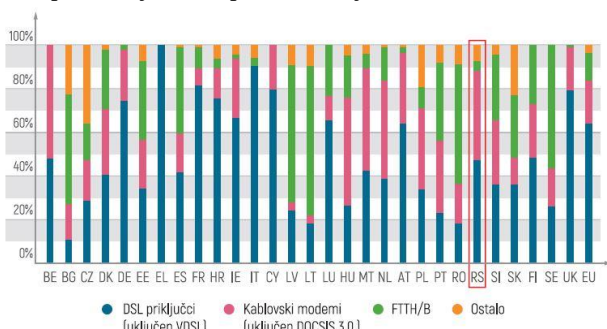
Ukupni prihodi ostvareni na tržištu elektronskih komunikacija Republike Srbije u 2017. godini iznose 1,58 milijardi evra i neznatno su viši u odnosu na prethodnu godinu (blizu 3% stopa rasta). U bruto društvenom proizvodu Srbije prihodi od elektronskih komunikacija su u 2017. godini imali udeo od 4,3%.

Najpre je analizirana usluga širokopojasnog pristupa Internetu u zemljama EU, kao i u Republici Srbiji i dato je poređenje. U strukturi fiksnog širokopojasnog pristupa različite tehnologije imaju učešće, gde je širokopojasni pristup putem kablovske mreže zastupljen sa 19% ukupnog broja pretplatnika na kraju 2017. godine, dok je taj procenat u Srbiji čak 41%.



Slika 1. Raspodela pretplatnika fiksnog širokopojsnog pristupa po tehnologijama U EU [1]

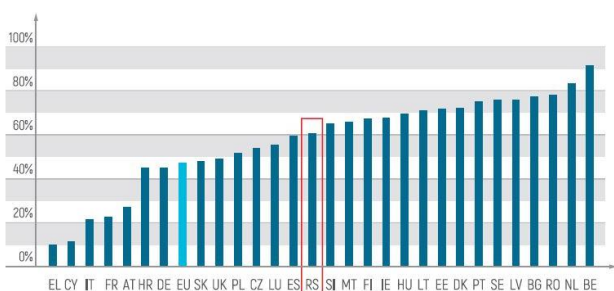
Zatim je prikazana zastupljenost različitih tehnologija fiksnog širokopojsnog pristupa Internetu u državama Evropske unije i u Republici Srbiji.



Slika 2. Raspodela pretplatnika fiksnog širokopojsnog pristupa po tehnologijama [1]

U Belgiji, Mađarskoj, Malti i Holandiji, najveće učešće ima pristup preko kablovskih distributivnih mreža. U Srbiji DSL tehnologije su i dalje vodeće, ali je u značajnoj meri zastupljen i kablovski pristup.

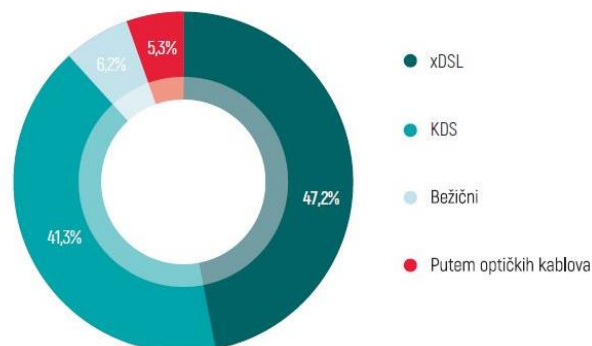
Što se tiče mreža sledeće generacije (engl. *Next generation access* - NGA), skoro 48% ukupnog broja pretplatnika fiksnog širokopojsnog pristupa Internetu u EU je realizovano preko NGA, pri čemu se pod NGA podrazumeva FTTH, FTTB, VDSL i kablovski DOCSIS 3.0. Belgija (92,6%) i Holandija (84,1%) su države sa najvećim učešćem NGA tehnologije, dok Grčka i Kipar imaju najmanje učešće. Srbija je sa učešćem od 61,2% iznad proseka EU, što je rezultat značajne zastupljenosti kablovskog DOCSIS 3.0. pristupa, koji čini 39% ukupnog broja pretplatnika fiksnog širokopojsnog pristupa Internetu u Srbiji, kao što se može videti sa sledeće slike.



Slika 3. Udeo NGA u ukupnom broju pretplatnika fiksnog širokopojsnog pristupa [1]

Struktura prihoda ostvarenih od fiksnog širokopojsnog pristupa Internetu, prati strukturu pretplatnika, pa u ukupnim prihodima najveće učešće imaju pristup

realizovan putem xDSL tehnologije i pristup putem kablovske distributivne mreže u iznosu od 47,2% i 41,3% respektivno, kao što se može primetiti na sledećoj slici [1].



Slika 4. Struktura prihoda fiksnog širokopojsnog pristupa Internetu prema načinu pristupa [1]

Takođe je izvršena analiza usluge distribucije medijskog sadržaja u Republici Srbiji i dati su podaci ostvareni u 2017. godini. RATEL kao nacionalno regulatorno telo za oblast elektronskih komunikacija, poštanskih usluga i informacione bezbednosti, vrši proveru parametara kvaliteta usluge distribucije medijskog sadržaja, na Euro DOCSIS 2.0/3.0 kablovskim distributivnim sistemima, što podrazumeva merenje nivoa signala, minimalnog odnosa signal/šum i *bit error rate*-a.

Uporedni prikaz broja korisnika, kao i stepen penetracije kablovskih sistema za 2014, 2015, 2016. i 2017. godinu dati su u sledećoj tabeli.

Tabela 1. Uporedni prikaz broja korisnika usluga distribucije medijskog sadržaja

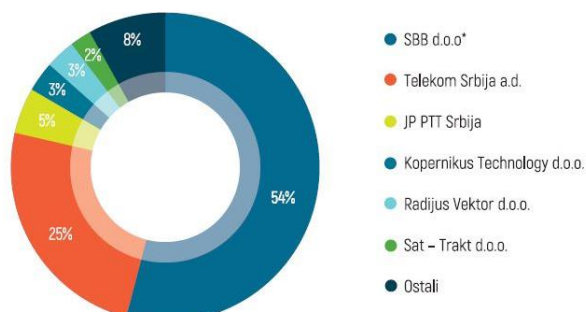
2014		2015		2016		2017	
Broj (u mil.)	Na 100 stan.	Broj (u mil.)	Na 100 stan.	Broj (u mil.)	Na 100 stan.	Broj (u mil.)	Na 100 stan.
1,50	20,95	1,60	22,55	1,66	23,51	1,70	24,13

Može se primetiti da je iz godine u godinu evidentan rast, kako broja korisnika, tako i prihoda koji se ostvaruju od usluga distribucije medijskog sadržaja.

Ukupan broj pretplatnika usluge distribucije medijskih sadržaja je u 2017. godini iznosio 1,70 miliona i povećan je u odnosu na prethodnu godinu za 2,4%, najviše zahvaljujući porastu pretplatnika usluga KDS. Oko 962.000 pretplatnika koristilo je uslugu distribucije medijskih sadržaja u okviru paketa usluga, i to najčešće sa uslugom širokopojsnog pristupa Internetu. Penetracija je iznosila 24,13% u odnosu na ukupan broj stanovnika, odnosno 68,28% ukupnog broja domaćinstava.

U 2017. godini bilo je 83 registrovanih operatera usluge distribucije medijskih sadržaja, koji su pružali usluge preko kablovske distributivne mreže (koaksijalne, hibridne i optičke), mreže bakarnih parica, satelitske distributivne mreže i bežične mreže [1]. Najveći operator distribucije medijskih sadržaja u Republici Srbiji u 2017. godini bilo je privredno društvo *Serbia Broadband – Srpske kablovske mreže d.o.o.* (SBB d.o.o.), sa tržišnim učešćem od 54% prema broju pretplatnika, uključujući i podatke operatora *IKOM d.o.o.*, koji je početkom 2018. godine pripojen operatoru *SBB d.o.o. Telekom Srbija a.d.*

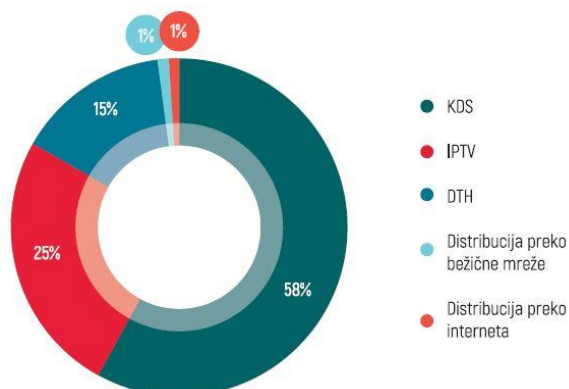
zauzima oko 25% tržišnog učešća u 2017. godini. Pored SBB d.o.o. i Telekom Srbija a.d, mogu se izdvojiti i privredna društva JP PTT Srbija, Kopernikus Technology d.o.o, Radijus Vektor d.o.o. i Sat - Trakt d.o.o. Mereno brojem pretplatnika, ovi operatori zajedno zauzimaju 92% tržišta distribucije medijskih sadržaja.



Slika 5. Tržišno učešće vodećih operatora u 2017. godini [1]

Najzastupljeniji način distribucije medijskih sadržaja u 2017. godini je i dalje bila distribucija preko kablovsko distributivnih sistema (KDS), sa oko 990 hiljada pretplatnika i beleži rast od 3% u odnosu na prethodnu godinu. Broj pretplatnika distribucije medijskih sadržaja preko bežične mreže takođe raste i u 2017. godini iznosi oko 21 hiljadu, kao i broj pretplatnika distribucije preko Interneta koji iznosi oko 15 hiljada.

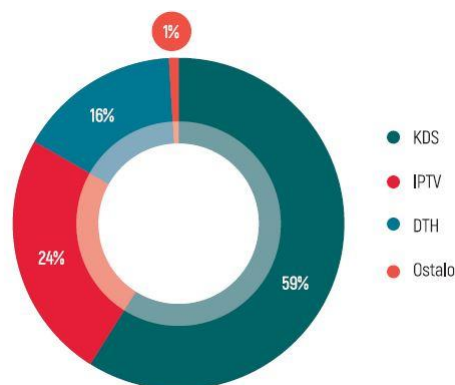
Procentualna raspodela pretplatnika prema vrsti distribucije se kod najzastupljenijih načina distribucije nije značajnije promenila u 2017. godini. Najviše pretplatnika je imala KDS sa 58% što se može videti na sledećoj slici.



Slika 6. Raspodela pretplatnika po vrsti distribucije u 2017. godini [1]

Raspodela prihoda prema načinu distribucije se takođe nije promenila u 2017. godini u odnosu na prethodnu godinu. Najveće učešće u 2017. godini u ukupnim prihodima od distribucije medijskih sadržaja imaju prihodi od KDS-a u visini od 59%, što je ilustrovano na sledećoj slici.

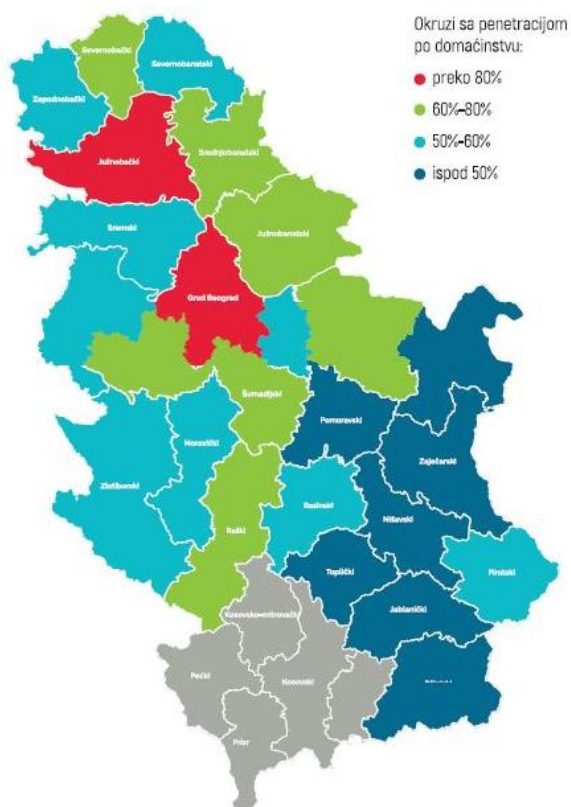
U tabeli 2, a zatim i na slici 8, dati su podaci o penetraciji usluge distribucije medijskih sadržaja po domaćinstvu, na nivou okruga u Republici Srbiji u 2017. godini [1].



Slika 7. Struktura prihoda od distribucije medijskih sadržaja u 2017. godini [1]

Tabela 2. Pregled broja pretplatnika distribucije medijskih sadržaja po okruzima Republike Srbije

Okrug	Broj domaćinstava	Ukupan broj pretplatnika	Penetracija po domaćinstvu (%)
Grad Beograd	606433	553879	91.33
Južnobački	223653	193296	86.43
Šumadijski	97096	67876	69.91
Srednjobanatski	68866	47895	69.55
Severnobački	71416	49170	68.85
Braničevski	59776	39406	65.92
Južnobanatski	101503	66581	65.60
Kolubarski	58973	36667	62.18
Raški	90515	55608	61.44
Severnobanatski	56800	34049	59.95
Mačvanski	100136	59811	59.73
Sremski	105031	61718	58.76
Zapadnobački	68888	40439	58.70
Podunavski	64155	37373	58.25
Rasinski	77270	43682	56.53
Moravički	72867	39899	54.76
Zlatiborski	94434	48497	51.36
Pirotski	34036	17260	50.71
Nišavski	128303	63841	49.76
Borski	45970	22676	49.33
Toplički	31184	14883	47.73
Pomoravski	71478	33903	47.43
Jablanički	66740	29303	43.91
Zaječarski	42445	18286	43.08
Pčinjski	49918	20819	41.71



Slika 8. Zastupljenost distribucije medijskih sadržaja po okruzima Republike Srbije [1]

3. ZAKLJUČAK

Ako bismo se zapitali gde bi kablovska industrija danas bila da nije DOCSIS-a i da kablovske mreže nisu sposobne za distribuiranje podataka, mogli bismo reći da za mnoge kablovske operatore danas najznačajniji procenat njihovog prihoda potiče direktno od distribucije podataka, iako je video tehnologija gradila kablovsku industriju.

Kablovske mreže nastoje da nastavu sa razvojem, da nastave da ispunjavaju tehničke zahteve za komunikacijom i da pružaju platformu za buduće inovacije.

Osnovna ideja kreatora DOCSIS-a bila je da se iskoristi već postojeća infrastruktura za distribuciju CATV signala. Zadovoljavajući zahteve korisnika, došlo se do poboljšanih verzija DOCSIS-a, koje omogućuju razmenu podataka sa većim protocima nego kod standardnih računarskih mreža.

Usavršavanjem postojećih CATV mreža doći će i do povećanja maksimalnog protoka podataka, omogućavajući korisnicima mnogo bolje servise nego do sada. Najnoviji STB uređaji koje nude kablovski operatori, sada imaju *cloud-based* korisnički interfejs i daljinsko video skladištenje, koji pomera kompleksne kompjuterske procese i keširanje podataka od korisničkog hardvera na mrežu i time potencijalno smanjuje potrošnju električne energije sa približno 25W na oko 15W, dakle oko 40%, što nije zanemarljiv procenat.

Kako se napredak tehnologije samo ubrzava, neizbežno je da će se DOCSIS i dalje razvijati kako bi se prilagodio novim uslugama, tehnologijama i zahtevima korisnika širom sveta. Kao informisani potrošači, naš je zadatak da budemo u toku sa ovim izmenama, kako bismo utvrdili šta bi nam najviše bilo od koristiti. Kako 4K i UHD video postaju sve češći, a bežične i žičane mreže se povećavaju u pogledu opsega i praktičnosti, verovatno je pretpostaviti da ima još mnogo uzbudljivih događaja koji će se tek odigrati u narednim decenijama, iako nije moguće u potpunosti predvideti razvoj različitih hardverskih i softverskih strategija, kojima će kablovske kompanije na kraju težiti. Nemoguće je precizno predvideti i tačan rast potražnje širokopojsnog pristupa. Cilj operatora će biti da razviju nove aplikacije, kako bi zadovoljili potražnju potrošača za novim uslugama.

4. LITERATURA

- [1] RATEL, „Pregled tržišta telekomunikacija i poštanskih usluga u Republici Srbiji u 2017. godini”, Beograd, 2018.

Kratka biografija:



Ljiljana Popović rođena je u Loznici 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Komunikacione tehnologije i obrada signala odbranila je 2019.god.
kontakt: ljiljana.popovic995@gmail.com



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

SIMULACIJA PRENOSA PODATAKA U 5G SISTEMU

SIMULATION OF DATA TRANSMISSION IN 5G SYSTEM

Nemanja Juzbaša, Željko Trpovski, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Rad prikazuje tehničke realizacije 5G sistema mobilne telefonije, sa osvrtom na softversku realizaciju prenosa podataka korišćenjem UI NYSIM simulatora.

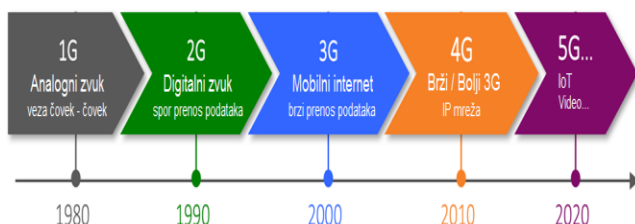
Ključne reči: 5G, New Radio, LTE-Advanced, OFDMA, gNB.

Abstract – The paper presents the technical realizations of the 5G mobile telephony system, with a focus on software implementation of data transmission using the UI NYSIM simulator.

Keywords: 5G, New Radio, LTE-Advanced, OFDMA, gNB.

1. UVOD

5G sistem je nastao kao namenski oblikovan sistem, postavljen u specifično-razvojnom prostoru, takavom da predstavlja makro potrebu modernog društva. Funkcionalnost sistema se ostvaruje primenom novih frekvencijskih opsega, te upotrebom razvijenih veoma složenih MIMO sistema, koji za krajnji cilj ostvaruje hiper brzu povezanost svih uređaja i korisnika. Sadržaj teksta u nastavku se odnosi na koncept implementacije upravo definisanog sistema, te predstavlja jedan vid kompromisa između želje korisnika i trenutnih tehničkih mogućnosti.



Slika 1. Kadenca razvoja mobilnih sistema prenosa

2. 5G OSNOVE

Osnove 5G New Radio sistema zasnivaju se na familiji čelijskih sistema među koje svrstavamo i prethodne tehnološke generacije poput GSM, EDGE, GPRS, HSPA, W-CDMA te LTE. 3GPP je standardizaciona organizacija na čijim protokolima su zasnovane prethodne generacije a na čijim tehničkim specifikacijama se vrši razvoj i održavanje novih tehnoloških komunikacionih sistema. 5G je naslednik trenutno eksploatisane 4. generacije mobilne telekomunikacione mreže, nadovezuje se na najnapredniji vid tehnologije pod nazivom LTE-A Pro.

NAPOMENA:

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

2.1 5G zahtevi

3GPP telo je postavilo zahteve koje 5G NR treba da ispunji, oni su sledeći:

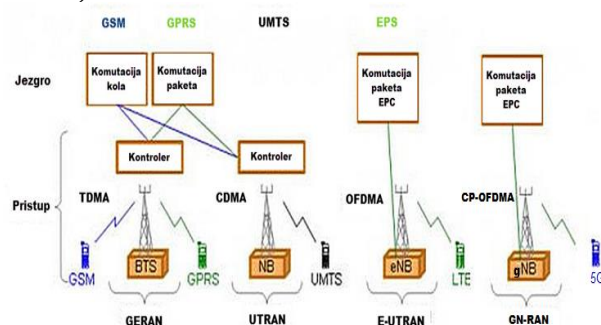
- ❖ Brzine prenosa za eMBB > 10 Gb/s,
- ❖ Masivne mašinske komunikacije - mMTC > 1M/km²,
- ❖ Odziv kod ultra pouzdanih komunikacija - URLLC < 1ms.

2.2 Faze standardizacije

Evolucija svih čelijskih mobilnih sistema prenosa podataka ogleda se u izdanjima 3GPP UMTS. Izdanja sa početnom oznakom Rel-4, do Rel-15 imaju sumirani uticaj na razvoj sistema koji danas kolokvijalno nosi naziv 5G.

2.3 Sistemska arhitektura

Prilikom evolucije mobilne tehnologije, tehnološka rešenja su zahtevala drugačiji pristup te se arhitektura sistema menjala. Potpuna komutacija paketa se koristi od četvrte generacije, čime se saobraćaj govora i podataka u potpunosti prenosi pomoću IP protokola a za pristup mreži se koristi OFDMA. Iskorak u odnosu na LTE-A Pro predstavlja peta generacija koja za pristup koristi CP-OFDMA (Cyclic-Prefix Orthogonal Frequency Division Multiple Access).

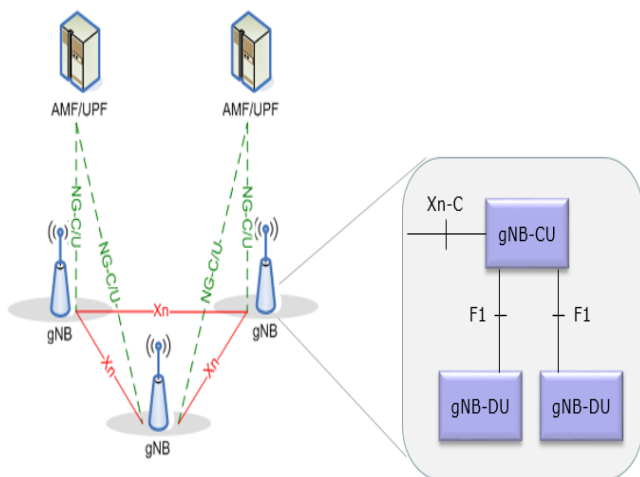


Slika 2. Arhitektura mobilnih sistema prenosa

Pristupnu mrežu 5G NR-a čini mreža međusobno povezanih baznih stanica pod nazivom gNB (next generation NodeB), koji su povezani preko Xn interfejsa, koji služi za signalizaciju i prosleđivanje paketa između gNB-ova tokom handover-a. Oni zajedno sačinjavaju flat arhitekturu.

Strukturu gNodeB bazne stanice delimo na:

1. gNodeB-CU (Centralized Unit), centralnu jedinicu koja obrađuje protokole i usluge koji nisu u stvarnom vremenskom toku,
2. gNodeB-DU (Distributed Unit), distribuiranu jedinicu koja obrađuje protokole i usluge na fizičkom sloju u realnom vremenu.

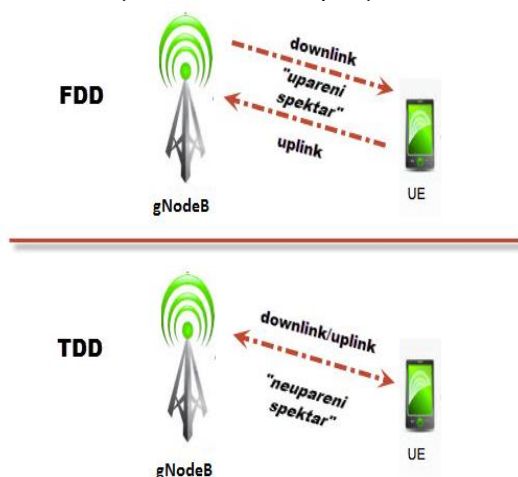


Slika 3. Struktura gNB

2.4 Aspekti 5G NR interfejsa

U 5G NR su podržana dva moda rada:

1. FDD (Frequency Division Duplex) i
2. TDD (Time Division Duplex).



Slika 4. FDD i TDD rad

Kod FDD moda imamo dva frekventijska nosioca, jedan za uplink transmisiju F_{ul} i jedan za downlink transmisiju F_{dl} . Obe transmisije mogu se obavljati simultano u okviru ćelije. Izolacija između date dve transmisije postignuta je korišćenjem predajno/prijemnih filtera tj. duplex filtera i dovoljno velikog razmaka između frekventijskih opsega. Kod TDD moda imamo jedan frekventijski nosilac, s tim što sada nije moguća simultana transmisija downlink-a i uplink-a tj. transmisije se dešavaju u različitim vremen- skim trenucima na nivou jedne ćelije.

Stoga kažemo da TDD radi u neuparenom spektru i zbog toga nije potreban duplex filter na terminalu.

Međutim s obzirom da uplink i downlink dele isti frekventijski opseg, signali ove dve transmisije mogu se međusobno interferirati.

Tada u TDD modu rada bazne stanice moraju biti međusobno sinhronizovane na nivou jedne ćelije. Ali zato kod FDD nemamo interferencije, zato što postoji frekventijski razmak između opsega i bazne stanice ne moraju biti sinhronizovane.

2.5 Frekventijski opsezi

5G NR nasleđuje sve frekventijske opsege definisane za LTE, konstantan rast potrebe za novim frekventijskim opsezima. Dva frekventijska opsega:

1. 5G FR1 - Niži frekventijski opsezi, do 6GHz, te
2. 5G FR2 - Viši frekventijski opsezi, koji nisu bili u upotrebi, alokacijom ovih opsega moguće je koristiti mm talase.

Niže frekventijske opsege, < 6 GHz (450 MHz – 6.000 MHz), karakteriše bolja teritorijalna bazna pokrivenost kao i propusni opseg kanala od 100 MHz.

Više frekventijske opsege, > 24 GHz (24.250 MHz – 52.600 MHz), karakteriše lošija teritorijalna bazna pokrivenost kao i propusni opseg kanala od 50 do 400 MHz. 5G teorijski može iskoristiti frekvencije do 300 GHz, zavisno od otvorenih frekventijskih opsega regulatornih tela.

3. 5G NR TEHNIKE PRISTUPA

Spektralna efikasnost ima presudnu ulogu, te su izabrani različiti pristupi za download i upload pristup, prevas- hodno zbog odnosa vršne i srednje vrednosti snage pre- netog signala.

U downloadu se koristi OFDMA gde je vrednost PAPR-a (Peak-to-Average Power Ratio) velika. Zbog potrošnje električne energije a time isplativosti, SC-FDMA (Single Carrier - Frequency Division Multiple Access) metoda se koristi za uplink.

3.1 OFDMA

Glavna prednost OFDMA u odnosu na druge tehnike prenosa sa jednim nosiocem jeste njegova sposobnost da se nosi sa selektivnom frekvencijom kanala za širokopojasnu komunikaciju. Podaci su podeljeni u paralelne tokove, a svaki je moduliran posebnim skupom podnosioca.

Širina opsega svakog podnosioca je postavljena na manju vrednost od širine opsega koherencije kanala. Drugim rečima, celokupan dodeljeni kanal je zauzet u vidu sume međusobno ortogonalnih uskopojasnih podnosioca.

Takođe, OFDM sistem vrlo efikasno koristi spektar, zbog načina pravougaonih preklapajućih nosioca čime se omogućava fleksibilno podešavanje frekvencija. Diskretni OFDM signal se izražava na sledeći način:

$$S_{OFDMA}[k] = \sum_{n=0}^{N-1} d_n \exp(j2\pi k \frac{n}{N}) \quad (1)$$

gde je d_n kompleksni simbol podataka podnosioca n , a N predstavlja ukupan broj podnosioca. OFDM može lako implementirati algoritmom inverzne brze furijeve transformacije (IFFT). Nakon toga se dodaje ciklični prefiks (CP), kopiranjem poslednjeg dela IFFT sekvence i dodavanjem na početak u funkciji zaštitnog intervala.

CP-OFDM sistemi su osetljiviji na greške u sinhronizaciji kao i OOB (Out-of-band-emissions).

Uklanjanje nedostataka predlažu sledeći kandidati za talasni oblik 5G NR sistema.

3.2 OFDMA 5G kandidati

Kandidate za 5G NR tehnike pristupa delimo na dve grupe:

1. šeme sa više nosioca,
2. šeme sa jednim nosiocem.

3.2.1 Šeme sa više nosilaca

1.) *Prozoriranje*: Prekide između susednih simbola, usled pravougaonog oblika prozora u vremenskom domenu izaziva visoku OOB za CP-OFDM. Prozorirani OFDM (W-OFDM) ravna ove oštre ivice na direktan način, sa malom složenosti. Osnovni opseg W-OFDM može se izraziti na sledeći način:

$$S_{W-OFDMA}[k] = \sum_{m=-\infty}^{+\infty} \sum_{n=0}^{N-1} d_{m,n} g[k - m(N + L_{CP} + L_{Ext})] \exp\left(j2\pi k \frac{n}{N}\right) \quad (2)$$

$d_{m,n}$ su složeni podaci koji se prenose na n -tom podnosiocu i m -ti OFDM simbol, L_{CP} predstavlja dužinu CP, L_{Ext} je vrednost proširenja prozora, a $g[k - m(N + L_{CP} + L_{Ext})]$ je funkcija prozoriranja.

2.) *Filtriranje podnosioca*, a^* FBMC (Filter bank multicarrier) daje dobru lokalizaciju frekvencijskog domena produžavanjem trajanja impulsa u vremenskom domenu nakon čega se koriste dizajnirani filtri za oblikovanje. Osnovni opseg FBMC/OQAM izražava se na sledeći način:

$$S_{FBMC-OQAM}[k] = \sum_{m=-\infty}^{+\infty} \sum_{n=0}^{N-1} d_{m,n} g\left[k - m\frac{N}{2}\right] \exp\left(j2\pi k \frac{n}{N}\right) \exp(j\Phi_{m,n}) \quad (3)$$

gde g predstavlja prototip filtra, $\Phi_{m,n}$ je dodatni fazni pojam podnosioca n i indeksu simbola m , koji se izražava kao $\frac{\pi}{2}(m + n)$. $d_{m,n}$ je stvarna vrednost jer se realni i imaginarni delovi prenose sa zakašnjenjem.

3.) *Filtriranje podnosioca*, b^* GFDM, Generalizovano frekvencijsko multiplexiranje, takođe primenjuje filtriranje podnosioca poput FBMC-a. Međutim, filteri za oblikovanje impulsa su kružno savijeni preko određenog broja simbola. GFDM simbol je predstavljen na sledeći način:

$$S_{GFDM}[k] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} d_{m,n} g_{m,n}[(k - mN) \bmod(MN)] \exp\left(j2\pi k \frac{n}{N}\right) \quad (4)$$

gde $d_{m,n}$ predstavlja kompleksne podatke koji se prenose pomoću n -tog podnosioca i m -tog podsimbola, $g_{m,n}$ prikazuje prototipni filter. Iako filteri prototipa FBMC moraju da ispunjavaju uslov ortogonalnosti, prototipovi GFDM nemaju data ograničenja.

4.) *Filtriranje MCM*, a^* UFMC, Univerzalni filter sa više nosilaca primenjuje podpojasno filtriranje kako bi se smanjio OOB. Podpojasno filtriranje, kompromis između filtriranja celog opsega i filtriranja podnosioca. bazni UFMC signal predstavlja se na sledeći način:

$$S_{UFMC}[k] = \sum_{b=0}^{B-1} \sum_{l=0}^{L-1} \sum_{n=0}^{N-1} d_{b,n}^b g_{b,l} \exp\left(j2\pi k \frac{n-1}{N}\right) \quad (5)$$

gde $d_{b,n}^b$ predstavlja kompleksne podatke koji se prenose pomoću n -tog podnosioca i b -tog podpojasa, a $g_{b,l}$ prikazuje frekvencijski ekvivalentnu funkciju prozoriranja filtra vremenskog domena (FIR). Stoga je svaka dužina bloka $L + N - 1$.

5.) *Filtrirani OFDM*, b^* f-OFDM filtrirajuća granularnost je fleksibilnija od UFMC-a. Podela u vremensko-frekvencijskoj mreži se podešava na osnovu različitog stanja kanala i upotrebe. Uzimajući u obzir ukupan broj blokova B , signal f-OFDM baznog opsega je predstavljen na sledeći način:

$$S_{f-OFDM}[k] = \sum_{b=0}^{B-1} \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} d_{b,m,n}^b g_{b,l} \exp\left(j2\pi k \frac{n-l-mL_{CP}}{N}\right) \quad (6)$$

gde $d_{b,m,n}^b$ predstavlja kompleksne podatke koji se prenose na b -tom bloku, n -tom podnosiocu i m -tom podsimbolu, $g_{b,l}$ prikazuje frekvencijski ekvivalentnu funkciju prozoriranja FIR filtra vremenskog domena na b -tom bloku, L_{CP} predstavlja CP vrednost. Kao što se može videti, f-OFDM održava CP za razliku od UFMC. Stoga je otporniji na ISI i treba manje komplikovan prijemnik.

3.2.2 Šeme sa jednim nosiocem

Šeme sa jednim nosiocem delimo na:

1. CP-DFT-s-OFDM, Predloženi su diskretni Furijeovi transformatori sa DFD-om kako bi se ublažili visoki PAPR problemi u CP-OFDM uz održavanje korisnih karakteristika.
2. ZT-DFT-s-OFDM, Interval zaštite je duboko implementiran u LTE sistemima, postoje dve opcije zaštite: normalani i produženi CP. Međutim, bazna stanica je podešena na samo jedan tip intervala zaštite jer se korišćenjem različitih intervala zaštite dobija različito trajanje simbola što posledično rezultira pogrešnim brojem simbola po frejmu.
3. UW-DFT-s-OFDM (Unique word DFT-spread OFDM) je još jedna šema sa jednim nosiocem koja koristi fleksibilne unutrašnje zaštitne pojase.

4. TRENUTNO STANJE

Analitički podaci datiraju od januara do marta 2019. godine, što je vremenski okvir neposredno pre pokretanja prvih komercijalnih 5G usluga u Južnoj Koreji, Sjedinjenim Američkim Državama i NR Kini. Testovi i analitički rezultati govore o velikim očekivanjima koje korisnici usluga baziranim na 5G sistemima očekuju. Statistika govori da evropske države predvode u implementaciji novih tehnologija, kako u kvantitetu tako i u kvalitetu LTE-A usluga.

Rani su dani za 5G, ali videli smo brojna pompezna komercijalna lansiranja u poslednjih nekoliko meseci, sa posebnim naglaskom na američko-azijsko rivalstvo. Većina tržišta je još uvek u procesu preraspodele opsega spektra za upotrebu 5G, među kojima je i tržište u Republici Srbiji.

4.1 5G mreže u eksploataciji

Imajući u vidu da je implementacija 4. generacije tekla usporeno, uvođenje komercijalne eksploatacije 5. generacije prvo uvode visokorazvijene zemlje zapadne Evrope. U oktobru 2019. godine sledeće evropske zemlje imaju aktivna komercijalna rešenja: Velika Britanija, SR Nemačka, Francuska, Italija, Švajcarska, Norveška, Austrija i Španija.

Sumirani efekat svih operatora sa prostora npr. Evrope nije dovoljan da se meri sa 3 mobilna operatora NR Kine: China Mobile, China Unicom i China Telecom. Pomenuti operatori imaju prednosti velikog broja korisnika koji se uvećava, kao i mogućnost da dobave i razviju proizvode po konkurentnijim cenama, u odnosu na zapadne operatore.

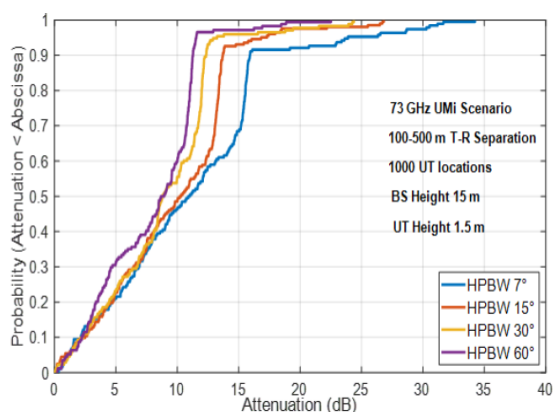
4.1.1. Test faze na prostoru Srbije

Sva tri operatora su pokrenula testne faze 5G NR. Telenor je pokrenuo prvu 5G baznu stanicu u Naučno-tehnološkom parku Beograd. Isti prostor je namenjen kao inkubator privredne aktivnosti, studenti Elektrotehničkog fakulteta Beograd koriste usluge NTP-a.

Na Univerzitetu Novi Sad, Fakultet tehničkih nauka se ističe kao predvodnik, sa sumiranim radom nekoliko katedrala i odseka. Izdvaja se rad ICONIC centra sa Katedre za telekomunikacije i obradu signala.

5G Mreža nije komercijalno puštena u Republici Srbiji.

5. SIMULACIJA 5G SISTEMA



Slika 5. SE postignut algoritmom hibridnog oblikovanja

Simulacija prenosa podataka vrši se korišćenjem softvera NYU WIRELESS. Simulacije se sprovode na polju milimetarskih talasa. Dostupne su opcije, kojim se može izabrati specifičan user-case, od kojih se može istražiti efekat ljudske blokade na generisane usmerene kanale, četiri seta Tx i Rx antena sa HPBW 7°, 15°, 30°, i 60° su odabrani u UMi scenariju na 73 GHz. Udaljenost T-R rastojanja se kreće od 100 m do 500 m. 1000 simulacija (tj. 1000 UT lokacija) izvode se za svako podešavanje Tx i Rx antene. Poređenje CDF-a senčenja gubitka za usmerene kanale u UMi NLOS scenariju, koristeći prikazana četiri seta Tx i Rx antene sa HPBW 7°, 15°, 30°, i 60° na slici 5 UT-ovi opremljeni užim HPBW antenama imaju veću verovatnoću da će doživeti ozbiljne

gubitke u blokadi. 31% UT-a sa 7° HPBW antenom doživljava gubitak senčenja veći od 15 dB.

6. ZAKLJUČAK

Prvi sistem 5. generacije, implementiran za korišćenje, je bio na raspolaganju u Rusiji 2018. godine tokom Svetskog fudbalskog kupa. U većini svetskih zemalja, ulaganja u uvođenje sistema 5. generacije se mere u desetinama milijardi dolara. Osim tehničkih aspekata, u nekoliko navrata je uvođenje sistema izazvalo političke nesuglasice između velike dvojke, Sjedinjenih Američkih Država i Narodne Republike Kine.

7. LITERATURA

- [1] 3GPP Technical Specification Group Radio Access Network; Evolved <http://www.3gpp.org/DynaReport/36-series.htm>
- [2] 5G intervju: Natali Delić, Vip mobile <https://pcpress.rs/5g-intervju-natali-delic-vip-mobile/>
- [3] Najbrži ikad prikazan internet u Srbiji - 5G! <https://mondo.rs/MobIT/Tech-Vesti/a1094432/mts-5G-mreza-5G-u-Srbiji-5G-mreza-u-Srbiji.html>
- [4] Telenor rolled-out first 5G station in Serbia <https://www.telenor.rs/en/about-telenor/news/news-and-press-releases/telenor-rolled-out-first-5g-base-station-in-serbia/#>
- [5] Uputstvo za upotrebu NYUSIM simulatora
- [6] Opensignal: Mobile Analytics and Insights <https://www.opensignal.com/global>

Kratka biografija:



Nemanja Juzbaša rođen je u Karlovcu, R. Hrvatska 1987. god. Upisao se na Fakultet tehničkih nauka u Novom Sadu, na Katedru za komunikacione tehnologije i obradu signala. Diplomski rad na temu »Perspektiva razvoja i uvođenja pete generacije mobilne telefonije« odbranio je 2016. godine.



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

**REALIZACIJA ELEKTRONSKOG SISTEMA ZA OČITAVANJE
JEDNODIMENZIONALNOG BAR-KODA****IMPLEMENTATION OF THE ELECTRONIC SYSTEM FOR READING THE ONE-
DIMENSIONAL BAR CODE**

Dragan Raca, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Tema ovog rada je realizacija sistema za očitavanje bar-kodova i analiza problema koji su se javili pri njoj. U radu je data mehanička konstrukcija, elektronski sistem i algoritam rada čitača. Prikazan je rezultat očitavanja i predložena su poboljšanja datog sistema.

Ključne reči: bar-kod, sistem za očitavanje, Atmega328P, STSPIN820, TSL250R, laser

Abstract – This article presents the realization of bar-code scanning system and analyses the problems that appeared. The paper presents the mechanical construction, electronic system and algorithm of reader operation. The result of the reading is presented and improvements of the given system are suggested.

Keywords: barcode, scanning system, Atmega328P, STSPIN820, TSL250R, laser

1. UVOD

Tema ovog rada zasnovana je na široko rasprostranjenom električnom uređaju, koji se koristi u svim trgovinskim centrima, a čiji rad pruža brzu i pouzdanu obradu informacija. Reč je o uređaju za očitavanje bar-kodova tj. bar-kod čitaču. Razvoj tehnologije bar-kodova i uređaja za njihovo očitavanje pružio je mogućnost zapisivanja velikog broja informacija na malom prostoru, kao i brzo i jednostavno očitavanje tih informacija i njihovo razumevanje.

Zapisivanje informacija upotrebom određenih elemenata prema unapred definisanim pravilima predstavlja kodovanje. Očitavanje tako formiranih kodova i njihovo prebacivanje u drugi oblik prikaza koji daje neko značenje je dekodovanje.

Ovde će biti realizovan niskobudžetni (engl. *Low-cost*) sistem za očitavanje kodova, čiji princip rada se zasniva na osnovnim principima rada savremenog bar-kod čitača. Za potrebe demonstracije realizovanog sistema definisan je pojednostavljeni kod, prilagođen mogućnostima realizovanog sistema.

2. BAR-KOD I NJEGOVO FORMIRANJE

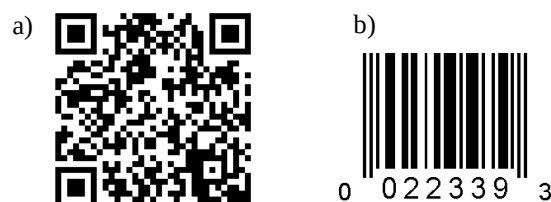
Bar-kod predstavlja grafički zapis određene informacije, čije se značenje može dobiti primenom odgovarajućeg algoritma za dekodovanje. On obično sadrži informaciju o proizvodu na kome se nalazi ali može sadržati bilo koji drugi podatak, internet adresu, e-mail adresu, lične podatke neke osobe i slično.

Postoji mnogo različitih tipova bar-kodova. Oni se razlikuju po količini i vrsti informacije koju skladište, prema algoritmu formiranja i dekodiranja, metodama izrade, primeni na tržištu, karakterističnim dimenzijama i još mnogo toga. Zbog tako velikih varijacija razvijene su simbologije, odnosno standardi koji definišu karakteristike štampanog simbola – njegovu strukturu i dimenzije, algoritam prema kome se taj simbol formira i dekoduje, kao i algoritam provere ispravnosti očitavanja. Svaka od simbologija prilagođena je različitim zahtevima tržišta.

Prema načinu zapisivanja bar-kodovi su podeljeni u dve osnovne grupe [1]:

1. Matrični ili dvodimenzioni (2D) kodovi - Predstavljaju dvodimenzionalni zapis informacije. To znači da se njihov sadržaj dobija čitanjem duž horizontalne i vertikalne ose. Na Slici 2.1a) prikazana je jedna od simbologija ove grupe kodova, QR (Quick Response).

2. Linearni ili jednodimenzioni (1D) kodovi – Predstavljaju jednodimenzionalni zapis informacije. Njihov sadržaj dobija se čitanjem duž horizontalne ose. Na Slici 2.1b) prikazana je jedna od velikog broja simbologija, po imenu UPC-E (Universal Product Code E) koji se može pronaći na proizvodima u trgovini.



Slika 2.1 Vrste bar-kodova: a) Matrični kod (Simbologija: QR). b) Linearni kod (Simbologija: UPC-E).

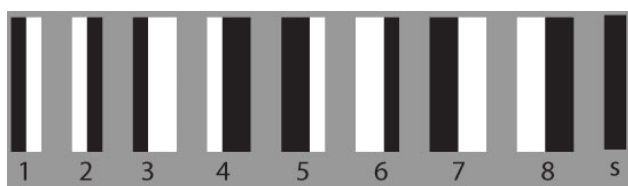
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Jovan Bajić, docent.

2.1 Formiranje bar-koda

Prateći strukturu kodova linearne simbologije i kod za potrebe ovog rada formiran je naizmeničnim postavljanjem crnih i belih linija duž horizontalne ose prostiranja. Definisane su dimenzije koda i njegovih elemenata, kao i algoritam formiranja. Međutim, on je jednostavniji u odnosu na standardne simbologije i nema algoritam za proveru ispravnosti očitavanja.

Sastoji od šest elemenata i njegova struktura se može prikazati kao niz "SBBBBBS". Prvi i poslednji element niza "S" je *start/stop* element koji označava početak i kraj koda, a preostala četiri elementa "B" čine njegov sadržaj, to su brojevi od 1 do 8. Brojevi su formirani kombinacijom jedne bele i jedne crne površine, a "S" element je predstavljen jednom crnom površinom. Na Slici 2.1 prikazani su osnovni elementi za formiranje bar-koda.

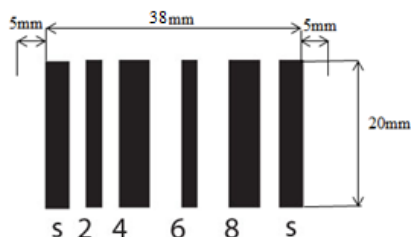


Slika 2.1 Osnovni elementi za formiranje bar-koda.

Za brojne elemente definisane su dve vrednosti širina pravougaonih površina čije vrednosti iznose 2mm i 4mm, a njihova visina 20mm. Dok je za start/stop element definisana širina 3mm i visina 20mm. Dakle, maksimalna širina bar-koda iznosi 38mm. Kako bi očitavanje bilo uspešno izvršeno, pre početka i nakon završetka koda, potrebno je ostaviti prazan prostor širine 3 do 5mm.

Algoritam za formiranje koda je veoma jednostavan. Potrebno je izabrati četiri brojna simbola i postaviti ih između *start* i *stop* elementa, pri čemu se crne i bele površine moraju naizmenično smenjivati.

Taj uslov mora biti ispunjen da se ne bi formirale širine veće od 4mm. Znači da se za formiranje koda mogu upotrebiti samo parni ili samo neparni brojevi, što znatno smanjuje broj kombinacija. Ali to u ovom slučaju ne predstavlja problem. Na slici 2.2 prikazan je kod formiran od parnih brojeva i njegove maksimalne dimenzije.



Slika 2.2 Bar-kod i njegove dimenzije.

3. OPIS I REALIZACIJA SISTEMA

Realizovani sistem može se rasčlaniti na tri osnovne celine:

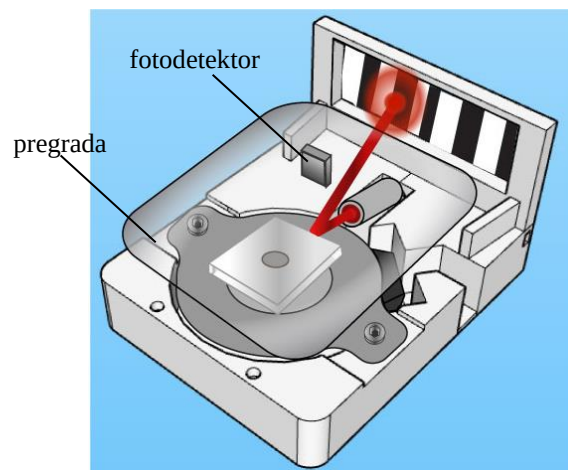
1. Mehanička arhitektura
2. Elektronski sistem
3. Programska podrška

3.1 Mehanička arhitektura

Mehanički deo sistema obuhvata kućište sa svojim elementima, na osnovu kojeg se informacija predstavljena grafičkim simbolom pretvara u električni signal.

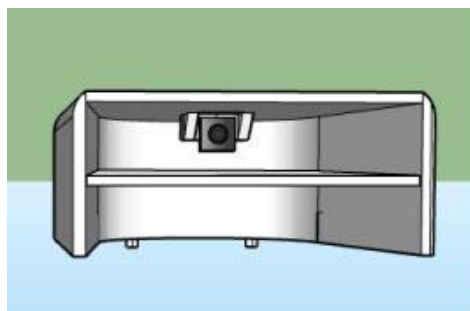
Model kućišta formiran je u softverskom alatu *SketchUp*, a zatim je izrađeno na 3D štampaču. Kućište se može podeliti na dve fizički odvojene celine.

Prva celina predstavlja osnovu kućišta. Na nju je pričvršćen motor sa ogledalom na svojoj osovinu. Ispred njega, u ravni motora, postavljen je laser talasne dužine 650nm. Iza lasera formiran je pravougaoni okvir za prislanjanje bar kodova. Raspored elemenata prikazan je na Slici 3.1.



Slika 3.1 Unutrašnjost kućišta i njegovi elementi: bipolarni koračni motor sa ogledalom, laser, zaštitna pregrada i fotodetektor.

Druga celina predstavlja poklopac kućišta. Sa njegove gornje strane postavljen je fotodetektor TSL250R [3] kao na Slici 3.2, a ispod njega postavljena je pregrada koja sprečava uticaj direktne laserske svetlosti na fotodetektor.



Slika 3.2 Poklopac kućišta sa fotodetektorom TSL250R i zaštitnom pregradom.

Očitavanje bar-koda zasnovano je na sledećem principu: Laserski snop pada na bočnu ivicu ogledala koje je postavljeno na osovinu koračnog motora i rotira oko svoje ose. Na taj način snop svetlosti reflektovan sa ogledala prelazi preko svake tačke bar-koda prislonjenog na otvor kućišta i kao rezultat kretanja snopa formira horizontalnu liniju. Svetlost reflektovana sa bele površine bar-koda više nije usmerena jer vrši difuznu refleksiju, pa se svaka tačka na bar-kodu može posmatrati kao tačkasti izvor svetlosti. Svetlost koja pada na crnu površinu koda biva apsorbovana ili pak delimično apsorbovana, pa je njen

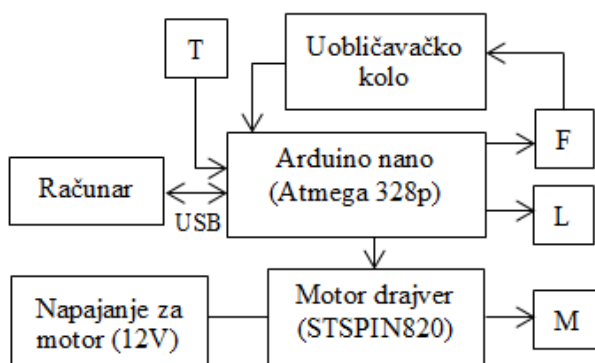
intenzitet približno jednak nuli [2]. Reflektovana svetlost dospeva do fotodetektora koji srazmerno sa njenim intenzitetom na svom izlazu daje signal određene amplitude. Analizom rezultata dobijenih očitavanjem na fotodetektoru određuje se širina crne ili bele površine, a na osnovu toga formiraju odgovarajući brojevi [4].

Iako se očitavanje zasniva na jednostavnom principu, potrebno je obratiti pažnju na veliki broj faktora koji utiču na njegovu uspešnost:

1. Raspored osnovnih elemenata (lasera, motora, ogledala, fotodetektora)
2. Širinu i intenzitet laserskog snopa
3. Vrstu motora i njegovu ugaonu brzinu (brzina obrtaja)
4. Uticaj okoline
5. Vrstu površine na kojoj se nalazi bar-kod

3.2 Elektronski sistem

Podeljena je na tri osnovna dela: mikrokontroler *Atmega328p* sa svojim okruženjem, integrisan na pločici *Arduino nano*, drajver za upravljanje koračnim motorom baziran na kontroleru *STSPIN820* [5] i električno kolo za uobličavanje signala. Napon napajanja celokupnog električnog kola iznosi 5V i dovodi se sa računara putem USB kabla. Napon napajanja koračnog motora iznosi 12V i dobija se pomoću integrisanog kola *buck/boost* konvertora. Blok šema sistema prikazana je na Slici 3.3.



Slika 3.3. Blok šema realizovanog sistema: *M*-bipolarni koračni motor, *L*-laser, *F*-fotodetektor *TSL250R*, *T*-taster.

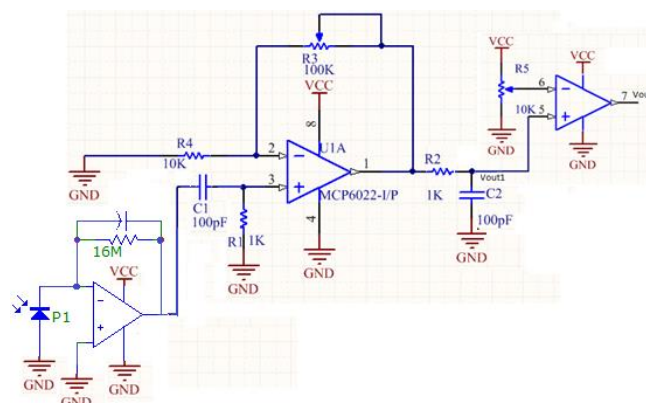
Proces očitavanja započinje se pritiskom na taster. Kontroler putem drajvera upravlja radom koračnog motora i na svom ulaznom pinu dobija signal sa uobličavačkog kola, koji se dalje analizira. Konačan rezultat očitavanja šalje se putem serijske komunikacije na terminal računara.

Električni signal dobijen na izlazu fotodetektora *P1* sa integrisanim transimpedansnim pojačavačem najpre se dovodi na visoko propusni filter u cilju otklanjanja *offseta* tj. *DC* komponente signala nastalog usled delovanja svetlosti spoljašnje sredine. Pošto je dobijeni signal veoma male amplitude on se dalje dovodi na pozitivan ulaz neinvertujućeg operacionog pojačavača. U povratnoj sprezi pojačavača postavljen je trimer potencijometar otpornosti $R_t=100k\Omega$ koji pruža mogućnost finog podešavanja pojačanja čija se vrednost dobija na osnovu izraza za pojačanje neinvertujućeg pojačavača:

$$A = 1 + \frac{R_t}{R} \quad (3.1)$$

pri čemu je $R=10k\Omega$, a R_t je podešena otpornost trimer potencijometra.

Pojačani signal se dalje propušta kroz nisko propusni filter čija je uloga uklanjanje šumova visokih frekvencija, a zatim na komparatorsko kolo čiji je izlaz postavljen na ulazni pin kontrolera *Atmega328p* koji je upotrebljen za analizu dobijenog signala. Komparatorsko kolo sastoji se od operacionog pojačavača koji je doveden na jednosmerni izvor napona napajanja vrednosti 5V. Na njegov pozitivan (+) ulaz doveden je signal sa niskopropusnog filtra V_{out1} , a na negativan (-) ulaz postavljen je višebrotni trimer potencijometar otpornosti $R_t=10k\Omega$ koji pruža fino podešavanje vrednosti referentnog napona komparatora u opsegu od 0 do 5V. Ako je signal doveden sa *RC* kola veći od referentnog napona, na izlazu komparatora dobijamo visoku vrednost, a ako je manji - nisku. Na ovaj način, na izlazu uobličavačkog kola formira se signal V_{out} pravougaonog oblika. Na Slici 3.4. prikazana je električna šema prethodno opisanog uobličavačkog kola.



Slika 3.4. Električna šema uobličavačkog kola. *P1*-fotodetektor sa transimpedansnim pojačavačem.

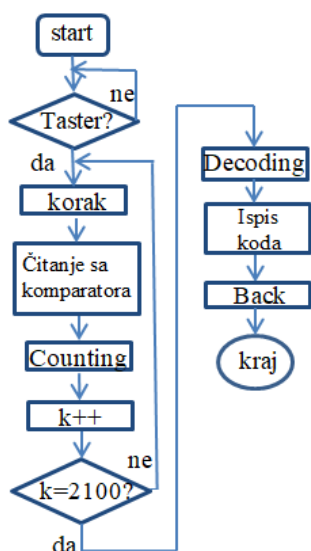
Zbog uzastopnog smenjivanja crne i bele površine na bar-kodu, na izlazu fotodetektora formira se signal frekvencije 1.8Hz. Ta frekvencija direktno zavisi od širine ovih površina i njihovog broja. Prelazak preko jedne bele i jedne crne površine čini jednu period signala. Upotrebom izraza 3.2 za određivanje frekvencije mogu se izračunati vrednosti za *R* i *C* komponente.

$$f = \frac{1}{2\pi RC} \quad (3.2)$$

Kod visokopropusnog filtra za graničnu frekvenciju uzeta je vrednost $f_{CR} = 0,18Hz$, a kod niskopropusnog filtra $f_{CR} = 18Hz$.

4. PROGRAMSKA PODRŠKA

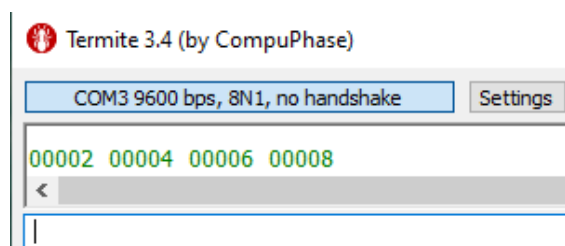
Program za kontolu rada mikrokontrolera je napisan u *C* programskom jeziku, primenom razvojnog okruženja *Atmel Studio 7.0*. Formiran je po standardnom protokolu: najpre je izvršena inicijalizacija logičkih blokova kontrolera a zatim su realizovane funkcije za rad sistema koje su kombinovane u glavnom (eng. main) delu programa. Na Slici 4.1 prikazan je algoritam rada.



Slika 4.1 Algoritam rada sistema.

Program proverava stanje ulaznog pina kontrolera koji je povezan sa tasterom. Njegovim prelaskom u nisko stanje pokreće se motor koji izvršava 2100 koraka, kako bi laserom prešao preko celog bar-koda. Nakon svakog koraka poziva se funkcija *Counting* koja broji visoka i niska stanja dobijena na izlazu komparatora i nakon svake promene stanja zapisuje vrednosti brojača u niz. Tako da elementi datog niza predstavljaju širinu bele ili crne površine. Motor se zaustavlja nakon prelaska preko bar-koda tj. kada brojač k dostigne vrednost 2100 i poziva se funkcija *Decoding* koja na osnovu elemenata prethodno popunjenog niza formira brojeve očitane sa bar-koda. Prvi i poslednji element se odbacuju jer su to *start* i *stop* simboli koda, a ostalih osam koristi se za formiranje brojeva. Na kraju se poziva funkcija *Back* za povratak motora u početni položaj.

Na Slici 4.2. prikazan je isečak terminala na kome su prikazani brojevi dobijeni kao rezultat očitavanja bar-koda sa Slike 2.2 sastavljenog od parnih brojeva 2,4,6,8.



Slika 4.2 Rezultat očitavanja bar-koda sa slike 2.2 prikazan u terminalu.

5. ZAKLJUČAK

Realizovani sistem pruža mogućnost analize velikog broja parametara koji utiču na uspešnost očitavanja bar-koda, te je jedan deo cilja uspešno izvršen.

Međutim, ako se posmatra funkcionalnost sistema, tj. ponovljivost uspešnog očitavanja bar-koda, ona nije ostvarena. Rezultati očitavanja ne poklapaju se uvek sa sadržajem bar-koda što je verovatno problem i u praksi. Te se koristi neka vrsta provere ispravnosti koda koja je ovde izostavljena, a koja pruža mogućnost odbacivanja pogrešnog rezultata očitavanja. Iz toga zaključujemo da su bar-kod i njegov čitač u velikoj meri povezani, tj. da očitavanje zavisi od simbologije.

Velika mana električnog kola za uobličavanje signala jeste ručno podešavanje praga komparacije tj. referentnog napona komparatora koji direktno zavisi od amplitude signala dobijenog na izlazu fotodetektora, a amplituda od položaja bar-koda i refleksije svetlosti. Najbolje rešenje bilo bi realizacija sistema sa električnim kolom za praćenje promene amplitude signala koje bi u odnosu na njenu vrednost prilagođavalo referentni napon na komparatoru.

Da bi se postiglo očitavanje realnog bar-koda, potrebno je smanjiti širinu laserskog snopa, da bude manja od minimalne širine bele ili crne površine koda.

6. LITERATURA

- [1] J. Phaniteja, P. Derin: „Evolution of Barcode”, International journal for development of computer science & technology (pristupljeno: 22.10.2019.)
- [2] S.O. Kasap: „Optoelectronics and Photonics”, 2 end ed. 2013, Harlow
- [3] Dokumentacija fotodetektora TSL250R: <http://www.farnell.com/datasheets/301932.pdf> (pristupljeno: 22.10.2019.)
- [4] J. Bajić, Beleške sa predavanja iz predmeta Primenjena optoelektronika: „Skeneri, Štampači, Primena lasera” (2018.)
- [5] Dokumentacija drajvera za motor STSPIN820: <https://www.st.com/resource/en/datasheet/stspin820.pdf> (pristupljeno: 22.10.2019.)

Kratka biografija:



Dragan Raca rođen je u Kikindi 1993. god. Master rad na Fakultetu tehničkih nauka iz oblasti Optoelektronika – Primenjena elektronika odbranio je 2019.god. kontakt: dragan044@rocketmail.com

MAGNETSKO POLJE SIMETRIČNOG TROFAZNOG VODA, SASTAVLJENOG OD „U” SABIRNICA**MAGNETIC FIELD PRODUCED BY SYMMETRIC THREE PHASE SYSTEM CONTAINING „U” SHAPED BUSBARS**Jadranka Ivković, Karolina Kasaš-Lažetić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je izvršena analiza raspodjele intenziteta vektora magnetske indukcije u okolini simetričnih trofaznih sabirnica u prisustvu neutralnog voda za četiri različita međusobna položaja „U” sabirnica i različite fazne stavove struja u njima. Dodatno je ispitan uticaj oklopljavanja sabirnica na raspodjelu magnetskog polja. Proračuni su izvršeni u programskom paketu „COMSOL Multiphysics”, koji radi na bazi metoda konačnih elemenata. Dobijeni rezultati su pokazali da promjena faznog stava jačina struja u sabirnicama kao i promjena položaja sabirnica dovodi do promjene raspodele magnetskog polja, a prisustvo oklopa smanjuje magnetsko polje izvan posmatranog sistema.

Ključne reči: Trofazne „U” sabirnice sa neutralnim provodnikom, Oklopljene sabirnice, Magnetsko polje, Metod konačnih elemenata (MKE)

Abstract – The paper analyzes the magnetic flux density distribution in the vicinity of symmetric three-phase busbars in the presence of a neutral line for four different intermediate positions of „U” busbars and different phase arrangement of currents. The effect of the busbars shielding on the distribution of the magnetic field is also examined. Calculations were performed in the software package „COMSOL Multiphysics” that is based on the finite element method. The obtained results showed that the current phase shift, as well as the position of the busbars change the magnetic field distribution and the enclosure reduces the magnetic field outside the observed system.

Keywords: Three phase „U” shaped busbars with neutral conductor, Shielded busbars, Magnetic field, Finite element method (FEM).

1. UVOD

Elektroenergetski sistem se sastoji od velikog broja dijelova koji su međusobno povezani i čine jednu složenu i funkcionalnu celinu. Sabirnice su jako važan dio svakog elektroenergetskog sistema i doprinose normalnom pogonu sklopnih postrojenja jer su svi vodovi povezani sa njima. Struje uspostavljene u energetske simetričnim trofaznim sabirnicama stvaraju oko sebe niskofrekventno (NF) magnetsko polje u kojima se mogu naći ljudi

i električni uređaji. Nastala magnetska polja mogu preko magnetskih sprega negativno da utiču na rad mnogih uređaja, tako da je praćenje nivoa magnetskog polja u okolini trofaznih sabirnica neophodno, radi očuvanja zdravlja ljudi i ispravnog funkcionisanja uređaja koji se u njima nalaze.

Sabirnice se najčešće izvode od neizolovanih bakarnih ili aluminijskih provodnika ili iz njihovih legura. Po obliku mogu biti pljosnati, okrugli ili savijeni u „U” profil. „U” profili su veoma pogodni za montažu u postrojenjima i imaju veoma dobre mehaničke karakteristike [1].

Zbog svega toga, cilj ovog rada jeste da se ispita uticaj, promene faznog stava struja u sabirnicama i uticaj prisustva oklopa oko sabirnica na raspodelu magnetskog polja u okolini trofaznih simetričnih „U” sabirnica u prisustvu neutralnog voda. Proračuni su izvršeni u programskom paketu COMSOL Multiphysics [2] posmatrajući dvodimenzionalan uprošćen model sistema. Rezultati su prikazani grafički, u vidu prostorne raspodele magnetskog polja kao i duž pravaca postavljenih paralelno sa osnovama sabirnica. Rezultati su ukazali na to da promenom faznih stavova struja u sabirnicama, može da se utiče na raspodelu magnetskog polja i da korišćenje oklopa značajno redukuje magnetsko polje van oklopa.

2. TEORIJSKA OSNOVA

Elektromagnetsko polje koje stvaraju sabirnice su mrežne učestanosti (na ovim prostorima 50 Hz) i kao takva spadaju u vremenski sporopromjenljiva polja. Na osnovu toga, analiza kako magnetske tako i električne komponente istog elektromagnetskog polja može da se vrši ponaosob, posmatranjem nezavisno vremenski sporopromjenljivog magnetskog polja i vremenski sporopromjenljivog električnog polja.

2.1. Osnove proračuna

Za opisivanje magnetskog polja, koristi se vektor magnetske indukcije, $\vec{B}$, dok se električno polje opisuje vektorom jačine električnog polja, $\vec{E}$. Određivanje ovih vektorskih veličina se najčešće vrši korišćenjem dva potencijala, električnog skalar potencijala V i magnetskog vektor potencijala, $\vec{A}$. Vektor jačine električnog polja i vektor magnetske indukcije, izraženi preko potencijala mogu da se predstave preko relacija:

$$\begin{aligned}\vec{E} &= -\nabla V - \frac{\partial \vec{A}}{\partial t} \\ \vec{B} &= \nabla \vec{A}\end{aligned}\quad (1)$$

Pošto cijeli sistem može da se smatra linearnim, sa prostoperiodičnim strujama u sabirnicama, definisanje problema i njegovo rješavanje mogu da se izvrše u kom-

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Karolina Kasaš-Lažetić, docent.

pleksnom domenu. Dekartov pravougli koordinatni sistem je najpogodnije postaviti tako, da z osa bude postavljena normalno na poprečni presjek sabirnice (Slika 1).

U tako izabranom koordinatnom sistemu, veličine od interesa, vektor gustine struje i magnetski vektor potencijal, imaju samo z komponentu, a mijenjaju se po koordinatama x i y. Magnetski vektor potencijal se dobija kao rješenje parcijalne diferencijalne jednačine [3], [4]:

$$\nabla^2 \vec{A} - j\omega\mu\sigma\vec{A} = -\mu\vec{J}. \quad (2)$$

U Dekartovom koordinatnom sistemu, jednačina (2) prelazi u skalarni oblik,

$$\frac{\partial^2 \underline{A}_z}{\partial x^2} + \frac{\partial^2 \underline{A}_z}{\partial y^2} - j\omega\mu\sigma \underline{A}_z = -\mu \underline{J}_z. \quad (3)$$

Za dobijanje partikularnog rješenja ove diferencijalne jednačine, neophodno je postaviti granične uslove. U prikazanom dvodimenzionalnom sistemu, intenzitet vektora elektromagnetskog polja opada sa rastojanjem od sistema, tako da se kao granični uslov usvaja da je magnetski vektor potencijal jednak nuli van zamišljenog kružnog prostora, čije su dimenzije mnogo veće od linearnih dimenzija poprečnog presjeka analiziranog sistema.

Usled indukovano električnog polja, u provodnicima se uspostavlja i indukovana komponenta struje, $\underline{J}_{indz}$, tako da u provodniku postoji struja ukupne gustine

$$\underline{J}_{ukz} = \underline{J}_z + \underline{J}_{indz}(x, y). \quad (4)$$

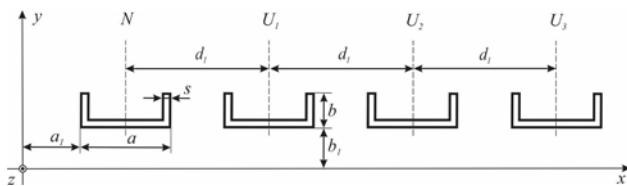
Indukovane struje nastaju usled istovremenog dejstva površinskog efekta i efekta blizine, i dovode do neravnomjerne strujne raspodele u sabirnicama a samim tim utiču i na raspodelu magnetskog polja u sabirnicama i njihovoj okolini.

3. MODEL

Da bi se izvršila analiza raspodjele intenziteta vektora magnetske indukcije u okolini simetričnih trofaznih „U” sabirnica sa neutralnim vodom, u razmatranje je uzet sistem sastavljen od „U6” sabirnica, čije su standardizovane dimenzije prikazane u tabeli 1, prema oznakama datih na slici 1 [5].

Tabela 1. Dimenzije standardizovanih „U6” sabirnica

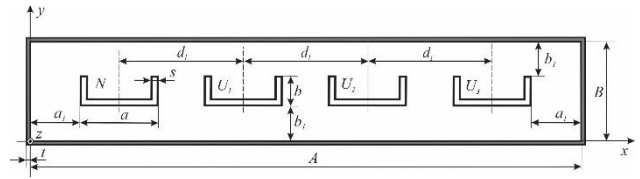
Tip sabirnice	a[mm]	b[mm]	s[mm]	d ₁ [mm]	S[mm ²]
U6	60	30,0	4	25	448



Slika 1. Raspored i dimenzije neoklopljenih „U6” sabirnica sa neutralnim vodom

Oznake N, U1, U2 i U3 odgovaraju neutralnom vodu i sabirnicama prve, druge i treće faze respektivno. Rastojanje neutralne sabirnice do y ose iznosi $a_1 = 35$ mm, dok su sve sabirnice na podjednakom rastojanju od x ose,

$b_1 = 45$ mm. Debljina oklopa iznosi $t = 3$ mm. Izgled oklopljene sabirnice prikazan je na slici 2.



Slika 2. Raspored i dimenzije oklopljenih „U6” sabirnica sa neutralnim vodom

Problem je rešavan korišćenjem programskog paketa COMSOL Multiphysics i pripadajućeg AC/DC modula. Postupak modelovanja prati određen šablon; postoji hronološki redoslijed koraka koje treba ispratiti. Prvo treba da se definiše prostor u kojem će se izvršiti analiza, a zatim modul koji pokriva fizičke karakteristike sistema. Za posmatran model to je 2D, Magnetic Fields (mf) – Frequency Domain simulacija. Kada je izabran odgovarajući modul i tip simulacije, prvo treba da se definiše geometrija sistema. Nakon toga se dijelovima sistema pridružuju njihove elektromagnetske karakteristike i definišu sve ulazne veličine od interesa. U narednom koraku se postavljaju granični uslovi, a nakon toga se definiše mreža konačnih elemenata i rješava problem. Program izračunava raspodjelu magnetskog vektor potencijala, pomoću kojeg se zatim mogu odrediti ostale veličine od interesa. U posmatranom modelu to je bila raspodjela intenziteta vektora magnetske indukcije unutar i u okolini sabirnica. U radu su posmatrane sabirnice načinjene od bakra. Specifična provodnost bakra je uzeta iz biblioteke paketa, i iznosi $\sigma_{Cu} = 5,998e7$ S/m. Pošto se u ovom modelu sabirnice nalaze u vazduhu, električne i magnetske karakteristike okruženja su podešene na odgovarajuće vrijednosti ($\epsilon_{r0} = 1$, $\mu_{r0} = 1$, $\sigma_0 = 0$). Cjelokupna posmatrana oblast je ograničena kružnim „oklopom” poluprečnika $R_{gr} = 2$ m, i debljine $d = 50$ mm u skladu sa opcijom definisanja domena „beskonačnih elemenata”, pomoću kojeg se beskonačan domen modeluje oblašću konačnih dimenzija.

Oklop koji se u nekim delovima ispitivanja postavlja oko sabirnica je u ovom primjeru od aluminijuma i njegova specifična provodnost je takođe uzeta iz biblioteke paketa, $\sigma_{Al} = 3,771e7$ S/m. Uloga oklopa je da smanji magnetsko polje izvan njega u oblasti od interesa.

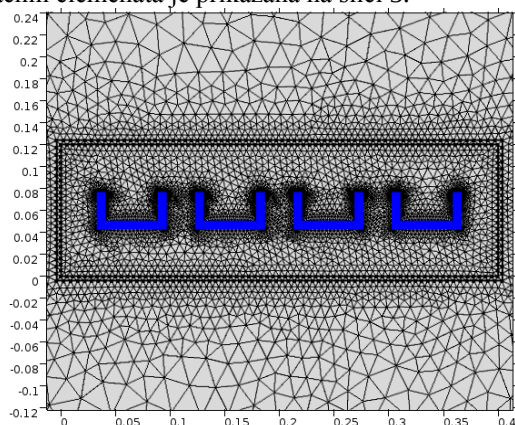
U posmatranom modelu, analiza je izvršena pretpostavljajući da je trofazni sistem simetričan, znači da su efektivne vrednosti jačina struja u sve tri sabirnice istog intenziteta, jednake nominalnoj jačini struje u posmatranim sabirnicama [5] i istog su smjera. Strujno opterećenje faznih provodnika je, u skladu sa tim, definisano kao

$$\begin{aligned} \underline{I}_1 &= I_1 e^{j0^\circ} [kA], \\ \underline{I}_2 &= I_1 e^{-j120^\circ} [kA], \\ \underline{I}_3 &= I_1 e^{-j240^\circ} [kA]. \end{aligned} \quad (5)$$

Kompleksna vrijednost jačine struje u neutralnom provodniku simetričnog sistema iznosi $\underline{I}_N = \underline{I}_1 + \underline{I}_2 + \underline{I}_3 = 0$. Sabirnice su posmatrane da su od neobojenog bakra.

Pri izgradnji mreže konačnih elemenata, formirna je adaptivna mreža, koja je gušća u pojedinim oblastima sistema (u provodniku, u oklopu, duž pravaca proračuna),

a rjeđa dalje od sabirnica unutar ostatka sistema. Gušća mreža se formira u domenima gdje se očekuje veća promjena veličina od interesa. Izgled formirane mreže konačnih elemenata je prikazana na slici 3.



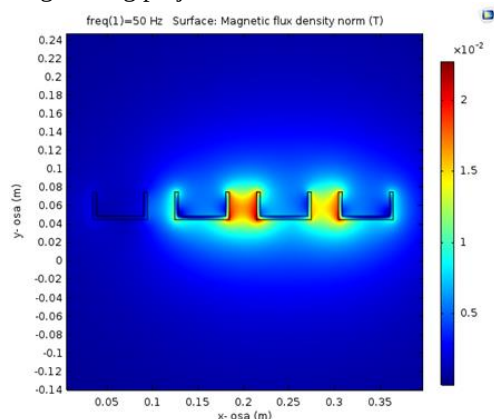
Slika 3. Dio modela sa formiranom mrežom

3. REZULTATI

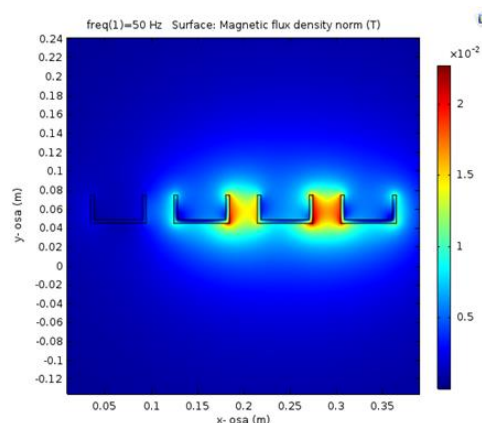
U radu je izvršena analiza raspodjele magnetskog polja u okolini neoklopljenih i oklopljenih "U6" sabirnica. Prvo je posmatran neoklopljen simetričan trofazni sistem "U" sabirnica i ispitan je uticaj promjene faza na raspodjelu intenziteta vektora magnetske indukcije. Na slikama 4 i 5 predstavljena je prostorna raspodjela intenziteta vektora magnetske indukcije kroz poprečni presjek neoklopljenih provodnika "U" profila u zavisnosti od faznog stava struja u provodnicima sistema.

Posmatranjem uticaja promjene svih šest različitih faznih stavova struja u sabirnicama na raspodjelu intenziteta vektora magnetske indukcije, utvrđeno je, da promjena faza utiče na raspodjelu magnetskog polja. Analize su pokazale, da po tri faza rasporeda daju istu raspodjelu intenziteta vektora magnetske indukcije: 123=231=312 i 132=213=321.

Upoređujući slike 4 i 5 može da se zapazi da se oblast najjačeg magnetskog polja javlja između faznih sabirnica, pod uticajem neravnomjerne raspodjele struja usljed površinskog efekta i efekta blizine [6]. Uočava se da najveće vrijednosti intenziteta vektora magnetske indukcije nastaju blizu uglova "U" sabirnica koji su okrenuti jedni prema drugima i te vrijednosti opadaju sa udaljavanjem od njih. U oblasti oko neutralnog voda, praktično nema magnetskog polja.



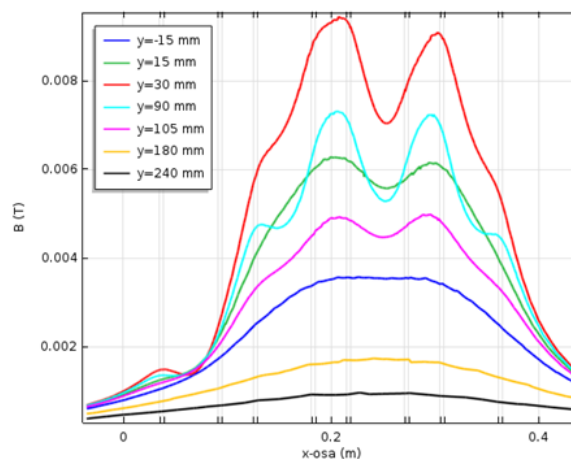
Slika 4. Prostorna raspodjela intenziteta vektora magnetske indukcije za raspored faza 123, 231, 312



Slika 5. Raspodjela intenziteta vektora magnetske indukcije za raspored faza 132=213=321

Posmatrajući slike 4 i 5 se vidi, takođe, da se oblast najjačeg polja preselio sa pozicije između prve i druge fazne sabirnice za prvu grupu faznih stavova struja, u oblast između druge i treće sabirnice, pri promeni faznih stavova struja na vrednosti druge grupe.

U nastavku analize, detaljnije je ispitana raspodjela intenziteta vektora magnetske indukcije u oblasti ispod i iznad sabirnica. Na slici 6 prikazana je raspodjela intenziteta vektora magnetske indukcije za raspored faza 123=231=312, duž pravaca koji prolaze kroz vazduh, ispod i iznad sabirnica, paralelno sa x osom, pri čemu je, donja ivica sabirnica na visini $y = b_1 = 45$ mm. Pomoćne vertikalne linije na slici 6 označavaju položaj vertikalnih delova sabirnica, radi lakše analize dobijenih rezultata.

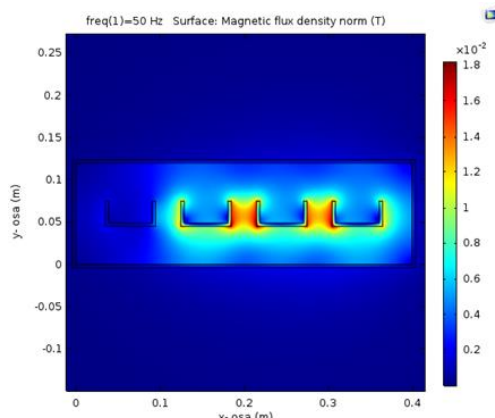


Slika 6. Raspodjela intenziteta vektora magnetske indukcije za raspored faza 123, 231, 312

Sa slike 6 se vidi da je raspodjela intenziteta vektora magnetske indukcije u vazduhu najveća ispod sabirnica ($y = 30$ mm), sa izrazitim maksimumom između prve i druge fazne sabirnice i nešto manjim maksimumom između druge i treće fazne sabirnice, a da se smanjuje se udaljavanjem od njih.

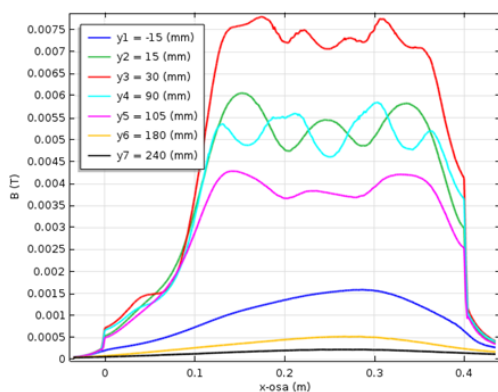
Nakon ispitivanja raspodjele intenziteta vektora magnetske indukcije u okolini neoklopljenih simetričnih trofaznih sabirnica sa neutralnim vodom, izvršena je analiza uticaja oklopa na raspodjelu intenziteta vektora magnetske indukcije u okolini sabirnica. Na slici 7 prikazana je prostorna raspodjela intenziteta vektora magnetske indukcije u okolini oklopljenih trofaznih sabirnica, za raspored faznih

struja 123=231=312, nakon što su sabirnice postavljene u oklop od aluminijuma.



Slika 7. Prostorna raspodjela intenziteta vektora magnetske indukcije oklopljenih sabirnica za raspored faza 123, 231, 312

Upoređujući slike 4 i 7, može da se zaključi da je prostorna raspodjela intenziteta vektora magnetske indukcije unutar oklopa izmenjena u odnosu na stanje pre oklopljavanja i da je u oblasti van oklopa intenzitet vektora magnetske indukcije mnogo manji nego pre oklopljavanja. Na slici 8 prikazani su rezultati dobijeni duž istih horizontalnih linija kao u slučaju neoklopljenog sistema sabirnica.



Slika 8. Raspodjela intenziteta vektora magnetske indukcije oklopljenih sabirnica za raspored faza 123, 231, 312

Sa slike 8 može odmah da se primjeti, da prisustvo oklopa skokovito smanjuje intenzitet vektora magnetske indukcije sa bočnih strana oklopa. Nagli pad intenziteta vektora magnetske indukcije iznad i ispod oklopa je takođe vidan u odnosu na neoklopljene sabirnice. Dovoljno je posmatrati vrednosti intenziteta vektora magnetske indukcije duž linije $y = -15$ mm koja je najbliža donjoj strani oklopa, da bi se videlo smanjenje intenziteta vektora magnetske indukcije izvan oklopa. Indukovane struje u oklopu dovode do smanjenja intenziteta vektora magnetske indukcije izvan oklopa ali i do njegovog porasta unutar oklopa. Posmatrajući liniju $y = 30$ mm koja je najbliža donjoj strani oklopa, uočava se da su se nakon oklopljavanja maksimalne vrednosti intenziteta vektora magnetske indukcije smanjile, ali je srednja vrednost najverovatnije porasla. Taj pravac je najdalji od oklopa a najbliži sabirnicama, što znači da je uticaj struja u sabirnicama veći od uticaja struja indukovanih u oklopu.

Pomoću prikazanih grafika je pokazano, da prisustvo oklopa smanjuje intenzitet vektora magnetske indukcije u oblasti van oklopa i da po potrebi oklopljavanje može da se pimjeni u cilju zaštite prostora oko sabirnica od neželjenog dejstva magnetskog polja sabirnica.

4. ZAKLJUČAK

Sabirnice su neizostavni dio distributivnih sistema električne energije, kao i pogona većih snaga, tako da je poznavanje parametara sabirnica od velikog značaja za njihov adekvatan izbor.

U okviru rada je izvršen proračun raspodjele intenziteta vektora magnetske indukcije u okolini simetričnih trofaznih “U” sabirnica sa neutralnim provodnikom pri različitim faznim stavovima struja posmatrani bez oklopa i sa oklopom.

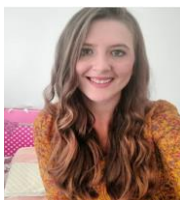
U radu je pokazano, da promena faznog stava struja u sabirnicama, menja raspodelu magnetskog polja, dok prisustvo oklopa značajno smanjuje intenzitet vektora magnetske indukcije izvan oklopa (na vrijednosti ispod 40% njene prvobitne vrijednosti). Time se pokazalo da se upotrebom oklopa može zaštititi prostor oko sabirnica od neželjenog dejstva magnetskog polja sabirnica.

Proračuni su izvršeni pomoću programskog paketa COMSOL Multiphysics. Izvršene analize su pokazale da se dati programski paket uspešno može primjeniti za rješavanje problema praktične, energetske elektromagnetike.

4. LITERATURA

- [1] J. Nahman, V. Mijailović, „Razvodna postrojenja”, Akademska misao, Elektrotehnički fakultet, Beograd, 2005.
- [2] COMSOL Muliphysics, documentation, 2016.
- [3] B. D. Popović, „Elektromagnetika”, Građevinska knjiga, Beograd, 2000.
- [4] M. Stojaković, „Matematička analiza 2“- drugi deo, 2. Izdanje, Fakultet tehničkih nauka Novi Sad, Novi Sad, 2007.
- [5] L.J. Gerić, “Razvodna postrojenja-zbirka zadataka, drugo izdanje”, Novi Sad, Fakultet tehničkih nauka, 2006.
- [6] Jadranka Ivković, “Određivanje frekvencijski zavisne strujne raspodele u “U” sabirnici”, diplomski rad, FTN Novi Sad, 2017.

Kratka biografija:



Jadranka Ivković rođena je 1993. godine u Trebinju. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo – Energetska elektronika i električne mašine je odbranila 2019.god.

kontakt: jadrankajacaivkovic@yaoo.com



Karolina Kasaš-Lažetić, rođena je u Novom Sadu 1964. Doktorirala je na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstva. Oblast interesovanja je primenjena elektromagnetika.

KLASIFIKACIJA SLIKA PRIMENOM FASTAI BIBLIOTEKE**CLASSIFICATION OF IMAGES USING FASTAI LIBRARY**

Jovana Antić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je razmatran problem klasifikacije slika primenom konvolucionih mreža realizovanih u sklopu fastai biblioteke. Trebalo je prikupiti podatke, obučiti model i prikazati rezultate.

Ključne reči: fastai, CNN, ResNet, learning rate, confusion matrix.

Abstract – The problem of image classification using CNN algorithm realize as part of fastai library is considered in this paper. It was requested to collect the data, train the model and display the results.

Keywords: fastai, CNN, ResNet, learning rate, confusion matrix.

1. UVOD

Klasifikacija slika je složen proces na koji mogu uticati mnogi faktori. Ovaj rad ispituje proces klasifikacije slike korišćenjem fastai biblioteke, kao i probleme i perspektive klasifikacije slika. Prolazeći kroz poglavlja rada steći će se šira slika o samoj klasifikaciji slika, funkcionisanja konvolucijskih neuronskih mreža, jupiter okruženja i fastai biblioteke.

Da bi sam proces bio jasniji prikazana su i dva primera, gde se može videti sa kakvim problemima se možemo suočiti tokom samog procesa kako ih rešiti ali i unaprediti.

2. KLASIFIKACIJA SLIKA I KONVOLUCIONA NEURONSKA MREŽA

Klasifikacija objekata za ljude je lak zadatak, ali pokazalo se da predstavlja složen problem za mašine [1]. Porast računara velikog kapaciteta, dostupnost visokokvalitetnih i jeftinih video kamera, kao i sve veća potreba za automatskom video analizom, izazvali su interesovanje za algoritme klasifikacije objekata.

Danas je uz pomoć učenja iz podataka, prepoznavanje slika postalo veoma efikasno. Ova je metoda mašinskog učenja dizajnirana po uzoru na funkcionisanje ljudskog mozga.

Tako se računari uče da prepoznaju vizuelne elemente unutar slike.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Mirna Kapetina, docent.

Konvolucijske neuronske mreže (engl. Convolutional neural network - CNN) predstavljaju aktuelan i istraživački atraktivan domen u širokoj oblasti veštačke inteligencije (eng. Artificial Intelligence – AI).

Konvolucijske neuronske mreže su biološki inspirisana verzija višeslojnog perceptrona (eng. Multilayer Perceptron – MLP). Smatra se da su od 2014. godine duboke konvolucijske neuronske mreže postale „glavni tok“ i predmet istraživačkih napora i zapaža se da od tada postižu značajne prednosti u raznim testovima [2].

3. KONVOLUCIONA NEURONSKA MREŽA

Konvolucionna neuronska mreža je dobila ime po konvoluciji, široko korišćenom operatoru, koji se primenjuje u obradi slike i signala.

Ideja konvolucione neuronske mreže jeste da se postavi veći broj slojeva za otkrivanje bitnih osobina ulaznih podataka. U skladu sa tom idejom se konvolucionni filteri primenjuju na sliku kako bi se izvukle korisne karakteristike i kreirale njihove mape (eng. feature maps).

Primenom filtera na ulaznu sliku, pored otkrivanja značajnih karakteristika, vrši se i redukcija rezolucije.

Nedostatak konvolucione neuronske mreže jeste potreba za značajnim hardverskim resursima, kao i to što su dosadašnji modeli obučavani velikim brojem slika, ali noviji rezultati [3] sugerišu da se visoke performanse mogu postići i sa relativnom malim brojem uzoraka.

4. FASTAI BIBLIOTEKA

Fastai su osnovali Rachel Thomas i Jeremy Howard. Rachel Thomas je profesor na univerzitetu u San Francisku, dok je Jeremy Howard naučnik koji se bavi istraživanjem na pomenutom univerzitetu.

Na osnovu PyTorch-a (biblioteka mašinskog učenja otvorenog koda zasnovana na biblioteci Torch, a koristi se za aplikacije kao što su računarski vid i obrada govora), fastai sadrži neke od najpopularnijih algoritama za klasifikaciju slika i zadatke na prirodnom jeziku. U praksi to znači da se modeli mogu kreirati i pokretati u samo nekoliko linija koda.

4.1 Realizacija konvolucione mreže u fastai -u

Osnovna ideja pri realizaciji konvolucionih mreža u sklopu fastai-a je da crvene, zelene i plave vrednosti piksela koji su u prevedenom značenju brojevi od 0 do 255, unose u jednostavno izračunavanje (tj. prvi sloj) a rezultat tog proračuna, odnosno izlaz iz prvog sloja, prelazi u drugi sloj, a rezultat toga prelazi u treći sloj i tako dalje. Može biti do hiljadu slojeva neuronske mreže.

Prvi sloj u osnovi nalazi grupe piksela koje imaju malu dijagonalnu liniju, drugi pronalazi dijagonalnu liniju u drugom pravcu, treći pronalazi gradijente koji idu od žute do plave boje i tako dalje.

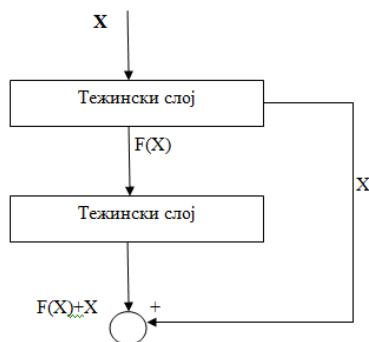
U prvom sloju imamo stvari koje mogu pronaći samo jednu liniju, a u drugom sloju možemo pronaći stvari koje imaju dve linije spojene.

Treći je mogao da preuzme sve stvari sa sloja 2 i kombinuje ih zajedno, dok četvrti sloj kombinuje stvari iz trećeg sloja

U sloju pet imamo nešto na čemu se mogu naći očne jabučice ptica i guštera ili lica određenih tipova pasa i tako dalje.

4.1.1. ResNet

ResNet je kratak naziv za *residual network*. Duboke konvolucione neuronske mreže postigle su rezultat klasifikacije slike na nivou čoveka. Duboke mreže izvlače karakteristike i klasifikatore niskog, srednjeg i visokog nivoa na višeslojni način, a broj složenih slojeva može obogatiti „nivo“ karakteristika.



Slika 1. Gradivni blok

4.1.1.1 Datasets

ImageNet je skup milion označenih slika visoke rezolucije, koje se mogu svrstati u otprilike 22 hiljada kategorija. Slike su sakupljene sa interneta i ljudi su ih etiketirali pomoću alata za prikupljanje publike.

Počevši od 2010. godine, u okviru Pascal Visual Object Challenge-a (PASCAL VOC), održava se godišnje takmičenje pod nazivom ImageNet Large-Scale Visual Recognition Challenge (ILSVRC2013). ILSVRC koristi podskup ImageNet-a sa otprilike 1000 slika u svakoj od 1000 kategorija.

Postoji otprilike 1,2 miliona slika za trening, 50k validacije i 150k testiranja slika.

PASCAL VOC pruža standardizovane skupove podataka o slikama za prepoznavanje klase objekata. Takođe pruža standardni set alata za pristup skupovima podataka i napomenama, omogućava procenu i poređenje različitih metoda i izvršava izazove za ocenjivanje performansi na prepoznavanju klase objekata.

4.1.1.2 Arhitektura

Plain baselines uglavnom su inspirisane filozofijom VGG mreža (konvolucionarna neuronska mreža koja se obučava na više od milion slika iz baze ImageNet). Konvolucionarni slojevi uglavnom imaju 3×3 filtera i slede dva jednostavna pravila:

- Za istu mapu izlaznih karakteristika, slojevi imaju isti broj filtera;
- Ako se veličina karte karakteristika prepolovi, broj filtera se udvostručuje kako bi se sačuvala vremenska složenost svakog sloja. Vredno je primetiti da ResNet model ima manje filtera i nižu složenost od VGG mreža.

Residual Network:

Na osnovu gore navedene jednostavne mreže ubačena je veza za prečicu koja mrežu pretvara u preostalu verziju. Prečice za identitet $F(x\{V\} + x)$ mogu se direktno koristiti kada su ulaz i izlaz istih dimenzija. Kada se dimenzije povećavaju), razmatra dve mogućnosti:

1. Prečica vrši mapiranje identiteta, sa dodatnim nultim unosima podložnim za povećanje dimenzija. Ova opcija ne uvodi dodatne parametre.
2. Prečica projekcije u $F(x\{V\} + x)$ koristi se za usklađivanje sa dimenzijama.

4.1.1.3 Implementacija

Slika se menja u veličinu tako da je kraća strana nasumično uzorkovana u $[256, 480]$ radi povećanja skale. "224×224 crop" nasumično se uzima iz slike ili njenog horizontalnog preokreta, pri čemu se oduzima srednja vrednost po pikselu. Stopa učenja počinje od 0,1 i deli se sa 10 kada se platoi grešaka i modeli obučavaju do 60×10000 iteracija. Koriste opadanje težine od 0,0001 i momentum 0,9. napona mreže.

4.1.1.4 Rezultat

Mreža sa 18 slojeva samo je potprostor u mreži sa 34 nivoa i još uvek deluje bolje. ResNet ima bolji napredak u slučaju da je mreža dublja.

4.2 JUPYTER okruženje

Za realizaciju CNN mreže i primenu algoritama iz fastai biblioteke korišteno je jupyter okruženje. Jupyter notebook (ranije IPython Notebooks) je veb aplikacija otvorenog koda. To je interaktivno računarsko okruženje za kreiranje Jupyter notebook dokumenata. Izraz notebook može kolokvijalno da se odnosi na mnogo različitih entiteta, u zavisnosti od konteksta, uglavnom na Jupyter veb aplikaciju, Jupyter Python veb server ili Jupyter document format. Jupyter notebook dokument je json dokument, koji sadrži listu ulaznih / izlaznih ćelija koje mogu sadržati kod, tekst matematiku, grafike i druge medije, obično sadrže „ipin“ ekstenziju.

5. KLASIFIKACIJA PRIMENOM "FASTAI" BIBLIOTEKE

U ovom poglavlju će biti opisani svi koraci koji su se koristili prilikom izgradnje modela.

5.1 Formiranje baze podataka

Prva i osnovna stvar u klasifikaciji slika je ništa drugo do prikupljanja samih podataka, tj. slika.

Postoji više mesta sa kojih možemo da prikupimo podatke, neki od njih su akademski skupovi podataka, *Kaggle competition datasets* i mnogi drugi.

Mesto sa kog su prikupljeni podaci koji su korišćeni u ovom radu je *Google*.

Prvi korak u procesu prikupljanja podataka je odlazak na <https://images.google.com/> i pronalazak se odgovarajućih slika. Sledeća stvar koja treba da se uradi jeste da se dobije spisak svih URL adresa. Da bi to bilo učinjeno, unutar prozora sa google slikama se pritisne komanda pristupi Javascript konzoli i zadavanjem putanje:

```
urls = Array.from(document.querySelectorAll('.rg_di .rg_me
ta')).map(el=>JSON.parse(el.textContent).ou);
window.open('data:text/csv;charset=utf-
8,' + escape(urls.join("\n")));
```

Na ovaj način se dobija URL slika koje smo želeli. U ovom slučaju to su slike tri vrste majmuna (orangutan, šimpanza i gorila) i ljudi (muškarac, žena, dete).

5.2 Kreiranje skupova podataka za obradu i validaciju

Kao što je već napomenuto pošto uzimamo podatke sa interneta, mogu da se podkadu neki neispravni podaci ili podaci koji nisu relevantni za naše potrebe. Te je naredna stvar pored brisanja neispravnih podataka sledeća stvar koja treba da se uradi je da se uklone slike koje u stvari i nisu slike. To se dešava stalno. U svakoj seriji uvek se nalazi nekoliko slika koje su iz bilo kog razloga oštećene. Vrlo često, kada se preuzima skup podataka iz nekog akademskog skupa podataka, postoje folderi nazvane trening, validacija i test [4] koji sadrže različite skupove podataka.

U ovom slučaju ne postoji zaseban skup provera validacije jer su slike uzete sa “Google” pretrage. Ali kako bismo znali da li naš model ide u dobrom pravcu potreban nam je validacioni set.

5.3 “data.show_batch”

Da bi imali uvid u podatke sa kojima radimo nije na odmet prikazati iste. Naredba “data.show batch” se može koristiti za prikazivanje nekih od sadržaja u grupi podataka

5.4 Treniranje modela

Posle pripreme podataka može se pristupiti obučavanju modela. Odnosno njegovom treniranu, što se u fastai-ju realizuje uz pomoć funkcije *learner*.

5.5 Otključavanje (Unfreezing), fine-tuning i learning rates

Kako poboljšati model? Može se poboljšati upotrebom precizne prilagodbe. Do sada su opremljene četiri epohe i to je prošlo prilično brzo. Razlog zbog kojeg je brzo prošla je taj što je korišćen mali trik. Ove konvolutivne mreže imaju mnogo slojeva.

Ono što je urađeno jeste da je dodato nekoliko dodatnih slojeva do kraja i samo su obučeni. Uglavnom je ostavljena većina modela upravo onakav kakav je bio, tako da je to zaista brzo. Ako se model gradi na nečemu što je slično, originalnom, unapred obučenom modelu, to deluje prilično dobro.

Ali ono što stvarno želimo je da se vratimo i obučimo čitav model. To je razlog zašto se prilično često koristi ovaj proces u dve faze. Kada podrazumevano, na

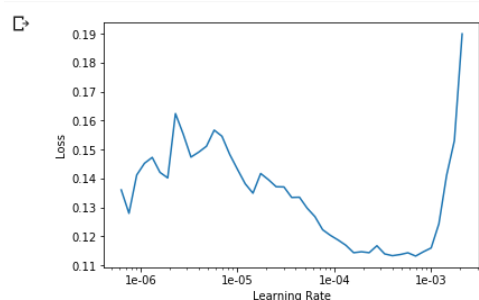
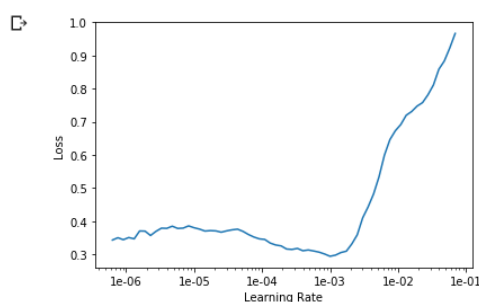
ConvLearner-u zovemo fit ili fit_one_cycle, jednostavno će se prilagoditi ovih nekoliko dodatnih slojeva koji su dodani do kraja i pokrenuće se vrlo brzo.

To u osnovi nikada neće biti prekomerno, ali da bi se to zaista postiglo, mora se pozvati *unfreezing* odnosno otključavanje. Otključavanje je stvar koja će da obučiti ceo model.

5.6 Pronalaženje learning rate-a

Konkretno, stopa učenja je podesivi hiperparametar koji se koristi u treningu neuronskih mreža koji ima malu pozitivnu vrednost, često u rasponu između 0,0 i 1,0. Stopa učenja kontroliše koliko brzo se model prilagođava problemu.

Iscrtaćemo dva grafika. Prvi grafik je grafik iz klase ljudi a drugi majmuna.



Korišćenjem ključne reči u Python-u koja se zove *slice* i koja može uzeti početnu i stop vrednost i u osnovi ono što kaže je treniranje prvih slojeva sa stopom učenja od 3e-5, a poslednji slojevi brzinom od 3e-4 u prvom slučaju i 3e-6 i 3e-3 u drugom, i rasporedite sve ostale slojeve preko toga (tj. Između te dve vrednosti podjednako). Ispostavi se da se najblji rezultati za klasu ljudi dobiju u prvoj tabeli a majmuna u drugoj.

epoch	train_loss	valid_loss	error_rate	time
0	0.126556	0.311108	0.097345	02:51
1	0.082458	0.323523	0.097345	02:51

epoch	train_loss	valid_loss	error_rate	time
0	0.142270	0.342863	0.080357	03:49
1	0.148123	0.273799	0.053571	03:51

5.7 Interpretacija

Da bi se videlo šta izlazi iz našeg modela, može da se koristi ovu klasu za interpretaciju klase.

Koristiće se ova fabrička metoda od učenika, tako da se prelazi u naučeni objekt.

Predmet učenja zna dve stvari:

1. Koji su naši podaci
2. Koji je naš model. Sada to nije samo arhitektura, već je obučeni model

To su sve informacije koje su nam potrebne za tumačenje tog modela.

5.7.1 Confusion matrix i Most confused

U oblasti mašinskog učenja, a posebno problema statističke klasifikacije, matrica konfuzije, poznata i kao matrica grešaka. Matrica konfuzije [9] je tabela koja se često koristi za opisivanje modela klasifikacije (ili „klasifikatora“) na skupu podataka ispitivanja za koje su poznate prave vrednosti.

Omogućava vizuelizaciju performansi algoritma. Omogućava lako prepoznavanje konfuzije između klasa, npr. jedna klasa se obično pogrešno označava kao druga. Većina mera performansi izračunava se iz matrice konfuzije.

Broj tačnih i pogrešnih predviđanja sažima se sa vrednostima brojanja i raščlanjuje ih o svakoj klasi. To nam daje uvid ne samo u greške koje radi klasifikator, već je važnije i vrste grešaka koje se čine.

Pored matrice konfuzije koristi se i komanda Most confused. Krajnji cilj obe stvari je isti, ali ako imamo mnogo klasa pogodnije je koristiti Most confused. ost confused će jednostavno izdvojiti iz matrice konfuzije posebne kombinacije predviđenog i stvarnog kod kojih se najčešće javlja greška .

5.8 Stavljanje modela u produkciju

Ono što bi moglo da bude interesantno je da na osnovu nasumično odabrane slike vidimo da li je i koliko naš model u stanju da pogodi o kojoj slici je reč.

U nastavku je dat po jedan primer za svaku od opisanih klasa prikazane na slikama 2 i 3. U prvom slučaju otvorena je slika gorile što je naš model pogodio, u drugom primeru je u pitanju devojčica što je sa jednakom tačnošću i utvrđeno.



Slika2. Pogađanje iz mreže koje je svrstalo sliku u kategoriju 'gorila'

☞



Slika 3. Pogađanje iz mreže koje je svrstalo sliku u kategoriju 'dete'

6. ZAKLJUČAK

Klasifikacija slika je kao tema veoma interesantna i često primenjivana u današnjem vremenu.

Kao što je u radu napomenuto domen primene same teme je veoma širok. Takođe smo se upoznali sa nekim od primera primene same klasifikacije, a koliko je tema rasprostranjena govori to da se koristi i za biomedicinsko snimanje, biometriju, video-nadzor, navigaciju vozila, industrijski vizuelni pregled, navigaciju robota i daljinsko istraživanje.

U ovom radu je prikazano kako možemo klasifikovati slike primenom fastai biblioteke.

Mogli smo da primetimo da korišćenje pomenute biblioteke dosta ubrzava i olakšava samu klasifikaciju, a da se pri tom ne gubi na tačnosti i pouzdanosti modela.

Prolazilo se kroz sve faze procesa klasifikacije, prikazani su i neki probleme do kojih se može doći tokom rada, takođe je pokazano kako te probleme rešiti kao i na koji način unaprediti sam model.

7. LITERATURA

[1] Image Classification. Dostupno na:

<https://www.sciencedirect.com/topics/computer-science/image-classification>

[2] Rethinking the Inception Architecture for Computer Vision. Dostupno na:

<https://arxiv.org/abs/1512.00567/>

[3] Dictionary of computer vision and image processing.

[4] About Train, Validation and Test Sets in Machine Learning. Dostupno na:

<https://towardsdatascience.com/train-validation-and-test-sets-72cb40c9e7>

Kratka biografija:



Jovana Antić rođena je u Beogradu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Adaptivno i napredno upravljanje odbranila je 2019.god.

Kontakt: anticj9@gmail.com

ANDROID APLIKACIJA ZA NADGLEDANJE SISTEMA BAZIRANIH NA AMAZON AWS PLATFORMI**ANDROID APPLICATION FOR MONITORING AMAZON AWS BASED SYSTEMS**Danilo Dimitrijević, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSTVO I AUTOMATIKA**

Kratak sadržaj – U radu je opisan sistem za nadgledanje servera pomoću AWS servisa Amazon Inspector. Specificiran je model sistema kroz UML dijagrame. Opisana je implementacija sistema na Android platformi.

Ključne reči: Amazon Inspector, AWS servisi, Android

Abstract – This paper presents the server monitoring by the AWS service called Amazon Inspector. Detailed description of a model through UML diagrams is given as well as implementation of the system in Android platform.

Keywords: Amazon Inspector, AWS services, Android

1. UVOD

Servisi za nadgledanje servera predstavljaju svakodnevnicu. Najčešći zahtev prilikom nadgledanja servera je da se primi upozorenje kada server nije u funkciji sa razlogom zbog kojeg je server prestao sa radom. Međutim, to nije dovoljno da se server održi bezbednim s obzirom da postoji mogućnost je da je server ranjiv na novu softversku grešku, da se nalazi pod nekom vrstom napada ili da je već kompromitovan. To su neke od stvari koje bi trebalo znati kako bi se preduzele akcije da se takve aktivnosti saniraju u najkraćem mogućem roku. Nadgledanje servera za bezbednosne događaje pruža mogućnost sprečavanja bezbednosnih problema, čak i ako se nešto loše desi može se kontrolisati kako se to odvija, a ne samo da se posmatra dok se sve „ruši“ [1].

U okviru ovog rada prikazana je Android aplikacija za nadgledanje sistema baziranih na Amazon AWS cloud platformi, sa posebnim osvrtom na servis Amazon Inspector.

2. AMAZON INSPECTOR

Amazon Inspector predstavlja automatizovani servis procene bezbednosti koja pomaže u poboljšanju bezbednosti i usklađenosti aplikacija postavljenih na AWS servisu. Amazon Inspector automatski procenjuje: izloženost na napade, odstupanje od preporučenih praksi i ranjivosti aplikacija. Nakon obavljanja procene Amazon Inspector daje detaljnu listu bezbednosnih nalaza sa prioritetom prema stepenu ozbiljnosti [2].

Procena bezbednosti Amazon Inspector-a pomaže u proveru ranjivosti Amazon Elastic Compute Cloud (Amazon EC2) instanci. Procena Amazon Inspector-a nudi predefinisane pakete pravila koji se sastoje od bezbednosnih

preporučenih praksi i definicija ranjivosti. Primeri ugrađenih pravila uključuju: proveru pristupa EC2 instancama sa interneta, mogućnost daljinskog korišćenja servera ili proveru ranjivosti instalirane verzije softvera [2]. Amazon Elastic Compute Cloud (Amazon EC2) je servis koji pruža siguran i promenljiv kapacitet računara na cloud-u. Amazon EC2 smanjuje vreme potrebno za pokretanje novog servera i omogućava brzu promenu kapaciteta servera u zavisnosti od potreba servera [3].

Pomoću Amazon Inspector-a može se automatizovati bezbednosna procena ranjivosti kroz razvoj pipeline-a. Ovo pruža mogućnost da bezbednosno testiranje postane redovan deo razvoja i IT operacija. Amazon Inspector nudi predefinisani softver koji se naziva agent i koji se može instalirati u operativni sistem podignut na EC2 instanci koju je neophodno nadgledati. Agent nagleda ponašanje EC2 instance uključujući mrežu, sistem datoteka i aktivnosti procesa, prikuplja širok skup o ponašanju i konfiguraciji [4].

Najvažniji elementi servisa Amazon Inspector su:

1. **Assessment Target** - predstavlja kolekciju AWS resursa koji deluju zajedno kao jedinica kako bi pomogli u ostvarivanju poslovnih ciljeva. Amazon Inspector procenjuje sigurnosti resursa koji predstavljaju assessment target. Trenutno, Amazon Inspector assessment target-i se mogu sastojati od EC2 instanci [5].
2. **Rules Packages and Rule** - upoređuje ponašanje i bezbednosnu konfiguraciju assessment target-a u odnosu na izabrane pakete pravila (rules packages) za proveru. U kontekstu Amazon Inspector-a, pravilo (rule) je bezbednosna provera koju Amazon Inspector vrši prilikom izvršavanja assessment run-a [5].
3. **Assessment Template** - predstavlja konfiguraciju koja se koristi prilikom pokretanja assessment run-a [5].
4. **Assessment Run** - predstavlja proces otkrivanja potencijalnih bezbednosnih problema kroz analizu konfiguracije i ponašanja assessment target-a na osnovu specificiranih bezbednosnih paketa pravila. Tokom rada assessment run-a Amazon Inspector nadgleda, prikuplja i analizira konfiguraciju i ponašanje podataka iz resursa specificiranog assessment target-a. Nakon toga, Amazon Inspector analizira podatke i upoređuje ih sa skupom bezbednosnih paketa pravila koja su specificirana u assessment template-u tokom pokretanja assessment run-a. Kompletirani assessment run proizvodi listu finding-a koji predstavljaju potencijalne bezbednosne probleme različitih nivoa ozbiljnosti [5].

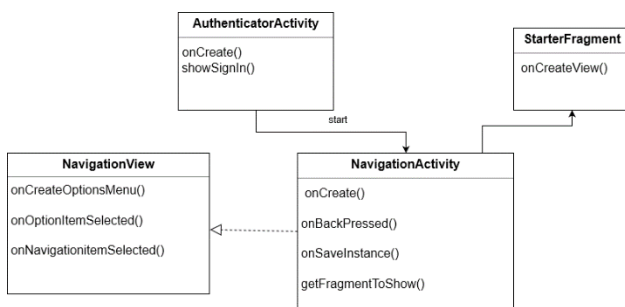
NAPOMENA:

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

5. **Finding** - predstavlja potencijalne bezbednosne probleme koje je *Amazon Inspector* otkrio prilikom procene nadgledanja *assessment target*-a. *Finding* se može prikazati pomoću *Amazon Inspector* konzole ili pomoću API-ja [5].
6. **Report** - predstavlja dokument koji opisuje šta se testira u *assessment run*-u i daje rezultate procene. Dobijeni *report*-i se mogu skladištiti, deliti sa timom kako bi se sanirali problemi ili koristiti za povećanje podataka o reviziji usklađenosti. Nakon završetka *assessment run*-a može se generisati izveštaj [5].

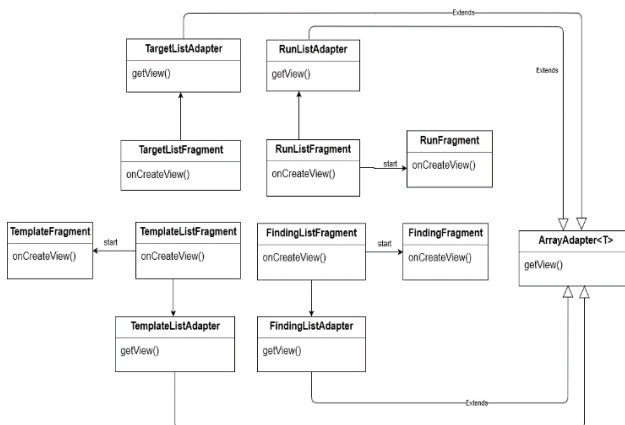
3. MODEL SISTEMA

Slika 1. prikazuje dijagram klasa glavnih komponenti *android* aplikacije. *AuthenticatorActivity* je klasa koja se pokreće prilikom inicijalizacije. Zadatak te klase je da proveriti da li je korisnik već prijavljen na sistem, ukoliko jeste automatski se prelazi na aktivnost *Navigation-Activity*, ukoliko nije otvara se stranica za prijavljivanje na aplikaciju. *NavigationActivity* predstavlja glavnu aktivnost aplikacije koja se sastoji iz dva dela, prvi deo je *StarterFragment* klasa koji služi kao prikaz u glavnoj aktivnosti, dok je drugi deo *NavigationView* koji omogućava navigacioni prikaz i pokretanje ostalih fragmenata.



Slika 1. – Dijagram klasa glavnih aktivnosti

Na slici 2 prikazan je dijagram klasa fragmenata koji služe za listanje elemenata *Amazon Inspector*-a. Fragmenti u sebi sadrže adaptere koji nasleđuju ugrađenu *android* klasu *ArrayAdapter<T>*. Namena adaptera je da svaki element iz liste prikažu na identičan način. Fragmenti za listanje elemenata mogu pokrenuti fragment za detaljan prikaz elemenata (npr. fragment *RunList-Fragment* može pokrenuti fragment *RunFragment*).



Slika 2. - Dijagram klasa elemenata

4. IMPLEMENTACIJA SISTEMA

U ovom odeljku će biti opisana implementacija *android* aplikacije za nadgledanje koja predstavlja sistem za nadgledanje server pomoću AWS servisa *Amazon Inspector*.

Amazon Web Services (AWS) je jedna od najrasprostranjenijih *cloud* platformi koja nudi preko 165 servisa iz *data* centara širom sveta [6].

Implementacija sistema se sastoji od dve aplikacije:

- Jedna aplikacija je urađena u programskom jeziku *Java* i predstavlja *AWS Lambda* izraze, od kojih jedan *Lambda* izraz služi za dodavanje *Amazon Inspector* elemenata u bazu, dok drugi služi za obradu *Amazon SNS* događaja i slanje notifikacija.
- Druga aplikacija je mobilna aplikacija urađena u okruženju *Android Studio* koji predstavlja zvanično okruženje za razvoj *android* aplikacija. Mobilna aplikacija služi za prikaz *Amazon Inspector* elemenata i primanje notifikacija.

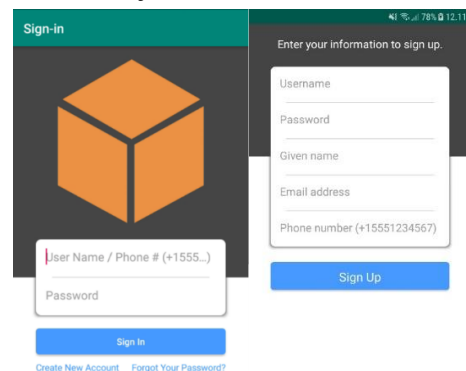
Amazon Lambda je servis koji omogućava pokretanje koda bez potrebe za obezbeđivanjem ili upravljanjem serverima [7]. *Amazon Simple Notification System (Amazon SNS)* je servis koji upravlja isporukom ili slanjem poruka pretplatničkim krajnjim tačkama ili klijentima. Ovaj servis se može koristiti za slanje mobilnih notifikacija, SMS poruka, mail-ova kao i za slanje događaja na druge AWS servise [8].

Kako bi korisnik mogao koristiti aplikaciju neophodno je da poseduje *AWS* nalog, kao i da ima kreiran bar jedan *assessment target* na *Amazon Inspector* servisu kako bi mogao da učita neophodne podatke u aplikaciju.

4.1. Prijava i registracija na sistem

Aplikacija zahteva od korisnika prijavu na sistem kako bi mogao da koristi aplikaciju. Za implementaciju stranice za prijavljivanje i registraciju je korišćena predefinisana stranica koju pruža *AWS Mobile Hub* (servis za mobile aplikacije) servis u kombinaciji sa servisom *Amazon Cognito* (servis koji predstavlja korisničku bazu) (slika 3.).

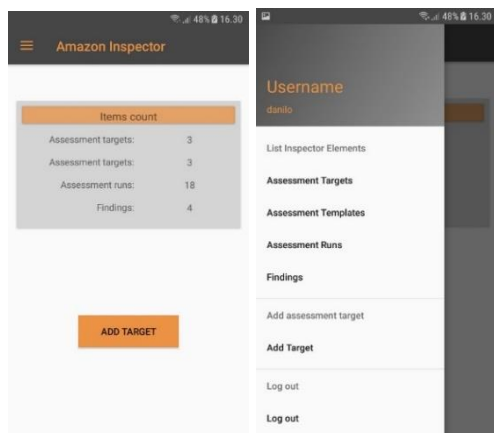
Ukoliko korisnik nema nalog, može ga napraviti odabirom opcije za pravljenje novog naloga (slika 3. opcija *Create New Account*) i otvoriće mu se prozor za registraciju. Kada se registruje njegovi kredencijali se automatski čuvaju u korisničkoj bazi.



Slika 3. - Predefinisane stranice za prijavljivanje i registrovanje na aplikaciju

4.2. Početna stranica

Početna stranica prikazuje listing svih elemenata sa ukupnim brojem tih elemenata u prvoj kartici i omogućava korisniku da napravi novi target (slika 4.). Početna stranica nudi i opciju za navigaciju kroz projekat (slika 4.).

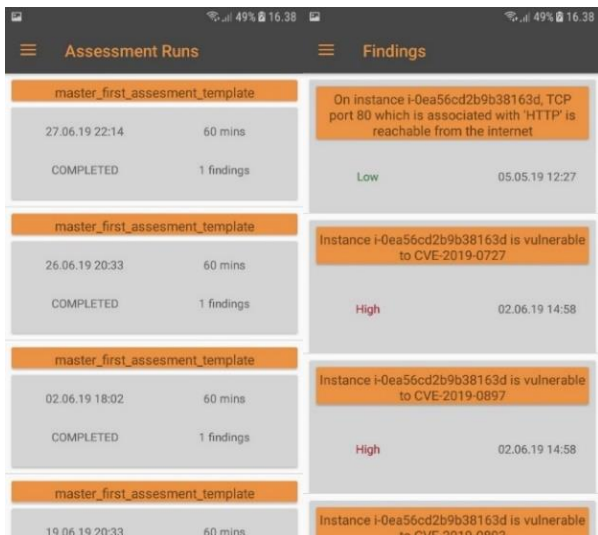


Slika 4. – Početna stranica i navigacioni prikaz

4.3. – Prikaz liste elemenata Amazon Inspector-a

Slika 5. prikazuje listing Amazon Inspector elemenata pod nazivom *assessment run* i *finding*. Prikazane su samo najbitnije informacije kako elementi liste ne bi zauzimali previše mesta na ekranu, za detaljnije informacije korisnik treba da klikne na element.

Ostali elementi Amazon Inspector-a su prikazani na sličan način kao i elementi prikazani na slici 5., jedina razlika je kod prikaza *assessment target*-a jer taj element nema mogućnost detaljnog prikaza.

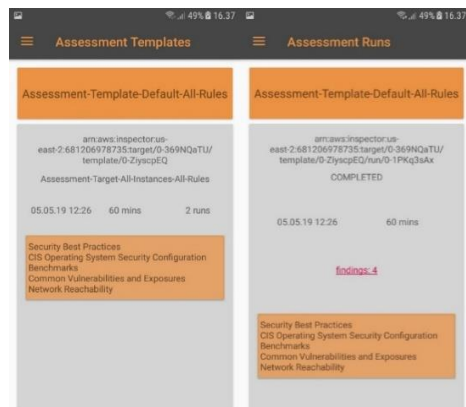


Slika 5. – Prikaz listanja assessment run-ova i finding-a

4.4. Detaljan prikaz elemenata Amazon Inspector-a

Slika 6. predstavlja prozor za detaljan prikaz *assessment template*-a i *assessment run*-a sa naslovima *template*-a, vremena pravljenja, paketima pravila koji su korišćeni prilikom pokretanja *run*-a.

Ostali elementi Amazon Inspector-a prikazani su na sličan način kao i dva prikazana na slici 6.



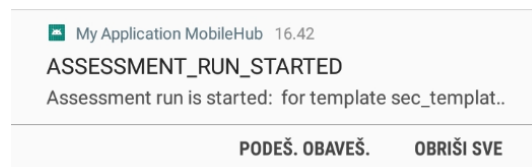
Slika 6. - Detaljan prikaz assessment template-a i assessment run-a

4.5. - Notifikacije

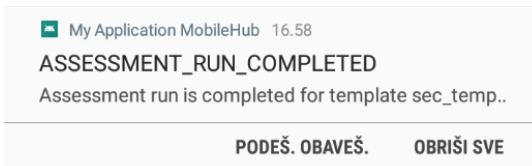
Za slanje notifikacija mobilnoj aplikaciji, korišćen je *Firebase Cloud Messaging (FCM)* sistem. *FCM* predstavlja *Firebase* platformu za slanje i primanje poruka i notifikacija [9].

Aplikacija prima notifikacije na dva događaja: početak i kraj *assessment run*-a.

Slika 6. i 7. predstavljaju prikaz notifikacija u mobilnoj aplikaciji. Korisnik ima mogućnost da klikne na notifikaciju i ode na aktivnost koja predstavlja detaljan prikaz *assessment run*-a.



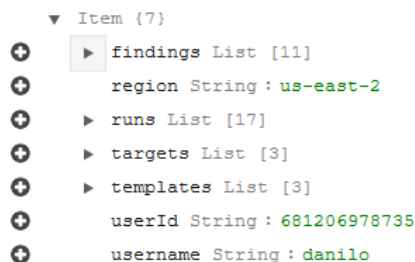
Slika 6. – Notifikacija za pokretanje assessment run-a



Slika 7. – Notifikacija za završetak assessment run-a

4.6 - Baza podataka

Za ovaj projekat kao baza podataka je korišćen *AWS* servis *Amazon DynamoDB* (*NoSQL* baza podataka). Na slici 8. predstavljen je jedan red u bazi podataka na kojem se vide elementi vezani za korisnika kao što su korisničko ime, region, identifikacioni broj korisnika na *AWS*-u i *Amazon Inspector* elementi vezani za tog korisnika.



Slika 8. – Primer reda u bazi podataka

Listing 1. predstavlja inicijalizaciju servisa *Amazon DynamoDB* u *android* aplikaciji i čitanje podataka pomoću metode *load*.

```
DynamoDBMapper dynamoDBMapper;  
AmazonDynamoDBClient dynamoDBClient = new  
AmazonDynamoDBClient(AWSMobileClient.getInstance()  
e().getCredentials());  
dynamoDBMapper = DynamoDBMapper.builder()  
    .dynamoDBClient(dynamoDBClient)  
  
    .awsConfiguration(AWSMobileClient.getInstance()  
    .getConfiguration())  
    .build();  
InspectorModel inspectorModel =  
dynamoDBMapper.load(InspectorModel.class,  
targetDTOS[0].getUserId());
```

Listing 1. Inicijalizacija servisa *Amazon DynamoDB*

5. ZAKLJUČAK

Nadgledanje sistema je važno pogotovu u posljednjem periodu kada se pojavljuje sve više i više sofisticiranih napada na različite aplikacije i servise. Gotovo sve veće *cloud* platforme poseduju proizvod koji nadgleda te platforme kako bi sprečile ranjivosti i ažurirali sistem sa najnovijim izmenama. *Amazon* predstavlja jednu od najsigurnijih i najčešće korišćenih *cloud* platformi sa konstantnim razvijanjem novih i ažuriranjem starih servisa.

U okviru ovog rada opisana je *AWS Amazon Inspector* platforma za nadgledanje servera sa svojim karakteristikama i elementima. Dat je prikaz specifikacije i implementacije *android* aplikacije za nadgledanje *AWS* sistema. Takođe, prikazani su osnovni elementi *Amazon Inspector*-a sa svojim osnovnim informacijama u vidu liste, kao i detaljni prikazi elemenata u posebnom prozoru. Opisano je na koje događaje aplikacija prima notifikacije.

Neki od daljih smerova nadogradnje i razvoja sistema su:

- prijavljivanje na aplikaciju pomoću provajdera *Google* i *Facebook*,
- pravljenje novih *Amazon Inspector* elemenata pomoću *android* aplikacije i
- nadograditi *android* aplikaciju da radi sa najnovijim *AWS* servisom koji je predviđen za rad sa mobilnim aplikacijama pod nazivom *AWS Amplify*.

6. LITERATURA

- [1] Server Monitoring Overview <https://bobcares.com/blog/server-security-monitoring/> (pristupljeno u oktobru 2019.)
- [2] Amazon Inspector <https://aws.amazon.com/inspector/> (pristupljeno u oktobru 2019.)
- [3] Amazon EC2 <https://aws.amazon.com/ec2/>
- [4] Amazon Inspector Introduction https://docs.aws.amazon.com/inspector/latest/userguide/inspector_introduction.html (pristupljeno u oktobru 2019.)
- [5] Amazon Inspector Concepts https://docs.aws.amazon.com/inspector/latest/userguide/inspector_concepts.html (pristupljeno u oktobru 2019.)
- [6] AWS <https://aws.amazon.com/what-is-aws/> (pristupljeno u oktobru 2019.)
- [7] AWS Lambda <https://aws.amazon.com/lambda/> (pristupljeno u oktobru 2019.)
- [8] Amazon SNS <https://docs.aws.amazon.com/sns/latest/dg/welcome.html> (pristupljeno u oktobru 2019.)
- [9] Firebase Cloud Messaging <https://firebase.google.com/docs/cloud-messaging> (pristupljeno u oktobru 2019.)

Kratka biografija:



Danilo Dimitrijević rođen je 28.04.1994. godine u Kragujevcu. Osnovnu školu „Stanislav Sremčević“ je završio 2009. godine, srednju školu „Prva tehnička škola“ u Kragujevcu, završio je 2013. godine. Iste te godine upisao je „Fakultet tehničkih nauka“ u Novom Sadu, odsek računarstvo i automatika. Diplomirao je u roku 2017. godine i upisao master studije iste godine. Od 2018. godine radi u kompaniji „Synchron“.

kontakt: choda.94@gmail.com

ANDROID АПЛИКАЦИЈА ЗА НАДГЛЕДАЊЕ СИСТЕМА БАЗИРАНИХ НА AZURE ПЛАТФОРМИ**ANDROID APPLICATION FOR MONITORING AZURE BASED SYSTEMS**

Данило Аћимовић, Факултет техничких наука, Нови Сад

Област – РАЧУНАРСТВО И АУТОМАТИКА

Кратак садржај – У раду је представљена апликација за надгледање система заснованих на Azure платформи. Описана је имплементација система и представљен је преглед система.

Кључне речи: *Cloud, Azure, надгледање, Android, обавештења, логови*

Abstract – *This paper presents application for monitoring Azure based systems. Implementation details and system overview are given.*

Keywords: *Cloud, Azure, monitoring, Android, notifications, logs*

1. УВОД

Cloud је могућност хостинга софтверске платформе или услуге са удаљене локације којој се може приступити и користити било где путем Интернета. Уместо да се инсталира пакет софтверских програма на више рачунара, потребно је само једно пријављивање на апликацију. Поред тога, будући да је *Cloud* хостован сервис, смањује потребу за физичком инфраструктуром што самим тим смањује финансијске трошкове и улагања са многим сервисима дизајнираним по моделу „pay-as-you-go“. Оно што је некада било доступно само великим мултинационалним компанијама сада је доступно знатно мањим компанијама [1].

Данас се *Cloud Computing* широко користи за пружање услуга путем интернета из техничких и економских разлога. Број услуга заснованих на *cloud*-у се последњих година брзо и снажно повећавао, па је самим тим повећана и сложеност инфраструктуре која стоји иза ових услуга. За правилно функционисање и управљање таквим сложеним инфраструктурама ефикасно и ефективно надгледање система базираних на *cloud*-у је стално потребно [2]. Надгледање представља процес прикупљања и анализе информација о раду хардвера и софтвера како би се осигурало да све функционише исправно унутар апликације или сервиса. Аутоматске или ручне методе управљања потврђују доступност и перформансе веб сајта, сервера, апликације и остале инфраструктуре засноване на *Cloud*-у [3].

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији је ментор био проф. др Горан Сладић.

Надгледање пружа помоћ у схватању како раде саме апликације, такође, помаже да се проактивно идентификују проблеми који погађају апликације и ресурсе који од њих зависе [4].

Тема овог рада јесте апликација за надгледање система заснованих на *Azure* платформи. Апликација служи да корисницима *Azure* платформе олакша добијање критичних упозорења на њихове телефоне независно од локације на којој се корисници налазе. Под критичним упозорењима се подразумевају упозорења креирана од стране самих корисника *Azure Monitor* платформе, односно, упозорења за које корисник сматра да су критична за његов систем. Идеја јесте да корисник буде што пре свестан могућих претњи и да благовремено реагује на активности које по њему представљају претњу за сам систем.

2. ПРЕГЛЕД ПОСТОЈЕЋИХ РЕШЕЊА

Модерне апликације засноване на *Cloud* технологији су високо дистрибуиране, састоје се од стотина компоненти и константно су изложене променама. Надгледање представља јако битну ставку у одржавању нормалног функционисања система и очувању његових перформанси. Све је већи број компанија које прелазе на коришћење *Cloud*-а, а самим тим се убрзава развој нових *Cloud* платформи доступних самим корисницима. Свака од тих платформи подразумева и постојање дела унутар ње која се бави надгледањем, примера ради на *Azure*-у се та платформа назива *Azure Monitor* [4].

У наставку су описане неке од основних карактеристика за *Amazon Cloud Watch*, *Azure Monitor* и *Stackdriver Monitoring*.

2.1. AMAZON CLOUDWATCH

Amazon CloudWatch представља сервис за праћење и надгледање. *CloudWatch* пружа податке за праћење апликација, реаговање на промене перформанси унутар система, оптимизацију употребе ресурса и добијање јединственог погледа на оперативно стање система. Прикупља оперативне и податке за надгледање у облику логова, метрика и догађаја, пружајући корисницима обједињени приказ *AWS* [5] ресурса, апликација и сервиса који су покренути на *AWS*-у или на локалним серверима. Може се користити за откривање аномалијског понашања у разним окружењима, постављање аларма, визуализацију логова и метрика, предузимање аутоматизованих радњи и решавање проблема [6].

2.2. AZURE MONITOR

Azure Monitor јесте уграђена платформа унутар Azure-а која омогућава увид у перформансе, здравље и стање Azure ресурса. Представља сервис који обезбеђује надгледање ресурса на једном месту. Пружа могућност визуализације, креирања упита, архивирања метрика и логова који долазе од ресурса [4]. Основа AzureMonitor-а јесте *TelemetryPipeline* која обезбеђује прикупљање података из преко 30 сервиса. Ово омогућује доследно искуство надгледања унутар граница Azure-а. Сви прикупљени подаци могу бити визуелизовани на командној табли или могу бити коришћени од стране разних апликација, које нису на Azure-у, коришћењем REST API-а. Два типа података су најбитнија: метрике и логови [4].

2.3. STACKDRIVER MONITORING

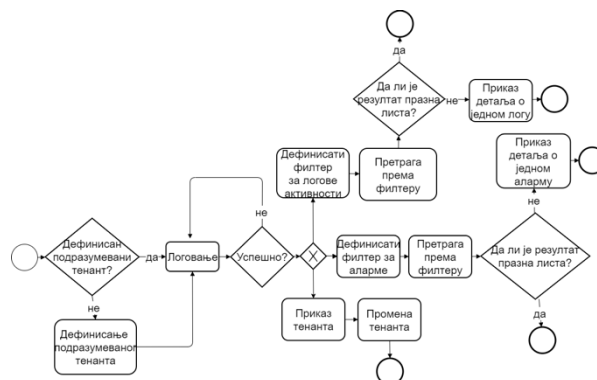
Представља алат за надгледање намењен за апликације које су покренуте на AWS-у или Google Cloud Platform-и. Пружа увид у перформансе и опште здравље апликација у cloud-у. Скупља метрике, догађаје и мета податке из Google Cloud Platform-е, Amazon Web Service-а и различитих уобичајених компоненти апликација. Слично као и претходно два описана алата, Stackdriver, генерише увиде у стање система помоћу контролне табле, графикана и упозорења.

3. МОДЕЛ СИСТЕМА

За динамички модел биће приказани дијаграми активности који се одвијају приликом коришћења апликације.

Слика 1 приказује дијаграм активности приликом коришћења апликације од стране корисника. Одмах на почетку се врши провера да ли је корисник дефинисао подразумевану конфигурацију. Ако није, иде се на акцију за дефинисање подразумеване конфигурације, а ако јесте прелази се на логовање. Уколико је логовање успешно наставља се даље, у супротном се враћа на поновно логовање. Након логовања бира се једна од следећих опција:

- Приказ листе логова активности према задатом филтеру. Потребно је да се дефинише филтер претраге који укључује жељену претплату и временски интервал, ако резултат претраге није празна листа корисник бира један лог, приказује његове детаље и ту је крај активности. Ако је резултат филтрирања празна листа активност се завршава
- Приказ листе аларма према задатом филтеру, потребно је да се дефинише филтер који укључује претплату за коју се врши филтрирање, ако је резултат претраге празна листа, активност се завршава. Ако резултат претраге није празна листа корисник бира један аларм из листе понуђених и приказује његове детаље што представља последњу акцију за дати избор након које се активност завршава
- Приказ листе тенанта где корисник врши промену тенанта на основу чијег активног директоријума ће се вршити аутентификација



Слика 1 - Дијаграм активности приликом коришћења апликације од стране корисника

4. ИМПЛЕМЕНТАЦИЈА СИСТЕМА

4.1 FIREBASE

Представља свеобухватну платформу за развој мобилних апликација. Поседује велики број функционалности које се односе на аналитику, базе података, слање порука и извештаје о престанку рада апликације. *Firebase* се заснива се на Google инфраструктури и аутоматски се скалира чак и за највеће мобилне апликације [7]. У апликацији за надгледање Azure активности користи се *Cloud Messaging* [8]. *Firebase* се користи за слање обавештења корисницима уколико дође до активирања аларма. Да би корисник примао обавештења он мора да дефинише правила која ће довести до активирања аларма. Када дође до активирања аларма обавештење се шаље кориснику на основу његове претплате и *Firebase-CloudMessaging (FCM)* токена [9].

4.2 AZURE REST API

У овом систему *Azure Rest API*[10] се користи за приступ подацима који су приказивани у апликацији. Ти подаци се налазе унутар *Azure Monitor*-а и *Azure Resource Manager*-а.

4.3 AZURE ACTIVE DIRECTORY AUTHENTICATION LIBRARY (ADAL)

Azure Active Directory (Azure AD) јесте сервис за креирање идентитета заснован на cloud-у који омогућава програмерима да развијају апликације које безбедно аутентификују кориснике са *Microsoft*-овим налогом за фирме и студенте [11]. Захваљујући *ADAL*-у, прво се врши провера да ли је токен у кешу, односно да ли је корисник већ улогован и да ли је токен, ако постоји, валидан. Код у Листингу 1 покушава да добије токен из кеша. Ако није успео да пронађе токен у кешу, форсира се интерактивно добијање токена (Листинг 2) тако што се кориснику приказује нови прозор који му омогућава да се улогује на *Microsoft*-овом сајту како би добио токен за приступ сервисима.

```
mAuthContext.acquireTokenSilentAsync(  
    RESOURCE_ID,  
    CLIENT_ID,  
    userId, getAuthSilentCallback());
```

Листинг 1 - Покушај да се токен добије из кеша

```
mAuthContext.acquireToken(
    getActivity(),
    RESOURCE_ID,
    CLIENT_ID,
    REDIRECT_URI,
    PromptBehavior.Auto,
    getAuthInteractiveCallback());
```

Листинг 2 - Захтев за добијање токена

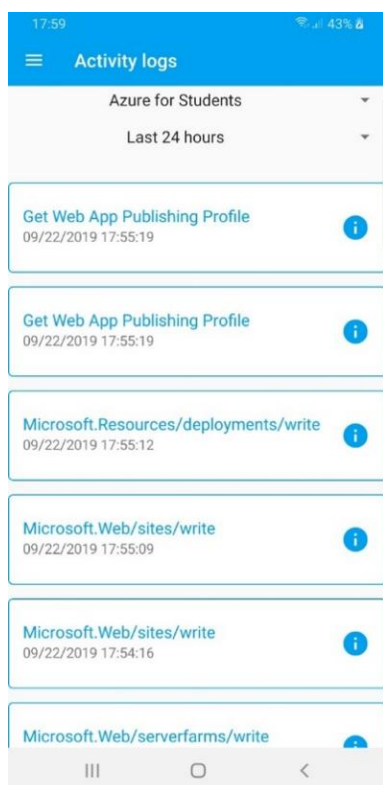
4.4 ПРЕГЛЕД ФУНКЦИОНАЛНОСТИ ANDROID АПЛИКАЦИЈЕ ЗА НАДГЛЕДАЊЕ СИТЕМА БАЗИРАНИХ НА AZURE ПЛАТФОРМИ

Уколико корисник први пут приступа апликацији, од њега ће бити захтевано да унесе идентификатор инстанце активног директоријума (тенант) која ће се користити за иницијалну аутентификацију.

Након што је дефинисао идентификатор тенанта, од корисника се тражи да се путем *Microsoft*-овог налога улогује како би добио токен за приступ сервисима и започео коришћење саме апликације.

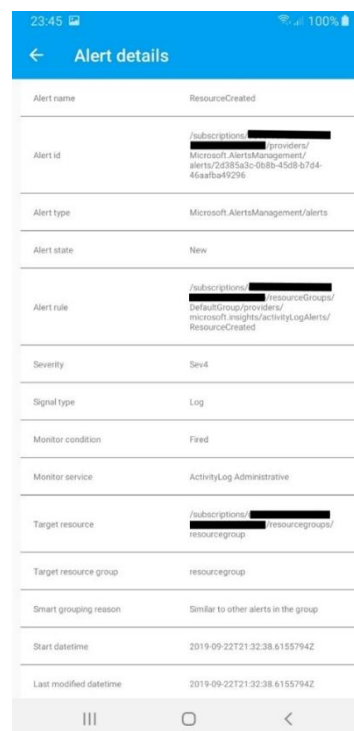
Након успешног логовања кориснику се приказује главна активност која садржи могућност приказа логова активности, приказ упозорења и приказ активних тенанта за кориснички налог.

Корисник може да изабере било коју од ставки да прегледа. Када изабере логове активности, пре приказа резултата, од њега се тражи да изабере претплату и временски опсег за који жели да прикаже резултате, када је унео потребне информације, ако има логова активности који одговарају критеријуму они се приказују кориснику (Слика 2).



Слика 2 - Логови активности

Кликом на неки од приказаних логова приказују се његове детаљне информације. Корисник може да прикаже сва упозорења за селектовану претплату. Притиском на неки од приказаних упозорења отвара се активност која приказује његове детаљне информације. (Слика 3)



Слика 3- Детаљан приказ једног упозорења

Следећа опција коју корисник може да изабере јесте приказ свих активних тенанта за његов налог. Уколико постоји више тенанта који се везују за кориснички налог, корисник може да промени тренутни, подразумевани тенант у апликацији који је дефинисао на самом почетку њеног коришћења. Уколико жели да дефинише нови тенант са новим алиасом корисник апликације биће приморан да се одјави и врати на почетни екран. Одјава се врши притиском на дугме *SignOut* које се налази у менију заједно са ставкама главних активности, логовима активности, упозорењима и тенантима.

5. ЗАКЉУЧАК

Cloud технологије су из дана у дан све више популарне и све више се развијају. Захваљујући *cloud*-у мале и велике компаније постижу већу флексибилност и брзину пословања. Неоспорно је да је добар систем за надгледање један од главних фактора исправног функционисања сервиса на *cloud*-у. Захваљујући добром систему за надгледање корисници су у могућности да веома брзо и лако детектују проблеме и отклоне их што је пре могуће.

У оквиру овог рада представљена је *android* апликација која може да побољша квалитет надгледања на *cloud*-у крајњим корисницима и допринесе унапређењу система за надгледање на *Azure* платформи.

Апликација такође може да утиче на брзину откривања проблема и отклањања истих смањујући потребу корисника да често одлази на портал како би проверавао да ли је дошло до неких већих проблема.

Приказана апликација садржи имплементирани основне функционалности са становишта надгледања *cloud* система и отвара многе могућности за унаређење. Једна од ставки са којом апликација може да се прошири јесте могућност да корисник дефинише нова правила за упозорења директно из апликације. Филтери приликом приказа логова и упозорења могу бити проширени да укључе неке нове критеријуме.

Како увек није неопходно да се приказују логови свих нивоа, могуће је проширити апликацију подршком за филтрирање логова по различитим нивоима.

За функционалности упозорења могу се извршити унапређења тако што ће се поред информација заједничких за сва упозорења у свим сервисима, приказивати и детаљи специфични за сервис које је довео до задовољења услова за активирање упозорења.

6. ЛИТЕРАТУРА

- [1] *What is Cloud Based Technology? - Front Desk & Reservation System for Small Hotels - Little Hotelier*, Приступљено Август 2019, Доступно на: <https://www.littlehotelier.com/what-is-cloud-based-technology/>
- [2] Aceto. G., Botta. A., Donato. W., Pescarè. A. (2013). *Cloud monitoring: A survey*. Доступно на: <https://www.sciencedirect.com/science/article/abs/pii/S1389128613001084>
- [3] *Learn The Importance of Cloud Monitoring, Before It's Too Late*, Приступљено Август 2019. Доступно на: <https://phoenixnap.com/blog/what-is-cloud-monitoring>
- [4] *Microsoft, Azure Monitor overview*. Приступљено Август 2019, Доступно на: <https://docs.microsoft.com/en-us/azure/azure-monitor/overview>

- [5] *Amazon, What is AWS*, Приступљено Август 2019, Доступно на: <https://aws.amazon.com/what-is-aws/>
- [6] *Amazon, Amazon CloudWatch - Application and Infrastructure Monitoring*, Приступљено Август 2019, Доступно на: <https://aws.amazon.com/cloudwatch/>
- [7] *Google, Firebase*, Приступљено Август 2019, Доступно на: <https://firebase.google.com/>
- [8] *Google, Firebase Cloud Messaging | Send notifications across platforms for free*, Приступљено Август 2019. Доступно на: <https://firebase.google.com/products/cloud-messaging/>
- [9] *Google, Credentials | About FCM messages | Firebase*, Приступљено Август 2019, Доступно на: <https://firebase.google.com/docs/cloud-messaging/concept-options#credentials>
- [10] *Microsoft, Azure REST API Reference*, Приступљено Август 2019. Доступно на: <https://docs.microsoft.com/en-us/rest/api/azure/>
- [11] *Microsoft, Azure Active Directory for developers (v1.0) overview*, Приступљено Август 2019, Доступно на: <https://docs.microsoft.com/en-us/azure/active-directory/develop/v1-overview>

Кратка биографија:



Данило Ахимовић рођен је 9.3.1994. године у Лозници. Основне академске студије завршио је 2017. године, смер рачунарство и аутоматика на ФТН-у. Мастер академске студије уписао је исте године, такође смер рачунарство и аутоматика. Положио је испите предвиђене студијским програмом.

OPTIMIZACIJA RADA VETROTURBINE UPOTREBOM MAŠINSKOG UČENJA OPTIMIZATION OF WIND TURBINE OPERATION USING MACHINE LEARNING

Nataša Rašeta, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Fokus ovog rada jeste na energiji vetra i optimizaciji rada vetroturbine. Konkretno, problematika je usmerena ka sistemu za praćenje smera vetra koji predstavlja jedan od ključnih faktora za proizvodnju energije. Promatrana je kalibracija i optimizacija ugla uređaja za merenje smera vetra koji se nalazi na kućištu vjetroturbine. Za rešenje ovog problema upotrebljeno je mašinsko učenje nakon čega su prikazani i rezultati simulacije sa zaključkom.

Ključne reči: vetroturbina, optimizacija, mašinsko učenje

Abstract – The focus of this paper is on wind energy and optimization of wind turbine operation. In particular, the problem is directed towards a yaw control system, which presents one of the key factors for energy production. The problem of yaw misalignment is especially observed (calibration and optimization of the angle of the wind direction measuring device located on the nacelle). Machine learning was used to solve this problem, followed by the simulation results with a conclusion.

Keywords: windturbine, optimization, machine learning

1. UVOD

Procenjena kratkotrajna potrošnja goriva na bazi uglja i značajano zagađenje atmosfere tokom poslednjih decenija uzrokovali su brzi razvoj obnovljivih izvora energije. Prelazak na održivu proizvodnju energije postao je pitanje od kapitalnog značaja kako bi se osigurala dugoročna stabilnost energetskog sistema. Među svim dostupnim obnovljivim izvorima energije, vetar i fotonaponska energija stoje kao najveći i najbrže rastući resursi.

Kako je tehnologija vetrogeneratorsa poznata, u ovom radu neće se ići u detaljno objašnjavanje i principe rada same vetroturbine (detaljan opis dat je u samom master radu). Pažnja je posvećena sistemu za praćenje smera vetra. Iako sistem za praćenje smera vetra (eng. yaw control system) nije detaljno proučen u literaturi od velikog je značaja u sistemu upravljanja vetroturbinama. Pravilno usmjeravanje kućišta (glave) turbine omogućava povećano iskorišćenje mehaničke snage izvučene iz vetra pa samim tim i povećanje proizvedene električne energije. Bitno je napomenuti kako je optimalan položaj kućišta vetroturbine onaj za koji se ima ugao od nula stepeni između kućišta vetroturbine i apsolutnog smera vetra. Ako dođe do odstupanja ovog ugla od nule duži vremenski period, dolazi do gubitka proizvedene energije odnosno

do nepotpunog iskorišćenja energije vetra. U ovom radu posebna pažnja usmerena je ka statičkoj neusklađenosti ose mernog uređaja i ose rotora odnosno nedovoljno dobroj kalibraciji sistema za praćenje vetra, dok se podrazumeva da sistem za praćenje smera vetra radi ispravno i da turbina u svakom trenutku prati smer vetra. U stvarnim aplikacijama, nulta tačka uređaja (senzora) za smer vetra može odstupati od ose rotora iz razloga ugradnje, rada u otežanim uslovima ili uticaja iskrivljenih vazдушnih masa iza ravni rotora.

U ovoj situaciji rezultat merenja smera vetra uključuje sistematsku grešku koja će dovesti do neusklađivanja između položaja kućišta turbine i smera vetra, što posledično dovodi do smanjenja proizvodnje energije vetra u radu vetroturbine. Za rešavanje ovog problema upotrebljen je poseban oblik mašinskog učenja nakon čije simulacije su prikazani rezultati.

2. SISTEM ZA PRAĆENJE SMERA VETRA

Moderne vetroturbine obično su projektovane kao vetroturbine „uz vetar“ kod kojih je površina lopatica okrenuta ka vetru i zato su pod aktivnom kontrolom sistema za praćenje smera vetra. Glavni razlog za ovakvu konstrukciju jeste interferencija struja vetra sa stubom, što je manje kod konstrukcija „uz vetar“. Sistem za praćenje smera vetra pokušava da poravna rotor prema vetru.

Ovaj sistem je spor u poređenju sa promenama smera vetra (koje su nepredvidive) i brze reakcije turbine mogu prouzrokovati veću potrošnju energije ovog elektromotornog sistema i naprezanje komponenti te se zbog toga prihvata određeni stepen greške [1]. Detaljan opis principa rada sistema za praćenje smera vetra dat je u master radu dok je u ovom radu predstavljen glavni problem kod ovog sistema, a to je sistematska greška nastala iz neusklađenosti nulte ose mernog uređaja i ose rotora koja prouzrokuje nedovoljno dobro poravnanje sistema za praćenje vetra sa smerom vetra.

Da bi se dobio što bolji uvod u problematiku, u narednom delu predstavljeni su ključni uglovi sistema za praćenje smera vetra.

2.1. Definicija ključnih uglova

Postizanje dobre usklađenosti turbine sa smerom vetra teško je iz nekoliko razloga. Jedan od njih je taj što kod standardnih turbina, merenja brzine i smera vetra vrše se iza rotora. To dovodi do izobličenja struja vetra uzrokovanih lopaticama rotora i samim kućištem. Kako bi se ovaj problem prevazišao postoje rešenja koja uključuju ugradnju mernih uređaja ispred turbine, ali ona podrazu-

NAPOMENA:

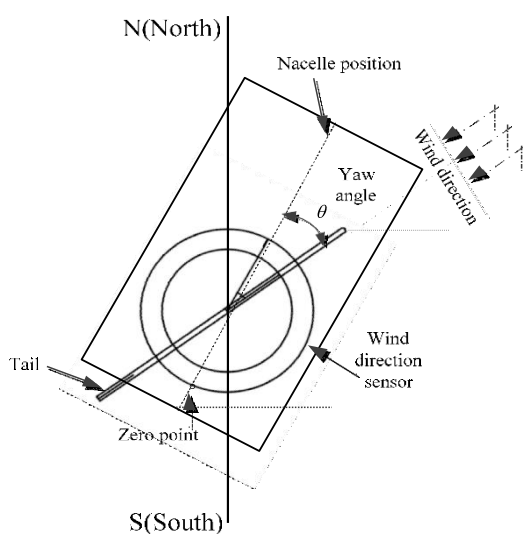
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Katić, red. prof.

mevaju ugradnju novog hardvera i samim tim povećane investicione troškove [1].

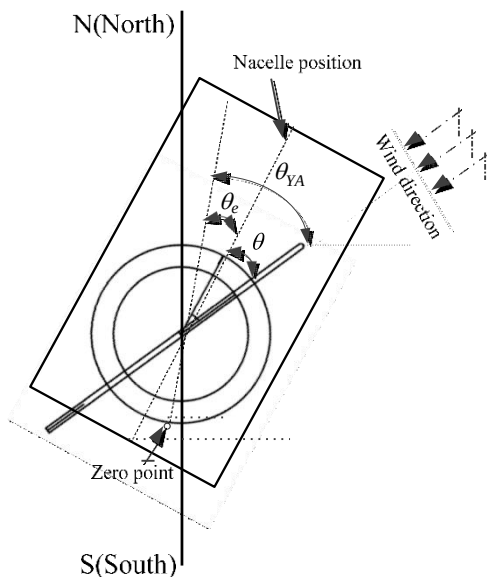
Na Slici 1. prikazan je ugao Θ koji predstavlja ugao između ose rotora i apsolutnog smera vetra. Sistem za praćenje smera vetra aktiviraće se da bi se smanjio ovaj ugao ako je isti veći od predefinisane praga za koji se ima aktivacija sistema za praćenje.

Turbine su nakon puštanja u rad usklađene sa pravim severom, ali oprema za merenje rotacije kućišta turbine često nije dovoljno tačna i dovodi do pomeranja oznake pravog severa tokom radnog veka [2].

Shodno tome, stare turbine često ne mogu pokazati ispravni smer vetra, jer je smer vetra određen iz zbira položaja kućišta i ugla sistematske greške. Ovaj problem ilustrovan je na Slici 2.



Slika 1. Definicija uglova u sistemu za praćenje smera vetra [1]



Slika 2. Definicija uglova u sistemu za praćenje smera vetra sa sistematskom greškom [1]

na kojoj su prikazani uglovi koji se imaju za već izloženi problem. Sa iste slike može se primetiti kako u ovom slučaju ose mernog uređaja i rotora nisu usklađene i ovaj ugao obeležen je sa θ_e .

U ovom slučaju za rezultat merenja ima se ugao θ_{YA} koji uključuje i sistematsku grešku za merenje ugla između kućišta turbine i smera vetra u SCADA sistemu. Ova sistematska greška, kao što je prethodno napomenuto, dovodi do neusklađenosti između pozicije kućišta i smera vetra te tako dovesti do smanjenja proizvodnje energije tokom rada vetroturbine.

2.2. Manuelno otkrivanje ugla sistematske greške

Na izlazu snage vetroturbine utiče nekoliko faktora, kao što su brzina vetra, ugao relativnog smera vetra, *pitch* ugao, stanje vetroturbine i tako dalje. Betzov zakon objašnjava kako brzina vetra i ugao relativnog smera vetra utiču na izlaznu snagu, kao što pokazuje jednačina (1) [1].

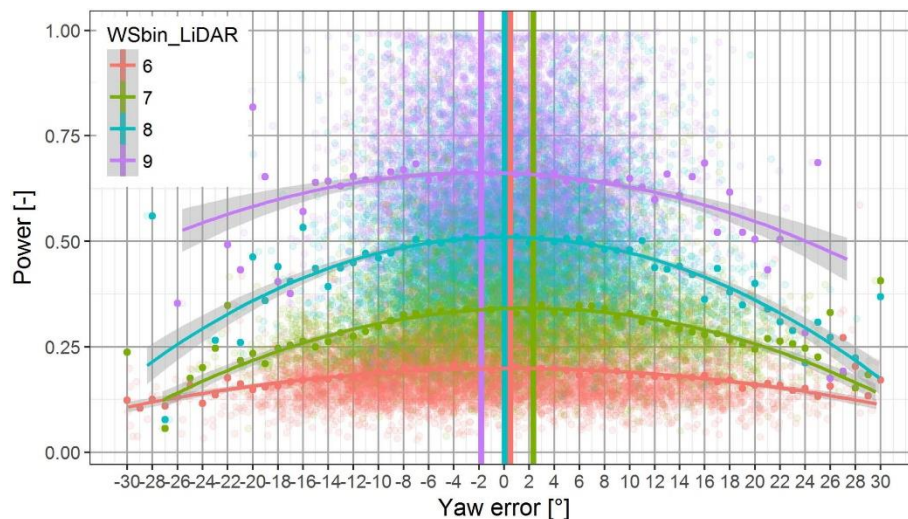
$$P = \frac{1}{2} \rho V^3 C_p \cdot (\cos \beta)^3 \quad (1)$$

Gde je P generisana snaga vetra, ρ je gustina vazduha, V je brzina vetra, C_p je koeficijent snage, dok je β ugao između kućišta turbine relativnog smera vetra. Iz jednačine (1) jasno se zaključuje kako se optimalna proizvodnja snage ima za ugao β jednak nuli. Promatrano s druge tačke gledišta, SCADA podaci s najboljim performansama snage trebaju odgovarati nultom uglu relativnog smera vetra.

Polazna tačka manualnog određivanja ugla sistematske greške je analiza performansi snage pod različitim uglovima relativnog smera vetra na osnovu SCADA podataka. Ugao za koji se ima najveća izlazna snaga smatra se za uglom sistematske greške (eng. *yaw misalignment*) koji je potrebno kompenzovati „pomeranjem“ sistema za praćenje smera vetra za upravo detektovan ugao [2]. To se obavlja tako što se za određeni rang brzine vetra, u kontrolni sistem, unose vrednosti ugla za koji se detektovala optimalna proizvodnja.

Kako bi se ovaj slučaj što lakše vizualizirao, na Slici 3. prikazane su krive snage u zavisnosti od ugla relativnog smera vetra za svaki rang brzine. Intervali brzine vetra definisani su sa korakom od po 0.5 m/s. Sa Slike 3. jasno se može uočiti kako je za brzine u intervalima 6 m/s i 8 m/s sistem za praćenje smera vetra dovoljno tačno poravnat doke se u slučajevima brzina od 7 m/s i 9 m/s uočava maksimalna izlazna snaga za uglove otprilike 2 i -2 stepena, respektivno.

Kako je navedeno u jednačini (1), tako i u literaturi, gubitak snage kao posledica ugla sistematske greške proporcionalan je kosinusu tog ugla na kub $((\cos \theta_e)^3$, gde je θ_e ugao sistematske greške sa Slike 2.) pa se za svaki ugao ove greške ima njemu proporcionalan gubitak snage.



Slika 3. Krive izlazne snage vetrotrubine u zavisnosti od relativnog smjera vetra za četiri inetrvala brzine i detektovanim uglom sistemske greške [2]

3. PRISTUP REŠAVANJU PROBLEMA UPOTREBOM MAŠINSKOG UČENJA

Glavna ideja za ovaj rad pa tako i sam master rad bila je upotreba mašinskog učenja u oblasti vetroindustrije u svrhu optimizacije rada vetrotrubine. Prethodno izložen problem simuliran je primenom posebne vrste mašinskog učenja odnosno tehnikom koja se zasniva na interakciji s okolinom, *Reinforcement learning*-om. Detaljnije objašnjenje principa rada pomenutog algoritma predstavljen je u samom master radu. U svrhu razvoja modela korišćen je programski jezik *Python*.

3.1. Mašinsko učenje – osnovni principi

Kako bi osnovni koncepti bili što jasnije predstavljeni, mašinsko učenje će biti posmatrano kao alternativa klasičnom inženjerskom pristupu. Konvencionalni inženjerski dizajn počinje sa sticanjem osnovnog znanja o problemu koji rešavamo. Nakon sakupljanja dovoljno teorijskog znanja definišemo matematički model, koji obuhvata fizičku strukturu posmatranog problema. Na osnovu modela formira se optimizovani algoritam koji garantuje očekivane performanse, pod uslovom da model predstavlja tačnu reprezentaciju stvarnosti.

Sa druge strane, mašinsko učenje (tj. njegova najosnovnija forma) umesto sticanja teorijskog znanja ima potencijalno lakši zadatak, a to je prikupljanje dovoljno velike količine primera (podataka) koji će opisati željena ponašanja algoritma od interesa. Ti primeri formiraju trening set.

Trening set se dalje prosleđuje odgo- varajućem algoritmu za trening, zahvaljujući kom dobijamo "istreniranu" mašinu za izvršavanje odgovarajućeg zadatka. Postupak učenja je moguć zahvaljujući dobrom izboru hipoteza, pomoću kojih algoritam vrši selekciju parametara tokom treninga.

Algoritmi mašinskog učenja su generalno bazirani na optimizaciji performansi, pri čemu se kao kriterijum za ocenu performansi uzima poklapanje stvarnih podataka sa onim koji su predviđeni pomoću mašine.

Postoje tri osnovne klase mašinskog učenja [3]:

- Nadgledano učenje (eng. *Supervised learning*)

- Nenadgledano učenje (eng. *Unsupervised learning*)
- Tehnika koja se zasniva na interakciji sa okolinom (eng. *Reinforcement learning* - RL), upotrebljena u ovom radu

3.2. Simulacija i rezultati

Kako rad ne bi poprimio softversku prirodu, u ovom delu kratko je pojašnjen princip tehnike učenja zasnovane na interakciji s okolinom. Glavna ideja koja se krije iza ovog principa jeste da se na okolinu, koja je modelovana kao vetrotrubina u ovom slučaju, primeni „agent“ odnosno algoritam koji će po principu poduzimanja akcija na trenutno stanje primati nagrade ili kazne i na taj način proizvesti naredno stanje [5]. Princip primanja nagrade (eng. *reward*) uveden je kako bi „agent“ naučio u kojem smeru treba da se kreće kako bi došao do željenog stanja. U ovom slučaju, željeno stanje jeste smanjiti gubitak snage na izlazu iz vetrotrubine. Na Slici 4. dat je prikaz jednog dela ponašanja „agenta“.

```

STATE BEFORE AN ACTION
Assembly error is: -8.265610518322255
Action is: 7.948283
  wind_speed  wind_direction
0    4.436451    -0.383333

STATE AFTER AN ACTION
  wind_speed  wind_direction  Power loss [%]
0    4.436451    -0.383333    0.02243

REWARD 10

```

Slika 4. Prikaz kompenzacije ugla sistemske greške od strane „agenta“

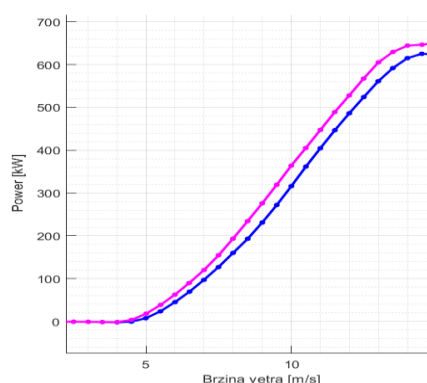
Ugao sistemske greške, u ovom slučaju nazvan i kao „ugao postavljanja mernog uređaja“ (eng. *assembly error*) zadan je u modelu nasumično od strane algoritma i u ovom konkretnom slučaju iznosi -8.27°. Glavni cilj jeste da „agent“ nauči da kompenzuje dati ugao i samim tim smanji gubitak snage koji bi se imao da isti ugao nije kompenzovan. Iz date Slike 4. može se primetiti kako je „agent“ uspešno kompenzovao nasumično zadat ugao i na taj način smanjio gubitak izlazne snage.



Slika 5. Kriva nagrade u odnosu na korake učenja (dokaz naučenog – konvergencija)

Da bi se dobila bolja predstava uticaja ove pojave na izlaznu snagu data je Slika 6. na kojoj se može primetiti razlika između dve krive snage (pre i posle kompenzacije) i uvideti da je kriva snage nakon kompenzacije ugla sistematske greške optimizovana što je i bio cilj simulacije.

Konkretno za ovaj ilustriran primer prikazan na Slici 6., „agent“ je uspešno naučio da kompenzuje ugao od otprilike 15° i samim tim povećao izlaznu snagu za 10%.



Slika 6. Krive snage pre (plava) i posle (roze) kompenzacije ugla sistematske greške

4. ZAKLJUČAK

Moderne vetroturbine obično su projektovane kao vetroturbine „uz vetar“ kod kojih je površina lopatica okrenuta ka vetru i zato su pod aktivnom kontrolom sistema za praćenje smera vetra. Pravilno usmeravanje kućišta (glave) turbine omogućava povećano iskorišćenje mehaničke snage izvučene iz vetra pa samim tim i povećanje proizvedene električne energije.

Optimalan položaj kućišta vetroturbine jeste onaj za koji se ima ugao od nula stepeni između kućišta vetroturbine i apsolutnog smera vetra. Ovaj rad daje uvid u problematiku ovog sistema vetroturbine.

Pored opisa standardnog rada sistema za praćenje smera vetra, predstavljen je problem pojave ugla sistematske greške ili ugla između ose mernog uređaja koji je postavljen na kućištu turbine i ose rotora za koji se ima pojava smanjenja izlazne snage.

U industriji se obično proces detekcije ugla sistematske greške obavlja manuelno te je svrha ovog rada da doprinese jednoj vrsti automatizacije tog posla, odnosno „pametnoj“ automatizaciji upotrebom mašinskog učenja. Polazna tačka manualnog određivanja ugla sistematske greške je analiza performansi snage pod različitim uglovima relativnog smera vetra na osnovu SCADA podataka. Ugao za koji se ima najveća izlazna snaga smatra se za uglom sistematske greške (eng. yaw misalignment) koji je potrebno kompenzovati „pomeranjem“ sistema za praćenje smera vetra za upravo detektovan ugao. U drugom delu rada predstavljena je simulacija koja je realizovana pomoću mašinskog učenja.

Mašinsko učenje u ovom domenu predstavlja način rešavanja problema tako što mašina „uči“ iz iskustva i samim tim optimizuje svoj rad tokom vremena. Detaljniji pristup ovoj tehnici mašinskog učenja izložen je u master radu. Rezultati koji se dobijaju simulacijom ukazuju na optimizaciju krive snage čija degradacija od strane ugla sistematske greške unutar sistema za praćenje vetra i predstavlja ključan problem. Predstavljena simulacija za sada je laboratorijski primer problema koji je u ovom radu izložen. Naredni napori biće usmereni ka primeni samog modela na pravoj turbini u polju i u tu svrhu zahteva dodatnu nadogradnju i optimizaciju.

5. LITERATURA

- [1] Yan Pei, Zheng Qian, Bo Jing, Dahai Kang, Lizhong Zhang, „Data-Driven Method for Wind Turbine Yaw Angle Sensor Zero-Point Shifting Fault Detection“, Energies 2018, 11(3), 553, March 2018.
- [2] N. Mittelmeier, M. Kühn, „Determination of optimal wind turbine alignment into the wind and detection of alignment changes with SCADA data“, Wind Energ. Sci., 3, 395–408, 2018
- [3] Y. Bengio, I. Goodfellow and A. Courville, „Deep learning“ In MA: MIT Press, 2016.
- [5] A.S. Aguirre, E. Zulueta, U.F. Gamiz, J. Lozano, J.M. Lopez-Guede, „Artificial Neural Network Based Reinforcement Learning for Wind Turbine Yaw Control“, Energies 12(3):436, January 2019

Kratka biografija:



Nataša Rašeta rođena je u Kninu, Republika Hrvatska, 1994. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Distribuirani elektroenergetski resursi odbranila je 2019. god.
kontakt: natasa.raseta23@gmail.com



dr. Vladimir Katić red. prof. rođen je u Novom Sadu 1954. godine. Doktorirao je na Univerzitetu u Beogradu 1991. godine. Oblasti interesovanja su mu energetska elektronika, obnovljivi izvori električne energije, kvalitet električne energije i električna vozila.

ИНВЕРТОРИ СА СИЛИЦИЈУМ – КАРБИДНИМ ТРАНЗИСТОРИМА

SILICON-CARBIDE TRANSISTOR INVERTERS

Никола Плавшић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – Дати су општи појмови о новим типовима полупроводника заснованим на материјалима са широким опсегом, а посебно су обрађени силицијум-карбидни (SiC) транзистори. Дато је њихово поређење са силицијумским компонентама и представљена примена у енергетској електроници. Урађен је модел SiC инвертора и низ симулација у MATLAB – у. Дат је преглед стања на тржишту, као и будући правци развоја.

Кључне речи: Инвертори, компоненте са широким опсегом, SiC

Abstract– General concepts about new types of semiconductors based on wide-band gap materials are given, and silicon carbide (SiC) transistors are specifically addressed. A comparison with silicon components is given and applications in power electronics are presented. A model of SiC inverters and a series of simulations in MATLAB have been developed. The market situation as well as future directions of development are given.

Keywords: Invertors, semiconductors based on wide-band gap materials, SiC

1. УВОД

У протеклој деценији у енергетској електроници се све чешће користе нови материјали за израду полупроводника као што су силицијум-карбид (SiC), галијум-нитрат (GaN) и многи други који су познати под називом полупроводници са широким опсегом (*wide-band gap*). Најбитније карактеристике ових полупроводника су побољшања у диелектричној чврстоћи, напону пробоја, ширини опсега, температурној издржљивости и др. у поређењу са стандардним силицијумским материјалима [1].

У овом раду је дат детаљан опис полупроводних компонената састављених од нових материјала (SiC), а нарочито њихова примена у уређајима енергетске електронике, конкретно у SiC инверторима.

2. КОМПОНЕНТЕ СА ШИРОКИМ ОПСЕГОМ (WIDEBAND GAP)

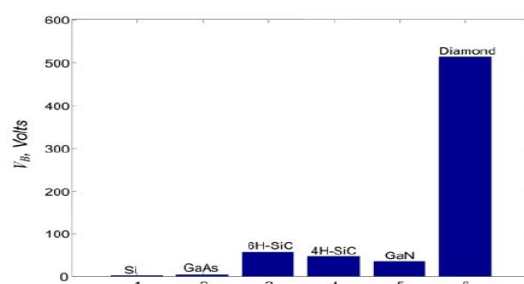
Полупроводнички материјали са широким опсегом могу се дефинисати као они чији енергетски процеп је већи од 2.2 eV. Полупроводничке компоненте са широким опсегом омогућавају побољшање ефикасности постојећих електричних кола са релативно једноставним контролним шемама [2].

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био проф др Владимир Катић.

Према недавним истраживањима предвиђено је да се ови уређаји користе на средњем напону, а такође се процењује да би могли наћи примене и у енергетским претварачима на високом напону [2]. На слици 1 приказане су вредности пробојних напона за различите материјале, што указује на већу напонску издржљивост.

Око 45% произведене електричне енергије троши се на електричне моторе. Процењује се да би се коришћењем компонената, тј. прекидачких елемената са широким опсегом, уместо доступних силицијумских IGBT, ефикасност електромоторних погона могла повећати за 20% - 30% [2].

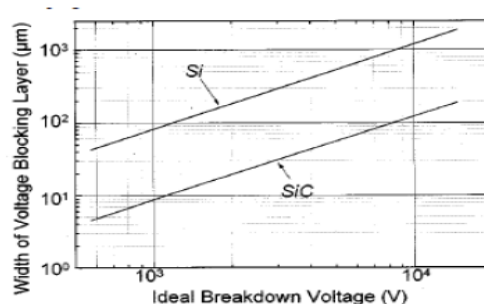


Сл. 1. Вредности пробојних напона за различите материјале. [2]

Издржљивост електричног поља полупроводника зависи од дебљине PN споја. Најмања дебљина PN споја се може одредити преко следеће једначине:

$$W \geq \frac{2V_b}{E_{max}} \quad (1)$$

где је V_b пробојни напон, E електрично поље, а W је дебљина p-n споја. Са слике 2 се примећује да SiC компоненте имају 10 пута већу вредност електричног поља E_{max} од Si компонената. Стога се закључује да SiC компоненте за исти напонски ниво имају 10 пута мању дебљину PN споја од силицијумских компонената.



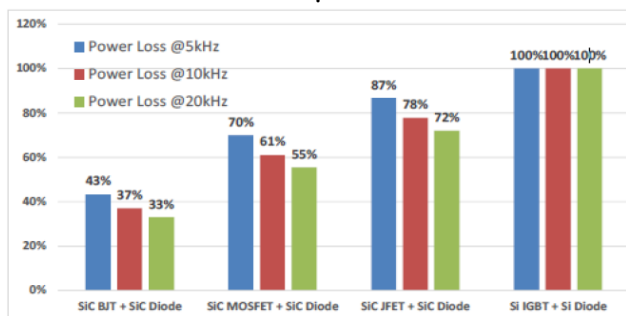
Сл. 2. Напон пробоја за SiC и за Si компоненте. [2]

Такође, ове компоненте имају скоро 10 пута већу диелектричну чврстоћу и око 5 пута већу топлотну проводљивост од силицијумских компонената. SiC компоненте имају пробојни напон реда kV, а Si компоненте су ограничене од 500V до 1000V [2].

3. КАРАКТЕРИСТИЧНЕ ПРИМЕНЕ

КОМПОНЕНАТА СА ШИРОКИМ ОПСЕГОМ

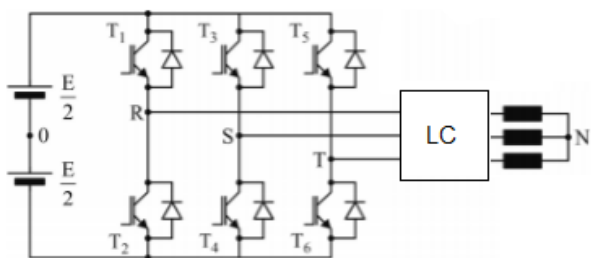
Карактеристичне примене компонената са широким опсегом су вучни системи, извори напајања, електрична возила, корекција фактора снаге. Своју примену све чешће налазе и у претварачима за повезивање обновљивих извора енергије (соларне или ветро електране) на мрежу. У јапанској компанији MITSUBISHI је установљено да се применом претварача са SiC компонентама укупни губици смањују за око 30 % [2].



Сл. 3. Поређење губитака за Si и SiC компоненте [2]

4. ИНВЕРТОРИ – ОПИС РАДА И ВРСТЕ

Из обновљивих извора електричне енергије добија се једносмерни напон (на пр. код соларних) или се он појављује у процесу претварања у једносмерном међуколу (на пр. код ветроелектрана). За прикључење на дистрибутивну наизменичну (AC) мрежу потребно га је претворити у мрежни напон одговарајуће учестаности (50 Hz или 60 Hz). Слична ситуација је и у електромоторним погонима код фреквентних претварача, када једносмерни напон међукола треба претворити у наизменични варијабилне учестаности. У таквим случајевима користе се инвертори (DC/AC) [3,4]. У зависности од области примене инвертори могу бити мрежни (*grid-tied*) и моторски, а примену налазе у соларним електранама, за конверзију при преносу електричне енергије једносмерном струјом и високим напонем – HVDC (High-Voltage Direct Current) [3,5]. Могу бити са два нивоа излазног напона, три нивоа и инвертори са више напонских нивоа (*Multi level inverters*) [6]. На слици 4 приказана је шема трофазног напонског инвертора.

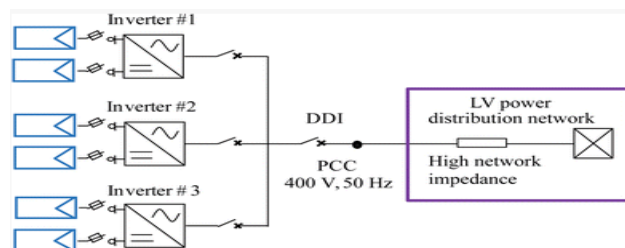


Сл. 4. Трофазни напонски инвертор [4]

Инвертор је преко LC филтера за сузбијање виших хармоника (блок LC) повезан на електричну мрежу или на неки други потрошач, нпр. електро мотор.

На слици 5 дат је пример примене у соларним електранама, где је приказана шема три паралелно везана низа фотонапонских (ФН) панела који су преко одговарајућих инвертора спрегнути са нисконапонском мрежом [7,8]. Могуће је и решење са једним, централним инвертором, који би покривао све ФН панеле. Снага инвертора бира се спрам снаге на излазу из ФН панела умањене за 10% до 15%.

На слици 6, приказана је примена инвертора у електромоторном погону АС асинхронног мотора, где се он налази у другом степену и директно врши регулацију брзине и момента овог мотора [9].

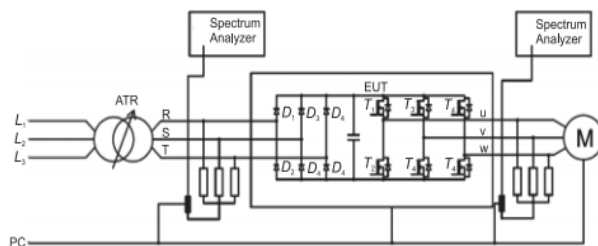


Сл. 5. Повезивање ФН панела на постојећу електричну мрежу [7]

5. ИНВЕРТОРИ СА SiC ТРАНЗИСТОРИМА

Компоненте засноване на SiC омогућавају да се развијају склопови енергетске електронике (у овом случају инвертори) који у поређењу са традиционалним претварачима са Si компонентама нуде већу ефикасност (мање губитке на транзисторима), мање димензије, већи праг толеранције у погледу загревања [10]. Обзиром на то да толеришу већи улазни напон (1200V до 1500V) од класичних инвертора базираних на силицијумским компонентама, ови инвертори се све чешће користе на средњем и високом напону. Повећањем напона приликом преноса електричне енергије смањује се јачина струје а самим тим и топлотни губици.

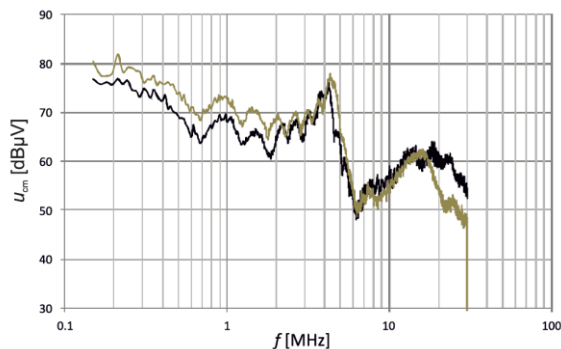
За тестирање SiC инвертора коришћена је шема погона са кавезним асинхронним мотором. Мерења су вршена помоћу осилоскопа. На слици 6 је приказана шема мерења [10]. Посматран је поремећај напона.



Сл. 6. Шема за мерење струје, напона и учестаности у погону са асинхронним мотором

Поремећај напона на излазу из претварача настаје услед изобличења (утицаја виших хармоника) мрежног напона, тј. напона на улазу у сам претварач.

На поремећај напона такође утичу и паразитивне индуктивности и капацитивности, електромагнетне сметње, рад транзистора на високим прекидачким учестаностима итд. На слици 7 приказан је поремећај напона за инверторе са Si и са SiC компонентама. Виде се мале разлике. Поред тога, таласни облик напона зависи и од алгоритма управљања претвараčem. Поремећаји напона зависе и од врсте потрошача. На пример, при стартовању мотора долази до пропада напона (када напон у кратком временском интервалу падне испод називне вредности).



Сл. 7. Поремећај напона на излазу претварача са Si (црна линија) и са SiC (зелена) транзисторима [10]

Поред наведених предности, постоје и недостаци претварача са SiC компонентама што се тиче електромагнетне компатибилности. Ови уређаји имају већи утицај на окружење у погледу електромагнетних сметњи. Тестирањем ових склопова енергетске електронике установљено је да су нивои електромагнетних сметњи знатно виши, посебно у случајевима када је носећа учестаност повећана и када се користи контрола помоћу импулсно-ширинске модулације. Ови недостаци негативно утичу на опрему која је прикључена на исту мрежу. Електромагнетне сметње се могу смањити коришћењем филтера на улазу и излазу из претварача [10].

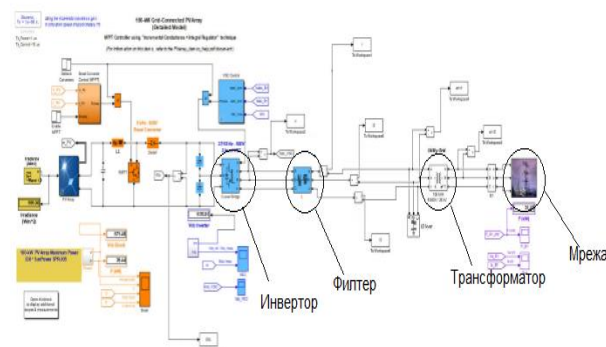
5. СИМУЛАЦИЈА РАДА SiC ИНВЕРТОРА

У овом поглављу биће представљена симулација рада инвертора са SiC транзисторима по шеми са слике 5. Са улазне стране на инвертор прикључени су соларни ФН панели, а са друге, излазне стране инвертор је преко трансформатора прикључен на постојећи електроенергетски систем. За симулацију је коришћен програмски пакет MATLAB 2011. На слици 8 приказан је MATLAB/Simulink модел инвертора прикљученог на мрежу.

Максимални излазни напон једне ФН ћелије се креће у опсегу од 600 до 700 mV. Спајањем на ред више ФН ћелија добија се жељени излазни напон једног модула. Соларни панел је састављен од више ФН модула везаних редно или паралелно у зависности од потребног излазног напона.

За симулацију је коришћена таква конфигурација соларног панела која се састоји од 17 редно везаних модула у једном низу и 19 паралелно везаних низова. Излазни напон је око 600 V. Укупна излазна снага панела је око 35 kWp. Напон на излазу из соларног панела се преко подизача напона доводи до самог

инвертора. На излазу из инвертора повезан је филтер за сузбијање виших хармоника. Затим се напон помоћу енергетског трансформатора повећава на одговарајући напонски ниво погодан за прикључење на постојећи електроенергетски систем.

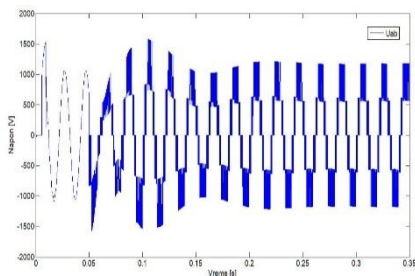


Сл. 8. Simulink модел инвертора прикљученог на електроенергетски систем

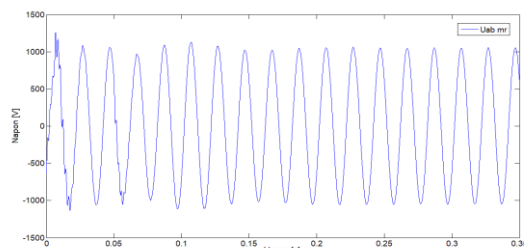
Са слике 9 може се приметити да напон на излазу из инвертора нема у потпуности синусни облик, а то је због утицаја виших хармоника. Са слике се такође може уочити да је амплитуда излазног међуфазног напона око 1250 V, што омогућава прикључења и на неке средње-напонске мреже. Поређења ради, на основу обрасца (2) може се израчунати амплитуда излазног међуфазног напона:

$$U_{abm} = \frac{2\sqrt{3}}{\pi} U_{DC} = 1,1 \cdot 1.175,01 = 1.292,6 \text{ V} \quad (2)$$

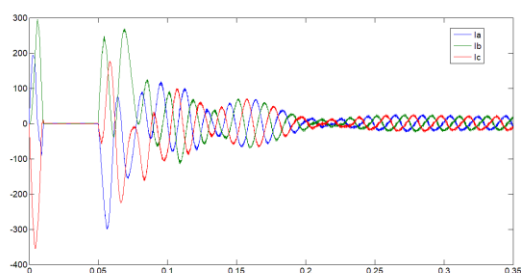
Филтрирањем напона са слике 9, тј. увођењем додатног склопа са LC филтером (слике 4 или 8), може се добити жељени квалитет излазног напона, што је приказано на слици 10. Излазна струја инвертора зависи од рада комплетног погона, а на слици 11 приказана је струја примара мрежног трансформатора при процесу укључивања на мрежу.



Сл. 9. Међуфазни напон на излазу инвертора



Сл. 10. Таласни облик међуфазног напона на излазу из филтера



Сл. 11. Таласни облик струје у све три фазе на нисконапонској страни трансформатора

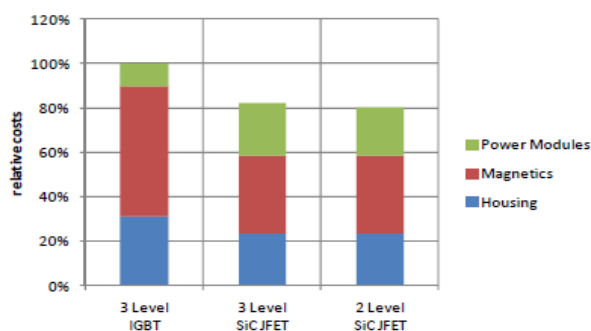
Проблем су укупни губици у инвертору, који се могу израчунати на основу следећег обрасца:

$$P_g = 3 \cdot (P_{\text{con-t}} + P_{\text{con-d}}) + P_{\text{sw}} \quad (3)$$

где су: $P_{\text{con-t}}$ – губици током провођења транзистора, $P_{\text{con-d}}$ – губици током провођења диоде и P_{sw} – прекидачки губици. Због виших фреквенција рада, прекидачки губици су доминантни.

6. ТРЖИШТЕ SiC ИНВЕРТОРА - ЦЕНЕ И БУДУЋИ РАЗВОЈ

Трошкови SiC инвертора анализирани су за три главне компоненте, модуле за напајање, магнетне компоненте и хладњак. Ово су компоненте које су технички повезане са самим модулом за напајање и они претежно доприносе у структури трошкова соларног претварача [2]. На слици 12 приказана је структура трошкова с обзиром на ове три главне компоненте. Види се да су трошкови магнетних компоненти доминантни. Даље, на основу испитивања тржишта, утврђено је да су они обрнуто сразмерни повећањем прекидачке учестаности. На пример, повећањем прекидачке учестаности са 16 kHz на 32 kHz трошкови се смањују за око 70 % [2]. Ако се узме инвертор са три напонска нивоа са IGBT транзисторима као референтно решење, уочава се да се употребом SiC транзистора трошкови смањују за 20% [2].



Сл. 12. Трошкови добијени анализом претварача са различитим прекидачким елементима [2]

7. ЗАКЉУЧАК

У овом раду разматране су новије полупроводничке компоненте, компоненте са широким опсегом, а посебно оне базиране на силицијум-карбиду. SiC компоненте осим у енергетској електроници своје

место проналазе и у телекомуникационим системима, савременој LED расвети итд.

С обзиром на то да су се компоненте са широким опсегом релативно скоро појавиле на тржишту као и да су њихове карактеристике знатно боље од класичних Si компонента, самим тим је и њихова цена на тржишту знатно већа.

8. ЛИТЕРАТУРА

- [1] <http://mikroelektronika.elfak.ni.ac.rs/files/materijali-za-elektroniku/7-poluprovodnici.pdf>, Електронски факултет Ниш, 2017.
- [2] B. Ahmad, „Wide Band Gap Power Semiconductor Devices and their Applications“, M.Sc. Thesis, Aalto University, Espoo, Finland, 2015.
- [3] N. Mohan, T. Undeland, W. Robbins, „Power Electronics“, 3rd Ed., John Wiley and Sons, 2003.
- [4] М. Недељковић, С. Срђић, „Енергетски претварачи 1 - Основне топологије енергетских претварача“, Електротехнички факултет, Београд, 2016.
- [5] J. Arillaga, Y.H. Liu, N.R. Watson, „Flexible Power Transmission – The HVDC Options“, John Wiley and Sons, 2007.
- [6] A. Nordvall „Multilevel Inverter Topology Survey“ Chalmers University of Technology, Göteborg, Sweden, 2011.
- [7] E. Ghiani, F. Pilo “Smart inverter operation in distribution networks with high penetration of photovoltaic system”, Journal of Modern Power Systems and Clean Energy, Vol.3, pp.504-511, 2015.
- [8] R. Teodorescu, M. Liserre, P. Rodriguez, „Grid converters for photovoltaic and wind power system“, John Wiley and Sons, 2011.
- [9] J.A. Alves, G. da Cunha, P. Torri, „Medium voltage industrial variable speed drives“, Brazil, 2009. <https://pdfs.semanticscholar.org/7df6/86b0e46280824d6f39ec313acc2fb5a7b9a1.pdf>.
- [10] W. Czuchra, W. Mysinski, B. Woszczyna, „Analysis of the use of transistors based on SiC technology in inverters in the context of electromagnetic compatibility“, Technical Transaction, Electrical Eng., Vol.1-E, 2016, doi: 10.4467/2353737XCT.16.027.5289.

Кратка биографија:



Никола Плавшић рођен је 21.05.1992. у Сомбору. Основну школу завршио је у Сивцу. Средњу Техничку школу завршио је у Сомбору 2011. године и исте године уписује Факултет техничких наука у Новом Саду, судијски програм Енергетика, електроника и телекомуникације. На студијама се определио за смер Електро-енергетика – енергетска електроника и електричне машине и дипломирао је 2016. године на тему „Симулација рада система тракастих транспортера угља у термоелектрани „Никола Тесла – А“ у Обреновцу“. Мастер рад одбранио је у октобру 2019. године. Живи и ради у Београду.

NATIVESCRIPT ОКВИР ЗА РАЗВОЈ МОБИЛНИХ АПЛИКАЦИЈА

NATIVESCRIPT MOBILE APP DEVELOPMENT FRAMEWORK

Дарко Тачић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИЧКО И РАЧУНАРСКО ИНЖЕЊЕРСТВО

Кратак садржај – У овом раду анализиран је *NativeScript* радни оквир за развој мобилних апликација. Затим су специфицирани захтеви и дизајн мобилне апликације за евиденцију пацијената. На крају је специфицирана апликација имплементирана и тестирана у *NativeScript* радном оквиру.

Кључне речи – *NativeScript*, *Cordova*, *Sidekick*, *Playground*, *JWT*, *MyPatientCare*, *MyDoctorCare*

Abstract – This paper analyses the *NativeScript* mobile app development framework. Then it specifies requirements and design of a mobile app for managing patients. Finally, it describes the implementation and testing of the specified app.

Keywords – *NativeScript*, *Cordova*, *Sidekick*, *Playground*, *JWT*, *MyPatientCare*, *MyDoctorCare*

1. УВОД

Мобилне технологије спадају међу технологије које се најбрже развијају. Тренутно 66.53% светског становништва поседује мобилни телефон [1]. У циљу олакшавања програмерима креирања мобилних апликација развијена је *NativeScript* технологија. У питању је платформа која се користи за писање нативних вишеплатформских мобилних апликација уз помоћ познатих веб технологија (HTML, CSS и JavaScript).

Поменута технологија ће бити представљена кроз две мобилне апликације, чија је главна улога да воде евиденцију прегледа пацијената у болници.

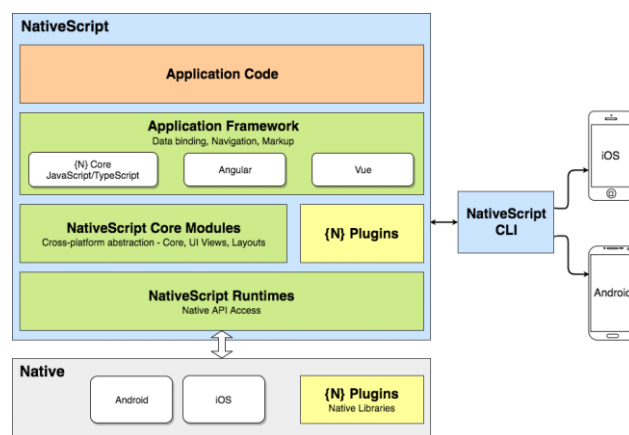
2. МОБИЛНЕ ПЛАТФОРМЕ

Мобилне апликације се могу груписати у четири велике категорије по врсти и окружењу: Native – Android и iOS, Hybrid – PhoneGap и Cordova, cross-compiled – Xamarin, JIT – *NativeScript*. Најзначајнија карактеристика ненативних апликација јесте могућност писања кода који може да се извршава на више платформи само једном. Технологије које ово омогућавају називамо *cross-platform*.

2.1 *NativeScript*

NativeScript [2], као један од савремених представника *cross-platform* технологија, спада међу JIT компајли-

ране апликације и користи JavaScript виртуелну машину као окружење за рад. *NativeScript* се састоји од следећих компоненти (слика 1): извршно окружење (енг. runtimes), модули (енг. core modules), кориснички интерфејс (енг. command line interface) и додаци (енг. plugins).



Слика 1. Компоненте које чине *NativeScript* [3]

2.2. JSON Web Token

JWT (JSON Web Token) представља стандард за креирање веб токена ради ауторизације корисника. Након што се клијент успешно аутентификује, сервер генерише овај токен и прослеђује га клијенту. JWT се састоји из три основна дела: заглавље (енг. header), садржај (енг. payload) и потпис (енг. signature).

3. СПЕЦИФИКАЦИЈА ЗАХТЕВА

Захтеви које апликација треба да испуни се могу груписати у функционалне и нефункционалне.

3.1 Функционални захтеви

У раду су приказани функционални захтеви као случајеви коришћења. Корисници система (лекари и пацијенти) имају низ заједничких и појединачних случајева коришћења.

Заједнички случајеви коришћења су:

- Логовање
- Регистрација
- Приказ посета
- Приказ детаља посете

Појединачни случајеви коришћења за пацијента су:

- Креирање посете
- Бирање третмана
- Бирање лекара

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Стеван Гостојић.

- Бирање датума
- Бирање времена
- Приказ контакта
- Слање email-а
- Слање sms поруке
- Телефонски позив
- Приказ информација здравствене установе

Појединачни случајеви коришћења за лекара су:

- Приказ посета
- Приказ детаља посета
- Потврда посете
- Одбијање посете
- Приказ годишњих одмора
- Додавање годишњег одмора
- Ажурирање годишњег одмора
- Приказ радних времена
- Додавање радног времена
- Ажурирање радног времена
- Ажурирање профила

3.2 Нефункционални захтеви

Најбитнији нефункционални захтеви које апликација треба да испуни су:

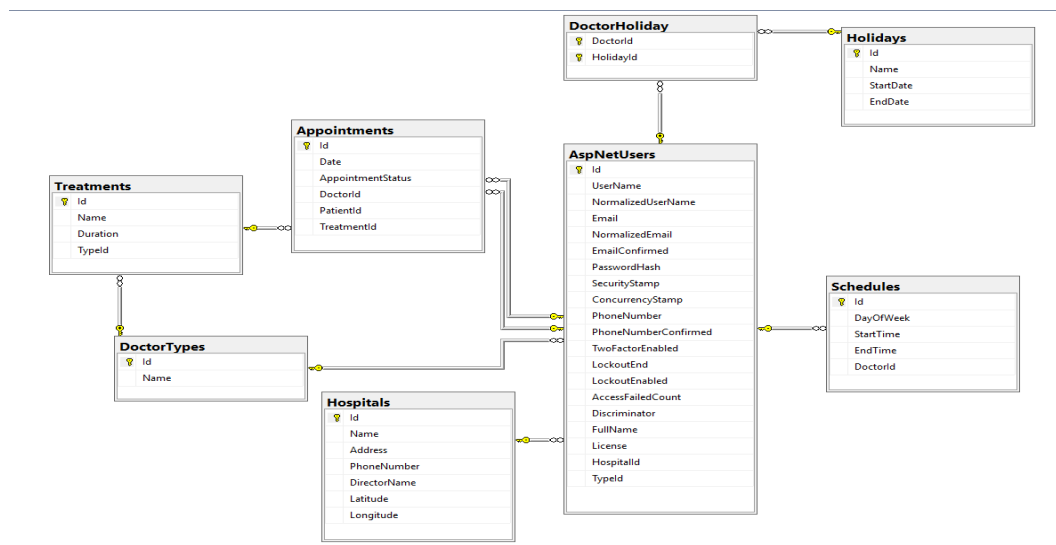
- Перформансе
- Безбедност
- Поузданост

4. СПЕЦИФИКАЦИЈА ДИЗАЈНА

Спецификацију дизајна апликације чине статички и динамички модели.

4.1 Статички модели система

Дијаграм класа представља врсту UML дијаграма који описује структуру система приказивањем класа система, њихових атрибута, операција (метода) и односа између објеката. Дијаграм класа за апликацију за вођења евиденције пацијената приказан је на слици 2.



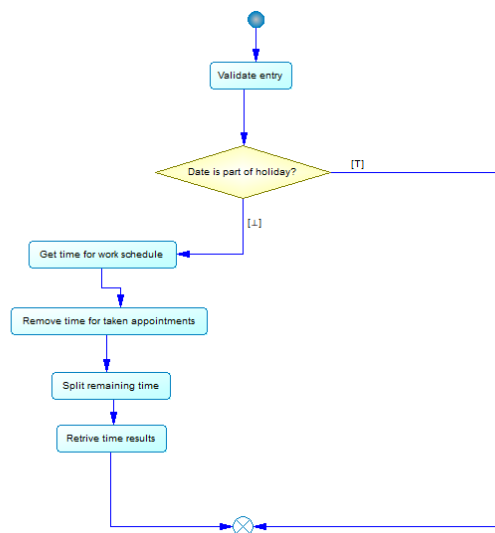
Слика 2. Дијаграм класа за пројекат вођења евиденције пацијената

4.2 Динамички модели система

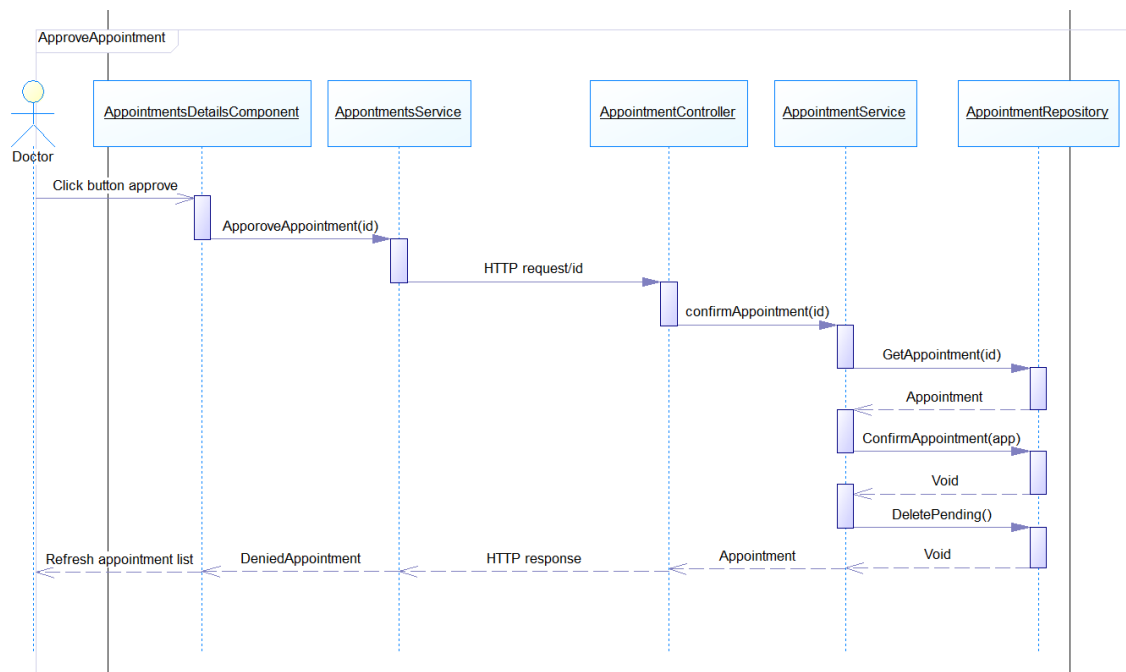
Дијаграм активности је дијаграм који истиче ток контроле од активности до активности. Користи се за детаљно описивање посебних случајева коришћења. Процес добављања слободних термина започиње валидацијом улазних података.

Уколико је валидација успешна, поставља се питање да ли за жељени датум има слободних термина. Следи провера да ли жељени датум припада неком годишњем одмору. Уколико не припада, добављају се сви термини за тај дан.

Међутим, уколико у бази већ постоје заказани термини за тај датум, потребно је елиминисати заузете термине из листе слободних времена. Коначно, кориснику се достављају одговарајући подаци (слика 3).



Слика 3. Дијаграм активности за добављање листе слободних термина



Слика 4. Дијаграм секвенце за процес потврде посете лекару

Дијаграм секвенце представља динамичку слику односа између објеката. Ова динамика се види из порука које они размењују. У пројекту је коришћен дијаграм секвенце за додатни опис случаја коришћења потврда посете лекару.

Процес потврде посете започиње притиском лекара на дугме "Approve Reservation". Даље, путем AppointmentsService-a се шаље HTTP захтев ка Backend серверу. Endpoint за овај захтев се обрађује у класи AppointmentController. Уколико је валидација података успешна, позива се confirmAppointment метода која се налази у класи AppointmentService. Уколико тражени Appointment постоји у систему, врши се промена стања у Approved, као и брисање свих оних резервација које су временски пресецале претходно потврђену резервацију.

Клијент на крају добија повратну информацију на освеженом екрану (Слика 4).

4.3 Опис сервиса система

Важно је дати осврт на сервис класе које сачињавају пословну логику апликације за вођење евиденције пацијената. Сервис класе су: UserService, TreatmentService, ScheduleService, HospitalService, ContactInfoService, HolidayService и AppointmentService.

5. ИМПЛЕМЕНТАЦИЈА

За реализацију пројекта коришћено је низ софтверских алата, који су међусобно повезани. Софтверски алати примењених у току израде пројекта су:

- Visual Studio [4]
- Visual Studio Code [5]
- NativeScript Playground [6]
- NativeScript Sidekick [7]
- SQL Management Studio [8]
- Android Virtual Device Manager [9]

Имплементација JSON Web токена обухвата више фаза. Пре него што сервер генерише токен, потребно је да у конфигурацији сервера омогућимо овај вид

комуникације. Жељени циљ се постиже додавањем CORS (Cross-Origin Resource Sharing) који представља механизам комуникације путем HTTP протокола између сервера различитог порекла. Порекло сервера може да зависи од његовог домена, протокола комуникација или од вредности порта.

// This method gets called by the runtime.
Use this method to configure the HTTP request pipeline.

```

/// <summary>
/// The Configure
/// </summary>
/// <param name="app">The
app<secref="IApplicationBuilder"/></param>
/// <param name="env">The
env<secref="IHostingEnvironment"/></param>
public void Configure(IApplicationBuilder
app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseAuthentication();
    app.UseCors(builder =>
        builder.AllowAnyOrigin()
            .AllowAnyHeader()
            .AllowAnyMethod());
    app.UseMvcWithDefaultRoute();
}
  
```

Листинг 1. Конфигурација сервера за CORS

6. ДЕМОНСТРАЦИЈА

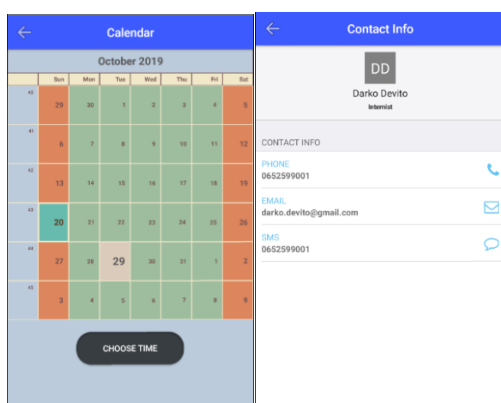
Апликација је демонстрирана приказом графичког корисничког интерфејса.

6.1 MyPatientCare

MyPatientCare је апликација за пацијенте. Основне функционалности које пацијент извршава помоћу ове апликације су заказивање прегледа, контактирање лекара и преглед информација о здравственој установи.

Основни екрани које чине MyPatientCare апликацију:

- Login Component
- Register Component
- Appointments Component
- Appointments Details Component
- Treatments Component
- Doctors Component
- Calendar Component (слика 5 – лево)
- Contact Component
- ContactInfo Component (слика 5 – десно)
- AboutUs Component



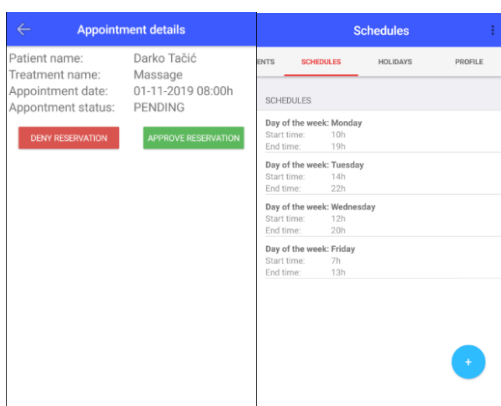
Слика 5. Изглед екрана са календаром и екрана са контакт подацима

6.1 MyDoctorCare

MyDoctorCare је апликацију за лекаре. Основне функционалности које лекар извршава су потврда или одбијање прегледа, ажурирање радног времена и ажурирање одсуства.

Основни екрани које чине MyDoctorCare апликацију су:

- Appointment Details Component (слика 6 – лево)
- Schedule Component (слика 6 – десно)
- Holiday Component
- Profile Component



Слика 6. Изглед екрана за потврду или одбијање резервације и екрана за приказ радног времена

7. ЗАКЉУЧАК

У раду је приказан NativeScript оквир, као алтернативни начин за развој мобилних апликација.

NativeScript сам по себи не нуди много нових могућности у поређењу са сличним технологијама. Међутим, захваљујући алатима који га окружују, комплетирани су све фазе развоја мобилних апликација.

Упркос високој интуитивности NativeScript-a, недостатак документације умногоме отежава његову примену. Ова мана се може објаснити његовом малој популарности, с обзиром да је оквир релативно нов. Ипак, тим који одржава NativeScript не одустаје од свог производа и наставља да надограђује постојеће функционалности.

8. ЛИТЕРАТУРА

- [1] <https://bankmycell.com/blog/how-many-phones-are-in-the-world> (Пристапљено: 17.09.2019.)
- [2] NativeScript (2019), Progres, Пристапљено: 17.10.2019. [Online] Доступно: <https://docs.nativescript.org/angular/core-concepts/technical-overview>
- [3] M. Branstein, N. Branstein, "The NativeScript ebook", 2016
- [4] VisualStudio (2019), Microsoft, Пристапљено: 05.08.2019. [Online] Доступно: <https://visualstudio.microsoft.com/>
- [5] VisualStudioCode (2019), Microsoft, Пристапљено: 03.09.2019. [Online] Доступно: <https://code.visualstudio.com/>
- [6] NativeScript Playground (2019), Progres, Пристапљено: 30.09.2019. [Online], Доступно: <https://play.nativescript.org/>
- [7] NativeScript SideKick (2019), Progres, Пристапљено: 28.09.2019. [Online], Доступно: <https://www.nativescript.org/nativescript-sidekick>
- [8] SQL Management Studio (2018), Microsoft, Пристапљено: 11.10.2019. [Online], Доступно: <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>
- [9] Android virtual device manager (2019), Google, Пристапљено: 15.10.2019. [Online], Доступно: <https://developer.android.com/studio/run/managing-avds>

Кратка биографија:

Дарко Тачић рођен је 15.10.1994. године у Суботици. Године. Гимназију „Светозар Марковић“ у Суботици завршио је 2013. године. Исте године уписао се на Факултет техничких наука, одсек Рачунарство и аутоматика. У четвртој години бира смер Интернет и електронско пословање. Положио је све испите предвиђене планом и програмом. Мастер студије је уписао 2017. године на истом одсеку, модул Електронско пословање.

РАЗВОЈ SDK БИБЛИОТЕКЕ ЗА ИМПЛЕМЕНТАЦИЈУ ИНДУСТРИЈСКИХ ПРОТОКОЛА

DEVELOPMENT OF THE SDK LIBRARY FOR THE IMPLEMENTATION OF INDUSTRIAL PROTOCOLS

Бранко Јелић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

2. ТЕОРИЈСКЕ ОСНОВЕ

Кратак садржај – У овом раду представљено је једно решење *Software Development Kit (SDK)* библиотеке за потребе развоја индустријских протокола. Поред описаног концепта решења, у раду су описане основе *SCADA* система, комуникационих и индустријских протокола.

Кључне речи: SCADA, RTU

Abstract – This paper introduces one solution of the *Software Development Kit (SDK)* library for industrial protocol development purposes. In addition to the described solution concept, the paper describes the basics of *SCADA* systems, communication and industrial protocols.

Key words: SCADA, RTU

1. УВОД

Непрекидан развој информационих технологија условио је паралелан напредак у индустрији. Многи индустријски процеси потпуно су аутоматизовани и укључени у системе за надзор и управљање подацима. Системи за надзор и управљање, познатији у свету као *SCADA (Supervisory Control And Data Acquisition)* системи, врше прикупљање и обраду велике количине података. Размена података између самог *SCADA* система и опреме за аутоматизацију у оквиру постројења врши се путем различитих индустријских комуникационих протокола. Само тестирање имплементације различитих комуникационих протокола је често отежано и ризично радити на самом индустријском систему. Због те чињенице се имплементирају симулатори индустријских система за различите комуникационе протоколе.

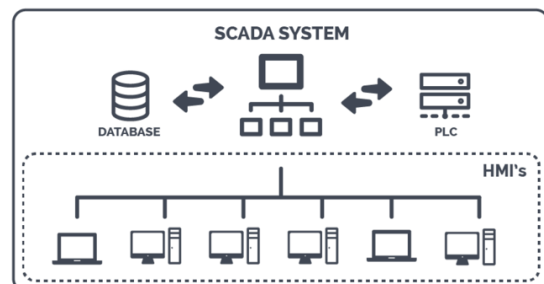
Идеја овог рада је развој *Software Development Kit (SDK)* библиотеке за потребе развоја индустријских протокола. Сам задатак има за циљ имплементацију и опис библиотеке која ће омогућити развој различитих индустријских протокола у сврху симулације индустријског система. Решење треба да омогући да развој библиотеке конкретног индустријског протокола обухвати само имплементацију специфичности самог протокола.

2.1 SCADA систем

Аквизиционо управљачки системи (*AUS*) представљају скуп просторно дистрибуираних и међусобно повезаних рачунарских модула који за заједнички циљ имају остварење функције надзора и управљања физичким процесом у реалном времену. Често се овакви системи (специфичне намене и структуре) у литератури означавају термином *SCADA (Supervisory Control And Data Acquisition)*. Кључни захтеви које аквизиционо управљачки систем опште намене мора испунити су:

- Рад у реалном времену,
- Дистрибуција рачунарских ресурса у оквиру индустријског система,
- Постизање максималне поузданости и расположивости.

Са становишта примењених алгоритама управљања и географске топологије постоје два основна типа аквизиционо управљачких система. Први тип представља управљање у географски размештеним системима (системима за пренос енергената и других сировина). Овакви системи имају инсталиране једноставне процесне контролере који се називају удаљене телеметријске станице (*RTU – Remote Telemetry Unit*). Код оваквих система комуникациони подсистем је често веома сложен и релативно спор у погледу брзине преноса података (укључује разнородне преносне медије). Други тип аквизиционо управљачких система се односи на управљање у просторно ограниченим индустријским постројењима (индустријска постројења код којих се углавном све налази у оквиру истог постројења). Код оваквих система комуникација се одвија поузданије и постоји висок ниво аутоматизације самих управљачких активности [1].



Слика 2.1.1 Архитектура SCADA мреже

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Бранислав Атлагић, доцент

Водећа процесна јединица или *MTU (Master Terminal Unit)* у *SCADA* систему представља уређај који се користи за издавање команди над *RTU* уређајима, прикупља захтеване податке, складишти податке, обрађује информације и исте приказује у различитим формама (сlike, графови, табеле) на корисничком приказном уређају (*HMI*) и на тај начин олакшава оператерима доношење одлука (слика 2.1.1). *MTU* се налази у оквиру контролног центра [1].

2.1.1 Прикупљање података у *SCADA* системима

Прикупљање података у *SCADA* системима се врши слањем поруке на *RTU* уређај и чекањем да стигне одговор. *RTU* уређај има везу са физичким светом у коме се врше мерења напона, струје, фазе... Одговор који стиже назад од *RTU* уређаја садржи конкретне вредности затражених тачака.

Постоје разни примери захтева, помоћу аналогних улаза се углавном добијају мерене вредности (струја, напон...), дигитални улази представљају најчешће стања прекидача, док се помоћу бројачких или импulsних улаза мери проток разних супстанци.

У зависности од типа опреме, улоге опреме у систему, као и ризика који носи са собом процес прикупљања података се може вршити сваких пар секунди, минута, сати, дана, месеци или година.

Увек треба рачунати на непредвиђене ситуације које су саставни део овако критичних система. *SCADA* систем треба да буде спреман да адекватно одреагује у складу са тренутно насталом непредвиђеном ситуацијом и да спречи упад система у неконзистентно стање. На пример постоји процес који прикупља податке сваких 20 минута, а у 10. минуту се догоди испад (прекoraчење). Део опреме где је измерено прекoraчење треба да обавести *RTU* уређај који ће након тога иницирати раније повлачење података [1].

2.2 Комуникациони протоколи

Комуникационим протоколом се означава скуп правила и процедура који контролишу ток комуникације и обезбеђују успешну интеракцију између удаљених процеса. Протокол дефинише формат и редослед порука које се размењују, као и сваку акцију која се иницира пријемом одређене поруке. За опис протокола се користи дијаграм промене стања или неки од виших програмских језика. Основне и најбитније функције сваког комуникационог протокола су контрола грешака и управљање током података у мрежи. Контрола грешака обезбеђује поуздан пренос и интегритет порука, док је примарни циљ управљања током избегавање загушења у саобраћају и дељење комуникационих ресурса између више корисника.

Структура комуникационих система је због своје комплексности строго хијерархијска и реализује се кроз више нивоа (слојева).

Неки од разлога који су условили слојевитост структуре протокола су следећи [1]:

- *Независност између нивоа.* Под овим се подразумева да виши ниво протокола није

условљен начином на који је имплементиран претходни.

- *Флексибилност.* Све док је постојећа спрега идентична, промена на једном нивоу не утиче на остале нивое.
- *Физичко одвајање.* Неки од слојева протокола могу бити реализовани на различитој физичкој основи, уз коришћење предности најсавременије технологије.
- *Једноставнија имплементација и одржавање.* Као последица модуларности слојева и декомпозиције укупних мрежних услуга, олакшан је развој и испитивање опреме и програмске подршке.
- *Могућност стандардизације.* Услуге и спреге појединих нивоа се могу стандардизовати, уколико се прецизно спецификују функције.

2.3 Индустрijски протоколи

Индустрijски протоколи и индустрijске мреже је уобичајено име за класу протокола специфичних за *SCADA* системе. Према типу учесника у комуникацији и њиховим улогама у АУС, индустрijски протоколи се могу класификовати у неколико типичних категорија. Неки протоколи имају примене у више суседних категорија.

- *Сензорски протоколи* омогућују повезивање са „smart“ сензорима и извршним уређајима, у циљу примарне аквизиције података из постројења.
- *Fieldbus протоколи* примарно су оријентисани комуникацији са *UI* модулима и контролерима повезаних процесном магистралом у индустрijску fieldbus мрежу.
- *Телеметријски протоколи* контролишу комуникацију између процесних контролера и централне *SCADA* станице. Због тога се код њих много пажње посвећује минимизацији обима података који се преносе.
- *Међусистемски протоколи* реализују спрегу између два аутономна али кооперативна система, без обзира да ли се ради о два *SCADA* система, вези *SCADA* система са симулатором, или нечему другом.

Поред наведених класа протокола, реални *SCADA* системи морају имплементирати клијентски протокол који преноси *SCADA* податке до оператерских радних станица или пословних апликација. *Modbus* је најпопуларнији индустрijски протокол.

У електроенергетским *SCADA* апликацијама, у Европи су доминантни протоколи *IEC 60870-5101/104*, а у Америци протокол *DNP3*. Оба протокола имају заједничко порекло и развијени су баш за примену у системима за пренос и дистрибуцију електричне енергије.

Спрега између аутономних *SCADA* система, стандардизована је углавном кроз два стандарда: *ICCP (Inter-Control Center Communications Protocol)*, и *OPC (OLE for Process Control)*. *ICCP* је ту једини релевантан стандард, јер је независан од програмске платформе и једини нуди адекватне перформансе.

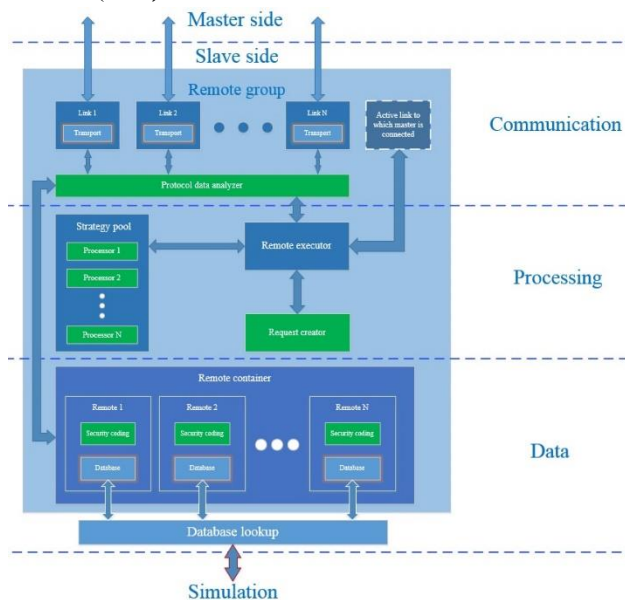
2.3.1 Комуникациони приступи

Комуникациони приступи подржавају ефикасну комуникацију преко мреже широког подручја до *RTU*, за прикупљање процесних података и пренос управљачких команди, који се могу оптимизовати и са становишта безбедности и трошкова. Комуникациони приступи подржавају различите конфигурације. Данас су најпопуларније у употреби [1]:

- *Multidrop* је радијална конфигурација где се *RTU* позивају у низу преко једног комуникационог канала. Ово даје јефтиније решење на рачун времена одзива.
- *Point-To-Point* намењује један комуникациони канал за један *RTU*. Обично се користе за главне трафостанице или концентраторе података који имају *RTU* са великим улазно/излазним захтевима.
- *Loop* (петља) ради у конфигурацији отворене петље подржане са два комуникациона приступа, где је сваки канал *multidrop* типа.
- *Star* (звезда) конфигурација је комбинација *point-to-point* до *RTU* концентратора података, који управља приступом подацима до подређених *RTU* конфигурираних као *point-to-point* или *multidrop*. Ове конфигурације се користе у аутоматизованим дистрибуцијама где мешовито време одзива може бити економски планирано.

3. КОНЦЕПТ РЕШЕЊА

На слици 3.1 је приказана архитектура самог решења подељена у више целина (делова). Целине су комуникација (*communication*), обрада (*processing*) и подаци (*data*).



Слика 3.1 Приказ везе између основних компоненти у архитектури решења

Разлог овакве архитектуре решења је у томе што је подржана *multi drop* веза у пољу између више *RTU*-ова једног комуникационог протокола. Дакле ако је реч о *Point-To-Point* вези између *master* стране и *RTU* онда ће *Remote container* садржати само један *Remote*.

Уколико се ради о *multidrop* вези више *RTU*-ова *Remote container* ће имати више *Remote*-ова.

Важна напомена је да су плавом бојом на слици 3.1 означене компоненте које су имплементиране у оквиру библиотеке, а зеленом бојом оне компоненте које ће корисник имплементирати по унапред дефинисаним интерфејсима приликом имплементације индустријског протокола.

Свака *Remote group* може имати више комуникационих канала (*Link*) путем којих размењује поруке са *master* страном.

Remote executor представља компоненту која је задужена за координацију осталих компонентама из више различитих целина.

4.1 Комуникација (*Communication*)

Као што се може видети на слици 3.1 комуникациони део је одговоран за размену (пријем и слање) података са *master* страном.

Као што је раније речено свака *Remote group* има своје *Link*-ове. Кључни разлог оваквог дизајна лежи у томе што сваки *RTU* уређај у пољу је повезан са остатком система путем више различитих комуникационих веза (*Link*-ова). У сваком тренутку *RTU* размењује податке са *master* страном путем искључиво једне од комуникационих веза (по природи *RTU* редно обрађује захтеве пристигле са *master* стране и размена података путем више комуникационих веза није могућа) и она се назива активна комуникациона веза (*Active link*). Изузетак је кад су у квару све комуникационе везе, тада *RTU* нема активну комуникациону везу.

Веома важан део сваког *Link*-а представља коришћени протокол за размену порука на транспортном нивоу (*TCP* или *UDP* су протоколи који се користе на транспортном нивоу). Он одређује на који начин се врши пријем и слање порука преко мреже.

Када се *master* страна повеже на неки од *Link*-ова он постаје активан (*Active link*) тек онда када *master* страна прекине везу са свим осталим *Link*-овима.

Оног тренутка кад неки од *Link*-ова постане активан, *Remote executor* почиње пријем, обраду и слање података са тог *Link*-а.

Основна замисао је да се кориснику омогући имплементација специфичности протокола кроз *Protocol data analyzer*.

Приликом пријема података константно ће бити консултован *Protocol data analyzer* који би требало да препозна да ли је пристигла порука примљена у потпуности и уколико јесте да изврши валидацију пристиглих података. У супротно треба да одбаци поруку или настави пријем док порука не буде примљена у потпуности. *Protocol data analyzer* има задатак да одреди за који *Remote* из *Remote container*-а је стигла порука.

Такође је омогућена подршка симулационом окружењу за укључивање и искључивање *Link*-ова по жељи што омогућава симулацију прекида везе.

4.2 Обрада (Processing)

Обрада примљене поруке почиње у оном тренутку када *Protocol data analyzer* заврши са њеном валидацијом.

Remote executor наставља са даљом обрадом поруке. Он позива *Request creator*-а који има задатак да од примљене поруке направи одговарајућу протокол захтев.

Request creator је компонента која се креира од стране корисника библиотеке и мора бити имплементирана на унапред дефинисан начин путем њеног универзалног интерфејса.

Након тога *Remote executor* консултује *Strategy pool* на који ће од *Processor*-а послати захтев на обраду. *Strategy pool* је садржи за одређени тип захтева *Processor* који је одговоран за његову обраду.

Даља обрада се одвија на *Processor*-у који поред самог захтева добија и *Remote* на ком захтев треба да се изврши. Корисник библиотеке има задатак да у потпуности имплементира за сваки тип захтева *Processor* који ће вршити његову обраду и креирање одговора. Када *Processor* обради захтев и креира одговор *Remote executor* ће преко активног линка поруку послати назад на *master* страну.

4.3 Подаци (Data)

Компоненте одговорне за чување података за сваку *Remote group*-у су приказани на слици 3.1 у оквиру секције *Data*.

Компонента одговорна за чување *Remote*-ова у оквиру *Remote group*-е се назива *Remote container*. Ова компонента је потпуно независна од протокола.

Дизајн самог *Remote* је такође осмишљен да буде независан од индустријског протокола али са могућности проширења са специфичности конкретног индустријског протокола. Сваки *Remote* може имати (али и не мора) *Security coding* чија је имплементација препуштена ономе ко имплементира конкретан индустријски протокол.

Приликом дизајнирања решења водило се рачуна о томе да манипулацију вредностима тачака може обављати више страна. Прва страна је *master*, који путем протокол захтева има могућност промена и читања вредности тачака (у зависности од типа тачке).

Другу страну представља симулационо окружење, коме је такође омогућена промена и читање вредности тачака. Битна ствар коју треба напоменути је да симулационо окружење има могућност мењања и читања вредности свих типова тачака.

Из претходно наведених разлога сваки *Remote* има своју *Database* у оквиру које се налазе све његове тачке. Приступ *Database* се очекује искључиво приликом извршавања протокол захтева.

Database lookup компонента омогућава приступ тачкама свих *Remote*-ова једног индустријског протокола. Осмишљен је у сврху приступа тачкама и мењања њихових вредности из симулационог окружења.

5. ИМПЛЕМЕНТАЦИЈА РЕШЕЊА

Само програмско решење имплементирано је у програмском језику *C#*, коришћењем *.NET 4.6.2* софтверске платформе у *Microsoft Visual Studio 2015* развојном програмском окружењу. Приликом израде самог решења коришћене су најбоље софтверске праксе попут: *TAP* програмирања [2], *S.O.L.I.D* принципа [3], дизајн патерна [3], *Test-Driven Development*-а [4].

6. ЗАКЉУЧАК

У овом раду је представљено програмско решење *SDK* библиотеке за потребе развоја индустријских протокола. Пре саме реализације рада било је потребно детаљно се упознати са *SCADA* системом и његовом архитектуром, као и специфичностима разних индустријских протокола. Идеја самог рада је била да се омогући лака и једноставна имплементација групе или појединачних удаљених телеметрисаних јединица (*RTU*) различитих индустријских протокола, који би се користили за опонашање реалног система.

У самом програмском решењу је остављено простора за размишљање о будућим унапређењима. Даљи правац развоја би подразумевао подршку за *unsolicited* поруке, као и пружање могућности кориснику да пресреће пристигле поруке и дефинише одговоре који ће се слати назад на *master* страну. Сама литература о индустријским протоколима је слабо јавно доступна па је самим тим и отежано узети у обзир све случајеве приликом развоја саме библиотеке. Једно од будућих унапређења би свакако било проналазак адекватне литературе и модификација постојећег решења у складу са стеченим знањем.

7. ЛИТЕРАТУРА

- [1] *Софтвер са критичним одзивом*, 2015, Бранислав Амлагић
- [2] <https://docs.microsoft.com/en-us/dotnet/standard/asynchronous-programming-patterns/task-based-asynchronous-pattern-tap>
- [3] *Beginning SOLID Principles and Design Patterns for ASP.NET Developers*, 2016, Bipin Joshi
- [4] *Test-Driven Development by Example*, 2002, Kent Beck

КРАТКА БИОГРАФИЈА

Бранко Јелић рођен је 07.09.1995 године у Сремској Митровици. Завршио је Електротехничку школу “Михајло Пупин” у Новом Саду 2014. године. Исте године је уписао основне академске студије на Факултету техничких наука у Новом Саду. Дипломски рад из области Електротехника и рачунарство – Примењени софтверски инжењеринг одбранио је 2018. године. Испунио је све обавезе и положио све испите предвиђене студијским програмом.

**KONVERZIJA PODATAKA ELEKTROENERGETSKE MREŽE DEFINISANE U CIM-u
U ESRI SHAPEFILE FORMAT****CONVERSION OF POWER GRID DATA DEFINED IN CIM TO ESRI SHAPEFILE
FORMAT**Rade Ilić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Ovim radom je obuhvaćeno istraživanje *Esri shapefile* formata podataka koji se koristi za čuvanje geoprostornih podataka i kako se definišu geoprostorni podaci elektroenergetskih resursa u CIM-u. Takođe je istražena mogućnost konverzije geoprostornih podataka definisanih u CIM-u u *Esri shapefile*. Kao rezultat istraživanja nastala je aplikacija *CIMToShapefileConverter* koja može da konvertuje geoprostorne podatke elektroenergetske mreže definisane u CIM verziji 16 u *shapefile* format. Performanse aplikacije su testiranje i analizirane, a ispravnost rezultujuće *shapefile* datoteke testirane u QGIS-u.

Ključne reči: *Shapefile*, CIM, konverzija podataka, geoprostorni podaci, elektroenergetska mreža

Abstract – This paper covers research on *Esri shapefile* data format which is used for storing spatial data and how spatial data is defined in CIM. Possibility of conversion of geospatial data defined in CIM to *Esri shapefile* is also researched. As a result of that research an application is developed named *CIMToShapefileConverter* which is able to convert geospatial data of power grid defined in CIM version 16 to *shapefile* data format. CPU usage of application has been analyzed, and resulting *shapefiles* were tested in QGIS.

Keywords: *Shapefile*, CIM, data conversion, geographical data, power grid

1. UVOD

Rastući broj aplikacija i kompanija koje razmenjuju podatke doveo je do značajnog porasta kompleksnosti integracija tih aplikacija. Dolazi do potrebe za zajedničkim formatom podataka koji se razmenjuju u svim segmentima elektroenergetike. Za razmenu ovih podataka se koristi CIM. CIM je definisan serijom standarda propisanih od strane međunarodne elektrotehničke komisije IEC (*International Electrotechnical Commission*) koji definišu model elektroenergetskog sistema. ENTSO-E (*European Network of Transmission System Operators for Electricity*) je mreža operatora prenosnih mreža širom Evrope napravljena sa ciljem da se poboljša njihova saradnja i omoguće integracije podataka. ENTSO-E je razvio CGMES (*Common Grid Model Exchange Standard*) standard koji se zasniva na IEC standardima i koji je prihvaćen od strane IEC [1].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Milan Gavrić.

CGMES sadrži profile *GeographicalLocation* i *Diagram-Layout* koji omogućavaju definisanje geoprostornih podataka elektroenergetskih resursa. *Shapefile* format je geoprostorni vektorski format podataka za softver geografskih informacionih sistema (GIS). Razvijen je i regulisan od strane *Environmental Systems Research Institute* (Esri) kao otvoren standard za interoperabilnost podataka između Esri i drugih GIS softverskih rešenja [2] [3]. Otvoren standard i interoperabilnost su omogućili *shapefile*-u da bude podržan od strane velikog broja GIS softverskih rešenja.

Pošto CIM sadrži klase koje opisuju geoprostorne podatke elektroenergetskih resursa, moguće je napraviti aplikaciju koja će da pročita ove podatke iz CIMXML datoteke i da ih konvertuje i serijalizuje u *shapefile* format. Cilj ovog rada je da se razvije ovakva aplikacija, kao i da se rezultujuće *shapefile* datoteke testiraju u nekom od GIS softverskih rešenja koji nije razvijen od strane Esri-ja i na taj način proveriti interoperabilnost *shapefile* formata.

2. OPIS KORIŠĆENIH TEHNOLOGIJA

Ovo poglavlje sadrži kratak osvrt na tehnologije, metode i principe koji su korišćeni za implementaciju projektnog rešenja. Opisane su samo tehnologije i principi koji su najviše uticali na implementaciju rešenja.

2.1. CIM

CIM (*Common Information Model*) je zajednički model podataka definisan nizom standarda propisanih od strane međunarodne elektrotehničke komisije IEC (*International Electrotechnical Commission*) koji omogućavaju aplikacijama razmenu informacija o resursima u prenosnoj i distributivnoj elektroenergetskoj mreži [4] [5]. CIM se održava kao UML dijagram. Za razmenu CIM podataka koriste se CIMXML datoteke. CIMXML datoteka koristi RDF (*Resource Description Framework*) model podataka za opis konkretnih instanci CIM resursa. Ove instance su serijalizovane XML sintaksom.

2.2. Shapefile

Shapefile je geoprostorni vektorski format podataka za softver geografskih informacionih sistema (GIS) [2]. Razvijen je i održavan od strane *Environmental Systems Research Institute* (Esri) kao otvoren standard za interoperabilnost podataka između Esri i drugih softverskih rešenja. Esri je međunarodna kompanija koja razvija softver za geografske informacione sisteme. Esri-jevi proizvodi imaju 40% udela u GIS tržištu, a posebno se ističe ArcGIS Desktop. Detaljan opis i tehnička dokumentacija *shapefile* formata se može pronaći na zvaničnom sajtu Esri-ja [2] [3]. Ovaj format podataka može prostor-

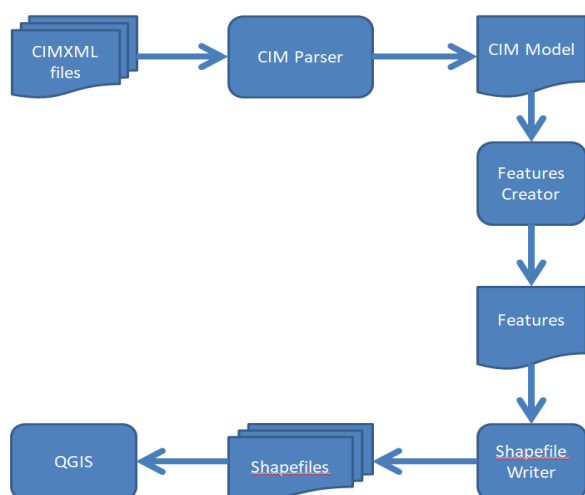
no da opiše vektorske karakteristike kao što su: tačke, linije i poligoni, koji mogu da predstavljaju objekte iz realnog sveta. Na primeru iz elektroenergetske mreže prekidači se mogu predstaviti kao tačke, a provodnici kao linije. Svaki *shapefile* može da sadrži više ovakvih elemenata koji se u datoteci čuvaju kao oblici (eng. *shape*), i svaki od njih može da sadrži dodatne atribute koji ih opisuju. *Esri shapefile* se sastoji od tri obavezne datoteke: *main file*, *index file* i *dbf file*. Ove tri datoteke moraju da imaju ekstenzije, respektivno: *.shp*, *.shx* i *.dbf*. Svaka datoteka mora da ima isti naziv. *Main* datoteka čuva geometrijske podatke, *index* datoteka čuva pozicije slogova unutar *main* datoteke a *dbf* datoteka sadrži bilo koji skup atributa koji se pridružuju oblicima iz *shapefile*-a.

2.3. QGIS

QGIS (Quantum GIS) je aplikacija otvorenog koda za rad sa *GIS*-om. *QGIS* podržava otvaranje *Esri shapefile* datoteka. Omogućava grafički prikaz u više slojeva (eng. *layer*) gde svaki sloj ima drugu *shapefile* datoteku kao izvor podataka i prikazuje ih sve na istom prozoru. Elementima iz svakog sloja može da se dodeli boja kojom će biti obojeni kako bi se razlikovali od elemenata iz drugih slojeva. Umesto tačke moguće je dodeliti neki od predefinisanih simbola ili simbola koje korisnik napravi. *QGIS* omogućava dodavanje mape kao novi grafički sloj. Na ovaj način je omogućeno da se vidi geografska lokacija elemenata na mapi. *QGIS* je korišćen kako bi se testirala interoperabilnost *shapefile* formata.

3. DIZAJN REŠENJA

Rešenje je implementirano kao *Microsoft .Net WPF (Windows Presentation Framework)* aplikacija. Prilikom razvoja *WPF* aplikacije korišćen je *MVVM* šablon koji razdvaja razvoj grafičkog korisničkog interfejsa od razvoja poslovne logike. Slika 3.1 prikazuje šematski dijagram toka podataka prilikom korišćenja aplikacije.



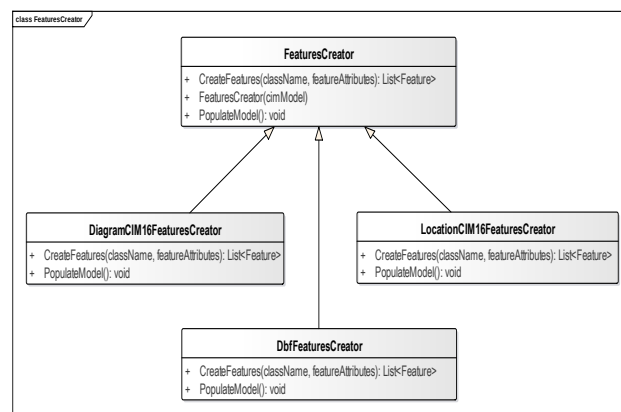
Slika 3.1 Arhitektura rešenja i šematski tok podataka

Kao ulazne podatke aplikacija prima *CIMXML* datoteke koje sadrže model elektroenergetske mreže kao i geoprostorne podatke resursa u mreži. Komponenta *CIMXML parser* učitava ulazne datoteke i parsira ih.

Parsirani podaci se šalju komponenti *Feature Creator* koja podatke učitane iz datoteke pretvara u listu oblika i njihovih atributa. *Shapefile Writer* prima ovu listu i konvertuje je u *shapefile*.

4. OPIS IMPLEMENTACIJE

Aplikacija se sastoji od tri komponente: *CIMXML parser*, *FeaturesCreatorCIM16* i *ShapefileWriter*. *CIMXML parser* učitava jednu ili više *CIMXML* datoteka, prolazi kroz njih liniju po liniju i parsira *XML* elemente i atribute. Parser zatim čita *CIM* resurse i njihove atribute koji su zapisani *XML* sintaksom. Kao rezultat parsiranja dobija se instanca klase *CIMModel*. Ova klasa u sebi sadrži kolekciju svih resursa, a za svaki resurs sadrži kolekciju vrednosti atributa tog resursa. Druga komponenta *FeaturesCreatorCIM16* je i najvažnija komponenta. Apstraktna klasa *FeaturesCreator* sadrži metode koje su potrebne da bi se kreirali *Feature*-i. *Feature* je klasa koja sadrži geoprostorne podatke u atributu *geometry* i atribute resursa u atributu *attributes*. Instanca *Feature* klase predstavlja model *shapefile* oblika u memoriji. U ovoj aplikaciji podržani su oblici *Point* i *MultiLine*. *Feature* predstavlja model podataka koji će biti jedan slog u *shapefile*-u. *FeaturesCreator* sadrži konstruktor koji prima *CIMModel* klasu kao ulazni parametar, jer se *Feature*-i stvaraju na osnovu pročitanih resursa iz datoteke. Apstraktnu klasu *FeaturesCreator* nasleđuju tri klase: *DiagramCIM16FeaturesCreator*, *LocationCIM16FeaturesCreator* i *DbfFeaturesCreator*. Klasni dijagram na slici 4.1 prikazuje kako se implementiraju *FeaturesCreator*-i.

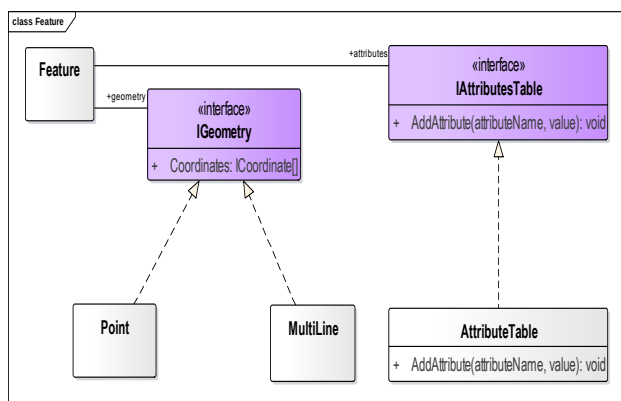


Slika 4.1 *FeaturesCreator* klasni diagram

DiagramCIM16FeaturesCreator klasa služi za kreiranje *Feature*-a tako što koristi *Diagram* klasu iz *CIMModel*-a kao izvor geoprostornih podataka. Ova klasa je implementirana prema *CIM*-u verziji 16, jer koristi klase *Diagram*, *DiagramObject* i *DiagramObjectPoint* koje su definisane u ovoj verziji. Instance ovih klasa kreira metoda *PopulateModel* na osnovu učitanih podataka iz *CIMXML* datoteke. Interna struktura ovih podataka se koristi za kreiranje *Feature*-a u metodi *CreateFeatures*. Ova metoda kreira *Feature*-e tako što prolazi kroz kolekciju *DiagramObject*-a, i za svaki *DiagramObject* se kreira jedan *Feature*. *DiagramObject* može da sadrži jedan ili više *DiagramObjectPoint*.

Ako sadrži jedan *DiagramObjectPoint* onda će *geometry* atribut da sadrži *Point* oblik, a ako sadrži više od jednog onda će *geometry* atribut da sadrži *MultiLine* oblik. *CreateFeatures* metoda kao parametar prima naziv klase, jer se *Feature*-i prave za svaku klasu posebno.

Razlog toga je što svaka klasa ima različite attribute, a slogovi u *dbf* datoteci moraju imati iste attribute, što znači da će različite klase biti eksportovane u različite *shapefile*-ove. *LocationCIM16FeaturesCreator* radi isto što i *DiagramCIM16FeaturesCreator*, jedino je razlika što ova klasa koristi *Location* klasu kao izvor geoprostornih podataka. *DbfFeaturesCreator* klasa se koristi kada korisnik u aplikaciji odabere opciju da kreira samo *dbf* datoteke. Ova klasa će da od ulaznih resursa iz *CIMXML*-a napravi *Feature*-e koji nemaju geoprostorne podatke, već samo listu atributa. *ShapefileWriter* komponenta služi za kreiranje *shapefile* datoteka. Podaci o oblicima koji će biti zapisani u *shapefile* se dobijaju u listi *Feature*-a od komponente *FeatureCreator*-a. Slika 4.1 prikazuje klasni dijagram koji opisuje kako su modelovani *shapefile* oblici pomoću klase *Feature*. *Feature* klasa sadrži dva atributa, *attributes* i *geometry*. *ShapefileWriter* za svaki *Feature* napravi jedan slog u *shapefile*-u.



Slika 4.2 *Feature* klasni dijagram

5. TESTIRANJE REŠENJA

Prilikom testiranja razvijene aplikacije testirana je brzina rada, odnosno koliko je opterećen procesor prilikom rada. Takođe naknadno je testirana validnost rezultujućih *shapefile*-ova koristeći QGIS.

5.1 Brzina rada

Brzina rada aplikacije je testirana pomoću *profiler*-a, alata u okviru programskog okruženja *Visual Studio*. Slika 5.1 prikazuje rezultat testiranja aplikacije pomoću *profiler*-a. *Profiler* rezultate prikazuje u obliku tabele koja ima kolonu *Function Name* koja izlistava imena metoda u aplikaciji koje su testirane, i kolonu *Total CPU (%)* koja pokazuje procentualno koliko se metoda vremenski izvršavala u odnosu na to koliko se cela aplikacija vremenski izvršavala. Tabela sadrži i neke druge kolone koje nisu prikazane na slici, na primer *Total CPU (ms)* koja prikazuje vreme izvršavanja metode u milisekundama. Kolona *Function Name* je organizovana kao stablo, tako da kada se izabere neka funkcija otvara se podstablo funkcija koje se iz nje pozivaju. Analizirajući rezultate vidi se da metoda *LoadXMLFiles* vremenski najviše opterećuje procesor. Sledeća metoda koja najviše

zauzima procesor jeste metoda *CreateFeatures*, međutim vreme izvršavanja je znatno manje u odnosu na metodu *LoadXMLFiles*. *LoadXMLFiles* metoda najviše vremena potroši prilikom parsiranja elemenata u *XML* datoteci kao i na rad sa datotekama. Takođe je testirana zavisnost vremena izvršavanja parsiranja od veličine datoteke. Korišćene su *CIMXML* datoteke koje sadrže definisane elektroenergetske resurse srednjenaponskih izvoda jedne severno-američke distribucije i njihove geoprostorne podatke koristeći *CIM 16*.

Function Name	Total CPU (%)
CIMToShapefileConverter.vshost.exe (PID: 13604)	100.00%
[External Code]	100.00%
Microsoft.VisualStudio.HostingProcess.HostProc!RunUsersAssembly	87.10%
[External Code]	87.10%
CIMToShapefileConverter.App!Main	49.65%
[External Code]	49.65%
CIMToShapefileConverter.Main!MainWindowRectangle_Drop	37.20%
CIMToShapefileConverter.ViewModel.Main!MainWindowViewModelLoadXMLFiles	37.14%
[External Code]	0.06%
dynamic!ClassIL_STUB_CLRtoCOM	0.91%
CIMToShapefileConverter.ViewModel.My!CommandSystem.Windows.Input.Command.Execute	0.41%
CIMToShapefileConverter.ViewModel.Main!MainWindowViewModelCreateShapefile	0.39%
CIMToShapefileConverter.Model.ShapefilesWriter!Write	0.38%
CIMToShapefileConverter.Model.DiagramLayout!DiagramCIM16FeatureCreatorCreateFeatures	0.37%
[External Code]	0.01%
[External Code]	0.01%
[External Code]	0.01%
Microsoft.VisualStudio.Debugger.Runtime.Implementation!0x007fb458d1c86	0.26%

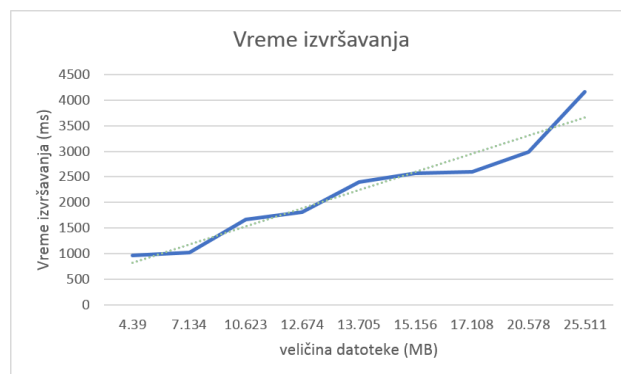
Slika 5.1 *Profiler* analiza rada aplikacije

Tabela 5.1 prikazuje nazive fidera koji su korišćeni za testiranje, veličina datoteke u megabajtima i vreme izvršavanja u milisekundama.

Tabela 5.1 *Vreme izvršavanja parsiranja*

Feeder	File size [MB]	Elapsed [ms]
Feeder 1	4.39	967
Feeder 2	7.134	1025
Feeder 3	10.623	1673
Feeder 4	12.674	1812
Feeder 5	13.705	2398
Feeder 6	15.156	2565
Feeder 7	17.108	2597
Feeder 8	20.578	2983
Feeder 9	25.511	4157

Na slici 5.3 je prikazan grafik koji se dobije korišćenjem podataka iz tabele 5.1.



Slika 5.3 *Vreme izvršavanja parsiranja*

Grafik predstavlja zavisnost vremena izvršavanja od veličine datoteke. Na grafiku se može videti tendencija promene vremena izvršavanja u zavisnosti od veličine datoteke. Vreme izvršavanja raste linearno kada se povećava veličina datoteke.

5.2. Testiranje shapefile-ova

Za testiranje rezultujućih *shapefile*-ova korišćena je aplikacija *QGIS*. *Shapefile*-ovi koji su testirani su dobijeni konverzijom *CGMES CIMXML* datoteka koje *ENTSO-E* koristi za *CIM interoperability* testove. Rezultat sadrži dva *shapefile*-a, jedan za klasu *ACLineSegment* i jedan za klasu *Substation*. Slika 5.3 prikazuje ova dva *shapefile*-a otvorena u *QGIS*-u. Instance *ACLineSegment* klase su prikazane kao zelene linije, a instance *Substation* klase kao plavi kvadrati.



Slika 5.3 *Shapefile* otvoren u *QGIS*-u

Svaki *shapefile* sadrži *dbf* datoteku koja može da sadrži dodatne attribute koji se pridružuju oblicima. *QGIS* za svaki *shapefile* nudi mogućnost otvaranja tabele atributa. Slika 5.4 prikazuje izgled ove tabele za *ACLineSegment*.

	ID	r0	shortCircuit	bch	length	x	r	name
1	_0447c6f1-c766...	2.87914181	80	0.000312213	101.782356	9.23472	2.822688	82-96
2	_04483c2a-c766...	5.633876	80	0.0006732093	155.154892	20.0898724	5.523408	23-32
3	_04486330-c766...	5.49169636	80	0.00059573	61.4510345	17.59824	5.384016	69-77
4	_04488a45-c766...	9.508277	80	0.0002341598	118.117485	28.314	9.32184	103-105
5	_04499bb6-c766...	8.637425	80	0.000196281	81.91258	23.8708782	8.468064	49-51
6	_044a10e4-c766...	0.6812784	80	0.00131818186	114.272278	7.744	0.66792	65-68
7	_044a37f2-c766...	4.22985029	80	0.0006083563	37.14677	17.371727	4.146912	89-90
8	_044a5f09-c766...	1.2736944	80	0.00254132226	130.187241	15.5848007	1.24872	9-10
9	_044b7078-c766...	9.41941452	80	0.0002708907	26.6151047	31.88592	9.23472	105-107
10	_044b9789-c766...	2.11492515	80	8.18411E-05	37.2328453	9.40896	2.073456	5-6
11	_044bbe91-c766...	1.20457923	80	0.00240082643	250.2668	14.762	1.18096	8-9
12	_044c5ad6-c766...	2.488147	80	8.23003E-05	64.94043	9.530928	2.43936	105-106
13	_044c5ad7-c766...	0.9672968	80	2.14646E-05	29.1255341	2.631024	0.850291	55-56
14	_044c5ads-c766...	4.21207762	80	0.0001365932	38.38349	16.4308319	4.129488	28-29
15	_044c81e1-c766...	7.108992	80	0.0001905418	44.9179039	23.6269436	6.9696	45-46
16	_044c81e3-c766...	3.94450331	80	0.001876033	149.0798	41.624	3.86716	26-30
17	_044cd003-c766...	2.77250671	80	0.0004958678	31.222496	13.9391994	2.718144	23-25
18	_044cd006-c766...	5.296199	80	0.0001440542	69.213974	17.16264	5.192352	31-32
19	_044e56a2-c766...	3.181274	80	7.21993E-05	74.0993347	8.79912	3.118896	48-49
20	_044ef2e1-c766...	3.00354886	80	0.000115932	32.4472427	12.3187675	2.944656	54-55
21	_044ef2e7-c766...	1.53909636	80	6.76079E-05	1	7.910496	1.508918	71-73

Slika 5.4 *ACLineSegment* tabela atributa

Svaka kolona predstavlja jedan atribut, a svaki red predstavlja vrednosti tih atributa za jedan oblik u datom *shapefile*-u. Ova tabela se otvara desnim klikom na sloj nekog *shapefile*-a u *QGIS*-u, a zatim na opciju *Open Attribute Table*.

6. ZAKLJUČAK

Kao rezultat ovog rada dizajnirana je i implementirana aplikacija *CIMToShapefileConverter*. Aplikacija je pokazala način na koji je moguće konvertovati geoprostorne podatke definisane u *CIMv16* u *shapefile*. Ispravnost dobijenih *shapefile*-ova je testirana u *QGIS*-u. Testiranjem performansi aplikacije zaključeno je da aplikacija najviše optereti procesor prilikom parsiranja datoteka i da vreme izvršavanja linearno raste kada se povećava veličina datoteke.

Pošto ova aplikacija podržava samo *CIMv16*, smer budućeg razvoja bi mogao biti usmeren ka dodavanju podrške i za ostale verzije *CIM*-a.

Pored ovog potencijalnog proširenja, aplikacija bi mogla da se proširi sa *plugin* arhitekturom. *FeaturesCreator*-i bi se u tom slučaju učitavali dinamički kao *dll plugin*. Ovaj *plugin* bi trebao da sadrži implementaciju apstraktne klase *FeaturesCreator*.

Na ovaj način bi se olakšalo proširivanje aplikacije i svako bi mogao da implementira svoj *FeaturesCreator plugin* za bilo koju verziju *CIM*-a.

7. LITERATURA

- [1] ENTSO-E, "Common Grid Model Exchange Standard," <https://www.entsoe.eu/digital/cim/#common-grid-model-exchange-specification-cgmes>. (pristupljeno u oktobru 2019.)
- [2] ESRI, "ESRI Shapefile Technical Description," <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. (pristupljeno u oktobru 2019.)
- [3] ESRI, "Spatial Data Standards and GIS Interoperability," <https://www.esri.com/library/whitepapers/pdfs/spatial-data-standards.pdf>. (pristupljeno u oktobru 2019.)
- [4] "Core IEC Standards," <https://www.iec.ch/smartgrid/standards/>. (pristupljeno u oktobru 2019.)
- [5] Mathias Usler et al., "The Common Information Model CIM: IEC 61968/61970 and 62325 - A practical introduction to the CIM" (Power Systems), Berlin, Springer, 2012.

Kratka biografija:



Rade Ilić, rođen je Zrenjaninu 1995. god. Školske 2014/2015. upisao se na Fakultet tehničkih nauka u Novom Sadu, odsek Elektrotehnika i računarstvo, smer Primenjeno softversko inženjerstvo. Školske 2018/2019. je upisao master akademske studije na Fakultetu tehničkih nauka u Novom Sadu, smer Primenjeno softversko inženjerstvo.

МОДЕЛИ ПОДАТАКА ЕЛЕКТРОЕНЕРГЕТСКОГ СИСТЕМА ДЕФИНИСАНИ У ГРАФ ОРИЈЕНТИСАНИМ БАЗАМА ПОДАТАКА**POWER DATA MODELS DEFINED IN GRAPH ORIENTED DATABASES**

Марко Гајић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – Овим радом обухваћено је истраживање рада граф оријентисаних база података са моделима података електроенергетских система. Улазни подаци су представљени у једној од RDF форми серијализације, док је као граф оријентисана база коришћена MarkLogic база података. Подаци се у бази података смештају у формату тројки (субјекат-предикат-објекат). Кроз развијену клијентску апликацију тестиране су CRUD (Create, Read, Update, Delete) операције ослањајући се на SPARQL језик. Такође, испитана је могућност закључивања коју MarkLogic база података поседује. Закључивање представља могућност MarkLogic базе података да на основу већ постојећих информација представљених као тројке генерише нове тројке. На крају је извршено тестирање перформанси базе података у зависности од величине улазних података и могућности закључивања над њима, као и тестирање конективности дефинисаних елемената електроенергетске мреже у случајевима са и без могућности закључивања.

Кључне речи – Модели података електроенергетског система, граф оријентисане базе података, закључивање, MarkLogic, SPARQL.

Abstract – Graph oriented databases and power data models are covered by this paper. Input data are presented as one of the forms of RDF serialization and graph-oriented database is MarkLogic database. Data in database are stored in triples format (subject-predicate-object). CRUD (Create, Read, Update, Delete) operations were tested with developed client application relying on SPARQL query language. Also, MarkLogic Inferencing option was tested. Inferencing presents capability of MarkLogic database to create new triples based on the existing information. An analysis of database performances based on input data size and inference capability is performed. Network connectivity test based of inference capability is performed, too.

Key words – Power data models, graph oriented databases, inferencing, MarkLogic, SPARQL.

1. УВОД

У данашњим информационим системима постоји велики број различитих типова података. У зависности од типа података, разликује се више врста база података.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био доц. др Милан Гаврић.

Поред традиционалних релационих база података, кључ-вредност организованих и документ оријентисаних база података, постоје и граф оријентисане базе података. Предност граф оријентисаних база података представља могућност прилагођавања променама података које се дешавају у реалном времену. Такође, пружају боље перформансе приликом претраге повезаних података, као и објектно оријентисан приступ приликом конструисања упита.

Данашњи информациони системи у електроенергетици презентују податке уз помоћ стандардизованог модела података који је заједнички за све системе. Такав модел података назива се CIM (Common Information Model) и његова појава представља решење проблема где је потребан један модел података који ће бити независан од апликације која га користи. Да би се са подаци преносили између апликација, CIM приступ предлаже њихово представљање у стандардизованој форми за размену података познатијој као RDF (Resource Description Framework). RDF податке представља у облику тројки субјекат-предикат-објекат.

Такви подаци се затим серијализују у један од формата RDF серијализације ради даљег рада са њима. Када подаци који су сада представљени у стандардизованом формату стигну у систем, могуће их је складиштити у одређену базу података. Осим самог складиштења, граф оријентисане базе података имају и друге могућности рада са подацима, као што је њихово конструисање, добављање и преглед доступних података. Језик који се користи при тим операцијама назива се SPARQL (Sparql Protocol and RDF Query Language) и представља стандардизован упитни језик над граф оријентисаним базама података.

2. ОПИС КОРИШЋЕНИХ ТЕХНОЛОГИЈА

Ово поглавље садржи кратак осврт на технологије, методе и принципе који су коришћени за имплементацију пројектног решења. Описане су само технологије и принципи који су највише утицали на имплементацију решења.

2.1 CIM

CIM (Common Information Model) представља апстрактни информациони модел који се користи за моделовање података електричне мреже и различите опреме која се користи у мрежи, као и моделовању података од значаја за управљање електроенергетском мрежом. Коришћењем овог модела података, компаније смањују трошкови

ве интеграција, што им омогућава да више ресурса усмере ка повећању функционалности и оптимизацији електричних система [1]. Креиран је да реши проблем који су предузећа имала са EMS (*Energy Management System*), зато што је сваки EMS користио своје технологије за креирање апликација. Као решење за тај проблем, произвођачи софтвера су прихватили CIM као основни информациони модел. CIM се ослања на XML и UML (*Unified Modeling Language*), док су две најкоришћеније серије стандарда IEC 61970 (модел мреже) и IEC 69168 (дистрибуција).

2.2 RDF

RDF (*Resource Description Framework*) представља стандард за размену података који је развијен од стране W3C [2]. У RDF се искази формирају у облику тројке (субјекат-предикат-објекат). Субјекат представља ресурс дефинисан преко URI-а, објекат представља особину субјекта и може бити дефинисан као ресурс или литерал, док предикат представља везу између субјекта и објекта. Да би се RDF подаци размењивали између различитих рачунарских система, потребно их је текстуално презентovati, односно серијализовати. Неке од стандардних форми серијализације RDF података су: Turtle, N-Triples, N-Quads, RDF/XML (CIMXML).

2.3 SPARQL

SPARQL (*Sparql Protocol and RDF Query Language*) представља стандардизован упитни језик за рад са RDF подацима, и као такав је погодан за рад са граф оријентисаним базама података. Користи се за додавање и манипулисање информацијама смештеним у бази података у RDF формату. Резултат упита може бити скуп тројки или граф [3]. Корисник са својом базом података комуницира уз помоћ SPARQL упита. Ако се погледа структура упита, она се може поделити на два дела, главу и тело упита. Глава разликује четири различите врсте које представљају израз за конструисање одговора упита. То су Select, Describe, Ask и Construct. Резултат упита се добија провером слагања тела упита са графом над којим се врши упит, односно повезивањем променљивих из тела упита са резултатима из графа. Тело упита може бити једноставна тројка субјекат-предикат-објекат ако су кориснику потребни најосновнији подаци, али се такође те једноставне тројке могу комбиновати са опционим деловима у случају да кориснику требају неке комплексније информације.

Поред упита, постоје и SPARQL Update опције. Синтаксно се незнатно разликују од упита и деле се у две групе: операције над графом (*create, drop, copy, move, add*) и операције над подацима (*insert data, delete data, clear, delete where, insert where, delete..insert where*).

2.4 Граф оријентисане базе података

Са повећањем броја података који захтевају чување као и операције претраге и управљања над њима, јавила се потреба за новим решењима у области складиштења података, односно потреба за базама података које поседују особине скалабилности како би могле да одговоре на све већи број информација које је потребно чувати. Такође, неопходно је да

поседују подршку за један од упитних језика који ће омогућити брзу претрагу базе података и који ће клијенту приказати тражене податке у прегледној форми. Форма субјекат-предикат-објекат одговара нечему што је у математици познато као граф, по дефиницији математичка репрезентација објекта (чворови) и веза између њих (гране). Граф оријентисане базе података се организационо ослањају на горепоменућу математичку дефиницију графа, где чвор представља објекат, док грана представља везу између два објекта које спаја. Сваки чвор је дефинисан јединственим идентификатором (UID), скупом грана који долазе и одлазе од њега као и чворовима који су са њим повезани [4].

2.4.1 MarkLogic база података

MarkLogic база података је NoSQL мултимодел база, што значи да подаци нису организовани на један начин, односно према једном моделу, већ подржава различите моделе организације података, међу којима је и граф оријентисани приступ. Не ослања се на шему, што значи да се подаци могу учитати у базу онакви какви су у оригиналном формату, без потребе за претходним дефинисањем шеме података. База пружа флексибилну интеграцију и чување података, као и њихов каснији преглед у виду документа, графа или релационе организације, што је предност мултимодел приступа.

2.5 Закључивање

Закључивање (*Inference*) се врши на основу доказа и образложења. Самим тим, закључивање представља технику извођења нових знања на основу постојећих. Са стране *MarkLogic* семантике, закључивање представља аутоматско откривање нових чињеница на основу података и правила понашања тих података. Када се та теорија пренесе на RDF тројке, добије се техника генерисања нових тројки на основу већ постојећих тројки [5]. Постоје два начина на који се нове тројке могу генерисати преко технике закључивања. Прва се назива *forward-chaining* закључивање и представља технику генерисања нових тројки тако да се оне додају у базу података. Друга техника се назива *backward-chaining* закључивање и она креира нове тројке за време извршавања самог упита. *MarkLogic* се ослања на аутоматско *backward-chaining* закључивање [5]. Аутоматско закључивање се примењује помоћу скупа правила и онтологија које корисник дефинише.

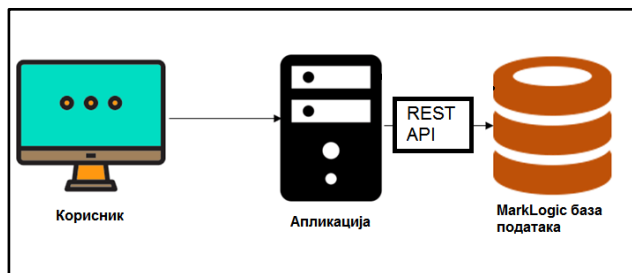
3. ДИЗАЈН РЕШЕЊА

Овим делом рада описана је архитектура решења, као и функционалности апликације.

3.1 Архитектура решења

Први корак представља подешавање радног окружења, односно инсталацију и покретање *MarkLogic* сервера. Затим је потребно направити базу података у оквиру *MarkLogic* сервера. Након тога је потребно направити корисничку апликацију која ће представљати везу између корисника и базе података. Корисник треба да буде у могућности да преко апликације комуницира са базом, врши операције читања и писања и креирања и извршавања SPARQL

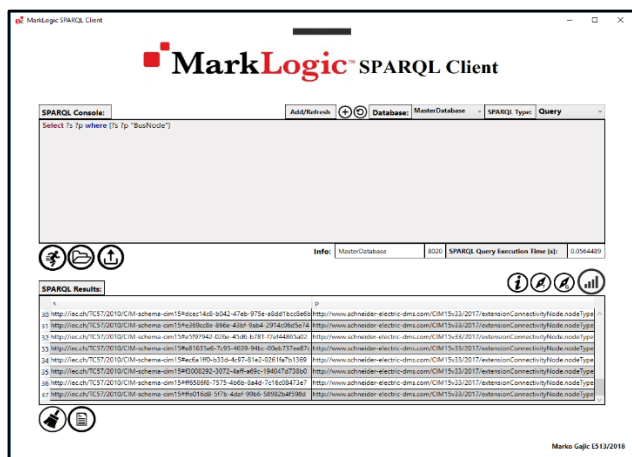
упита. Да би веза између базе и апликације била могућа, искоришћен је REST API интерфејс. Он омогућава комуникацију са *MarkLogic* базом података независно од програмског језика. Слика 1 приказује изглед архитектуре решења.



Слика 1. Архитектура решења

3.2 Функционалности апликације

Клијентска апликација је дизајнирана тако да омогући клијенту што лакшу интеракцију са базом података која се налази на *MarkLogic* серверу. Даље су набројане неке од најважнијих функционалности које поседује клијентска апликација. Корисник је у могућности да дефинише SPARQL Query и SPARQL Update исказе, ручно или учитавањем из фајла. Такође, корисник може у саму базу података учитати податке у једном од облика серијализације RDF формата. Могуће је направити нову базу података као и REST API инстанцу, као и изабрати жељену базу са којом корисник жели да ради. Такође, постоји посебан део који кориснику приказује заузеће процесорске јединице и RAM меморије од стране *MarkLogic* сервера за време рада апликације. Кориснику се резултати интеракције са базом података приказују у формату табеле. Корисника интересује и време извршавања SPARQL исказа, тако да се и та информација приказује у оквиру клијентске апликације. Слика 2 приказује изглед апликације у раду са *MarkLogic* сервером.



Слика 2. Почетна страна апликације

4. МЕРЕЊЕ ПЕРФОРМАНСИ РЕШЕЊА

Виртуелној машини на којој се налази *MarkLogic* сервер обезбеђен је *Intel(R) Core(TM) i3-3220 CPU @ 3.30 GHz* процесор и *13.7 GB* RAM меморије. Као улазни подаци за тестирање су коришћена пет CIMXML фајлова који представљају пет фидера из једне северноамеричке дистрибуције.

Тестирањем се приказују перформансе апликације у зависности да ли је код базе података укључена могућност закључивања и како та могућност утиче на пораст броја тројки, време извршавања самог упита заузеће процесора.

Такође, тестирана је и конективност учитаних елемената. Као улазни подаци код тестирања конективности коришћен је модел мреже у CIMXML формату обезбеђен од ENTSO-E (*European Network of Transmission System Operators for Electricity*).

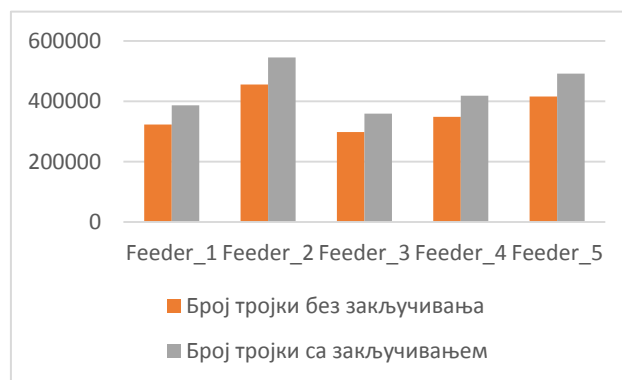
4.1 Перформансе са и без закључивања

Особине које су тестиране су: број тројки, време извршавања SPARQL упита који добавља све тројке и заузеће процесора током добављања тројки. Циљ је био упоредити резултате који су добијени без укљученог закључивања са резултатима који су добијени са укљученим закључивањем.

Закључивање се остварује уз помоћ онтолошког фајла насталог на основу *CIM16v33legacy-rdfs* и *rdfs.rules* скупа правила који се учитава у базу података. Табела 1 приказује број тројки док Слика 3 приказује број тројки добијених са и без укљученог закључивања.

Табела 1. Број тројки са и без закључивања

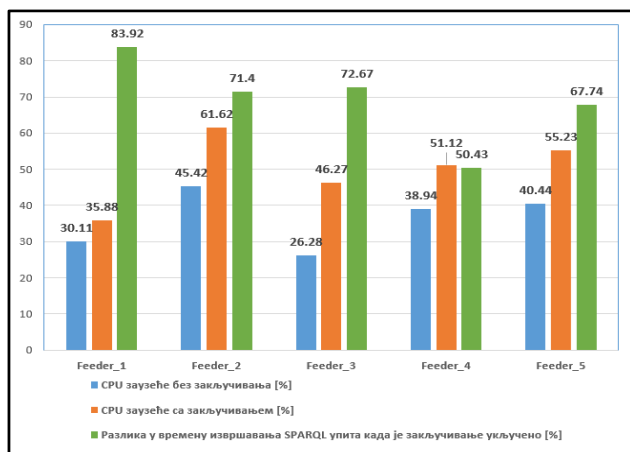
Фајл	Број тројки без закључивања	Број тројки са закључивањем
Feeder_1	323473	387093
Feeder_2	455931	545704
Feeder_3	298256	359292
Feeder_4	348550	418563
Feeder_5	416041	491921



Слика 3. Поређење и тренд броја тројки са и без могућности закључивања

Приметан је пораст у броју тројки, што је и главна особина закључивања. Слика 4 приказује заузеће процесора и процентуалну разлику у времену извршавања SPARQL упита са и без укљученог закључивања. Време које је потребно да би се извршио SPARQL упит када је закључивање укључено процентуално је дуже од времена извршавања SPARQL упита када је закључивање искључено.

Такође, вредност заузећа процесора је већа када је закључивање укључено. Такви резултати су очекивани јер упит враћа више тројки када је закључивање укључено.



Слика 4. CPU заузеће и SPARQL време извршавања

4.2 Тестирање конективности

Конективност је прво тестирана у случају када је сва опрема повезана. Затим су укањањем одређених *Breaker* елемената и њихових терминала добијена четири одвојена острва повезане опреме смештена у 4 различита именована графа. Након тога је над сваким од њих тестирана конективност. Циљ је био добити времена извршавања SPARQL упита конективности за случај када је опрема подељена на више острва и упоредити та времена са временом добијеним у случају када је сва опрема повезана и упоредити резултате за случај када је закључивање укључено и када закључивање није укључено. Табела 2 приказује број проводне опреме и број тројки док табела 3 приказује времена извршавања SPARQL упита.

Табела 2. Број проводне опреме и тројки

Повезаност	Број проводне опреме	Број тројки без закључивања	Број тројки са закључивањем
Све повезано	1547	40013	64898
Острво 1	959	26851	45848
Острво 2	125	1951	3056
Острво 3	58	907	1457
Острво 4	392	6119	9482

Табела 3. SPARQL време извршавања

Повезаност	Разлика у времену извршавања када је закључивање укључено [%]
Све повезано	204
Острво 1	466
Острво 2	433
Острво 3	1700
Острво 4	1100

Очекивано, број тројки који се добије у случају укључене могућности закључивања је већи у односу на број тројки који се добије када закључивање није укључено. Такође, времена извршавања SPARQL упита за конективност су процентуално већа у случају укљученог закључивања. Дељењем повезане опреме на више острва и претрагом појединачних острва добијају се знатно мања времена извршавања. Ово је практично у случају великих мрежа када корисник зна у ком делу мреже се неки елемент налази.

5. ЗАКЉУЧАК

Радом је покривена анализа *MarkLogic* базе података као граф оријентисане базе за рад са моделима електроенергетских система. Предност *MarkLogic* базе података представља могућност извршавања CRUD операција над подацима, као и над графовима који садрже те податке преко SPARQL исказа. Предност представља и могућност закључивања, односно генерисања нових знања на основу постојећих. *MarkLogic* кориснику нуди опцију да сам дефинише правила на основу којих ће база података да извршава процес закључивања. Тестирањем су добијени резултати перформанси које база података остварује у случајевима када је закључивање укључено и када закључивање није укључено.

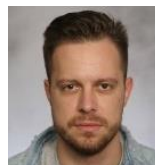
Као закључак се намеће то да ако корисник жели могућност закључивања, јавиће се пораст времена извршавања, као и веће заузеће процесора.

Клијентском апликацијом кориснику је омогућено писање и учитавање SPARQL упита из фајла, учитавање података у одабрану базу, креирање базе као и приказ заузећа рачунарских ресурса у току рада апликације од стране *MarkLogic* сервера. Предлог за даље усавршавање представља графичку визуелизацију података над којим корисник ради у циљу повећања свести корисника о моделу електроенергетског система над којим се извршавају неопходне операције.

6. ЛИТЕРАТУРА

- [1] L.King, The Common Information Model for Distribution:An Introduction to the CIM for Integrating Distribution Applications and Systems, EPRI, 2008.
- [2] Ontotext, „What is RDF?“, <https://ontotext.com/knowledgehub/fundamentals/what-is-rdf/>. [Последњи приступ 28 Октобар 2019].
- [3] P. Leslie F. Sikos, Mastering Structured Data on The Semantic Web.
- [4] WhatIs.com, „Graph database“ <https://whatIs.techtarget.com/definition/graph-database/>. [Последњи приступ 28 Октобар 2019].
- [5] MarkLogic Corporation, MarkLogic Server:Semantic Graph Developer's Guide, 2019.

7. ПОДАЦИ О КАНДИДАТУ



Марко Гајић рођен је 31. марта 1995. год. у Шапцу. Завршио је основну школу “Јеврем Обреновић” у Шапцу 2010. год. Завршио је средњу школу “Техничка школа Шабац” у Шапцу, 2014. године. Исте године је уписао основне академске студије на Факултету техничких наука, смер примењено софтверско инжењерство. У септембру 2018. год. дипломирао је и уписао мастер студије. Испунио је све обавезе и положио све испите у оквиру мастер студија у року са просечном оценом 8.67.

ПРЕПОЗНАВАЊЕ ГРАФИЧКИХ СИМБОЛА НА ШЕМАТСКИМ ДИЈАГРАМИМА IDENTIFICATION OF GRAPHIC SYMBOLS ON SCHEMATIC DIAGRAMS

Вукашин Иличић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – Овим радом обухваћено је истраживање различитих приступа и алгоритама који су везани за препознавање симбола на шематским дијаграмима. Описан је поступак на којем се заснива имплементација апликације која омогућава кориснику да проналази жељене симболе на изабраној шеми. Приликом имплементације коначног решења коришћен је C# програмски језик. Током развоја апликације урађене су детаљне анализе и тестирања која испитују њене перформансе.

Кључне речи: Вештачка интелигенција, машинско учење, детекција објеката, шематски дијаграм

Abstract – This paper covers the study of different approaches and algorithms related to the identification of symbols in schematic diagrams. The paper describes in detail the procedure on which the implementation of an application is based, which allows the user to detect desired symbols on an input scheme. The C# programming language was used in the final solution. During the development of the program, detailed analyzes and tests were conducted to test the application performances.

Keywords: Artificial Intelligence, Machine learning, Object-detection, Schematic diagram

1. УВОД

Анализа слика као област је позната од раније и тренутно је врло популарна пошто је нашла широку примену у индустрији и различитим пословима који су постали саставни део наших живота. У овој области постоје различити приступи и технике које се примењују. Препознавање симбола на сликама може да се реализује на различите начине помоћу технике машинског учења или помоћу принципа директне претраге. За реализацију решења коришћена су оба приступа, где је установљено да решење које се заснива на употреби неуронских мрежа не даје очекиване резултате. Главни разлог за недовољно добре резултате је ограничена количина слика за обуку неуронске мреже. Код реализације решења помоћу неуронских мрежа коришћена је Google-ова платформа *TensorFlow*, намењена за машинско учење. Она пружа разноврсне могућности код обучавања неуронских мрежа. Након што је установљено да због недовољне количине података резултати остварени помоћу машинског учења нису довољно добри у обзир је узето решење које се заснива на принципу „клизача“ који пролази кроз улазну шему

и тражи елементе одређеног типа. Крајње решење реализовано је помоћу принципа директне претраге која користи структуру пирамиде слика. Структура пирамиде слика омогућава брже извршавање процеса тражења одређеног типа јер користи посебну структуру која смањује опсег претраге. Имплементирана апликација је осмишљена да омогући кориснику да учита жељену слику на којој је представљена шема са различитим елементима које је потребно препознати. После одабира дела слике на којој се налази жељени тип елемента извршава се претрага означеног типа. Као резултат претраге добија се учитана слика са претраженим елементима који припадају истом типу као и означени елемент којег је одабрао корисник.

2. ОПИС АНАЛИЗИРАНИХ ТЕХНОЛОГИЈА

Поред краћег описа имплицитно или експлицитно су наведене неке од предности и мана које долазе са њиховим коришћењем. Описане су само технологије и приступи који су највише утицали на имплементацију решења.

2.1. COMPUTER VISION

Научници и истраживачи који су се бавили темом *computer vision*-а, развили су различите технике засноване на математици за враћање тродимензионалног облика и изгледа објеката на сликама [1]. Захваљујући њима, данас постоје поуздане технике које омогућавају да се на основу одређеног броја фотографија које су дате у две димензије, створи 3D модел који верно репрезентује објекат са фотографија. Поред наведене функционалности могуће је пратити и кретање особе на видео снимку па чак пребројати и именовати људе са фотографије помоћу технике препознавања лица.

2.2 AI, МАШИНСКО УЧЕЊЕ И DEEP LEARNING

Вештачка интелигенција је као област у рачунарству настала педесетих година прошлог века. Вештачка интелигенција се може посматрати и као напор да се аутоматизују интелектуални задаци које људи могу уобичајено да обављају. Као таква, област *AI (artificial intelligence)* обухвата машинско учење и *deep learning* [3]. Машинско учење представља програмирање рачунара тако да могу да уче из података. Сличну дефиницију о машинском учењу написао је и Arthur Samuel (1959), „Машинско учење је област која рачунарима даје могућност учења без експлицитног програмирања“ [4].

Deep learning представља специфично поље машинског учења, где се усвајање новог знања

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био доц. др Милан Гаврић.

заснива на принципу неуронских мрежа које се састоје из више слојева. Модерне методе **deep learning**-а укључују десетине или чак стотине узастопних слојева који се обучавају помоћу одабраних података за тренинг. Да би се слике могле обрађивати помоћу рачунара, могуће је применити концепте машинског учења и неуронских мрежа. Процес препознавања објеката на сликама помоћу машинског учења је сложен и захтева доста времена и примера за учење.

2.3. ПИРАМИДА СЛИКА

За имплементацију апликације која може да препознаје одабране симболе, коришћена је посебна структура у којој се улазна слика за претрагу скалира на мање слике. На тај начин се добија структура у облику пирамиде помоћу које је могуће ефикасније урадити претрагу жељених симбола. Овај принцип представља оптимизацију за такозвану директну претрагу шаблона на сликама.

Пирамида се састоји од одређеног броја нивоа. Први односно основни ниво пирамиде представља оригиналну слику, док сваки следећи ниво настаје умањивањем слике са претходног нивоа [2]. Процес корелације, односно поређења шаблона који се претражује и саме слике на којој треба пронаћи шаблон започиње од врха пирамиде. Због тога што највиши ниво садржи најмању слику, омогућено је брже извршавање претраге. Кроз процес поређења утврђују се такозване хипотезе о томе где би тражени шаблон могао да се налази.

Хипотезе које су добијене на вишим слојевима, проверавају се на нижим. Овакав приступ омогућава да се мали део око грубе процене са вишег нивоа испитује у оквиру нижих нивоа. Овај алгоритам пружа смањени опсег претраге и помоћу њега могу брзо и лако да се разврстају делови слике што доводи до оптимизације која резултује краћим временом извршавања процеса претраге.

3. ДИЗАЈН РЕШЕЊА

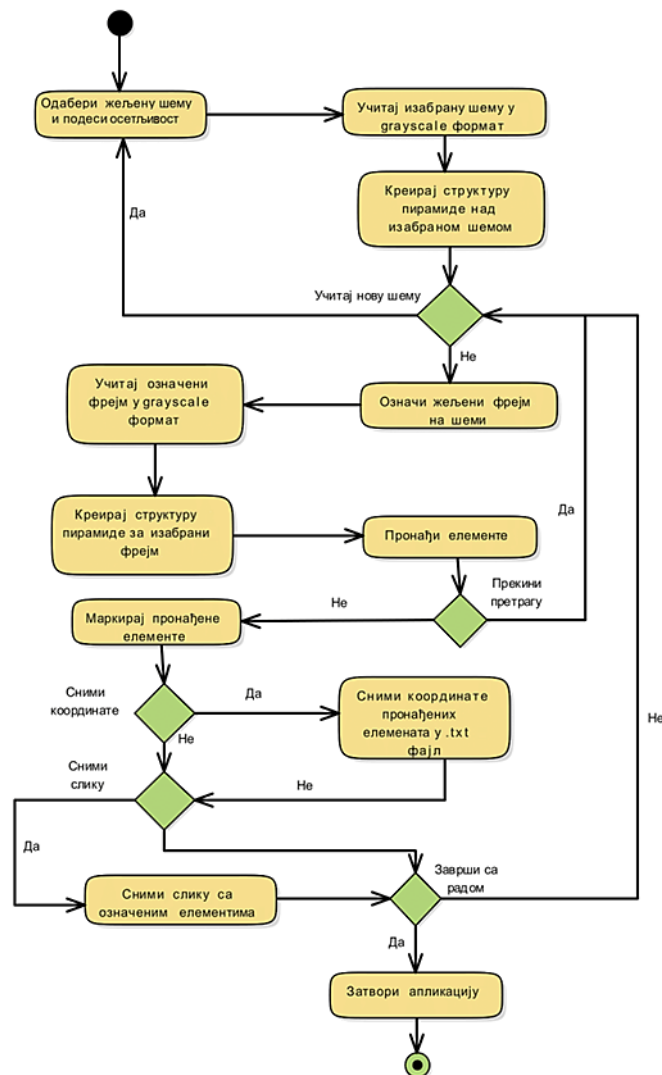
У оквиру овог поглавља укратко је представљена реализација решења за задати проблем. Треба напоменути да се током имплементације решења мењао приступ решавања због одређених ограничења која су наведена у наставку. Улаз у програм представљају слике, које се конвертују у **grayscale** формат. Апликација је направљена тако да ради са **grayscale** сликама (Слика 1), међутим корисник може да бира који приказ слике жели приликом рада у програму (**Grayscale** или **RGB**). Интерно у меморији учитана слика је представљена помоћу матрице, чија поља имају вредности у опсегу од 0 до 255.

Свака вредност у матрици представља **grayscale** вредност одређеног пиксела на изабраној шеми. Када се слика за претрагу учита, кориснику је омогућено да одабере елемент који жели да се претражи. Означени фрејм се такође претвара у **grayscale** матрицу вредности интерно у меморији.

После учитавања жељене слике и **frame**-а следећи корак представља креирање пирамиде. Пирамида се

посебно креира за учитану слику и за селектовану област која се претражује. Креирање пирамиде се ради тако што се вредности четири суседна пиксела сабирају и деле са четири. На тај начин се добија њихова средња вредност.

Са овим кораком добијају се слојеви који су по висини и по ширини за дупло мањи од свог родитељског слоја од којег су настали. На тај начин новонастали слој је по броју пиксела четири пута мањи у односу на



Слика 1. Алгоритам рада апликације

родитељски слој. Број слојева у структури пирамиде се подешава ручно у зависности од резолуције улазне слике. Оваква структура је погодна због перформанси претраге која представља следећи корак у целокупном процесу. Процес претраге започиње од врха пирамиде учитане слике, односно од најмањег слоја. У оквиру најмањег слоја у којем се ради претрага, користи се фрејм са врха пирамиде фрејмова.

Фрејм који одговара врху пирамиде учитане шеме пролази кроз комплетну шему на врху структуре са кораком померања од једног пиксела. Процес поређења фрејма са одговарајућим делом шеме се ради тако што се пореде **grayscale** вредности боја код пиксела. Сваки пиксел на фрејму има одговарајући

пиксел на посматраном делу шеме са којим се пореди. На основу разлике вредности пиксела са шеме и фрејма, прорачунава се укупна разлика помоћу које се одлучује да ли је на шеми пронађен задати тип елемента. Помоћу параметра осетљивости поставља се горња граница која одређује колико процентуално посматрани део шеме може да се разликује од задатог елемента. Уколико је проценат различитости поређења мањи од вредности дефинисаног параметра осетљивости, онда се анализира део шеме сматра припадником траженог типа. Када се претрага горњег слоја уради добијају се области, односно координате потенцијалних елемената који одговарају изабраном фрејму. У нижим слојевима уместо да се претрага ради од почетка до краја, она се врши само у областима који су добијени у вишим слојевима.

На тај начин се смањује област претраге на области које су од значаја за тражени резултат. Последњи корак представља постављање маркера који уоквирују пронађене елементе на слици. Резултат претходног корака представља листу координата пронађених елемената. На сваку од пронађених координата поставља се по један маркер (правоугаоник). Као резултат добија се слика са уоквиреним елементима који су претраживани. Поред слике са означеним елементима у склопу апликације корисник има опцију да експортује добијене координате у излазни текстуални фајл. У наставку ће бити објашњена имплементација компоненти које су од значаја за рад апликације.

4. ИМПЛЕМЕНТАЦИЈА

Апликација је имплементирана по MVVM (model-view-viewmodel) патерну. Што се тиче view компоненти, апликација се састоји од основног *MainView*-а на којем се налазе сви потребни *widget*-и преко којих корисник остварује жељене радње. У склопу *MainView*-а користе се различите класе дефинисане у моделу које се користе ради постизања жељених функционалности. Једна од главних класа из модела представља *DataWorker* класа која садржи податке о учитаној слици и означеном фрејму који су представљени у облику матрице са вредностима од 0 до 255. Претходно споменуте две матрице презентују *grayscale* формат слика са којима се ради у програму. Кориснику је омогућено да учитава растерске слике које могу бити у .jpg или .png формату.

Што се тиче метода у *DataWorker* класи, она садржи основне методе за учитавање података за изабрану слику у интерну структуру која презентује *grayscale* формат те слике. Након што се дефинишу сви потребни подаци везани за означени фрејм и слику на којој се врши претрага, могуће је започети процес претраге. Процес претраге се започиње тако што се прво креира структура пирамиде помоћу класе *PyramidImageHelper*. Споменута класа садржи две листе у којима су смештене матрице које презентују слику шеме и слику фрејма у различитим нивоима. Пирамида слика се креира тако што се на основу оригиналне слике шеме и слике фрејма креирају наредни скалирани нивои структуре пирамиде. Након креирања структуре пирамиде, започиње се процес

тражења елемената који су означени помоћу изабраног фрејма. Процес претраживања креће од врха пирамиде где се добијени резултати користе у нижем слоју ради постизања истог циља код слике са већом резолуцијом. Процес претраге се понавља док се не заврши претрага на слици са највишом резолуцијом која се налази на најнижем слоју пирамиде. Сваком слоју пирамиде са скалираном сликом која се претражује одговара један слој пирамиде у којој су скалиране слике изабраног фрејма.

На овај начин се омогућава да се одбаце делови за претрагу који нису од значаја, односно да буду претражени само делови на слици који потенцијално садрже тражене елементе. Помоћу оваквог приступа претраге омогућава се мањи број итерација, што доводи до тога да целокупан процес претраге буде временски краћи. Слика 2 приказује пикселизацију елемената по слојевима. Пример А са слике приказује пикселизацију дела шеме, док је у примеру Б приказана пикселизација означеног елемента.

У оба примера структура пирамиде садржи четири нивоа. Са слике (Слика 2) се може уочити губитак својстава код приказаних елемената са порастом броја нивоа.



Слика 2. Шематски приказ пикселизације (А) Пикселизације дела шеме, (Б) Пикселизација означеног елемента

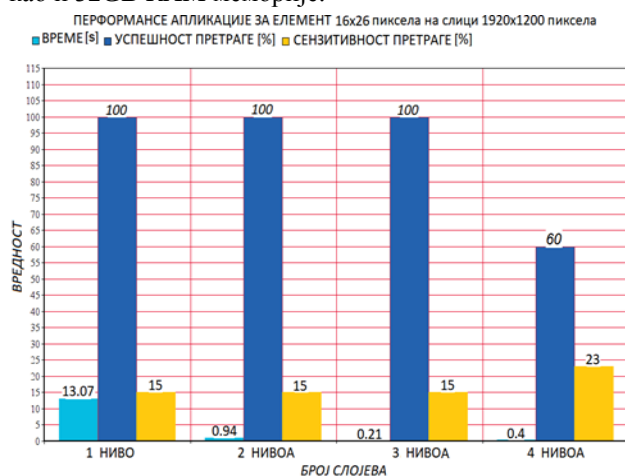
Губитак својстава код елемената утиче на успешност претраге, што је приказано и на графику перформанси (Слика 3). Са графика се може видети да код претраге елемената помоћу структуре пирамиде са четири нивоа успешност је смањена на 60%. Употреба структуре пирамиде смањује време извршавања али пикселизација која се јавља услед скалирања може да утиче на успешност претраживања. Због тога је битно проценити број нивоа за претрагу на основу величине и

својстава претраживаних елемената на изабраној шеми.

5. МЕРЕЊЕ ПЕРФОРМАНСИ РЕШЕЊА

У овом поглављу размотрене су перформансе рада апликације. Брзина извршавања претраге зависи од више фактора (Слика 3). Основни фактор који утиче на брзину претраживања јесте величина улазне слике. Апликација је тестирана за слике које су у **full HD** резолуцији. У зависности од резолуције потребно је подесити одређени број нивоа који ће се користити за претрагу. Код резолуције 1920 x 1080 пиксела, тестирање је показало да се најбоље перформансе могу постићи када се користе пирамиде са два или три нивоа. Осим резолуције улазне слике на брзину претраге утиче и величина претраживаног елемента. Уколико је резолуција претраживаног елемента већа онда је и време претраживања дуже.

Осим времена претраге, на графику су приказане и информације о успешности претраге као и осетљивости. Поред споменутих фактора и параметара на претрагу утиче и осетљивост коју подешава корисник. Подешавање осетљивости претраге је корисно, јер поспешује саму претрагу и на тај начин се остварују бољи резултати рада. На пример претраживани елементи истог типа имају своју специфичност која их разликује од других елемената тог типа. Због међусобне разноликости између елемената истог типа, кориснику је обезбеђено да сам може да подеси колика ће осетљивост да се користи током претраге. У зависности од геометрије и разноликости претраживаних елемената биће употребљена већа или мања осетљивост. Да би се искористили номинални параметри за претрагу, врло често је потребно урадити претрагу неколико пута да би могло да се закључи које вредности параметара су најбоље за разматрани случај претраге. Перформансе имплементираних апликација су тестиране на рачунару који има Intel-ов i3 процесор (3.2 GHz) са два језгра као и 32GB RAM меморије.



Слика 3. Измерене перформансе

6. ЗАКЉУЧАК

Стручњаци који се баве машинским учењем и неуронским мрежама сматрају да ће вештачка интелигенција

бити све присутнија у животу човека. Да би се ова визија остварила на прави начин врло је важно разумети начин на који поједине ствари функционишу, јер у супротном последице могу бити трајне, поготово када је реч о **AI**. Овај рад обухвата само део целокупне области и као такав може да послужи као основа и инспирација за даља проучавања. Одређене специфичности које карактеришу решавани проблем, довеле су до тога да се уз употребу **C#** програмског језика имплементира алгоритам који ће омогућити потребну функционалност без да се користи машинско учење и **deep learning**. Главни разлог због којег се користи другачији приступ је мали број слика које могу да се употребе за обуку неуронске мреже. Као прихватљиво решење за овај проблем показала се и оптимизована директна претрага која користи структуру пирамиде слика која је објашњена у претходном поглављу.

Решење које је имплементирано по споменutom принципу садржи и одређене недостатке. Приликом претраге користе се слике у **grayscale** формату, што уводи одређено ограничење. Боје које се јасно разликују у **RGB** формату, врло су сличне приликом конверзије у одговарајућу нијансу сиве боје. Овај недостатак долази до изражаја када су елементи истог типа представљени различитим бојама у **RGB** формату, које приликом конверзије у **grayscale** формат дају сличне нијансе сиве боје. Овај недостатак може да се реши помоћу различитих апликација као што је **Photoshop** који може помоћу **color-range** опције да одбаци делове слике обојене у одређену боју. Тиме би се омогућило да се претрага жељеног елемента ради независно од боје у RGB формату. Тема није у целости затворена и може да послужи као платформа за нова истраживања и радове.

7. ЛИТЕРАТУРА

- [1] Szeliski Richard, "Computer Vision", Springer, 2010.
- [2] Marco Alexander Treiber, "An Introduction to Object Recognition", 2010.
- [3] F. Chollet, "Deep Learning with Python", Manning Publications, 2017.
- [4] A. Géron, "TensorFlow" 2nd Edition, O'Reilly Media, 2019.

Кратка биографија:



Вукашин Илић рођен је у Новом Саду 1995. године. Завршио је основну школу „Светозар Марковић Тоза“ 2010. године. Завршио је гимназију „Светозар Марковић“, општи смер, у Новом Саду 2014. године. Исте године је уписао основне академске студије на Факултету техничких наука, смер примењено софтверско инжењерство, у Новом Саду. На основним академским студијама дипломирао је 2018. године са просечном оценом 9,13. Мастер студије је уписао 2018. године и испунио је све обавезе и положио све испите предвиђене студијским програмом са просечном оценом 9,83.

**UTICAJ NAČINA UZEMLJENJA DISTRIBUTIVNE MREŽE NA PRORAČUN
LOKACIJE KVARA****THE IMPACT OF DISTRIBUTION NETWORK GROUNDING ON THE FAULT
LOCATION**Ivan Stanojević, Savo Đukić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu dati su osnovni pojmovi vezani za upravljanje kvarom. Takođe, teorijski je opisan algoritam za lociranje kvara. Testiran je uticaj načina uzemljenja distributivne mreže na funkciju lokacije kvara na test mreži i izneti su zaključci o dobijenim rezultatima.

Ključne riječi: Lokacija kvara, Uzemljenje, Distributivna mreža.

Abstract – This paper gives basic concepts about the fault management. Also, algorithm for fault location are described theoretically. Influence of the distribution network grounding on the fault location is tested using the simple network and obtained conclusions are provided.

Keywords: Fault location, Grounding, Distribution network.

1. UVOD

Prekid napajanja potrošača je nepoželjno stanje, te je potrebno preduzeti odgovarajuće mere da se ono što pre eliminiše. Zato je prvo neophodno ustanoviti gde se nalazi mesto sa kvarom, zatim ga izolovati od ostalog dela distributivne mreže, da bi se potom, obnovilo napajanje kod potrošača kod kojih je došlo do prekida napajanja, a kod kojih je to moguće izvesti. Skup prethodno navedenih akcija predstavlja upravljanje kvarom [1].

Tradicionalni pristup podrazumeva izlazak ekipe na teren, vizuelni pregled deonice sa kvarom i ručnu manipulaciju rasklopnom opremom. Potrebne informacije o lokaciji kvara se dobijaju telefonskim putem od potrošača [2].

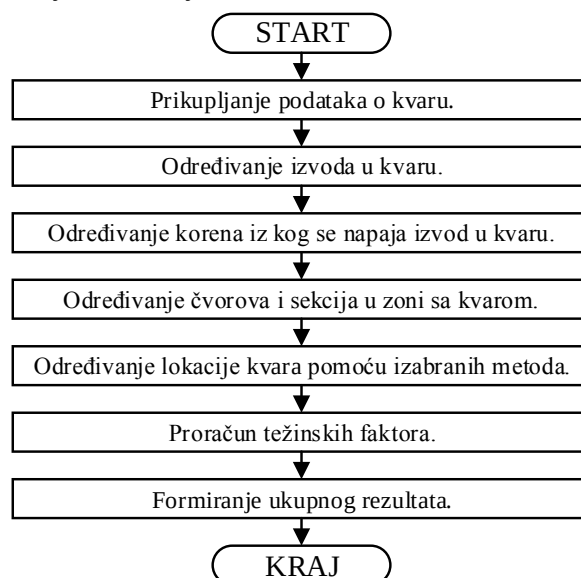
Velika mana tradicionalnog pristupa problemu eliminacije kvara jeste vreme trajanja prekida napajanja potrošača, koje može biti skraćeno efikasnijim upravljanjem kvarom. Višestruko efikasnije upravljanje kvarom se postiže korišćenjem različitih indikatora i rasklopne opreme, koja je povezana sa odgovarajućim softverom u dispečerskom centru, preko nadzorno upravljačkih sistema. Princip rada funkcije za lociranje kvara opisan je u sledećem poglavlju. U trećem poglavlju, izvršeno je ispitivanje uticaja načina uzemljenja distributivne mreže na funkciju za lociranje kvara. Dobijenu rezultati su predstavljeni i prokomentarisani. Na kraju rada dati su odgovarajući zaključci i korišćena literatura.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Savo Đukić, docent.

2. FUNKCIJA ZA LOKACIJU KVARA

Rezultat proračuna funkcije za lokaciju kvara jeste verovatnoća na kojoj sekciji se dogodio kvar, pri čemu to može biti jedna, ali i više sekcija na kojoj se potencijalno nalazi mesto kvara. Uprošćeni algoritam funkcije za lokaciju kvara dat je na slici 2.1.



Slika 2.1. Uprošćeni algoritam funkcije za lokaciju kvara

Prikupljanje ulaznih podataka odvija se u prvom koraku algoritama, a neki od podataka koji su od interesa za ovu funkciju su tip kratkog spoja, broj obuhvaćenih faza, izabrane metode za lokaciju kvara, itd.

Pri pojavi kvara u mreži određujuće odgovarajuća relejna zaštita, koja će poslati komandu za otvaranje prekidača na početku izvoda. Iz tog razloga prvo je potrebno odrediti na kom izvodu se dogodio kvar.

Sledeće što je potrebno uraditi jeste određivanje korena distributivne mreže sa kog se napaja izvod sa kvarom. Pod korenom distributivne mreže podrazumeva se jedna od sabirnica u transformatorskoj stanici visoki na srednji napon.

Pošto se proračun lokacije kvara ne vrši nad celom mrežom već samo nad izvodom u kvaru, potrebno je odrediti sve čvorove i sekcije koje se nalaze u zoni sa kvarom.

Postoje različite metode za lociranje kvara i svaka od njih pruža rezultate koji su manje ili više tačni. Zbog toga se koristi više metoda za lociranje kvara. Najpoznatije metode za lociranje kvara su: Fault Detector, Current method, Impedance method i Failure Rate method.

Proračun lokacije kvara pomoću Fault Detector metode odvija se tako što se vrši pretraživanje detektora kroz koje je protekla struja kratkog spoja. U slučaju kada je npr. kroz detektor na početku sekcije protekla struja kvara, a kroz detektor na samom kraju sekcije nije protekla struja kvara, tada je kvar sigurno na toj sekciji.

Current metoda vrši simulaciju kvarova na početku i kraju svake sekcije iz zone kvara. Pomoću izračunatih vrednosti struje kratkog spoja na mestu ugradnje merne opreme i izmerene struje (I_{mereno}) vrši se određivanje broja sekcija koje pripadaju tzv. crvenoj i žutoj zoni. Smisao uvođenja crvene i žute zone jeste uvažavanje grešaka koje se javljaju pri merenju struje.

Kada se vrednost izmerene struje kvara nalazi između izračunatih struja kvara na početku i kraju neke sekcije, to znači da se kvar upravo nalazi na toj sekciji i da ona pripada crvenoj zoni. Treba naglasiti da broj sekcija koje se nalaze u crvenoj može biti veći od jedan.

Za određivanje broja sekcija u žutoj zoni koristi se isti uslov kao za određivanje broja sekcija u crvenoj zoni, ali se umesto I_{mereno} koristi $0.98 I_{mereno}$ odnosno $1.02 I_{mereno}$. Takođe, broj sekcija u žutoj zoni može biti veći od jedan.

Impedance metoda koristi ekvivalentnu merenu impedansu kvara (direktnog redosleda) od mesta merne jedinice do mesta kratkog spoja. Impedansa kvara dobija se kao količnik odgovarajućeg napona i struje izvoda u kvaru. Nakon toga vrši se simuliranje kvarova na početku i kraju svake sekcije pri čemu se proračunavaju impedanse direktnog redosleda.

Kada se dogodi da se vrednost modula merene impedanse (Z_{mereno}) nalazi u opsegu izračunatih vrednosti modula impedansi sa početka i kraja neke sekcije to znači da je upravo na toj sekciji kvar i da se ona nalazi u crvenoj zoni. Treba naglasiti da broj sekcija koje se nalaze u crvenoj zoni može biti veći od jedan. Za određivanje broja sekcija u žutoj zoni se koristi isti uslov kao i za određivanje broja sekcija u crvenoj zoni, ali se umesto Z_{mereno} koristi $0.9 Z_{mereno}$ odnosno $1.1 Z_{mereno}$. Takođe, broj sekcija u žutoj zoni može biti veći od jedan.

Podužni intezitet otkaza trajnih kvarova za sekciju je pokazatelj pouzdanosti, koji predstavlja broj kvarova u toku jedne godine po jedinici dužine. Množenjem tog pokazatelja i dužine sekcija dobija se verovatnoća kvara za svaku sekciju tog izvoda.

U prethodnom koraku izvršen je proračun lokacije kvara za izabrane, koje su se ujedno mogle izvršiti. Rezultati proračuna pojedinačne metode dobijeni su uz pretpostavku da je njihova pouzdanost 100%, pri čemu će ovi rezultati biti korigovani. Pouzdanost metode predstavlja verovatnoću da će ta metoda za svaki tip i mesto kvara u mreži dati kvalitetne rezultate.

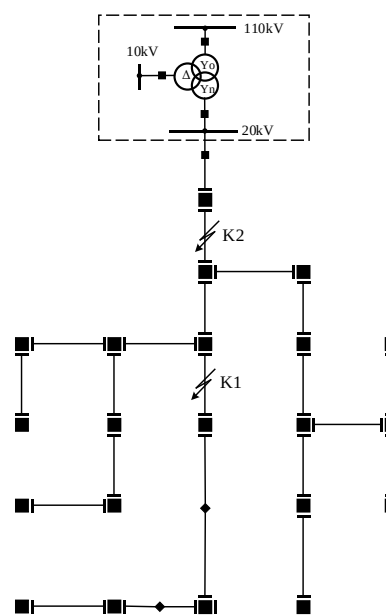
Vrednost pouzdanosti svake ponaosob metode može biti, u opsegu od 0% do 100%, pri čemu 0% znači da je metoda potpuno nepouzdana, a 100% znači da je metoda apsolutno pouzdana. Za strujnu i impedantnu metodu definišu se dve vrednosti pouzdanosti metode (po jedna za crvenu i žutu zonu). Korekcija rezultata proračuna lokacije kvara vrši se pomoću težinskih faktora. Težinski

faktor pojedinačne metode računa se kao količnik pouzdanosti te metode i zbira pouzdanosti svih metoda.

U poslednjem koraku se ukupan rezultat za neku sekciju dobija kao proizvod rezultata proračuna pojedinačnih metoda i izračunatih težinskih faktora.

3. UTICAJ NAČINA UZEMLJENJA MREŽE NA LOKACIJU KVARA

U ovom poglavlju predstavljeni su rezultati ispitivanja uticaja načina uzemljenja napojnog transformatora na rezultate proračuna lokacije kvara strujnom i impedantnom metodom. Ispitivanje je izvršeno za dve različite lokacije kvara (obeležene sa K1 i K2), na test mreži prikazanoj na slici 3.1.



Slika 3.1. – Izgled test mreže

U napojnoj transformatorskoj stanici nalazi se tronomotajni transformator sprege $YoYnD$. Na sekundarnoj strani napojnog transformatora se nalazi sva potrebna merna oprema sa zaštitnim relejima. Vrednosti otpora otpornika za uzemljenje, koje su korišćene pri ispitivanju, iznose 0Ω , 20Ω , 40Ω i 70Ω .

Iako se sredjenaponska mreža ne uzemljuje direktno, u ovom radu je to urađeno radi poređenja dobijenih rezultata. Što se vrste kvarova tiče obrađeni su kratki spojevi sa zemljom, odnosno jednopolni i dvopolni kratak spoj sa zemljom.

3.1. Dobijeni rezultati proračuna lokacije kvara pomoću strujne metode

U ovom poglavlju, u tabelama 3.1.1, 3.1.2, 3.1.3 i 3.1.4 prikazani su rezultati proračuna lokacije kvara pomoću strujne metode za različite vrednosti otpora otpornika za uzemljenje. U tabelama 3.1.2 i 3.1.4 I_{min} i I_{max} predstavljaju najmanju i najveću vrednost izračunate struje kvara koja protiče kroz sekundar napojnog transformatora dok je sa I označena merena vrednost struje kvara. Tabele 3.1.2 i 3.1.4 se razlikuju samo po

vrednostima merene struje kvara što je posledica različitih lokacija mesta kvara (K1 i K2).

Tabela 3.1.1. – Broj detektovanih sekcija po zonama za lokaciju kvara K1 pomoću strujne metode

R [Ω]	Broj sekcija u crvenoj zoni		Broj sekcija u žutoj zoni	
	1PKS	2PKSZ	1PKS	2PKSZ
0	3	3	3	3
20	3	3	15	7
40	3	3	20	7
70	3	3	20	7

Tabela 3.1.2. – Promena struja kvara za lokaciju kvara K1 pomoću strujne metode

R [Ω]		0	20	40	70
1PKS	I _{min} [A]	3272	576	304	178
	I [A]	4690	605	311	180
	I _{max} [A]	7593	631	318	182
2PKSZ	I _{min} [A]	4326	4054	3991	3962
	I [A]	5314	4648	4575	4543
	I _{max} [A]	7530	5283	5203	5169

Tabela 3.1.3. – Broj detektovanih sekcija po zonama za lokaciju kvara K2 pomoću strujne metode

R [Ω]	Broj sekcija u crvenoj zoni		Broj sekcija u žutoj zoni	
	1PKS	2PKSZ	1PKS	2PKSZ
0	1	1	0	3
20	1	1	12	3
40	1	1	19	3
70	1	1	22	3

Tabela 3.1.4. – Promena struja kvara za lokaciju kvara K2 pomoću strujne metode

R [Ω]		0	20	40	70
1PKS	I _{min} [A]	3272	576	304	178
	I [A]	5393	613	313	181
	I _{max} [A]	7593	631	318	182
2PKSZ	I _{min} [A]	4326	4054	3991	3962
	I [A]	5798	4851	4776	4743
	I _{max} [A]	7530	5283	5203	5169

Ono što je možda na prvi pogled čudno, jeste činjenica da je broj detektovanih sekcija u crvenoj zoni (za obe lokacije kvara) ostao isti bez obzira na promenu R. Iako povećanje R izaziva smanjenje struje kvara (naročito kod 1PKS), njena raspodela unutar test mreže je ostala ista, pa je zbog toga logično zašto je broj sekcija u crvenoj zoni ostao isti.

Broj detektovanih sekcija u crvenoj zoni pri lokaciji kvara K2 je manji zato što je lokacija kvara bliža izvoru napajanja izvoda.

Kada je razmatrana mreža direktno uzemljena, vrednosti I_{min} i I_{max} se dosta razlikuju i zato je broj sekcija u žutoj zoni relativno mali. Pri otporu uzemljenja većim od 20Ω razlika između I_{min} i I_{max}, pri 1PKS gotovo da ne postoji, a to znači da je vrednost struje simuliranog 1PKS u bilo kom čvoru mreže praktično ista.

Upravo iz tog razloga se broj sekcija u žutoj zoni povećao. Ovaj efekat postoji i kod 2PKSZ, ali je mnogo je manje izražen, jer su struje 2PKSZ mnogo veće od struja pri 1PKS. Zato je broj sekcija u žutoj zoni pri 2PKSZ manji nego pri 1PKS.

Iz dobijenih rezultata za lokaciju K2 i dalje se uočava uticaj otpornosti uzemljenja na broj sekcija u žutoj zoni, kao i na vrednosti struje 1PKS na lokaciji K2. Broj detektovanih sekcija u crvenoj zoni je sada manji u odnosu na kvar na lokaciji K1.

To je zbog toga što je kvar K2 bliži izvoru napajanja, u odnosu na kvar K1. Analogno važi i za 2PKSZ na lokaciji K2.

3.2. Dobijeni rezultati proračuna lokacije kvara pomoću impedantne metode

U ovom poglavlju, u tabelama 3.2.1, 3.2.2, 3.2.3 i 3.2.4 prikazani su rezultati proračuna lokacije kvara pomoću impedantne metode za različite vrednosti otpora otpornika za uzemljenje. U tabelama 3.2.2 i 3.2.4 Z_{min} i Z_{max} predstavljaju najmanju i najveću vrednost izračunate impedanse kvara (direktnog redosleda) dok je sa Z označena merena vrednost struje kvara.

Tabele 3.2.2 i 3.2.4 se razlikuju samo po vrednostima merene impedanse kvara što je posledica različitih lokacija mesta kvara (K1 i K2). Vrednost Z_{min} dobija se pri kratkom spoju na početku izvoda i njena vrednost je nula, jer je vrednost napona za tu lokaciju nula.

Za merenje impedanse direktnog redosleda pri 1PKS faze L1 koristi se relacija (3.2.1), a za merenje impedanse direktnog redosleda pri 2PKSZ faza L1 i L2 formula (3.2.2):

$$Z = \frac{U_{L1}}{I_{L1} + 3kI_{L1}}, \quad (3.2.1)$$

$$Z = \frac{U_{L1} - U_{L2}}{I_{L1} - I_{L2}}, \quad (3.2.2)$$

gde su:

U_{L1}, U_{L2} – mereni naponi faza L1 i L2 za vreme kvara,
I_{L1}, I_{L2} – merene struje faza L1 i L2 za vreme kvara,
Z – merena impedansa kvara.

Tabela 3.2.1. – Broj detektovanih sekcija po zonama za lokaciju kvara K1 pomoću impedantne metode

R [Ω]	Broj sekcija u crvenoj zoni		Broj sekcija u žutoj zoni	
	1PKS	2PKSZ	1PKS	2PKSZ
0	3	3	5	6
20	3	3	5	6
40	3	3	3	6
70	3	3	3	6

Tabela 3.2.2. Promena impedanse kvara za lokaciju kvara K1 pomoću impedantne metode

R [Ω]		0	20	40	70
1PKS	Z _{min} [Ω]	0	0	0	0
	Z [Ω]	229.2	232.3	235.3	240
	Z _{max} [Ω]	420.5	433.5	446.5	466.1
2PKSZ	Z _{min} [Ω]	0	0	0	0
	Z [Ω]	229.2	229.2	229.2	229.2
	Z _{max} [Ω]	446.7	446.7	446.7	446.7

Tabela 3.2.3. Broj detektovanih sekcija po zonama za lokaciju kvara K2 impedantne metode

R [Ω]	Broj sekcija u crvenoj zoni		Broj sekcija u žutoj zoni	
	1PKS	2PKSZ	1PKS	2PKSZ
0	1	1	3	3
20	1	1	3	3
40	1	1	3	3
70	1	1	3	3

Tabela 3.2.4. Promena impedanse kvara za lokaciju kvara K2 pomoću impedantne metode

R [Ω]		0	20	40	70
1PKS	Z _{min} [Ω]	0	0	0	0
	Z [Ω]	154.1	154.6	155.2	156
	Z _{max} [Ω]	420.5	433.5	446.5	466.1
2PKSZ	Z _{min} [Ω]	0	0	0	0
	Z [Ω]	154.1	154.1	154.1	154.1
	Z _{max} [Ω]	446.7	446.7	446.7	446.7

Iz tabela 3.2.2 i 3.2.4 se vidi da se pri 1PKS i povećanju impedanse za uzemljenje javlja blago povećanje merene impedanse. Zanimljivo je što se broj sekcija u žutoj zoni smanjuje. U imeniocu relacije (3.2.1) figuriše član $3kI_{L1}$, koji upravo predstavlja merenu nultu komponentu struje 1PKS. Koeficijent k je takozvani koeficijent zemljospoja, koji treba da uzme u obzir koji procenat nulte komponente struje, sa mesta kvara, protiče kroz merne instrumente na sekundaru napojnog transformatora. U razmatranom slučaju, ta "kompenzacija" nulte komponente pomoću koeficijenta zemljospoja nije baš idealna i zato se pojavila promena merene impedanse direktnog redosleda. Broj sekcija u žutoj zoni detektovanih pomoću impedantne metode je višestruko manji u odnosu na broj sekcija u žutoj zoni detektovanih pomoću strujne metode za isto mesto kvara.

Pošto je mesto kvara K2 na samom početku izvoda, detekcija sekcija po zonama je daleko efikasnija u odnosu na kvar K1. To se vidi po broju detektovanih sekcija i u crvenoj i u žutoj zoni koji je sada manji. Može se još primetiti da mnogo veći uticaj na dobijene rezultate ima mesto kvara u odnosu na način uzemljenja mreže.

4. ZAKLJUČAK

Iz priloženih rezultata može se zaključiti da postojanje otpornika za uzemljenje izaziva smanjenje vrednosti struja kvara, ali može proizvesti i značajno povećanje broja detektovanih sekcija (pri korišćenju strujne metode) u okviru žute zone. Ovaj efekat je najizraženiji pri jednopolnim kratkim spojevima koji su relativno udaljeni od izvora napajanja distributivne mreže.

Takođe, zaključeno je da vrednost otpornika za uzemljenje ne utiče na broj detektovanih sekcija u okviru crvene zone. Dominantni efekti koji utiču na broj sekcija u crvenoj zoni su mesto kvara i tačnost merenja struje.

Iako impedantna metoda koristi ekvivalentnu impedansu kvara direktnog redosleda, postoji uticaj uzemljenja mreže na vrednost merene impedanse i broj detektovanih sekcija u okviru žute zone. Za razliku od strujne, kod impedantne metode ovaj efekat postoji samo pri 1PKS koji su udaljeni od izvora napajanja. Štaviše, pri povećanju vrednosti otpornosti otpornika za uzemljenje smanjuje se broj detektovanih sekcija u okviru žute zone. Na broj detektovanih sekcija koje pripadaju crvenoj zoni najviše utiče mesto kvara, dok vrednost otpornika za uzemljenje nema nikakav uticaj.

Poređenjem rezultata dobijenih pomoću impedantne i strujne metode, potvrđena je veća preciznost impedantne metode u svim razmatranim slučajevima.

5. LITERATURA

- [1] Dragan Popović, Duško Bekut, Valentina Dabić: *Specijalizovani DMS algoritmi*, DMS Group, Novi Sad, 2011.
- [2] M. M. Saha, J. Izykowski, E. Rosolowski: *Fault Location on Power Networks*, Springer, London, 2010.

Kratka biografija:

Ivan Stanojević rođen je u Valjevu 1993. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranio je 2018. godine.

Savo Đukić rođen je u Novom Sadu 1983. godine. Doktorsku disertaciju odbranio je 2014. godine na Fakultetu tehničkih nauka iz oblasti Elektroenergetski sistemi.

ZAŠTITA VODOVA U DISTRIBUTIVNIM I PRENOSNIM MREŽAMA**PROTECTION OF POWER LINES IN DISTRIBUTION AND TRANSMISSION NETWORKS**

Aleksa Ristić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su izvršene analize za različite načine zaštite vodova u distributivnim i prenosnim mrežama. Dati su podaci koji su neophodni za adekvatno podešavanje i funkcionisanje relejne zaštite. Objašnjene su prekostrujna i distantna zaštita. Izvršen je proračun podešavanja prekostrujne i distantne zaštite.

Ključne reči: Releji, prekostrujna zaštita, distantna zaštita

Abstract – Analysis of different ways to protect lines in distribution and transmission networks is given in this paper. Also, this paper provides data necessary for proper setting and operation of power system protection. Overcurrent and distance protection are explained. Calculation of overcurrent and distance protection has been performed.

Keywords: Relays, overcurrent protection, distance protection

1. UVOD

Da bi elektroenergetski sistem (EES) mogao da funkcioniše potrebno je obezbediti zaštitu svakog njegovog elementa, a takođe je potrebno obezbediti i zaštitu pojedinih celina u okviru EES od kvarova. U distributivnim preduzećima evidentan je porast potrebe za što sigurnijom isporukom električne energije potrošačima, što zahteva mrežu sa malim brojem kvarova i ispada opreme. Veliku ulogu u sprovođenju ovih zahteva ima relejna zaštita.

Relejna zaštita podrazumeva skup zaštitnih uređaja i postupaka za štićenje elemenata i EES. Koristi se sa ciljem da se uklone trajni kvarovi i minimizira šteta koja bi u suprotnom nastala pri delovanju prekidača.

Tema ovog rada odnosi se na zaštitu vodova u distributivnim i prenosnim mrežama.

U drugoj glavi objašnjena je uloga koju relejna zaštita ima u elektroenergetici i nakon toga je u nastavku opisana podela releja. Treća glava odnosi se na zaštitu mreža pomoću relejne zaštite. Navedeni su tipovi releja koji se koriste za zaštitu vodova u distributivnim i prenosnim mrežama.

Četvrta glava je posvećena faznoj prekostrujnoj zaštiti, dok je peta glava posvećena pofaznoj distantnoj zaštiti.

NAPOMENA:

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

U šestoj glavi urađen je proračun podešavanja distantne i prekostrujne zaštite. U sedmoj glavi dati su osnovni zaključci rada, dok je u osmoj glavi naveden spisak korišćene literature.

2. ULOGA RELEJNE ZAŠTITE

Uloga relejne zaštite jeste da reaguje prilikom nastanka kvara, da spreči nastanak sledećeg kvara i da smanji štetu koju bi taj kvar izazvao.

Kako bi EES mogao da funkcioniše potrebno je obezbediti zaštitu svih njegovih elemenata i pojedinih celina od kvarova. Zaštita celokupnog EES od kvarova ostvaruje se relejnom zaštitom.

EES se štiti od sledećih nenormalnih stanja:

- stanja sa kvarom (kvarovima),
- opasnih pogonskih stanja.

Prethodno pomenuta stanja predstavljaju stanja u kojima parametri (napon, struja, frekvencija, itd) izlaze iz okvira predviđenih projektima i konstrukcijom EES.

Pred relejnu zaštitu se postavlja nekoliko opštih zahteva:

- selektivnost,
- brzina reagovanja,
- osetljivost,
- pouzdanost,
- sigurnost,
- ekonomičnost.

3. RELEJI I OSETLJIVOST RELEJNE ZAŠTITE

Za zaštitu vodova koriste se prekostrujna, diferencijalna, distantna zaštita i osigurač.

Prekostrujna zaštita – predstavlja jedan od najstarijih oblika zaštite. Deli se na brzu prekostrujnu zaštitu koja deluje bez vremenskog kašnjenja, prekostrujnu zaštitu sa definisanim vremenskim kašnjenjem i zaštitu sa inverznom karakteristikom reagovanja.

Diferencijalna zaštita – veoma brza i selektivna zaštita. Može biti podužna i poprečna.

Distantna zaštita vodova – pošto je primena podužne diferencijalne zaštite moguća samo na kratkim vodovima, selektivnu zaštitu vodova u upetljenim mrežama moguće je postići primenom distantnih releja, koji kao kriterijum koriste udaljenost kvara od releja. Da bi distantna zaštita bila selektivna, distantni releji moraju biti usmereni, moraju imati vreme reagovanja u zavisnosti od otpora (impedanse, reaktanse) koji mere i štićenog voda. To vreme reagovanja mora zadovoljiti princip: veća impedansa – veće vreme reagovanja.

Automatsko Ponovno Uključenje (APU). Ukoliko se kvar dogodi na vodu, trenutni prekostrujni relej će otvoriti prekidač i APU će nakon pauze ponovo zatvoriti prekidač i vratiti napajanje vodu. Nakon prvog ciklusa trenutni prekostrujni relej je blokiran. Ako je kvar i dalje prisutan, strujno zavisni prekostrujni relej će delovati, i otvaranjem prekidača ukloniti kvar. Ako je strujna zavisnost takva da je vreme delovanja nešto duže, može doći do topljenja osigurača pre delovanja releja. Na taj način osigurač će otkloniti kvar [1].

Za zaštitu vodova u distributivnim i prenosnim mrežama koriste se sledeći tipovi releja (za svaki tip releja u zagradi su navedene oznake po ANSI standardu):

- trenutni prekostrujni releji (50),
- prekostrujni releji sa vremenskom zadržskom (51),
- trenutni uzemljeni prekostrujni releji (50N),
- vremenski uzemljeni prekostrujni releji (51N),
- usmereni fazni prekostrujni releji (67),
- usmereni uzemljeni prekostrujni releji (67N),
- fazni distantni releji (21),
- uzemljeni distantni releji (21N).

Trenutni prekostrujni releji su dizajnirani da reaguju bez vremenskog odlaganja, kada struja premaši struju podešenja releja. Vreme delovanja može da varira između 0.016 s i 0.1 s. Podešenje ovih releja je nekoliko puta veće od nominalne struje. Njihova uloga je da za jako kratko vreme otklone kvarove čije su struje izuzetno visoke. Odlaganje otklanjanja takvih kvarova bi značajno ugrozilo bezbednost i velika šteta nad opremom bi bila načinjena [2, 3].

4. FAZNA PREKOSTRUJNA ZAŠTITA

Ova vrsta zaštite predstavlja najjednostavniju i najčešće korišćenu zaštitu za vodove naponskog nivoa od 10 kV do 35 kV. Stepem vremenske selektivnosti se mora postići između parova zaštite (svake dve na red postavljene zaštite) i on iznosi od 0.25 s do 0.5 s kod statičkih i elektromehaničkih releja, a od 0.6 s do 0.8 s kod primene primarnih okidača. Ovaj stepen vremenske selektivnosti se uvodi da bi se sprečila neselektivnost zaštite u slučaju bržeg delovanja zaštitom koja je bliža izvoru napajanja, dok se kasni sa delovanjem zaštitom koja je dalja od izvora napajanja.

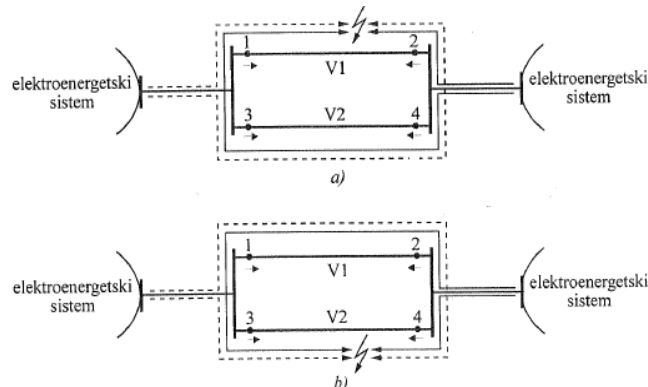
Prednosti prekostrujne zaštite su u njenoj jednostavnosti, sigurnosti i lakom ostvarenju principa rezervisanja. Osnovni nedostatak prekostrujne zaštite jeste taj da je za uklanjanje kvarova bližih izvoru vreme delovanja zaštite sve duže. Upravo zbog toga neophodno je proveriti da li je moguća primena ovog tipa zaštite [4].

5. POFAZNA DISTANTNA ZAŠTITA

Izlaganje u ovom delu počinje sa još jednim nedostatkom prekostrujne zaštite, koji je prikazan na slici 1.

Na slici 1 zapažena su dva paralelno dvostrano napajana voda. Punim i isprekidanim linijama su označeni mogući tokovi struja kod kvarova na jednom od vodova. Zaštite su na vodu V1 obeležene slovima 1 i 2, na vodu V2 slovima 3 i 4, dok su strelicama označeni smerovi delovanja zaštitama.

Pri kvaru na vodu V1 (slika 1 pod a) treba da se deluje zaštitama 1 i 2. U slučaju kvara zaštite 1 treba da se deluje zaštitom 4, iz čega sledi da je vreme delovanja zaštitom 4 veće od vremena delovanja zaštitom 1.



Slika 1. Primer kvara na paralelnim dvostrano napajanim vodovima: a) kvar na vodu V1 i b) kvar na vodu V2

Pri kvaru na vodu V2 (slika 1 pod b) treba da se deluje zaštitama 3 i 4. U slučaju kvara zaštite 4 treba da deluje zaštita 1, iz čega sledi da je vreme delovanja zaštitom 1 veće od vremena delovanja zaštitom 4.

Ova dva prethodna izraza su kontradiktorna, što znači da u ovakvim slučajevima nije moguće primeniti prekostrujnu zaštitu. Prema tome, prekostrujnu zaštitu nije moguće primeniti u složenim upetljenim mrežama. Iz tog razloga se u složenim mrežama primenjuje distantna zaštita.

5.1. Prvi stepen distantne zaštite (21P1)

Impedantno podešavanje prvog stepena distantne zaštite mora biti manje od impedanse voda koji se štiti, jer bi u suprotnom moglo da dođe do delovanja te zaštite i pri kvarovima na susednim vodovima.

Prvi stepen distantne zaštite podešava se tako da se štiti od 80 % do 90 % dužine voda (ova vrednost je viša za duže vodove). U našoj zemlji štiti se 85 % dužine voda.

Prvi stepen distantne zaštite ima kvadrilateralnu karakteristiku. Ukoliko su sledeći kriterijumi zadovoljeni smatra se da zaštita zadovoljava normalan nivo kriterijuma osetljivosti. U suprotnom, zaštita se neće smatrati prihvatljivom na nivou efikasnosti zaštite:

$$Z_{reach} \leq 0.8 \cdot Z_1, \quad (1)$$

$$Z_{RB} = 0.8 \cdot Z_{max.load}, \quad (2)$$

$$Z_{LB} \leq Z_{reach}, \quad (3)$$

gde je:

Z_{reach} – doseg prvog zaštićenog područja,

Z_1 – impedansa voda na kojem je zaštita,

Z_{RB} i Z_{LB} – leva i desna impedansa blindera, respektivno. Blinderi predstavljaju ograničenja na karakteristici zaštite po X osi.

5.2. Drugi stepen distantne zaštite (21P2)

Drugim stepenom distantne zaštite treba da se „sigurno” obuhvate kratki spojevi na suprotnom kraju voda, a da se istovremeno ne premaši ni jedan prvi stepen zaštite na susednim vodovima.

Drugi stepen distantne zaštite ima MHO karakteristiku. Zaštita je postavljena tako da se detektuju kratki spojevi na fideru, uključujući glavni vod i bočne priključke. Dakle, vrednost podešavanja se nalazi između neke minimalne i maksimalne vrednosti.

Kriterijum za normalni nivo efikasnosti jeste:

$$1,25 \cdot Z_{apparent} \leq Z_{reach} \leq 1,5 \cdot Z_{apparent}, \quad (4)$$

gde je:

$Z_{apparent}$ – impedansa vodova gde su DG povezani. Druga zona je postavljena za najnepovoljniji slučaj impedanse voda, uzimajući u obzir otpor kvara i uticaj kada su DG u službi.

5.3. Treći stepen distantne zaštite (21P3)

Treći stepen zaštite treba da predstavlja rezervnu zaštitu za susedne vodove i da se njim obuhvati zaštita sabirnica na kraju tih vodova.

Treća zona distantne zaštite ima MHO karakteristiku. Obično obezbeđuje dodatnu kontrolu za vremenske prekostrujne releje. Podesi se da obezbedi dodatnu pokrivenost zaštitom za kvarove koji imaju veliku impedansu.

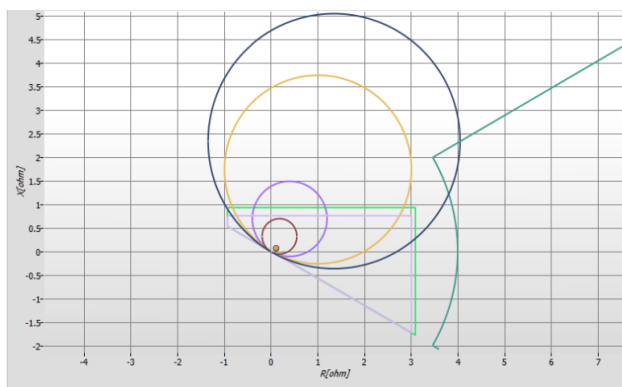
Kriterijum za normalni nivo efikasnosti jeste:

$$1,5 \cdot Z_{apparent} \leq Z_{reach} \leq 2 \cdot Z_{apparent}. \quad (5)$$

Provera distantne zaštite u softveru vrši se izborom tipa distantnog releja koji će se koristiti i odabirom kvara, tj. tipa kratkog spoja. Postoje sledeći tipovi kratkog spoja:

- LG – jednopolni kratak spoj (zemljospoj),
- LL – dvopolni kratak spoj,
- LLG – dvopolni kratak spoj sa zemljom,
- LLL – tropolni kratak spoj.

Na slici 2 prikazane su karakteristike distantnih releja.



Slika 2. Karakteristike distantnih releja

Za distantne releje koristi se R/C ravan za prezentaciju. Karakteristike za svaku zonu zaštite distantnih releja će biti predstavljene kao funkcije tačaka odgovarajuće merene impedanse posmatranih distantnih releja.

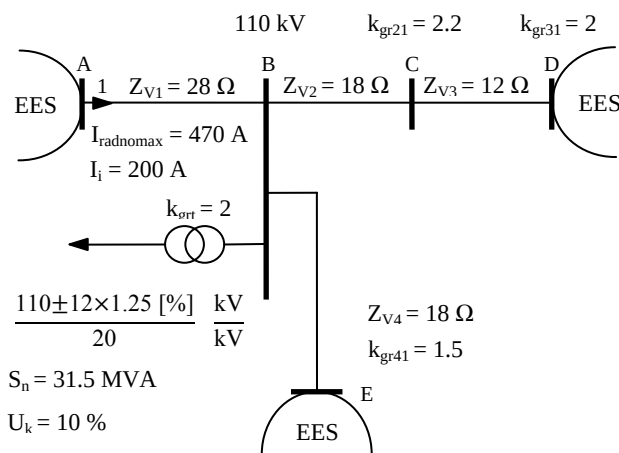
Korisnik ima mogućnost da u legendi izabere za koje od navedenih distantnih releja da se prikaže karakteristika. Svaki relej je na grafiku prikazan različitom bojom (recimo, ljubičasta predstavlja relej 21, braon 21N, itd).

6. PRORAČUN PODEŠENJA DISTANTNE I PREKOSTRUJNE ZAŠTITE

U programskom jeziku C++ urađena su dva zadatka, od kojih se prvi odnosi na podešenje distantne zaštite, dok se drugi odnosi na podešenje prekostrujne zaštite.

6.1. Proračun podešenja distantne zaštite

Zadatak: Podesiti distantnu zaštitu na mestu označenom brojem 1, na osnovu podataka sa slike 3. Stepenn selektivnosti iznosi $d_t = 0,4$ s. Svi potrebni podaci za izbor podešenja dati su na slici.



Slika 3. Deo EES od interesa

Da bi se obezbedio uslov selektivnosti rada zaštite potrebno je da bude ispunjen i sledeći uslov:

$$Z_{IV1} \leq Z_{IIV1} \leq Z_{IIIV1} \leq 0,9 \cdot Z_{pob}, \quad (6)$$

gde su:

$Z_{IV1}, Z_{IIV1}, Z_{IIIV1}$ – impedantna podešenja prva tri stepena, Z_{pob} – pobudni stepen.

Određena su impedantna podešenja i vremena delovanja prva tri stepena zaštite.

Transformator koji je vezan za sabirnicu B je distributivni. Neophodno je proveriti da li se izabranim impedantnim podešenjem prva tri stepena distantne zaštite ispunjava zahtev da se ne detektuju kratki spojevi iza distributivnog transformatora, na 20 kV sabirnicama.

Dobijeno je da impedantno podešenje trećeg stepena zaštite ima nižu vrednost od impedanse koja bi se zaštitom V1 merila pri kratkom spoju na 20 kV sabirnicama, što znači da se kratki spojevi na nižepodskoj strani distributivnog transformatora ne mogu detektovati.

Podešenje prvog stepena distantne zaštite iznosi 23.8 Ω . Vreme delovanja prvog stepena distantne zaštite iznosi 0 s.

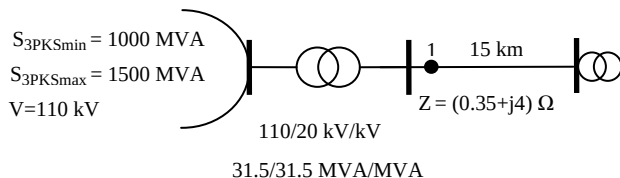
Podešenje drugog stepena distantne zaštite iznosi 43.3075 Ω . Vreme delovanja drugog stepena distantne zaštite iznosi 0.4 s.

Podešenje trećeg stepena distantne zaštite iznosi 69.183 Ω . Vreme delovanja trećeg stepena distantne zaštite iznosi 0.8 s.

Podešenje pobudnog stepena distantne zaštite iznosi 76.87Ω . Vreme delovanja pobudnog stepena distantne zaštite iznosi 3 s.

6.2. Proračun podešenja prekostrujne zaštite

Zadatak: Podesiti prekostrujnu zaštitu na mestu označenom brojem 1 i proveriti osetljivost zaštite. Svi potrebni podaci dati su na slici 5.



Slika 5. Razmatrana distributivna mreža

Struja podešenja računa se na osnovu sledećeg izraza:

$$I = \frac{k_{sigurnosti} \cdot k_{samopuštanja} \cdot I_{radno\ maksimalno}}{a}, \quad (7)$$

gde su:

$k_{sigurnosti}$ – koeficijent sigurnosti,

$k_{samopuštanja}$ – koeficijent samopuštanja,

$I_{radno\ maksimalno}$ – maksimalna radna struja,

a – koeficijent otpuštanja releja.

Za koeficijent sigurnosti usvaja se vrednost 1.2, dok se za vrednost koeficijenta samopuštanja usvaja proizvoljna vrednost od 1 do 6. Radi lakšeg proračuna uzeto je da je $k_{samopuštanja} = 1$, tj. podrazumeva se da je uticaj asinhronih motora u potrošnji zanemarljiv.

Za izbor strujnog podešenja zaštite potrebno je raspolagati vrednošću $I_{radno\ maksimalno}$. U ovom slučaju za tu vrednost se uzima 70 A.

Za koeficijent otpuštanja releja usvaja se vrednost 0.85.

Podešenje releja se proračunava na sledeći način:

$$i = \frac{k_{spoja} \cdot I}{p}, \quad (8)$$

gde su:

k_{spoja} – koeficijent spoja,

p – prenosni odnos strujnog transformatora.

Ako se primenjuje trofazna ili zaštita u dve faze, svaki od releja se napaja preko posebnog strujnog transformatora, što znači da je $k_{spoja} = 1$.

Neophodno je proveriti osetljivost zaštite, da bi dobijene vrednosti strujnih podešenja predstavljale i stvarne vrednosti podešenja zaštite. Za navedenu proveru potrebno je odrediti minimalne vrednosti struja kratkih spojeva na kraju zone šticećenja. Pošto su struje dvopolnog kratkog spoja manje od struje trolnog kratkog spoja, usvojiće se vrednost struje dvopolnog kratkog spoja. Može se proračunati vrednost struje trolnog kratkog spoja, u cilju lakšeg proračuna, i množenjem te vrednosti

sa $\frac{\sqrt{3}}{2}$ dobijaju se tražene vrednosti struja dvopolnog kratkog spoja. Kako bi dobili minimalnu vrednost struje dvopolnog kratkog spoja, neophodno je koristiti i minimalnu vrednost snage kratkog spoja.

Struja podešenja zaštite iznosi 98.8235 A.

Koeficijent osetljivosti iznosi 8.38418.

Dobijeno je da koeficijent osetljivosti ima veću vrednost od minimalne zahtevane vrednosti, koja za osnovnu zonu iznosi 1.5, što znači da je zaštita dobro podešena.

7. ZAKLJUČAK

U ovom radu obrađeni su tipovi releja koji se koriste u zaštiti vodova u distributivnim i prenosnim mrežama.

Objašnjen je značaj relejne zaštite i njena uloga u zaštiti elektroenergetskih mreža. Svrha relejne zaštite jeste da se uklone kvarovi i minimizira šteta koja bi u suprotnom nastala. Od nje se očekuje da bude selektivna, osetljiva, pouzdana, sigurna, ekonomična i da ima veliku brzinu reagovanja.

Detaljno je objašnjeno podešenje fazne prekostrujne, kao i podešenje pofazne distantne zaštite. Nakon toga su u programskom jeziku C++ urađena dva zadatka, koja se odnose na proračun podešenja ovih zaštita.

8. LITERATURA

- [1] M. D. Nimrihter, P. N. Đapić: *Proračuni u distributivnim električnim sistemima*, Fakultet Tehničkih Nauka, Novi Sad, 2008.
- [2] *D60 Line Distance Protection*, System Instruction Manual, General Electric Industrial Systems, Manual P/N: 160-0089-R1(GEK-113343) Ontario, Canada, 2009
- [3] J. G. Andrichak, G.E. Alexander, *Distance Relays Fundamentals*, GER-3966, General Electric Co, Malvern, PA
- [4] D. Bekut: *Relejna zaštita*, Fakultet Tehničkih Nauka, Novi Sad, 2009.

Kratka biografija:



Aleksa Ristić rođen je u Smederevu 1993. god. Diplomirao je 2018. god. na Fakultetu tehničkih nauka, na temu „Mogućnosti skladištenja energije“. Iste godine upisuje Master akademske studije na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi. Master rad odbranio je 2019. god.

GENERISANJE SLIKA SA VIŠE OZNAČENIH MNIST CIFARA UPOTREBOM GENERATIVNIH NEURONSKIH MREŽA

GENERATING IMAGES WITH MULTIPLE LABELED MNIST DIGITS USING GENERATIVE NEURAL NETWORKS

Luka Maletin, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Za treniranje generativne neuronske mreže za generisanje slike već postoje ustaljeni šabloni i principi, ali trenutna istraživanja bave se pre svega generisanjem slike sa jednim objektom. Generisanje slike sa više željenih objekata koji se nalaze na određenim pozicijama, i koji su u koheziji sa pozadinom, značajno bi proširilo primenu ovakvih modela. U ovom radu predstavljena je arhitektura modela za generisanje slike sa više označenih cifara na jednostavnoj pozadini. Kao ulaz modela prosleđuju se labele i granični okviri cifara koje je potrebno iscrtati. Istrenirani model uspešno generiše slike sa redosledima cifara koje je video tokom treniranja, a u slučaju novih redosleda ne generiše uvek očekivane cifre. Predloženi su koraci za poboljšanje i dalji razvoj arhitekture.

Ključne reči: GAN, MNIST, Generisanje slike, Generisanje objekta, Generativni modeli

Abstract – When it comes to generating images using generative neural networks, there are already many common methods and principles. However, the current research is focused primarily on generating an image with a single object. Generating images with multiple desired objects on specified locations, and that are in cohesion with the background, would significantly increase the use of such models. In this paper, we present an architecture for generating images with multiple labeled digits on a simple background. The model's inputs are the labels and bounding boxes of the digits that should be painted. The trained model successfully generates images for the orders of digits it has seen during training, while in the case of new orders, it doesn't always generate the specified digits. Steps for further improving the architecture are discussed.

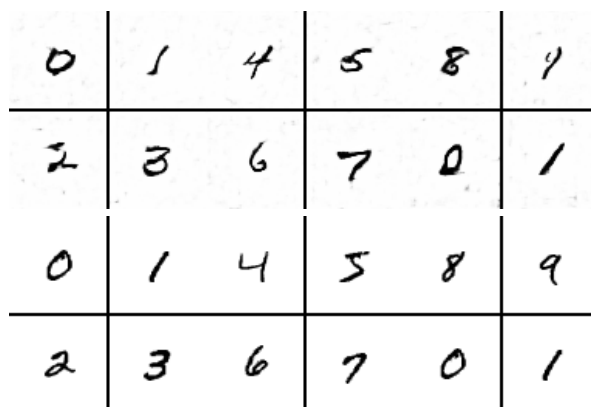
Keywords: GAN, MNIST, Generating images, Generating objects, Generative models

1. UVOD

Dok diskriminativni modeli mašinskog učenja modeluju uslovnu verovatnoću klase Y datog uzorka x , odnosno $P(Y|X=x)$, generativni modeli modeluju zajedničku raspodelu, $P(X, Y)$.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Jelena Slivka, docent.



Slika 1. Primer generisanih slika (gornji red) i primer slika iz skupa podataka sa istim ciframa (donji red).

Labele slika su redom 0-1-2-3, 4-5-6-7 i 8-9-0-1.

Ukoliko je na ovaj način dobro modelovan određeni skup podataka, model je moguće koristiti za generisanje novih uzoraka koji bi odgovarali raspodeli posmatranog skupa podataka. Neki od generativnih modela su model Gausovih mešavina, *Latent Dirichlet allocation* [1], *Variational autoencoder* [2], a u ovom radu korišćene su generativne suparničke mreže (eng. GAN – *Generative Adversarial Network*) [3].

GAN čine generator, koji generiše nove uzorke, i diskriminator, koji odlučuje da li je uzorak pravi (iz nekog skupa podataka) ili lažni (generisan od strane generatora). U cilju generisanja obeleženih podataka, odnosno uzoraka određene klase, korišćen je uslovljeni GAN (eng. *Conditional GAN*) [4], u kojem i generator i diskriminator na ulazu očekuju labelu.

Jedna od mogućih upotreba GAN-ova je generisanje slike primenom arhitekture poput dubokih konvolucionih GAN-ova (eng. *DCGAN – Deep Convolutional GAN*) [5]. Kombinovanjem uslovljenog GAN-a sa ovakvom arhitekturom, moguće je generisanje slike na kojoj se nalazi željeni objekat, dok je generisanje slike sa više obeleženih objekata značajno manje istražena oblast.

U [6] je predloženo rešenje koje razdvaja generator i diskriminator na po dve putanje (podmreže): globalnu putanju, zaduženu za pozadinu; i lokalnu putanju, zaduženu za pojedinačne objekte. Izlazi konvolucionih slojeva obe putanje spajaju se (konkateniraju) i potom nastavljaju.

Tabela 1. Arhitektura modela. Za svaki sloj mreže data je dimenzija njegovog izlaza.

	GLOBAL	LOCAL
G E N E R A T O R	Noise vector	100
	Label vector	/
	Dense, BN, PReLU	32x32x256
	Reshape	7x7x256
	Conv, BN, PReLU	7x7x256
	Upsample	14x14x256
	Conv, BN, PReLU	14x14x128
	Upsample	28x28x128
	Conv, BN, PReLU	28x28x64
	Conv	28x28x1
D I S C R I M I N A T O R	Write object images	128x128x1
	Input image	128x128x1
	Label vector	/
	Conv, PReLU, DO	28x28x32
	Conv, PReLU, DO	28x28x64
	Conv, PReLU, Flatten, DO	28x28x128
	Dense, PReLU, DO	128
	Dense	2
	Reshape	2 + 8
	Write object decisions	10
	Dense	2

Rešenje predloženo u našem radu takođe primenjuje koncept globalne i lokalne putanje. Dok je u [6] zadatak ovih putanja određivanje *feature* mapa, koje se ubrzo konkateniraju i dalje nastavljaju jednim tokom, u našem rešenju obe putanje predstavljaju gotovo ceo generator/diskriminator iz klasične GAN arhitekture, čiji rezultati se spajaju na samom kraju. Rezultati treninga nad skupom podataka baziranom na MNIST [7] slikama prikazani su u gornjem redu slike 1. U narednom poglavlju detaljnije je predstavljena postojeća literatura, kao i poređenje sa našim rešenjem. Arhitektura modela i korišćene tehnike opisane su u poglavlju 3, a poglavlje 4 sadrži analizu dobijenih rezultata i mogućih poboljšanja. Poglavlje 5 predstavlja sumarizaciju celog rada.

2. PRETHODNA REŠENJA

Generisanje slike na kojoj se nalazi više objekata moguće je i primenom običnog GAN-a [3], s tim da tada nemamo potpunu kontrolu nad time koji objekti se generišu i gde. U [8] je predloženo rešenje koje ovaj problem delimično prevazilazi. Predloženi model primenom rekurentne neuronske mreže određuje karakteristike prosleđenog teksta (opisa slike), na osnovu kojih se potom primenom DCGAN arhitekture dolazi do slike. Na ovaj način omogućeno je upravljanje osobinama objekata, ali ne i njihovih položaja i veličina na slici.

Uslovljeni GAN [4] pruža kontrolu klase generisanog objekta, ali njegova klasična arhitektura podržava generisanje samo jednog objekta na slici. Arhitektura predložena u [6] proširuje ovaj koncept time da model na ulazu prihvata proizvoljan broj labela objekata, kao i njihovih graničnih okvira (eng. *bounding box*). Ove informacije se potom pretprocesiraju, kako bi se dobila ulazna slika za generator, na kojoj su iscrtani granični okviri objekata obojeni u različite boje zavisno od njihovih labela. Globalni tok generatora (eng. *global*

pathway) na osnovu ove slike, kao i vektora šuma (eng. *noise*), generiše karakteristike slike, a pre svega njene pozadine. Za svaku od labela, primenom lokalnog toka generatora (eng. *local pathway*), generišu se karakteristike objekata. Primenom odvojene neuronske mreže [9], *feature* mape objekata se transformišu u dimenzije odgovarajućih graničnih okvira. Izlazi globalnog i lokalnog toka mreže se potom konkateniraju, nakon čega sledi nekoliko slojeva standardnih za GAN arhitekturu. Diskriminator se takođe sastoji iz globalnog i lokalnog toka, gde globalni izvlači karakteristike cele slike, a lokalni, na osnovu labela i graničnih okvira izvlači pojedinačno karakteristike svakog objekta. Ponovo, izlazne karakteristike oba toka se konkateniraju, nakon čega slede slojevi kojima se dolazi do konačne odluke diskriminatora.

Rešenje predstavljeno u ovom radu oslanja se na arhitekturu sa globalnim i lokalnim putanjama iz [6], ali uz razliku u tome šta izlazi tih putanja predstavljaju. Izlazi putanja generatora direktno se upisuju u rezultujuću sliku, najpre izlaz globalne putanje, a potom izlazi lokalnih putanja svakog objekta posebno. Izlazi putanja diskriminatora direktno predstavljaju odluke o pozadini i pojedinačnim objektima, na osnovu kojih se potom donosi konačna odluka za celu sliku.

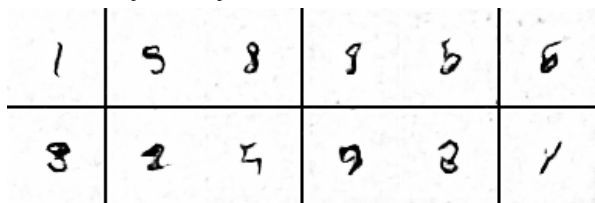
3. METOD

U narednim poglavljima izloženi su skup podataka, arhitektura i trening modela.

3.1. Skup podataka

Kako je cilj bio isprobati novu arhitekturu za generisanje više objekata na jednoj slici, najpre je kreiran jednostavan skup podataka. Izgenerisano je 15000 slika dimenzija 128x128, sa po četiri 32x32 slike cifara iz MNIST skupa podataka [7] na svakoj od njih. Cifre se nalaze u

posebnim ćelijama, međusobno razdvojenim crnim okvirima. Uz svaku sliku zabeležene su i labela sve četiri cifre, kao i njihovi granični okviri. Primer slika iz skupa podataka dat je u donjem redu slike 1.



Slika 2. Primeri slika sa loše generisanim ciframa. Labela slika su redom 1-9-8-6, 3-2-5-8 i 5-2-7-4

3.2. Arhitektura

Glavni izazov bio je napraviti model koji omogućava generisanje različitih obeleženih objekata na slici sa pozadinom. Arhitektura se oslanja na ideju globalne i lokalne podmreže predloženu u [6], a slojevi mreža prikazani su u tabeli 1. Globalni generator, počevši od vektora šuma (**Noise vector**) ima zadatak da napravi pozadinu. Ovo se postiže naizmeničnim primenjenjem konvolucionih slojeva (**Conv**) i uveličavanjem prostornih dimenzija, tj. širine i visine (**Upsample**).

Konvolucionni slojevi praćeni su *batch* normalizacionim slojevima (**BN**) [10], a za aktivacionu funkciju korišćen je parametrizovani **ReLU (PReLU)** [11]. Potom se lokalni generator primenjuje posebno za svaki objekat, pri čemu su korišćene iste težine u slojevima (eng. *shared weights*). Generator objekata, odnosno lokalni generator, koristi šablon uslovljenog GAN-a [4], gde se na ulazu prosleđuje labela željenog objekta (**Label vector**).

Labela se prosleđuje u vidu vektora dužine broja klasa (10 u slučaju cifara), gde se na poziciji koja odgovara željenoj klasi nalazi vrednost 1, a na ostalim pozicijama vrednost 0. Izlazi lokalnog generatora se potom mapiraju na izlaz globalnog generatora na pozicije odgovarajućih graničnih okvira objekata, što daje konačan rezultat mreže generatora.

Diskriminator je takođe podeljen na globalnu i lokalnu podmrežu. Globalni tok posmatra celu sliku i donosi odluku (u vidu 2 neurona) o tome da li je slika prava ili lažna (generisana). Lokalni tok primenjuje se redom na svaki od graničnih okvira i donosi posebnu odluku o verodostojnosti svakog objekta (po 2 neurona).

Na ovaj način, u slučaju četiri objekta, dobija se 10 neurona koji primenom još jednog potpuno povezanog sloja (**Dense**) učestvuju u konačnoj odluci diskriminatora da li se radi o pravoj ili lažnoj slici. Radi boljeg odlučivanja, lokalni diskriminator na ulazu takođe očekuje labelu posmatranog objekta. Za smanjenje rizika od prilagođavanja (eng. *overfitting*) između konvolucionih slojeva intenzivno su korišćeni dropout slojevi (**DO**) [12].

3.3. Trening

Za dobijanje rezultata sa odgovarajućom pozadinom i jasnim ciframa potrebno je oko 25000 epoha, ali je u svrhu eksperimentisanja različitih konfiguracija trenirano i značajno više od toga. Za veličinu *batch*-a isprobane su različite vrednosti, a najbolje se pokazala vrednost 32.

Tokom treninga korak učenja (eng. *learning rate*) je menjan, a najbolje rezultate ostvaruje korak između $1e-6$ i $1e-5$ za mrežu generatora i $1e-5$ i $1e-4$ za mrežu diskriminatora. Uprkos korišćenju *batch* normalizacionih slojeva trening je osetljiv na inicijalizaciju parametara, te su početne vrednosti parametara određene empirijski.

4. REZULTATI I DISKUSIJA

Model je sposoban da generiše slike sa redosledom cifara koji je video više puta tokom treninga, odnosno, ako se neki redosled (npr. 0-1-2-3) nalazi na slikama u skupu podataka, model će naučiti da generiše slike sa tim redosledom (slika 1, gornji red). U slučaju generisanja slike sa potpuno novim, neviđenim redosledom dešava se da model ne generiše uvek očekivane cifre (slika 2). Ovo ukazuje na potrebu za stavljanjem većeg akcenta na labela tokom treninga.

U [6] se labela ne prosleđuju samo kao vektor, već se generiše slika na kojoj su iscrtani granični oblici objekata obojenih na osnovu njihovih labela, što služi kao ulaz u mrežu. Ovim proširenjem naš model bi naučio da bolje razlikuje cifre.

Trenutan skup podataka sadrži veoma jednostavnu pozadinu koja ne pokriva pozadinu pojedinačnih cifara. Kao sledeći korak trebalo bi napraviti slike sa različitim pozadinama različitih šablona/dezena koji pokrivaju celu površinu. Postojeća arhitektura bi morala da se promeni da na pametniji način kombinuje izlaze lokalnog generatora sa izlazom globalnog generatora, a verovatno bi se pojavila i potreba za dodavanjem dodatnih konvolucionih slojeva nakon spajanja tokova, kako bi se osigurala kohezija objekata sa pozadinom.

5. ZAKLJUČAK

U ovom radu predstavljen je model koji omogućava generisanje slike sa više označenih objekata. U svrhu validiranja ideje napravljen je jednostavan skup podataka korišćenjem primeraka iz *MNIST* skupa podataka [7]. Arhitektura modela bazirana je na GAN arhitekturi [3], a kombinuje ideju uslovljavanja mreža iz [4], principe važne za uspešno generisanje slike iz [5] i koncept globalne i lokalne podmreže iz [6].

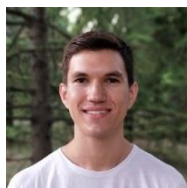
Rezultati pokazuju da model uspešno generiše slike sa konfiguracijama objekata koje je video tokom treninga, ali da pravi greške u slučaju novih konfiguracija. Predloženo je proširenje kojim bi se ovaj problem prevazišao, kao i mogući dalji koraci razvoja u ovom smeru, koji bi omogućili generisanje slika sa kompleksnijim pozadinama.

6. LITERATURA

- [1] Blei, D.M., Ng, A.Y. and Jordan, M.I., 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan), pp.993-1022.
- [2] Kingma, D.P. and Welling, M., 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- [3] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

- [4] Mirza, M. and Osindero, S., 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.
- [5] Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
- [6] Hinz, T., Heinrich, S. and Wermter, S., 2019. Generating Multiple Objects at Spatially Distinct Locations. arXiv preprint arXiv:1901.00686.
- [7] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.
- [8] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B. and Lee, H., 2016. Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396.
- [9] Jaderberg, M., Simonyan, K. and Zisserman, A., 2015. Spatial transformer networks. In Advances in neural information processing systems (pp. 2017-2025).
- [10] Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
- [11] Nair, V. and Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 807-814).
- [12] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), pp.1929-1958.

Kratka biografija:



Luka Maletin rođen je 1995. godine u Novom Sadu. Osnovne akademske studije završio je 2018. godine na Fakultetu tehničkih nauka, na kom brani i master rad 2019. godine iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo i informacione tehnologije.

kontakt: lukadjmaletin@gmail.com

**OTKRIVANJE SIGURNOSNIH PROPUSTA U WEB APLIKACIJAMA METODOM
PENETRACIONOG TESTIRANJA****DETECTING SECURITY VULNERABILITIES IN WEB APPLICATIONS USING
PENETRATION TESTING**

Doroteja Anđelić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Ovaj rad sadrži osnovne koncepte penetracionog testiranja, opisujući sve faze u metodologiji, od dobijanja informacija za identifikovanje mogućih slabih tačaka do eksploataisanja ranjivosti. Opisane su pronađene ranjivosti u realnim web aplikacijama Juice Shop i Peruggia. Za sprovođenje testiranja i otkrivanje potencijalnih sigurnosnih ranjivosti upotrebljeni su alati DIRB, Nikto, Dotdotpwn i Burp Suite.*

Ključne reči: *penetraciono testiranje; sigurnost i bezbednost web aplikacija; hakovanje*

Abstract – *This paper contains basic concepts of penetration testing, describing all stages in the methodology, from obtaining information to identify possible weak points to exploiting vulnerabilities. Found vulnerabilities in real-world web applications Juice Shop and Peruggia are described. DIRB, Nikto, Dotdotpwn and Burp Suite tools were selected to conduct testing and detect potential security vulnerabilities.*

Keywords: *Penetration testing; Web Application security; Hacking*

1. UVOD

Imajući u vidu da je prioritet isporučiti proizvod na vreme, programeri ne mogu da testiraju svoje aplikacije u potpunosti sa aspekta sigurnosti. Samim tim stvara se potreba za profesionalcima koji su specijalizovani za testiranje sigurnosti. Ovi profesionalci nazivaju se penetracioni tester.

Njihovim uključivanjem u razvojni proces, oni preuzimaju odgovornost testiranja aplikacije. Gledajući na sistem sa tačke gledišta napadača, izvršio bi penetracionog testa identifikaciju i eksploatišu postojeće ranjivosti.

U ovom radu su opisani osnovni koncepti penetracionog testiranja, kao i sve faze u metodologiji, od dobijanja informacija za identifikovanje mogućih slabih tačaka do eksploataisanja ranjivosti. Pronađene su neke od postojećih ranjivosti u realnim web aplikacijama Juice Shop i Peruggia.

Penetraciono testiranje predstavlja proaktivan način testiranja aplikacija i raznih sistema.

NAPOMENA:

Ovaj rad proistekao je iz master rada, čiji je mentor bio dr Darko Čapko, vanr. prof.

Cilj testiranja jeste pronalaženje što više nedostataka, propusta i ranjivosti u sigurnosti aplikacije i izbegavanje rizika koje izazivaju ovi nedostaci. Jedan od zadataka penetracionog testera jeste i da, kada pronađe i verifikuje ranjivost, posavetuje programere kako da isprave uočenu grešku i spreče njeno ponovno javljanje. Testiranje, zahvaljujući korišćenju istih alata i tehnika koje bi koristio i zlonameran napadač, nalikuje stvarnom napadu. U najboljem slučaju propusti će se otkriti i ublažiti pre nego što napadač uoči i iskoristi slabost. Testiranje se vrši na kontrolisan način.

Veliki posao penetracionog testiranja više je umetnost nego nauka. Penetraciono testiranje, pored veštine i iskustva, iziskuje specijalni vid shvatanja koje se ne može sistematizovati [1].

2. SIGURNOST SOFTVERA

Softverski problem predstavlja centralni i kritični aspekt problema računarske sigurnosti. Kvar softvera sa posledicama u sigurnosti uključuje greške u implementaciji, kao što su *buffer overflow* i nedostaci u dizajnu (loše upravljanje greškama). Sve češće se koriste i eksploatišu softverske greške ne bi li se pristupilo ciljanom sistemu [2].

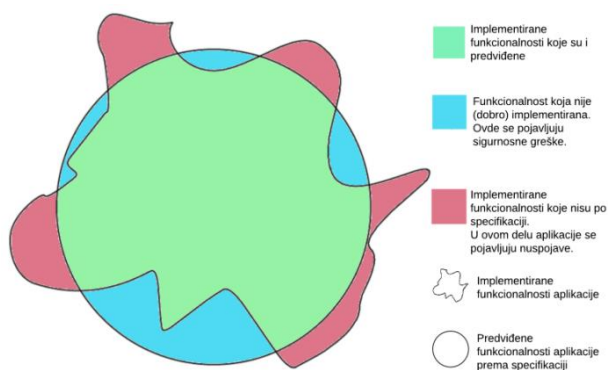
Pod sigurnim softverom podrazumeva se njegovo pravilno funkcionisanje uprkos zlonamerim napadima.

Jednostavnije je zaštititi softver koji nema grešaka nego softver pun ranjivosti. Zato je bitno obratiti pažnju na dizajn softvera, njegovu sigurnost, kao i edukovanje programera i arhitekata kako se gradi siguran softver. Nastajanje sigurnog softvera uključuje procese dizajniranja, izrade i testiranja softvera. Na ovaj način se identifikuju i brišu problemi u samom softveru. Dobija se softver koji je spreman da izdrži i podnese napad proaktivno [2]. U praksi se vrlo često izvode penetracioni testovi nakon faze razvoja softvera, ne bi li se naknadnom analizom utvrdio potencijalni stepen ranjivosti softvera.

2.1 Sigurnosni propusti u softveru

Razvoj sigurnog softvera podrazumeva uključivanje sigurnosnog aspekta u životni ciklus razvoja softvera.

Slika 1 ilustruje prirodu nuspojava većine softverskih ranjivosti. Krugom je predstavljeno predviđeno ponašanje softvera koje se obično definiše specifikacijom. Amorfan oblik nanesen na krug predstavlja stvarnu implementiranu funkcionalnost aplikacije i njeno pravo ponašanje. Idealno bi bilo da se implementirane funkcionalnosti preklapaju sa funkcionalnostima iz specifikacije, što se u praksi skoro nikada ne dešava [3].



Slika 1. Predviđeno i implementirano ponašanje softvera [3]

Zahvaljujući lošem kodiranju, funkcionalnosti programa nehotično utiču jedna na drugu i greške neizbežno ulaze u sistem. Područja kruga koja kodirana aplikacija ne pokriva (prikazano plavom bojom) predstavljaju prostor gde se pojavljuju greške. Kada se jednom ispolji nekorektno ponašanje aplikacije, te greške je prilično lako otkriti. Implementirane funkcionalnosti koje nisu predviđene specifikacijom predstavljaju nenamerne i potencijalno opasne funkcionalnosti softvera u kojima se javljaju nuspojave aplikacije. Ove funkcionalnosti su slikovito predstavljene kao područja koja izlaze van kružne regije (prikazano crvenom bojom) [3].

Implementirane funkcionalnosti koje nisu planirane specifikacijom, kao i nuspojave normalnog funkcionisanja aplikacije, ostavljaju prostor za pojavu sigurnosnih propusta [3].

2.2 OWASP Top 10 – 2017

Što je softver složeniji, postizanje sigurnosti je teže, a poteškoće eksponencijalno rastu. Identifikovanjem najozbiljnijih sigurnosnih rizika web aplikacije, OWASP (Open Web Application Security Project) je probudio svest programera i menadžera. Na ovaj način postao je sigurnosni standard aplikacija. Pored identifikovanja rizika, OWASP ukazuje i na njihove uticaje i protivmere. Rizici koji se nalaze u okviru projekta OWASP Top 10 su [4]:

1. *Injection* – ubrizgavanje koda
2. *Broken Authentication* – slaba (neadekvatna) autentifikacija
3. *Sensitive Data Exposure* – otkrivanje poverljivih podataka
4. *XML External Entities (XXE)* – XML spoljno ubrizgavanje entiteta
5. *Broken Access Control* – neovlašćena kontrola pristupa
6. *Security Misconfiguration* – pogrešna sigurnosna konfiguracija
7. *Cross-Site Scripting (XSS)* – ubrizgavanje skripte
8. *Insecure Deserialization* – nezaštićena deserijalizacija
9. *Using Components with Known Vulnerabilities* – korišćenje komponenti sa poznatim ranjivostima
10. *Insufficient Logging and Monitoring* – nedovoljno beleženje akcija i nadgledanje rada sistema

Najkritičnije ranjivosti u web aplikacijama su greške ubrizgavanja koda. Interaktivne web aplikacije koriste korisnički unos, obrađuju ga i vraćaju rezultat korisniku. Kada je aplikacija ranjiva na ubrizgavanje, ona prihvata

unos od korisnika bez pravilne ili čak bilo kakve validacije i obrađuje ga. Zlonamerni unos će prevariti aplikaciju primoravanjem osnovnih komponenata da izvrše zadatke za koje aplikacija nije programirana.

Funkcije aplikacije koje se odnose na autentifikaciju i upravljanje sesijom često su implementirane pogrešno, omogućavajući napadačima da kompromituju lozinke, ključeve, tokene sesija [4]...

Jedan od glavnih ciljeva bezbednosti informacija je zaštita poverljivosti podataka. Mnoge web aplikacije ne štite na pravi način kredencijale, brojeve kreditnih kartica, zdravstvene informacije i ostale lične podatke. Napadači tako slabo zaštićene podatke mogu ukrasti ili izmeniti. Osetljivi podaci bez dodatne zaštite, kao što je enkripcija, mogu biti ugroženi i zahtevaju posebne mere predostrožnosti kada se razmenjuju sa pretraživačem. Kriptografija, praksa komunikacije i dešifrovanje tajnih zapisa ili poruka, koriste se za zaštitu poverljivosti i integriteta podataka [5].

XXE predstavlja ranjivost web sigurnosti koja omogućava napadaču da se meša u obradu i analiziranje XML podataka. Do napada dolazi kada slabo konfigurabilni XML parser obrađuje ulaz koji sadrži referencu na eksterni entitet. Na ovaj način napadač može da pregleda datoteke na serveru, komunicira sa sistemima kojima i sama aplikacija pristupa, otkrije poverljive podatke, falsifikuje zahtev na strani servera [4]...

Ograničenja u vezi sa onim što autentifikovani korisnici mogu da rade često se ne sprovedu pravilno. Napadači koriste ove mane kako bi pristupili neovlašćenim funkcijama ili podacima. Za sprečavanje ranjivosti najčešće se koristi autorizacija koja se zasniva na ulogama.

Pogrešna sigurnosna konfiguracija je jednostavan način za eksploataciju web stranice. Definiše se kao neuspešna implementacija svih bezbednosnih kontrola web aplikacije ili kao implementacija sa greškama. Ovo je obično rezultat podrazumevane ili nepotpune konfiguracije, otvorenog cloud storage-a, pogrešno konfigurisanog HTTP zaglavlja ili poruke o greškama koja sadrži osetljive informacije [4].

XSS predstavlja slabost koja se oslanja na validaciju ulaza, slično SQL injection-u, a nastaje kada se korisnički unos nepravilno filtrira [6]. To omogućava napadaču da ubrizga zlonameran kod koji se kasnije izvršava u pretraživaču žrtve.

Nesigurna deserijalizacija je ranjivost koja se javlja kada se nepouzdana podaci koriste za zloupotrebu logike aplikacije, sprovođenje *Denial-of-Service (DoS)* napada ili izvršavanje proizvoljnog koda nakon serijalizacije. Primer eksploatacije ranjivosti predstavlja uspešno deserijalizovanje objekta, njegovo modifikovanje, a zatim i ponovno serijalizovanje.

Sve češće i jednostavne web stranice sadrže komponente od kojih zavise. Biblioteke i drugi softverski moduli rade sa istim privilegijama kao i aplikacija. Ako se iskoristi ranjiva komponenta, ovakav napad dovodi do gubitka podataka ili preuzimanja servera [4].

Nedovoljno logovanje i nadgledanje se donekle razlikuje od prethodnih devet rizika. Iako ne dovodi do direktnog upada, zakasnelo i nepravovremeno otkrivanje upada predstavlja neuspeh. Premda stopostotna sigurnost nije

realan cilj, postoje načini da se aplikacija redovno prati kako bi se određene mere preudzele usled neočekivanih događaja.

3. PENETRACIONO TESTIRANJE

Penetraciono testiranje predstavlja jednu od najopsežnijih metoda za pronalaženje ranjivosti i povećanje zaštite računarskih sistema i aplikacija. Iako se koriste alati, trikovi i tehnike koje koriste i zlonamerni hakeri, penetraciono testiranje je legalno uz dozvolu cilja [7].

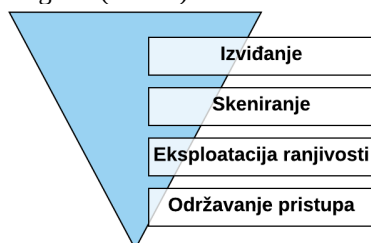
3.1 Faze penetracionog testiranja

Proces penetracionog testiranja se može *razbiti* na niz koraka ili faza. Svi koraci zajedno čine obimnu metodologiju za upotpunjavanje penetracionog testa. Detaljnom analizom izveštaja koji predstavljaju odgovore na incidente, zaključuje se da *black boxing* hakeri prate proceduru kada napadaju ciljani metu. Hakeri sakupljaju informacije, identifikuju ranjivost, a zatim je eksploatišu, odnosno izvršavaju napad. Nakon prodiranja u ciljani sistem, izazov je ostati što duže skriven. Ovakav organizovan pristup testera drži fokusiranog i usmerava ga ka sledećem koraku koji je olakšan zahvaljujući rezultatima iz prethodne faze [8].

Metodologija deli kompleksan proces na seriju manjih, bolje upravljivih zadataka. Postoje različite metodologije i sve sadrže između četiri i sedam faza. Iako broj faza varira, svaka metodologija obezbeđuje kompletan pregled celog procesa penetracionog testiranja [8].

3.1.1 Četvorofazni proces penetracionog testiranja

U okviru četvorofazne metodologije izdvajaju se sledeće faze: izviđanje, skeniranje, eksploatacija ranjivosti i održavanje pristupa **Error! Reference source not found..** Slikovit primer ova četiri koraka može se predstaviti okrenutim trouglom (slika 2).



Slika 2. Slikovit prikaz četiri faze penetracionog testiranja [8]

Izlaz svake faze je širok, a i koristi se kao ulaz naredne faze. Pomerajući se ka poslednjoj fazi, dobijaju se sve specifičniji detalji koji testera vode do cilja. Svaka informacija i svaki detalj o meti se sakuplja i skladišti.

Početna faza, koja je potpuni kontrast od faza koje slede, predstavlja sprovođenje temeljnog istraživanja o svim javno dostupnim informacijama.

Skeniranje uključuje pronalaženje mogućih otvora ili ranjivosti pomoću ručnog testiranja ili automatizovanog skeniranja.

Treća faza obuhvata eksploataciju ranjivosti, kompromitovanje cilja i dobijanje pristupa.

Održavanje pristupa ili posteksploatacija podrazumeva podešavanje sredstava za eskalaciju privilegija na eksploatisanom elementu ili instaliranje zadnjih vrata.

Nakon završetka četvrte faze, neophodno je sumirati sve pronalazke u obliku izveštaja penetracionog testiranja.

4. NAPADANJE NEDOSTATAKA U WEB APLIKACIJAMA

Zbog velikog broja *web* stranica na internetu i povećanja broja organizacija koje svoje poslove obavljaju *online*, *web* aplikacije i *web* serveri su atraktivne mete za napadače.

Investiranje resursa u pisanje bezbednog koda je efikasan metod za minimiziranje ranjivosti *web* aplikacija. Međutim, pisanje bezbednog koda je lako objasniti, ali teško implementirati.

4.1 Alati

Izvršioци testa procenjuju sigurnost sistema primenjujući različite alate za penetraciono testiranje. Korišćenje alata nudi više prednosti. Ako se efikasno koriste, mogu da obave veći deo posla koji je neophodan za kasniju analizu. Alat nikad ne može i ne treba u potpunosti da odmeni testera. Svaku vrstu ranjivosti koju je pronašao alat, neophodno je ručno proveriti. Glavne prednosti alata su brzina i jednostavan način otkrivanja određene sigurnosne ranjivosti.

Alati koji su korišćeni prilikom testiranja *web* aplikacija su *DIRB*, *Nikto*, *Dotdotpwn* i *Burp Suite*.

4.2 Testiranje ranjivih *web* aplikacija *Juice Shop (JS)* i *Peruggia*

Ukoliko penetracioni tester nema izričito pismeno odobrenje vlasnika, skeniranje, testiranje ili eksploatisanje ranjivosti na serverima i aplikacijama na internetu su ilegalni u većini država.

OWASP održava *Juice Shop*, *web* aplikaciju otvorenog koda, koji podržava učenje zasnovano na izazovima. Aplikacija je namerno ranjiva na deset najčešćih ranjivosti koje je identifikovao OWASP [4].

Pred navedena četiri alata postavljena su tri izazova – pronalazak putanje */ftp*, preuzimanje sakrivene datoteke *eastere.gg* i pronalaženje pravog *easter egg*-a (tabela 1).

Tabela 1. Alati i eksploatisane ranjivosti JS aplikacije

	Pronalazak putanje <i>/ftp</i>	Preuzimanje datoteke <i>eastere.gg</i>	Pronalazak pravog <i>easter egg</i> -a
<i>DIRB</i>	✓	✗	✗
<i>Nikto</i>	✓	✗	✗
<i>Dotdotpwn</i>	✗	✗	✗
<i>Burp Suite</i>	✓	✓	✓

Putanju */ftp* su otkrili svi osim *Dotdotpwn* alata. *Dotdotpwn* je koristio datoteku sa dužim spiskom putanja koje su dodate na glavnu putanju, a samim tim je i testiranje iziskivalo više vremena. Alati *DIRB* i *Nikto* su za kraće vreme uradili efektivniji posao. Uspeli su da detektuju zanimljivu putanju */ftp* i penetracionom testeru ukazali na tokove daljeg testiranja. *Burp Suite*, osim izlistanih svih potencijalnih putanja, nudi mogućnost prikazivanja *HTTP* zahteva.

U okviru pronađene putanje *http://localhost:3000/ftp* izlistane su datoteke sa različitim ekstenzijama. Moguće je preuzeti i pregledati samo one sa ekstenzijom *.pdf* ili *.md*. Jedan od izazova predstavlja preuzimanje i pregledanje sadržaja datoteke *eastere.gg*. Presretno je

zahtev sa URL-om *http://localhost:3000/ftp/eastere.gg* i zahvaljujući *Burp Suite*-ovom *Repeater*-u ekstenzija datoteke je u putanji promenjena u *.pdf* nakon čega se datoteka uspešno preuzima. U *HTTP* odgovoru i u datoteci *eastere.gg* nalazi se poruka sa kodom koji krije pravi *easter egg*.

Poslednji zadatak – pronalaženje pravog *easter egg*-a uspeo je da eksploatiše samo *Burp Suite*. Kod dobijen iz *HTTP* odgovora, prilikom preuzimanja datoteke *eastere.gg*, dekodiran je iz formata *base64*. Dobijena *ASCII* putanja ne vodi do rešenja, pa je pronađena putanja dekodovana uz pomoć *ROT13* tehnike maskiranja. Otkrivena putanja otvara pravo *vaskršnje* jaje u obliku *3D* planete.

Web aplikacija *Peruggia* predstavlja društvenu mrežu na kojoj se postavljaju slike i komentari.

Alati *DIRB*, *Nikto* i *Burp Suite* su detektovali potencijalno zanimljive putanje, dok ih *Dotdotpwn*, koristeći *directory traversal* napad, nije pronašao (tabela 2).

Tabela 2. *Uspešnost testiranja Peruggia aplikacije*

	Pronalazak skrivenih putanja
<i>DIRB</i>	✓
<i>Nikto</i>	✓
<i>Dotdotpwn</i>	✗
<i>Burp Suite</i>	✓

DIRB je detektovao devet zanimljivih putanja i na nekima od njih pronađeni su kredencijali korisnika sa administratorskim pravima.

Nikto je generisao pregledan izveštaj u kojem se navodi verzija servera, nedostatak *HttpOnly* tag-a na *cookie*-u, zastarele verzije softvera, kao i nedostatak adekvatne zaštite u zaglavlju.

Dotdotpwn nije pronašao potencijalno ranjive lokacije koje se nalaze iza korenskog direktorijuma.

Burp Suite je potvrdio postojanje putanja koje su već pronađene uz pomoć alata *DIRB* i *Nikto*. *Burp Intruder* automatizuje slanje više zahteva na server, zamenjujući selektovane vrednosti korisničkim unosom, i zapisuje sve odgovore kako bi se analizirali kasnije.

5. ZAKLJUČAK

Penetraciono testiranje omogućava da se softver ili određena moguća ciljna tačka hakera posmatra očima neprijatelja – zlonamernih hakera. Proces testiranja iziskuje dodatno vreme da bi se *zakrpio* sistem pre potencijalnog stvarnog napada i udarca.

Kroz ovaj rad je predstavljeno testiranje sigurnosti web aplikacija metodom penetracionog testiranja. Za sprovođenje testiranja i otkrivanje potencijalnih sigurnosnih ranjivosti izabrane su *Juice Shop* i *Peruggia* web aplikacije. Opisane su pronađene ranjivosti i alati pomoću kojih se došlo do njih.

Pokazano je da u datim primerima alati *DIRB*, *Nikto* i *Burp Suite* uspešno pronalaze ranjivosti, dok se *Dotdotpwn*, koji koristi *directory traversal* napad, nije pokazao kao prikladan. *Burp Suite* sa svojom širokom lepezom funkcionalnosti predstavlja veoma efektivan alat.

Ukoliko se ne otkrije ni jedna ranjivost prilikom testiranja, dokazano je samo da ne postoje ranjivosti za date scenarije, ali ne i da ranjivosti ne postoje. Prolazak penetracionog testa pruža vrlo malo sigurnosti da je aplikacija imuna na sve napade. Aplikacija je imuna samo na napade na koje je testirana.

Skrivena priroda sigurnosnih grešaka je razlog zašto su neophodne specifične tehnike za testiranje. Penetraciono testiranje prkosi tradicionalnom modelu verifikacije specifikacije aplikacije i umesto toga identifikuje neodređene i nesigurne nuspojave ispravne funkcionalnosti aplikacije [9].

Imajući u vidu da su u ovom radu predstavljeni pronalasci i načini eksploatacije samo nekih postojećih ranjivosti aplikacija, dalji rad bi podrazumevao identifikovanje i eksploataciju preostalih slabosti. Pohod na ranjivosti i eksploatacija nedostataka mogu se izvršiti nad ostalim ranjivim web aplikacijama koje se nalaze u okviru projekta OWASP BWA. Jedan od izazova predstavlja testiranje realne aplikacije, ali uz pristanak i dozvolu mete. Kako bi se postigla veća efikasnost, skup alata za testiranje mogao bi se proširiti.

Jedan od najvećih atributa penetracionog testiranja i hakovanja jeste što se nikada ne doseže kraj.

6. LITERATURA

- [1] James P. McDermott, „Attack net penetration testing”, *NSPW*, pp. 15-21, 2000.
- [2] Gary McGraw, *Software security: building security in*, Addison-Wesley Professional, vol. 1, 2006.
- [3] Herbert H. Thompson, „Why security testing is hard”, *IEEE Security and Privacy*, vol. 1, no. 4, pp. 83-86, 2003.
- [4] OWASP Top 10 – 2017, *The Ten Most Critical Web Application Security Risks*
- [5] Gilberto Najera Gutierrez i Juned Ahned Ansari, *Web penetration testing with Kali Linux*, Packt Publishing, 2018.
- [6] Joseph Muniz i Aamir Lakhani, *Web penetration testing with Kali Linux*, Packt Publishing, 2013.
- [7] Ajinkya A. Farsole, Amruta G. Kashikar i Apurva Zunzunwala, „Ethical Hacking”, *International Journal of Computer Applications (IJCA)*, vol. 1, no. 10, pp. 14-20, 2010.
- [8] Patrick Engebretson, *The basics of hacking and penetration testing: Ethical hacking and penetration testing made easy*, Elsevier, 2013.
- [9] H. H. Thompson, „Application penetration testing”, *IEEE Security and Privacy*, vol. 3, no. 1, pp. 66-69, 2005.

Kratka biografija



Doroteja Andelić rođena je 1995. godine u Novom Sadu. Diplomirala je 2018. godine na Fakultetu tehničkih nauka (studijski program Elektroenergetski softverski inženjering) sa prosečnom ocenom 9,71. Master akademske studije (studijski program Primenjeno softversko inženjerstvo) upisala je školske 2018/2019. godine.

КОНЦЕПТ НАПАЈАЊА СЛОЖЕНИХ ТЕХНОЛОШКИХ ПРОЦЕСА НА ПРИМЕРУ ПОГОНА ЗА ОДСУМПОВАЊЕ ДИМНИХ ГАСОВА**CONCEPT OF POWER SUPPLYING AN COMPLEX TECHNOLOGICAL PROCESS ON EXAMPLE OF FLUE GAS DESULPHURIZATION FACILITY**

Ђорђе Радовановић, Борис Думнић, Факултет техничких наука, Нови Сад

Област –ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду приказан је концепт и специфичности у напајању електричном енергијом потрошача погона за одсумповање димних гасова у термоелектранама. Предложено је техничко решење које обезбеђује повећање поузданости у напајању предметног технолошког решења. Дат је осврт и на законску регулативу која дефинише неопходне мере заштите животне средине у предметној области а приказан је и сам технолошки процес термоелектране као и постројења за одсумповање димних гасова.

Кључне речи: енергија/одсумповање/електрана

Abstract – This paper describes the concept of power supplying flue gas desulphurisation plant in thermal power plant, with focus on the specifics in it's realisation. It is proposed one technical solution, which provides increasing in reliability in it's working. It is given a review of legislation which defines a necessary measures of environmental protection. Also it is described a working principle of thermal power plant and flue gas desulphurisation plant.

Keywords: energy/flue gas desulphurization/power plant

1. УВОД

Годинама уназад, основни задатак електропривреде била је производња електричне енергије. Технологија имплементирана пре неколико деценија у оквиру термоелектрана, није узимала у обзир деструктиван утицај производа сагоревања на основне услове за живот и околни живи свет. Управо таква постројења, један су од најзначајних произвођача, али и тешко наслеђе електропривреде [1].

Усвајањем нових националних закона у области заштите животне средине а посебно обзиром на преузете међународне обавезе намеће се потреба за реализацијом погона за смањење емисије штетних гасова из енергетских објеката. Резултати бројних истраживања, покушали су оправдати енергију и новчана средства која треба потрошити за овакве погоне.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Борис Думнић, ван. проф.

Производња електричне енергије у Републици Србији остварује се највећим делом у оквиру Јавног предузећа Електропривреда Србије.

Термоенергетске капацитете чини осам термоелектрана са 25 блокова, који као погонско гориво користе лигнит укупне инсталисане снага од 5.171 MW као и три термоелектране-топлане са 6 блокова, укупне инсталисане снаге 353 MW [1]. Просечна старост термоенергетских јединица на угаљ у Републици Србији износи 29 година. Оваква старост и технолошка застарелост постројења, проузрокују велику потрошњу горива. Карактеристично за ове погоне је и хетерогеност у погледу опреме. У већини постројења која користе линит у Србији, емисија чврстих честица, сумпорних и азотних оксида вишеструко превазилази дозвољене граничне вредности емисије, а такође ни одлагање пепела и шљаке, као и третман вода нису задовољавајући.

У овом раду објашњена је улога и дато једно конкретно технолошко решење погона за одсумповање димних гасова. Централно место у раду, посвећено је концепту напајања погона за одсумповање димних гасова у термоелектрани, са фокусом на специфичности у његовој реализацији, с обзиром на одабрану технологију. Концепција представљања одабраног решења, започиње избором напојних трансформатора на средњем напону, како би се постепено, стигло до крајњих нисконапонских потрошача. Представљено решење напајања, је у највећој мери обликовала сама технологија процеса.

2. ПОСТРОЈЕЊЕ ЗА ОДСУМПОВАЊЕ ДИМНИХ ГАСОВА У ПОГОНУ ТЕРМОЕЛЕКТРАНЕ

Примарне и секундарне мере ограничења емисија оксида сумпора и азота, користе се у котловима у експлоатацији или у новоизграђеним котловима са сагоревањем у спрашеном стању. Постојења за одсумповање димних гасова, која омогућавају испуњење строга ограничења по питању емисије SO₂, у оквиру усвојених стратегија очувања животне средине су:

1. поступак дозирања сорбента непосредно у ложиште котла (FSI),
2. комбинација претходног поступка и хлађења димних гасова распршивањем воде у

реактору који се налази између котла и филтара (FSI + Q),

3. полусуви поступак, који се састоји од три фазе: хлађење димних гасова, мешање димних гасова са сувим сорбентом у облику $\text{Ca}(\text{OH})_2$ и рецикулација сорбента и производа реакције сорбента и димних гасова.
4. влажни метод – испирање димних гасова у сепаратору воденом суспензијом сорбента при чему се као крајњи производ добија гипс. Овај метод познат под називом „кречњачки-мокри поступак“ се у пракси показао као најефикаснији и техно-економски најоптимизованији.

2.1 Кречњачки-мокри поступак одсумпоровања

Најпознатији и најефикаснији метод одсумпоровања димних гасова је кречно-мокри поступак. Овом методом достиже се ефикасност од 90-95%. Основни принцип је, испирање димних гасова воденом суспензијом креча или кречњака у апсорбиционом реактору, чиме се добија сулфит креча CaSO_3 . Додатно засићење кисеоником, омогућава конверзију CaSO_3 у CaSO_4 , даљим испирањем и сушењем добија се гипс $\text{CaSO}_4 \cdot 2\text{H}_2\text{O}$.

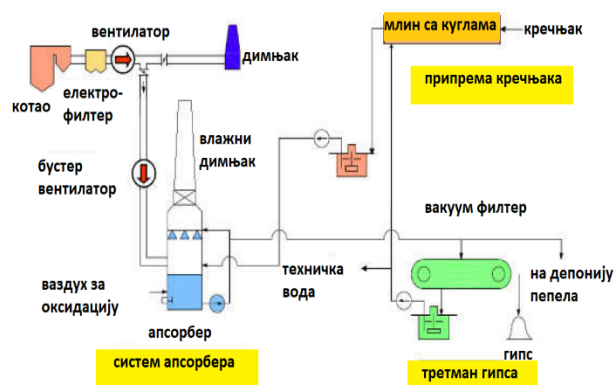
Прах кречњака, или негашеног креча се помоћу пумпи убацује у апсорбер. Циркулационе пумпе, цевоводи и систем млазница обезбеђују интензивно испирање димних гасова у апсорпционом торњу. У зависности од степена интензивности испирања, зависи и ефикасност. У оваквом поцесу одсумпоровања се додатно уклањају једињења HCl и HF , као и пепео. Из отпада се, даље, издвајају тешки метали. Рахлађен димни гас, потребно је загрејати, и тек након загревања упутити у димњак [2]. Висока ефикасност ових постројења, релативно ниска потрошња сорбента и могућности искоришћавања отпадног продукта – гипса, омогућили су широку примену технологије овог типа.

2.2 Подсистеми постројења за одсумпоровање димних гасова мокрым поступком

Постројење за одсумпоровање димних гасова, представља један сложен систем, који чине следећи сегменти:

- систем апсорбера,
- систем димног гаса,
- систем за пријем и складиштење кречњака,
- систем за припрему и складиштење кречњачке суспензије,
- систем за одводњавање суспензије гипса,
- системи дренаже,
- помоћни системи (систем технолошке воде, напајање електричном енергијом, систем управљања, систем компримованог ваздуха итд.) и
- систем за транспорт и одлагање гипса.

Основни технолошки процес који се одвија у овом поступку представљен је на слици 1 ().



Слика 1. Одсумпоровање димних гасова влажним, кречњачким поступком [4]

3. СПЕЦИФИЧНОСТИ НАПАЈАЊА ПРОЦЕСНО КРИТИЧНИХ ПОТРОШАЧА

3.1 Димензионисање и избор уљних трансформатор

Напајање електричних инсталација постројења за одсумпоровање димних гасова, врши се са генераторских сабирница енергетских блокова термоелектране. Прикључење мреже система за одсумпоровање остварује се помоћу оклопљених сабирница блокова, са којих се напајају уљни трансформатори.

Након расподеле потрошње, дате од стране испоручиоца технологије одсумпоровања, у циљу реализације напајања најпре се приступа анализи потребне енергије и димензионисању опреме. Најпре је потребно доћи до једновремене снаге потрошача. Она се добија као улазна снага потрошача помножена са фактором једновремености. Из табела, приложених у самом мастер раду из кога је приистекао овај рад, за одабрану снагу трансформатора од 31,5MVA, може се закључити да ће у нормалном раду најоптерећенији трансформатор бити искоришћен са само 37% свог капацитета.

Важан параметар који је било потребно одредити је напон кратког споја. У зависности од називне снаге, различите су карактеристичне вредности напона кратког споја. Он се изражава у процентима, и као такав је једнак импеданси кратког споја. У режиму кратког споја, доминантни су флуксеви расипања примарног и секундарног намотаја, и ови параметри су кључни за прорачун струје кратког споја, и димензионисање опреме на струјна, термичка и механичка напрезања.

$$\underline{U}_k = (R_k + jX_k)I_k \quad (3.1)$$

$$u_k = \frac{Z_k I_n}{U_n} \quad (3.2)$$

$$u_r = \frac{R_k I_n}{U_n} \quad (3.3)$$

$$u_x = \frac{X_k I_n}{U_n} = \sqrt{u_k^2 - u_r^2} \quad (3.4)$$

$$u_k(\%) = 100 u_k = 100 \frac{U_k}{U_n} = z_k[\%] \quad (3.5)$$

Импеданса треба да ограничи струју кратког споја на секундару трансформатора испод вредности од 50kA, што је минимум за димензионисање прекидне опреме. С обзиром на велики број мотора на 6kV, у систему за одсумпоравање, неопходно је уважити њихов утицај на струју кратког споја. Струја која дотиче у место квара од асинхроног мотора износи:

$$I''_{am} = \frac{1,1 U_n}{X_k + X_T} \quad (3.6)$$

Укупна субтранзијентна струја квара:

$$I'' = \frac{1,1 U_n}{X''} + \frac{1,1 U_n}{X_k + X_T} \quad (3.7)$$

При чему је

- X'' -еквивалентна реактанса мреже и
- X_T -реактанса трансформатора.

Узевши у обзир утицај мотора који једно време доприносе струји кратког споја на истим сабирницама, долази се до збира њихових називних струја и називних струја покретања:

- збир називних струја [A] = 2.428
- збир називних струја покретања [A] = 14.762, што је узето као укупан допринос мотора струји кратког споја, једновремено прикључених на исти сабирнички систем.

На основу израза за укупну субтранзијентну струју квара, може се доћи до компоненте струје кратког споја из мреже, која треба да буде ограничена на: 50kA-14,76kA=35,24kA. На основу израза за импедансу трансформатора:

$$Z_{tr} = \frac{C_{max} U_L}{\sqrt{3} I_{ks}} = 0,1081 \Omega \quad (3.8)$$

где је:

- Z_{tr} - минимална захтевана импеданса трансформатора,
- C_{max} -напонски фактор (1.1 за напоне веће од 1kV, према стандарду IEC 60909),
- U_L -напон мреже [kV] и
- I_{ks} -расположива струја кратког споја.

Из израза:

$$Z_{tr} = \frac{u_k U_{nt}^2}{100\% S_{nt}} \quad (3.9)$$

Где је:

- Z_{tr} - минимална захтевана импеданса трансформатора,
- u_k -називни напон кратког споја [%],

- U_{nt} -називни напон трансформатора [kV] и
- S_{nt} -називна снага трансформатора.

Долази се до вредности $u_k=9,46\%$, док је стандардом (IEC 60076-5) препоручена вредност износила 10%, и за ту вредност омска вредност импедансе износи $Z_T=0,1143 \Omega$. Када је у питању примена трансформатора са теретном склопом, примењује се и фактор корекције:

$$K_t = \frac{0,95 C_{max}}{1 + 0,6 X_t} = 0,978 \quad (3.10)$$

Сада прерачуната вредност импедансе износи: $Z_t=K_t \times Z_T=0,1120 \Omega$, и то је мање од препоручене вредности за $u_k=10\%$. Како се у колу кратког споја налази и импеданса мреже, препоручена вредност од 10% ће бити довољна да ограничи почетну вредност струје квара испод 50kVA на сабирницама 6kV. Овај прорачун, потврђен је у прорачуну струје квара у мрежи [3].

3.2 Непрекидност напајања СН разводног постројења – брза преклопна аутоматика

Ради одржавања максималног континуитета процеса, постројење је опремљено брзом преклопном аутоматиком (FT-fast transfer). Основна предност примене брзог аутоматског трансфера, огледа се у могућности наставка рада потрошача без прекида технолошког циклуса. Стања у којима се јавља потреба за пребацивањем напајања на резервни довод су:

- појава квара на напојном доводу,
- одржавање напојног трансформатора или доводне ћелије и
- одступање параметара квалитета напона.

На сабирницама са којих се напајају мотори јавља се резидуални напон. Он је последица постојања флуksа у ротору, услед протицања струје кроз роторско коло, која не нестаје тренутно. Основни уређај је микропроцесорска јединица која делује паралелно са заштитним релеом на укључење и искључење калемова прекидача. Основне информације кључне за управљање овог уређаја су напони на напојним водовима и уклопно стање прекидача.

Аутоматски режим, изводи се на четири начина:

- брзи трансфер (*fast transfer*),
- трансфер на прво поклапање фазе (*transfer at 1 st phase coincidence*),
- трансфер према резидуалном напону (*residual voltage*) и
- временски контролисан трансфер (*long time transfer*).

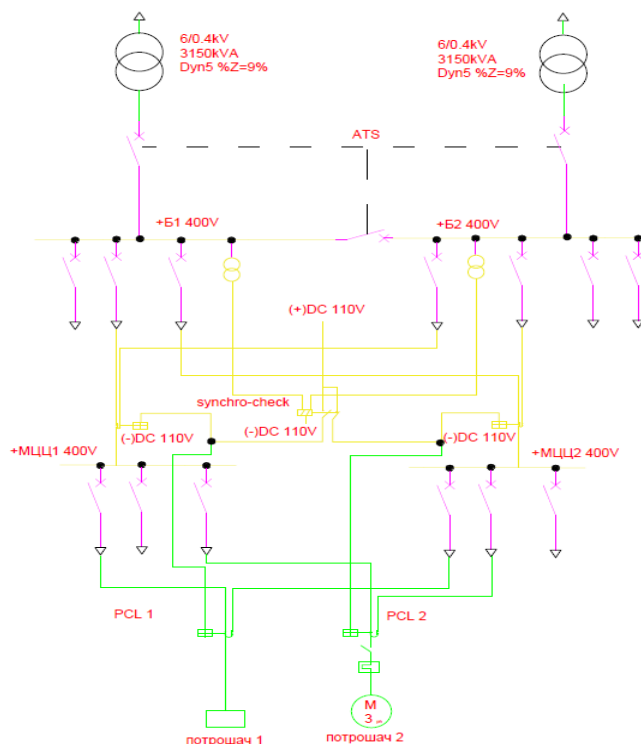
3.2 Поуздано снабдевање нисконапонских потрошача електричном енергијом

Различити су начини којима се може остварити управљање прекидачким јединицама у функцији брзог аутоматског пребацивања напајања (*eng. Automatic transfer switch*). Често се у те сврхе користе

префабрикована решења познатих произвођача, или се одговарајућа логика имплементира у неку управљачку јединицу (нпр. ПЛЦ). Систем је опремљен функцијом провере синхронизма, која условљава моменат преласка са једног извора на други. Сам концепт заснива се на сигнаlima који дају информације о статусу прекидачких јединица, заштитних релеа, релеа за синхронизацију (synchro-check relays) и интегрисаног система дистрибуиране контроле (eng. DCS - distributed control system). Са главних постројења напајају се моторни контролни центри (eng. MCC – motor control center), који су изведени као касетна разводна постројења. У оквиру ормана са касетама, групишу се потрошачи према потребама технолошког процеса. Нови степен повећања поузданости напајања уводи се двоструким напајањем поменутих MCC развода.

Постоји још један додатни степен расклопне опреме, који омогућава несметано снабдевање појединих потрошача, чак и у случају одржавања неког извода касетног развода ниског напона (MCC-a).

Дакле, поједини потрошачи који су процесно критични, обезбеђени су управљачком техником помоћу посебних пребацача, контактора и термичке заштите. За те потребе формиране су посебне касете (PCL - process critical load). Ове касете се напајају, као MCC-ови, двострано [5].



Слика 3. Концепт напајања крајњих нисконапонских потрошача [5]

4. ЗАКЉУЧАК

Постројења термоелектране, данас незаобилазно морају да садрже постројење за одсумпоравање

димних гасова уколико желе да задовоље важеће критеријуме из области заштите животне средине.

Погон, који је на било који начин интегрисан у оквиру термоелектране, и има ма колико миноран утицај на њен рад, потребно је на најбољи могући начин обезбедити у смислу напајања електричном енергијом. Представљено технолошко решење за одсумпоравање димних гасова у термоелектрани, представља тренутно најефикаснију методу.

Специфично решење напајања потрошача погона за одсумпоравање, представљено кроз овај рад, обезбеђује поуздано снабдевање електричном енергијом и представља техно-економско оптимално решење.

5. ЛИТЕРАТУРА

- [1] Зелена књига Електропривреде Србије, Београд, јун 2009.
- [2] Енергопројект ентел, Постројење за одсумпоравање димних гасова – ажурирана студија о процени утицаја на животну средину, јул 2015.
- [3] SUE 3000 High Speed Transfer Device Product Description, ABB.
- [4] Пројекат технологије постројења за одсумпоравање димних гасова термоелектране Никола Тесла А, Београд, 2018.
- [5] Ђорђе Радовановић, "Концепт напајања сложених технолошких процеса на примеру погона за одсумпоравање димних гасова", Мастер рад, Факултет техничких наука у Новом Саду, 2019.

Кратка биографија:



Ђорђе Радовановић – рођен у Ваљеву 1995. године. Дипломирао на Факултету техничких наука из области Електротехника и рачунарство – Енергетска електроника и електричне машине, 2018. године. Област интересовања: високнапонска постројења, електромоторни погони у индустрији, а велика љубав – пчеларство.

контакт:
radovanovicdjordje95@gmail.com



Борис Думнић дипломирао је 2003. године на Универзитету у Новом Саду, Факултету техничких наука из области електротехника и рачунарство. Од 2004. године запослен је на Факултету техничких наука у Новом Саду. Магистарски рад је одбранио 2007. године, а 2013. године је одбранио и докторску тезу. Област интересовања су му електромоторни погони, енергетска електроника и обновљиви извори електричне енергије.

MODELOVANJE TROFAZNE KAVEZNE ASINHRONE MAŠINE UZ UVAŽAVANJE KONSTRUKCIONIH OSOBENOSTI**MODELING OF A THREE-PHASE SQUIRREL CAGE INDUCTION MACHINE CONSIDERING THE CONSTRUCTION FEATURES***Ivana Ćirić, Dejan Jerkan, Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je razvijen dinamički model trofazne kavezne asinhronne mašine sa uvaženim efektima kojima doprinosi specifičnost konstrukcije, kao što je uticaj žljebova na statoru i rotoru razmatrane mašine. Na modelu koji je razvijen u MATLAB Simulinku su izvršene karakteristične simulacije. Takođe, tako dobijeni rezultati su upoređeni sa rezultatima dobijenim eksperimentalnim putem.

Ključne reči: Kavezna asinhronna mašina, metoda konačnih elemenata, MATLAB Simulink, dinamički model, tranzijentni procesi.

Abstract – In this paper a dynamic model of a three-phase squirrel cage induction machine is developed while considering the effects which are contributed by the specificity of the construction, such as the influence of the stator's and rotor's grooves of the considered machine. On the model developed in MATLAB Simulink, characteristic simulations were performed. Also, the results thus obtained were compared with the experimental results.

Keywords: Squirrel cage induction machine, Finite Element Analysis, MATLAB Simulink, dynamic model, transient processes.

1. UVOD

Zahvaljujući maloj masi i zapremini, jednostavnoj konstrukciji, većoj specifičnoj snazi (snaga po jedinici mase) i mogućnostima rada na većim brzinama obrtanja u odnosu na mašine jednosmerne struje, asinhronne mašine predstavljaju najrasprostranjeniju vrstu električnih mašina. Asinhronne mašine su danas široko zastupljene, naročito u savremenoj industriji gde se u pogonima za različite namene koriste kao pogonske mašine.

Ukoliko se govori o primeni asinhronnih motora u industrijskim pogonima, nalaze se u okviru raznih mešalica, mlinova, pumpi, presa, valjaonica, ventilatora, bušilica, ali i raznih drugih uređaja u okviru proizvodnih sistema. U domaćinstvima se primenjuju u različitim vrstama kućnih aparata, kao što su veš mašine, frižideri, klima uređaji, zamrzivači, mikseri, ali i u okviru računara i računarske opreme.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dejan Jerkan, docent.

Poznavanje dinamičkog modela mašina je potrebno za rešavanje problema upravljanja generatorima i motorima, za projektovanje zaštita i nadzora, za određivanje strukture i parametara regulacije u robotici, kao i za rešavanje problema u automatizaciji proizvodnje, električnim vozilima i drugim sličnim primenama [1]. Osnovni cilj matematičkog modelovanja jeste dobijanje adekvatnog, upotrebljivog i korisnog matematičkog modela koji će zadržati bitne fizičke karakteristike modelovane mašine i suštinski dobro oslikavati prirodu pojava u njoj. Matematičko modelovanje je značajno sa aspekta proučavanja karakteristika, analiziranja različitih radnih režima mašine, kao i razvijanja složenih algoritama za upravljanje istim. Zahvaljujući dovoljno detaljnom matematičkom modelu mašine, pruža se mogućnost računarskih simulacija raznih režima rada i praćenje kako će se u tim, ali i mnogim drugim okolnostima i situacijama modelovana mašina ponašati.

2. MATEMATIČKI MODEL TROFAZNE KAVEZNE ASINHRONE MAŠINE

Matematički model koji će biti izložen u nastavku je dinamički jer omogućava proučavanje tranzijentnih pojava u električnim mašinama, dok se ulazni parametri modela dobijaju primenom metode konačnih elemenata. Sam model se oslanja na skup relacija koje su poznate pod imenom višestruko spregnuta električna kola (eng. multiple coupled circuit approach) [2]. Ovakav model, predstavljen u originalnom, netransformisanom domenu je dovoljno detaljan da prikaže sve pojave od značaja koje su važne za proučavanje trofazne kavezne asinhronne mašine. Kako bi se dobio dovoljno detaljan i upotrebljiv model koji predstavlja matematičku reprezentaciju sistema koji se posmatra, neophodno je uvažiti određene aproksimacije [3]: pojave u električnoj mašini mogu dovoljno detaljno opisati pomoću skoncentrisanih parametara, zanemaruju se pojave kapacitivnog karaktera, nelinearnost feromagnetskog materijala, kao i gubici u magnetskom kolu mašine. Omski otpori statora i rotora su konstantni, kao i momenat inercije rotirajućih masa. Smatra se da nema struja curenja koje se zatvaraju mimo provodnika. Zakošenje štapnih provodnika rotora se izuzima iz razmatranja.

S obzirom na to da rotorski kavez nema električnih pristupa, potrebno je pronaći adekvatan način za definisanje i opisivanje svih pojedinačnih električnih kola kaveza. Najbolje rešenje ovog problema je da se rotorsko kolo predstavi preko rotorskih petlji. Jednu rotorsku petlju

sačinjava zatvoren provodni put koji se sastoji od dve susedne rotorske šipke i dva naspramna segmenta kratkospojnih prstenova [4].

U nastavku će biti izložene relacije koje opisuju kako električni, tako i mehanički podsistem matematičkog modela trofazne kavezne asinhronne mašine. Diferencijalna jednačina naponske ravnoteže u matričnoj formi može se predstaviti preko sledeće relacije:

$$\mathbf{u} = \mathbf{R} \cdot \mathbf{i} + \frac{d\boldsymbol{\psi}}{dt} \quad (1)$$

$\mathbf{u}$ – vektor napona svih električnih kola kojima je razmatrana električna mašina modelovana

$\mathbf{i}$ – vektor struja statorskih namotaja i svih rotorskih petlji, uključujući i struju kratkospojnog prstena

$\boldsymbol{\psi}$ – vektor svih fluksnih obuhvata statora i rotora

$\mathbf{R}$ – matrica svih statorskih i rotorskih otpornosti

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_s & \\ & \mathbf{R}_r \end{bmatrix} \quad (2)$$

$\mathbf{R}_s$ – dijagonalna submatrica statorskih otpornosti dimenzija 3×3

$\mathbf{R}_r$ – submatrica rotorskih otpornosti dimenzija $(N_B+1) \times (N_B+1)$

Ukoliko se sa N_B označi broj rotorskih petlji, odnosno broj rotorskih štapnih provodnika, uvažavajući i kratkospojne prstenove, tada postoji N_B+1 nezavisnih strujnih petlji čije struje su posledice postojanja indukovanih elektromotornih sila u njima. Generisanje submatrice statorskih otpornosti $\mathbf{R}_s$ je trivijalno i ona je prosta, dijagonalna matrica otpornosti namotaja statora. Submatrica rotorskih otpornosti $\mathbf{R}_r$ nije dijagonalna matrica jer ona opisuje topologiju rotorskog kaveza, odnosno način uspostavljanja naponske ravnoteže u rotorskim petljama. Uvažavanjem jednačina naponske ravnoteže za sve nezavisne rotorske petlje i uzimajući u obzir ukupnu otpornost tekuće rotorske petlje koja je data kao zbir otpornosti dva kratkospojna prstena i otpornosti dve susedne rotorske šipke, dobija se razvijeni oblik submatrice rotorskih otpornosti.

Naredna matrična jednačina koja predstavlja sastavni deo električnog podsistema matematičkog modela daje vezu između flukseva i struja preko nestacionarne matrice induktivnosti i predstavlja jednačinu za fluksni obuhvat:

$$\boldsymbol{\psi} = \mathbf{L} \cdot \mathbf{i} \quad (3)$$

Matrica induktivnosti $\mathbf{L}$ trofazne asinhronne mašine se sastoji od četiri submatrice, te se ona može napisati u sledećem obliku:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{ss} & \mathbf{L}_{sr} \\ \mathbf{L}_{rs} & \mathbf{L}_{rr} \end{bmatrix} \quad (4)$$

$\mathbf{L}_{ss}$ – submatrica svih sopstvenih i međusobnih induktivnosti namotaja statora

$\mathbf{L}_{sr}$ – submatrica statorsko-rotorskih induktivnosti

$\mathbf{L}_{rr}$ – submatrica svih sopstvenih i međusobnih rotorskih induktivnosti

$\mathbf{L}_{rs}$ – submatrica rotorsko-statorskih induktivnosti

Submatrica statorskih induktivnosti $\mathbf{L}_{ss}$ je matrica dimenzija 3×3 na čijoj glavnoj dijagonali se nalaze elementi sopstvenih induktivnosti namotaja statora, dok vandijagonalni elementi predstavljaju koeficijente međusobnih induktivnosti. S obzirom na činjenicu da se u ovom radu uzima u obzir postojanje razlike u magnetskoj permeabilnosti po obodu vazdušnog zazora usled uvažavanja zljebova na statoru i rotoru, elementi matrice

$\mathbf{L}_{ss}$ nisu nepromenljivi, već su funkcija vremena, odnosno promene relativnog položaja rotora u odnosu na stator. Submatrica rotorskih induktivnosti $\mathbf{L}_{rr}$ je znatno složenije forme, te su zbog izuzetno velikog broja međusobnih sprega koje postoje unutar rotorskog kaveza, dimenzije ove matrice $(N_B+1) \times (N_B+1)$. Pored činjenice da se i u njoj mora uvažiti način opisivanja rotorskog kaveza uvođenjem konturnih struja, uzimanje u obzir promenljivosti ove matrice u funkciji položaja rotora dodatno komplikuju njenu formu. Submatrice statorsko-rotorskih $\mathbf{L}_{sr}$ i rotorsko-statorskih induktivnosti $\mathbf{L}_{rs}$ su od suštinskog značaja jer se pomoću njih opisuje dominantna komponenta ostvarenog elektromagnetnog momenta. Zbog simetrije submatrica međusobnih induktivnosti, važi da je $\mathbf{L}_{sr} = \mathbf{L}_{rs}^T$. Dimenzija submatrice statorsko-rotorskih induktivnosti je $3 \times (N_B+1)$. Svi elementi ovih submatrica međusobnih induktivnosti imaju identičan talasni oblik, pri čemu je fazni pomak između elemenata kolona matrice $\mathbf{L}_{sr}$ jednak $2\pi/N_B$, dok su elementi vrsta međusobno fazno pomereni za ugao $2\pi/3$. Pritom, važno je napomenuti da to ne važi za elemente poslednje kolone matrice $\mathbf{L}_{sr}$ koji opisuju spregu između statorskih namotaja i petlje kratkospojnog prstena, koja je od zanemarivog značaja.

Kako bi matematički model bio potpun, neophodno je prethodno definisanim jednačinama električnog podsistema pridružiti i relacije koje opisuju mehanički podsistem. Izraz za elektromagnetni momenat, odnosno momenat konverzije, dat je sledećom relacijom:

$$M_{em} = \frac{1}{2} \cdot p \cdot \begin{bmatrix} \mathbf{i}_s \\ \mathbf{i}_r \end{bmatrix}^T \cdot \frac{d}{d\theta} \left(\begin{bmatrix} \mathbf{L}_{ss} & \mathbf{L}_{sr} \\ \mathbf{L}_{rs} & \mathbf{L}_{rr} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{i}_s \\ \mathbf{i}_r \end{bmatrix} \right) \quad (5)$$

M_{em} – elektromagnetni momenat mašine

p – broj pari polova trofazne asinhronne mašine

θ – električni ugao koji opisuje trenutni položaj rotora u odnosu na stator

Usled dejstva elektromagnetskih sila, na rotor deluje elektromagnetski momenat M_{em} . Rotor je preko vratila povezan sa radnom mašinom ili pogonskom turbinom. Vratilo prenosi mehanički momenat M_m . U mehaničkom podsistemu postoje gubici usled trenja i ventilacije, kao i akumulacija kinetičke energije u obrtnim delovima mašine. Stoga mehanički momenat M_m , koji se ima na izlaznom vratilu nije jednak elektromagnetskom momentu M_{em} koji deluje na rotor [1]. Pojave u mehaničkom podsistemu se opisuju i Njutnovom diferencijalnom jednačinom kretanja koja opisuje promenu brzine obrtanja mašine i određena je izrazom:

$$M_{em} - M_m = J \cdot \frac{d\omega_m}{dt} + K_m \cdot \omega_m \quad (6)$$

M_m – mehanički momenat mašine

J_m – ukupni momenat inercije pokretnog dela pogona

K_m – koeficijent i ventilacije odnosno koeficijent gubitaka

ω_m – trenutna vrednost mehaničke ugaone brzine

3. ODREĐIVANJE ELEKTRIČNIH PARAMETARA KAVEZNE ASINHRONE MAŠINE

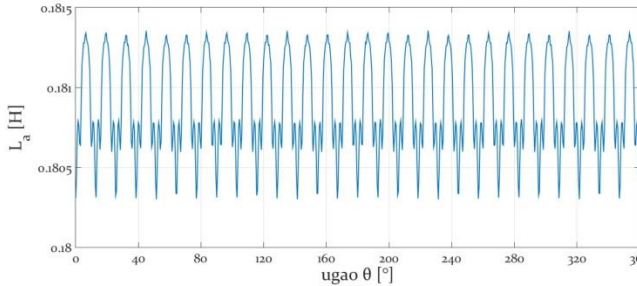
Kako bi se mogle izvršiti računarske simulacije korišćenjem navedenog matematičkog modela, neophodno je poznavati vrednosti elemenata svih matrica induktivnosti i otpornosti. Otpornosti namotaja se dobijaju eksperimentalno ili vršenjem adekvatnih proračuna. Za izračunavanje elemenata matrice induktivnosti koristi se metoda

konačnih elemenata koja podrazumeva sprovođenje niza magnetostatičkih simulacija. To se najčešće realizuje pomoću programskog paketa FEMM 4.2 [5]. Prilikom izvođenja magnetostatičkih simulacija, određeni namotaj ili rotorska petlja se pobuđuje jednosmernom strujom. Zbog uvažavanja postojanja žljebova i promene relativnog međusobnog položaja namotaja statora i rotorskih petlji usled kretanja rotora koje za posledicu ima periodične zavisnosti svih elemenata matrice induktivnosti od položaja rotora, potrebno je izvršiti čitave serije simulacija za različite položaje rotorskog kaveza.

Kada se izračunaju fluksni obuhvati, iz njih se primenom metode konačnih elemenata dobijaju vrednosti nepoznatih induktivnosti u vidu odbiraka. S obzirom na složenoperiodičnu zavisnost induktivnosti od položaja rotora, one se mogu predstaviti preko koeficijenata Furijeovog reda. Prvo se dobijene diskretne vrednosti induktivnosti linearnom interpolacijom moraju predstaviti kao kontinualne funkcije, s obzirom na činjenicu da je razvoj u Furijeov red definisan isključivo za kontinualne funkcije.

Izraz za razvoj sopstvene induktivnosti statorskog namotaja faze a u Furijeov red dat je relacijom (7), a primer izračunavanja ove induktivnosti dat za konkretnu mašinu prikazan je na slici 1. Kada se izračunata sopstvena induktivnost statorskog namotaja faze a fazno pomeri za $-2\pi/3$, dobije se sopstvena induktivnost faze b, a kada je fazni pomeraj jednak $+2\pi/3$, dobija se sopstvena induktivnost statorskog namotaja faze c.

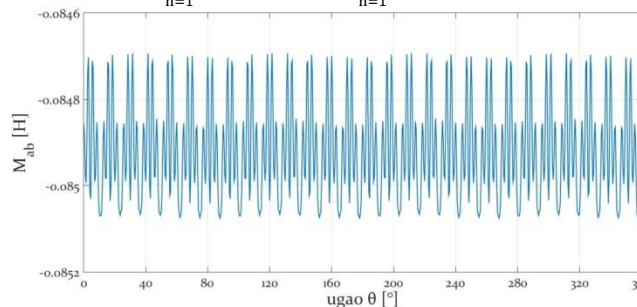
$$L_a(\theta) = L_{ss0} + \sum_{n=1}^{\infty} L_{ssan} \cdot \cos(n\theta) + \sum_{n=1}^{\infty} L_{ssbn} \cdot \sin(n\theta) \quad (7)$$



Slika 1. Sopstvena induktivnost statora faze a

Izraz za razvoj međusobne induktivnosti statorskih namotaja M_{ab} u Furijeov red dat je relacijom (8), a primer izračunavanja ove induktivnosti dat za konkretnu mašinu prikazan je na slici 2. Talasni oblici induktivnosti M_{ac} i M_{bc} su isti, ali fazno pomereni za $-2\pi/3$, odnosno $+2\pi/3$, respektivno.

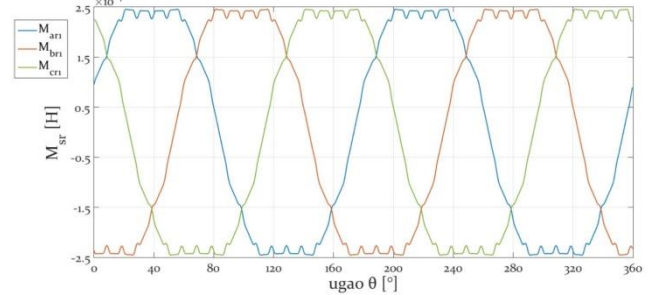
$$M_{ab}(\theta) = M_{ss0} + \sum_{n=1}^{\infty} M_{ssan} \cdot \cos(n\theta) + \sum_{n=1}^{\infty} M_{ssbn} \cdot \sin(n\theta) \quad (8)$$



Slika 2. Međusobna induktivnost statorskih namotaja M_{ab}

Slika 3 prikazuje talasne oblike međusobnih induktivnosti statorskih namotaja faza a, b i c i prve u nizu rotorskih petlji unutar kratkospojenog kaveznog rotora razmatrane mašine (rotorska petlja sa indeksom 1), a izraz za razvoj induktivnosti M_{ar1} u Furijeov red je dat relacijom (9). S obzirom na činjenicu da su elementi vrsta submatrice statorsko-rotorskih induktivnosti L_{sr} međusobno fazno pomereni za $2\pi/3$, talasni oblici predstavljenih induktivnosti su takođe pomereni za isti ugao.

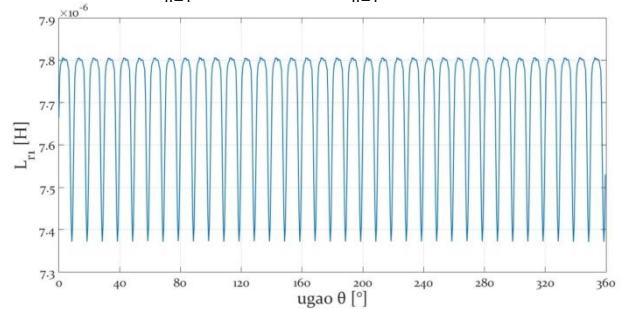
$$M_{ar1}(\theta) = \sum_{n=1}^{\infty} M_{sr an} \cdot \cos(n\theta) + \sum_{n=1}^{\infty} M_{sr bn} \cdot \sin(n\theta) \quad (9)$$



Slika 3. Statorsko-rotorske induktivnosti

Izraz koji predstavlja razvoj u Furijeov red sopstvene induktivnosti prve u nizu rotorskih petlji unutar kaveznog rotora mašine dat je relacijom (10), dok je primer izračunavanja ove induktivnosti prikazan na slici 4.

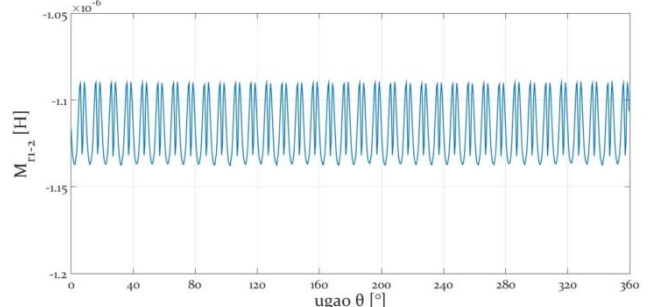
$$L_{r1}(\theta) = L_{rr0} + \sum_{n=1}^{\infty} L_{rran} \cdot \cos(n\theta) + \sum_{n=1}^{\infty} L_{rrbn} \cdot \sin(n\theta) \quad (10)$$



Slika 4. Sopstvena induktivnost prve rotorske petlje

Sve preostale sopstvene induktivnosti rotorskih petlji imaju identičan talasni oblik, uzimajući u obzir odgovarajući fazni pomak između susednih rotorskih petlji za ugao $2\pi/N_B$. Na slici 5 prikazana je zavisnost međusobne induktivnosti rotorskih petlji sa indeksima 1 i 2. Identičan talasni oblik imaju i sve ostale međusobne statorsko-rotorske induktivnosti koje se odnose na dve susedne rotorske petlje. Izraz za razvoj pomenute induktivnosti u Furijeov red, dat je sledećom relacijom:

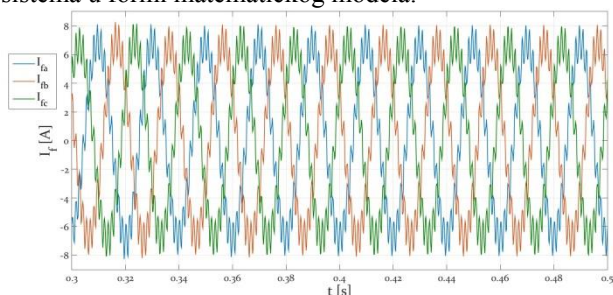
$$M_{rjk}(\theta) = M_{ljk0} + \sum_{n=1}^{\infty} M_{ljk-an} \cdot \cos(n\theta) + \sum_{n=1}^{\infty} M_{ljk-bn} \cdot \sin(n\theta) \quad (11)$$



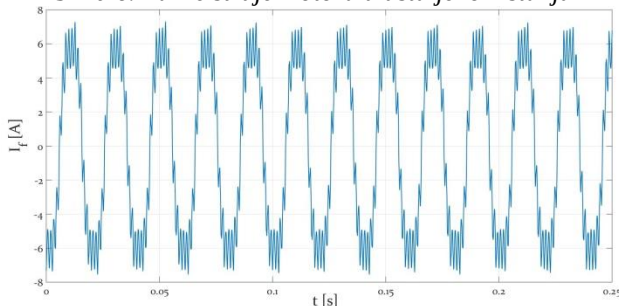
Slika 5. Međusobna induktivnost rotorskih petlji 1 i 2

4. MATLAB SIMULACIJE NA RAZVIJENOM MODELU MAŠINE

Dinamički matematički model trofazne kavezne asinhronne mašine realizovan je u programskom paketu MATLAB Simulink koji omogućava da se putem najrazličitijih računarskih simulacija istraže sve relevantne pojave i fenomeni koji se odvijaju unutar mašine kao sredstva elektromehaničke konverzije energije. Simulacije služe za pribavljanje podataka i informacija o različitim pojavama koje se odvijaju unutar električne mašine bez stvarnih događanja, odnosno, na primeru logičke zamene realnog sistema u formi matematičkog modela.



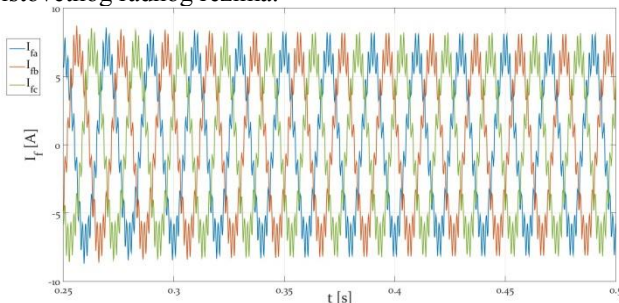
Slika 6. Fazne struje motora u ustaljenom stanju



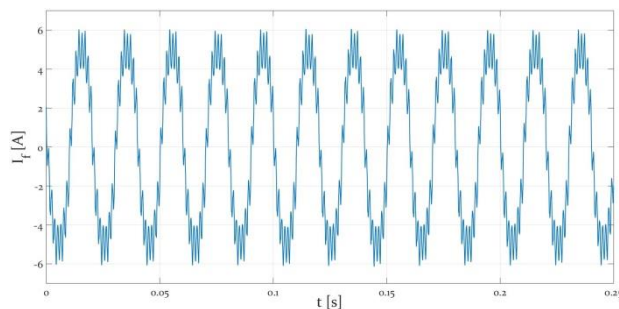
Slika 7. Eksperimentalni odziv fazne struje motora

Na slici 6 prikazan je talasni oblik fazne struje motora u režimu praznog hoda tokom ustaljenog, normalnog pogonskog stanja mašine. Ono što je karakteristično za složenoperiodične talasne oblike prikazanih struja jeste i primetno prisustvo viših harmonika. Oni su posledica uvažavanja konstrukcionih osobenosti mašine koji su na detaljan način uvaženi u matrici induktivnosti. Na slici 7 prikazan je eksperimentalno dobijen talasni oblik fazne struje tokom istovetnog radnog režima.

Na slici 8 prikazan je talasni oblik fazne struje motora u režimu praznog hoda u ustaljenom stanju u slučaju kvara kaveznog rotora. U složenoperiodičnom obliku struja se jasno uočavaju posledice električne neuravnoteženosti mašine nastale usled kvara. Slika 9 prikazuje eksperimentalno dobijen talasni oblik fazne struje tokom istovetnog radnog režima.



Slika 8. Fazne struje motora u slučaju kvara rotora



Slika 9. Eksperimentalni odziv fazne struje motora u slučaju kvara rotora

5. ZAKLJUČAK

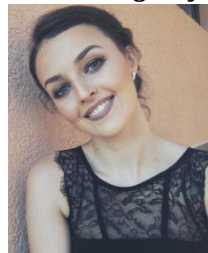
U ovom radu je razvijen dinamički matematički model trofazne kavezne asinhronne mašine čiji se ulazni parametri dobijaju primenom metode konačnih elemenata. Model uvažava uticaj postojanja žljebova na statoru i rotoru mašine. Razvijeni model je na više primera upoređen sa eksperimentalno dobijenim odzivima, pri čemu su uočena veoma zadovoljavajuća poklapanja rezultata.

Model omogućava simulacije različitih scenarija koji se mogu javiti tokom eksploatacije mašine, uključujući i pojavu različitih vrsta kvarova.

6. LITERATURA

- [1] Slobodan N. Vukosavić, „Električne mašine“, Univerzitet u Beogradu, Elektrotehnički fakultet, „Akademska misao“, 2010.
- [2] Dejan Reljić, „Otkrivanje kvara rotora kaveznog asinhronog motora primenom tehnika analize terminalnih veličina“, doktorska disertacija, Fakultet tehničkih nauka, Novi Sad, 2017.
- [3] Vladan Vučković, „Opšta teorija električnih mašina“, IP „Nauka“, 2009.
- [4] Dejan Jerkan, „Dinamički model trofazne kavezne asinhronne mašine zasnovan na metodi konačnih elemenata“, doktorska disertacija, Fakultet tehničkih nauka, Novi Sad, 2016.
- [5] <http://www.femm.info/wiki/HomePage>

Kratka biografija:



Ivana Čirić rođena je u Novom Sadu 1994. god. Diplomirala je na Fakultetu tehničkih nauka u Novom Sadu 2018. godine. Master studije iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine je upisala školske 2018./19. godine.



Dejan Jerkan je docent na Fakultetu tehničkih nauka u Novom Sadu, na Katedri za Energetsku elektroniku i pretvarače. Oblast interesovanja su mu modelovanje i dijagnostika električnih mašina, kao i metoda konačnih elemenata.

OPTIMIZACIJA RADA WPF APLIKACIJA UPOTREBOM VIRTUALIZACIJE OPTIMIZATION OF WPF APPLICATIONS USING VIRTUALIZATION METHODS

Jovan Topolić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu predstavljene su metode koje je moguće primeniti u WPF aplikacijama radi poboljšanja njihovih performansi. Akcenat rada je stavljen na optimizacione metode virtualizacije, koje poboljšavaju performanse prilikom rada sa velikim setovima podataka.. Postoje dve vrste virtualizacije: UI virtualizacija i virtualizacija podataka. Prikazan je način postizanja najboljih performansi aplikacije primenom ovih optimizacionih metoda.

Ključne reči: Optimizacija, virtualizacija, velike kolekcije podataka, WPF

Abstract – This paper presents the optimization methods that can be utilised in WPF applications for improving application performance. Focus of this paper is on optimization methods called virtualization, which help improve performance when dealing with large data sets. There are two types of virtualization: UI virtualization and Data virtualization. It shows the way of achieving best application performance when using these optimizations methods.

Keywords: Optimization, virtualization, large data collections, WPF

1. UVOD

WPF (Windows Presentation Foundation) je UI (User Interface) radni okvir (framework) koji se koristi za izradu desktop aplikacija, a razvijen od strane Microsoft-a. Prilikom razvoja WPF aplikacije potrebno je poseban osvrt napraviti na način na koji se aplikacija implementira, to jest potrebno je razmišljati o mogućim načinima optimizacije koji bi doveli do boljeg iskorišćenja resursa i samim tim i boljeg rada aplikacije. Postoji veliki broj različitih načina optimizacije, a ovaj rad najveći akcenat stavlja na vid optimizacije koji se naziva virtualizacija.

Postoje dva tipa virtualizacije kada se govori o WPF aplikacijama, a to su UI virtualizacija i virtualizacija podataka (*data virtualization*). Oba ova vida virtualizacije se odnose na situacije kada je potrebno prikazati veliki set podataka u okviru korisničkog interfejsa.

Osnovna ideja je učitavanje samo onih podataka koji će biti vidljivi na ekranu i tako smanjiti negativan uticaj na performanse učitavanjem celokupne kolekcije podataka.

NAPOMENA:

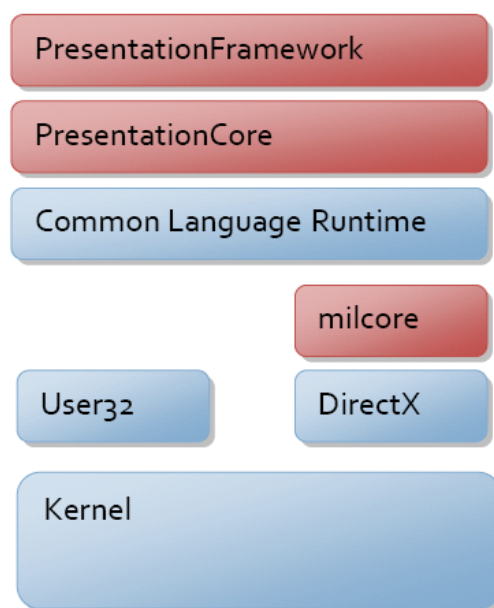
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Čapko, vanr. prof.

2. TEORIJSKE OSNOVE

2.1. WPF

WPF (Windows Presentation Foundation) je Microsoft-ova tehnologija za kreiranje grafičkih korisničkih interfejsa. Moguće je kreirati širok spektar aplikacija, od onih koje koriste bazične forme pa sve do bogatih 3D okruženja. On je takođe i osnova Silverlight-a koji je proširio WPF tehnologiju na Web i na uređaje kao što su Windows mobilni telefoni [1].

Arhitektura WPF-a je prikazana na slici 1 gde crvene sekcije na dijagramu predstavljaju WPF komponente, dok plave sekcije predstavljaju Windows komponente. Sav prikaz u WPF-u se vrši pomoću *DirectX engine*-a, koji omogućava efikasno hardversko i softversko iscrtaivanje.



Slika 1. WPF arhitektura [2]

Korišćenje WPF-a ima mnoge prednosti a nabrojane su najvažnije [1]:

- Nezavisnost od rezolucije ekrana – pod ovim se misli na to da će korisnički interfejs izgledati bolje čak i na ekranima sa niskom rezolucijom od korisničkog interfejsa razvijenog u *Windows Forms applications*, a to je zbog toga što WPF koristi *DirectX* komponente.
- Kontrola unutar kontrole – WPF omogućava da se definiše kontrola kao sadržaj neke druge kontrole kao što je recimo *Button*.
- Dostupnost različitih *layout*-ova – *layout* se koristi da se logički razgraniče kontrole na

korisničkom interfejsu i da se uredno prikažu na prozoru. U WPF-u neke od *layout* kontrole su: *StackPanel*, *WrapPanel*, *DockPanel*, *Grid*, *Canvas*.

- 2D, 3D grafika i animacije – u WPF-u se mogu koristiti animacije, multimedijalne datoteke i grafika. Podržano je puštanje video ili audio datoteka.
- *Data binding* – prednost WPF-a u odnosu na *Windows Forms applications* je ta što se ne mora voditi računa ručno o sinhronizaciji podataka između izvora podataka i UI elementa.

2.2. XAML

XAML (*Extensible Application Markup Language*) je dizajniran kao efikasan jezik za razvoj korisničkog interfejsa aplikacije [1]. Specifična prednost koju XAML donosi WPF aplikacijama je to što je on u potpunosti deklarativan jezik, time dopuštajući *developer*-u ili dizajneru da opiše ponašanje i integraciju komponenti bez korišćenja proceduralnog programiranja. XAML se ekstenzivno koristi u .NET Framework verzijama 3.X i 4.X, posebno u WPF-u. XAML formira *markup* jezik korisničkog interfejsa za definisanje UI elemenata, *data binding*-a, događaja i ostalih mogućnosti.

Sve što se kreira i implementira u XAML jeziku takođe je moguće izraziti koristeći tradicionalnije .NET jezike, kao što je C#. Međutim, ključni aspekt XAML tehnologije je smanjena kompleksnost koja je potrebna alatima da procesira XAML, jer je baziran na XML jeziku [1]. Postoje četiri glavne implementacije XAML tehnologije :

- Verzija namenjena za WPF, koja se koristi od .NET Framework 3.0 i nadalje.
- Silverlight 3 verzija.
- Silverlight 4 verzija.
- Windows 8 XAML/Jupiter verzija.

3. UI I VIRTUALIZACIJA PODATAKA

Virtualizacija je centralna tema ovog rada, te je potrebno detaljno opisati koncept na kome se ona zasniva. Kao što je napomenuto u uvodu razlikujemo dve vrste virtualizacije: virtualizacija korisničkog interfejsa (UI virtualizacija) i virtualizacija podataka. U oba slučaja radi se o optimizaciji prilikom rada sa velikim kolekcijama podataka tako što se vrši učitavanje samo onih podataka koji su vidljivi na ekranu.

3.1 UI virtualizacija

WPF kontrole kao što su *ListView* i *ComboBox* se koriste da prikažu listu podataka u aplikaciji. Ukoliko je lista podataka velika, može doći do pada performansi aplikacije. Ovo se dešava zbog toga što standardni *layout* sistem kreira *layout* kontejner za svaki podatak koji je asociran sa UI kontrolom [3], i računa njegovu *layout* veličinu i poziciju. Tipično nije potrebno prikazivati sve podatke u isto vreme, već se prikazuje podskup podataka a korisnik se najčešće kroz *scroll* mehanizam kreće kroz listu.

U ovakvim slučajevima ima smisla koristiti UI virtualizaciju, što znači da će se generisanje i komputacije za kontejner podataka (*item container*) odložiti dok podatak ne postane vidljiv na ekranu.

Container recycling je optimizacija dodata UI virtualizaciji u .NET Framework verziji 3.5 za kontrole koje nasleđuju *ItemsControl*, takođe ova optimizacija može poboljšati performanse kod *scroll* mehanizma. Kada se *ItemsControl* koja koristi UI virtualizaciju puni podacima, ona kreira *item* kontejner za svaki podatak koji se skroluje u prikaz i uništava *item* kontejner za svaki item koji se skroluje van prikaza [3]. *Container recycling* omogućava UI kontrolama da ponovo iskoriste postojeće kontejnere za različite podatke, time se *item* kontejneri ne kreiraju i ne uništavaju konstantno prilikom skrolovanja. Ova metoda optimizacije se omogućava postavljanjem vrednosti *attached property*-a *VirtualizationMode* na *Recycling*. *Deferred scrolling* (odloženo skrolovanje) predstavlja odloženo učitavanje podataka prilikom skrolovanja. Po *default*-u, kada korisnik vuče *thumb* na *scrollbar*-u, prikaz sadržaja se kontinualno ažurira. Ako je skrolovanje sporo u UI kontroli, može se iskoristiti *deferred scrolling*. Ovakvim pristupom sadržaj se ažurira samo kada korisnik otpusti *thumb*. Da bi se omogućio ovaj tip skrolovanja potrebno je podesiti *property IsDeferredScrolling* na *true*.

.NET Framework verzija 4.5 je uvela značajne promene koje olakšavaju rad sa velikim setovima podataka kao i poboljšane opcije prilikom *scroll*-a dok je virtualizacija uključena. Jedna od novina je da je od ove verzije omogućena virtualizacija prilikom grupisanja podataka (*grouping*) i to pomoću *IsVirtualizingWhenGrouping attached property*-a. U prošlim verzijama je takođe bio prisutan problem kod skrolovanja u smislu da ukoliko je bilo potrebno da se virtualizacija koristi, moralo se koristiti i logičko skrolovanje koje se podešava postavljanjem *CanContentScroll property*-a na *true*. Ovo je bilo problematično u slučaju kada se u *ListView*-u kao podaci nalaze ekspanteri koji prilikom ekspanovanja zauzimaju veliki vertikalni prostor prikaza. U ovakvom slučaju dolazi do naglih skokova između elemenata što korisniku predstavlja loše iskustvo prilikom korišćenja aplikacije. U .NET Framework verziji 4.5 ovaj problem je prevaziđen uvođenjem *ScrollUnit attached property*-a koji je moguće koristiti sa *Virtualizing* panelom. *ScrollUnit* ima dve moguće vrednosti: *Item(default)* koji suštinski predstavlja logičko skrolovanje element po element i *Pixel* koji nam upravo omogućava skrolovanje zasnovano na pikselima bez isključivanja virtualizacije.

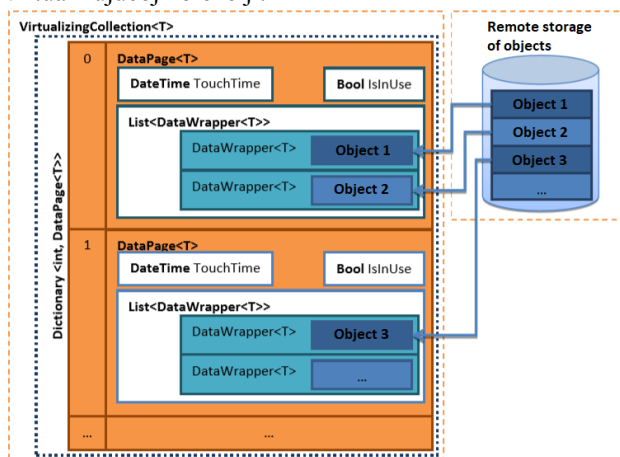
3.2 Virtualizacija podataka

Ukoliko se povećava broj podataka u listi, performanse će sve više opadati zbog toga što aplikaciji postaje teže da manipuliše podacima, sve je više memorije potrebno za njihovo skladištenje i mnogo vremena za učitavanje podataka. Ovaj problem se rešava upotrebom virtualizacije podataka.

Virtualizacija podataka je mehanizam koji omogućava aplikaciji da ne skladišti sve podatke odjednom u memoriji, već samo neophodan deo. Na ovaj način količina memorije koja se koristi za skladištenje podataka u klijentskoj aplikaciji ostaje mala i konstantna za bilo koji broj podataka u kolekciji. WPF je dizajniran za pravljenje korisničkog interfejsa i nije vezan za metode skladištenja i procesovanja podataka, te ne postoji ugrađena podrška za virtualizaciju podataka unutar WPF-a. Stoga je potrebno osmisliti način implementacije virtualizacije podataka. Virtualizacija podataka se vrši

tako što se kreira specijalna kolekcija objekata, koja sadrži samo deo elemenata i učitava objekte iz izvora podataka samo kada oni postanu potrebni UI kontroli. Ovaj mehanizam se može implementirati na različite načine, virtualizacioni mehanizmi se razlikuju u načinu učitavanja podataka. Ideja je da se u memoriji čuva samo deo podataka koji je vidljiv korisniku ali se interakcija nad podacima ponaša kao da su svi podaci učitani. *Scroll bar* indikator će uvek pokazivati poziciju prikazanih podataka u punoj kolekciji elemenata, što dalje omogućava korisniku da može da se pozicionira u bilo kojem delu pune kolekcije. Kolekcija koja virtualizuje podatke treba da simulira skladištenje svih elemenata, na primer kada UI kontrola zahteva broj podataka u listi, kolekcija treba da vrati broj objekata u punoj kolekciji [4].

Preporučuje se da virtualizaciona kolekcija koja se dizajnira koristi *IList<T>* interfejs kao bazni. *IList* je interfejs kolekcija objekata kojima se pristupa pomoću indeksa, jer pruža najbolje performanse prilikom *data binding*-a podataka na WPF kontrole. Javlja se pitanje skladištenja podataka u virtualnoj kolekciji. Izvor podataka se deli na stranice (*pages*), fragmente fiksirane veličine. Kada eksterni kod traži element kolekcije, učitaće se stranica na kojoj je lociran i vratiće se željeni element. Ove akcije se izvršavaju pomoću indeksa kolekcije. Osim učitavanja stranica, takođe se moraju i brisati one koje se ne koriste više od strane aplikacije. Da bi se ovo postiglo, svaka stranica bi imala informaciju o poslednjem pristupu. U određenom trenutku, proteklo vreme od poslednjeg pristupa stranici bi se računalo. Ukoliko bi ovo proteklo vreme premašilo neki zadati prag, stranica bi se obrisala iz kolekcije [4]. Na slici 2 prikazana je šema za skladištenje podataka u virtualizujućoj kolekciji.



Slika 2. Šema za skladištenje podataka u virtualizujućoj kolekciji [4]

4. OPIS PROBLEMA

Problem koji se javlja kao tema ovoga rada je kako optimizovati UI kontrolu koja služi za prikaz kolekcije podataka u WPF-u da bi negativan uticaj na performanse bio što manji. Potrebno je iskoristiti opcije UI virtualizacije na najbolji način, kao i pružiti sopstvenu implementaciju mehanizma virtualizacije podataka, jer u WPF-u ne postoji ugrađena podrška za nju. U današnjem vremenu sve veći akcenat se stavlja na brzinu rada aplikacija te je ovaj problem veoma aktuelan.

5. REŠENJE PROBLEMA

Za rešenje prethodno opisanog problema napravljena je test aplikacija u WPF-u i C# programskom jeziku koja će na primeru *ListView* kontrole prikazati na koje načine upotrebom virtualizacionih metoda možemo doći do optimalnih performansi same kontrole. Metode koje će biti opisane je moguće primeniti na bilo koju UI kontrolu koja nasleđuje *ItemsControl* te nije strogo vezana za *ListView*. Sama aplikacija se sastoji od virtualizacionih opcija koje je moguće podesiti preko korisničkog interfejsa, kako onih vezani za UI virtualizaciju tako i za virtualizaciju podataka, kao i *ListView* kontrole koja se generiše u zavisnosti od odabranih opcija. Takođe, najbitnije informacije koje dobijamo upotrebom aplikacije su potrošnja memorije u megabajtima kao i vreme koje je bilo potrebno prilikom generisanja *ListView* kontrole kao i prilikom skrolovanja kroz nju.

5.1 Implementacija UI virtualizacije

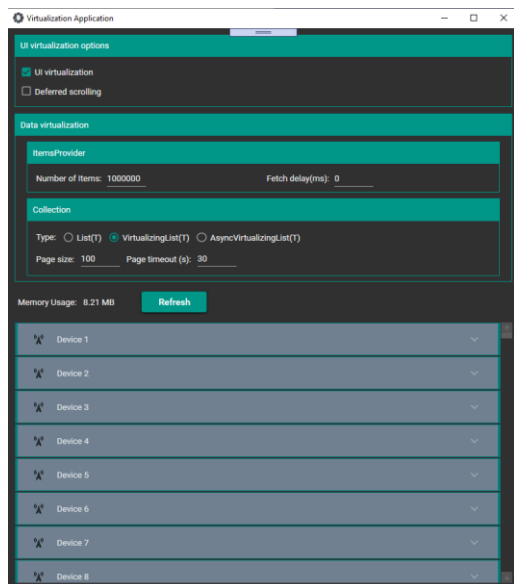
Potrebno je napomenuti da od UI virtualizacije nije podržano menjanje *ContainerRecycling* svojstva *ListView* kontrole, zbog toga što se pomenuti *property* ne može menjati nakon *Measure* poziva za sam *ListView*. Ukoliko je potrebna promena vrednosti za *ContainerRecycling* mora se definisati u XAML-u jer nije podržana njena promena u *runtime*-u. Dakle od UI virtualizacije moguće je konfigurisati da li je UI virtualizacija omogućena kao i da li je omogućen *DeferredScrolling* (odloženi *scroll*), koji značajno može pomoći brzini skrolovanja ukoliko nam nije potrebno kontinualno ažuriranje sadržaja prilikom samog skrolovanja. Podrška za UI virtualizaciju je ugrađena u WPF te nema značajnog napora za njeno omogućavanje. Performanse su najbolje kada je uključena UI virtualizacija a *ContainerRecycling* postavljen na *Recycling* vrednost, dok *DeferredScrolling* može pomoći performansama *scroll*-a ukoliko nam nije potrebno kontinualno ažuriranje sadržaja.

5.2 Implementacija virtualizacije podataka

Što se tiče ponuđenih opcija za virtualizaciju podataka prvo što možemo definisati je broj podataka koji će se nalaziti u izvoru podataka, što veći broj podataka to će biti vidljiviji benefiti samih mehanizama virtualizacije. *Fetch delay* poljem možemo simulirati vreme potrebno za dobavljanje podataka u milisekundama, kao recimo iz baze podataka. Proces dobavljanja podataka iz nekog *remote* izvora može biti vremenski zahtevan te ovde dolazi do izražaja mehanizam virtualizacije podataka koji limitira broj podataka koje je potrebno dobiti i instancirati u memoriji. Dalje opcije se tiču samog izbora kolekcije koji će se vezati za UI kontrolu. Odabirom *List<T>* opcije virtualizacija podataka se ne koristi te će se ceo izvor podataka instancirati odjednom u listi. Opcije - *VirtualizingCollection<T>* i

- *AsyncVirtualizingCollection<T>*

predstavljaju kolekcije koje podržavaju virtualizaciju podataka. Poslednje dve konfigurabilne opcije se tiču definisanja veličine stranice i vremena isteka stranice za virtualizovanu kolekciju. Na slici 3 prikazan je korisnički interfejs test aplikacije.



Slika 3. Korisnički interfejs test aplikacije

Da bi se iskoristilo rešenje za virtualizaciju podataka, izvor podataka mora biti u mogućnosti da pruži ukupan broj podataka unutar kolekcije, i mora biti u mogućnosti da pruži fragmente (stranice) od cele kolekcije. Ovi zahtevi su enkapsulirani u *IItemsProvider* interfejsu. *VirtualizingCollection<T>* je *IList* implementacija koja vrši virtualizaciju podataka. Ona deli čitavu kolekciju u određeni broj stranica. Stranice se zatim učitavaju u memoriji po potrebi, i otpuštaju kada više nisu potrebne. *AsyncVirtualizingCollection<T>* klasa je izvedena iz *VirtualizingCollection<T>* klase, i *override*-uje *Load* metode da bi se učitavanje podataka vršilo asinhrono. Ključna stvar kod asinhronog izvora podataka u WPF-u je ta da se UI mora obavestiti pomoću *data binding*-a kada se podaci dobavljaju. Kod regularnog objekta, ovo se postiže sa *INotifyPropertyChanged* interfejsom. Kod implementacija kolekcije je neophodno koristiti *INotifyCollectionChanged*. Stranice se skladište u *Dictionary*, gde se indeks stranice koristi kao ključ. Još jedan *Dictionary* se koristi da skladišti podatke o poslednjem pristupu stranicama.

5.3 Rezultati testiranja

Prilikom izvršavanja generisanja *ListView* pomoću opisane aplikacije, a pri korišćenju različitih virtualizacionih opcija dobijamo rezultate prikazane na tabeli 1.

Tabela 1. Rezultati testiranja

UI virt.	Br. Podataka [10 ⁶]	Tip kolekcije	Veličina stranice [br.el]	Istek stranice [s]	Potrošnja memorije [MB]
NE	0.02	List	/	/	1586
DA	1	List	/	/	2175
DA	1	Virt.List	100	30	12
DA	1	AsyncVirt.List	100	30	10

Primećuje se da upotrebom kombinacije UI virtualizacije i virtualizacije podataka se dolazi do najboljih rezultata potrošnje memorije, te da korišćenje virtualizacije

uopšteno može drastično poboljšati performanse aplikacije.

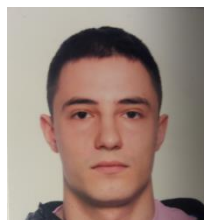
6. ZAKLJUČAK

Opisana je test aplikacija kojom se meri potrošnja memorije prilikom korišćenja raznih opcija UI virtualizacije i virtualizacije podataka, kao i vreme potrebno za generisanje *ListView* kontrole i vreme potrebno da se prođe kroz celu listu pomoću *scroll* mehanizma, te je izvršena analiza koje to opcije dovode do najboljih performansi aplikacije. Sama aplikacija nema neku realnu primenu već samo služi kao test scenario za pokazivanje performansi prilikom korišćenja različitih opcija. Ono što je primenjivo je sama implementacija mehanizma virtualizacije podataka koja može značajno pomoći kada aplikaciji nije potrebno da učita celu kolekciju podataka odjednom. Takođe ima i dosta prostora da se sam mehanizam virtualizacije podatka unapredi. Rešenje koje je prezentovano smatra da je izvor podataka *read only* i da se ne menja. Moguće je proširiti rešenje tako da periodično ili na zahtev ponovo učitava *Count* i stranice. Postoji prostor za proširenje funkcionalnosti i u *IItemsProvider* interfejsu u smislu da podržava na primer izmene i sortiranje.

7. LITERATURA

- [1] A.Nathan, "WPF 4.5 Unleashed", Sams Publishing, August 2013.
- [2] <https://docs.microsoft.com/en-us/dotnet/framework/wpf/advanced/wpf-architecture> (pristupljeno u novembru 2019.)
- [3] S.Yuen, "Mastering Windows Presentation Foundation", Packt Publishing, February 2017.
- [4] A.S. Chajkovskaja, K. Ya. Kudrjautsev, E. K. Pavlov, "Displaying large sets of dynamically changing remote data in WPF applications", *International Journal of Open Information Technologies*, vol. 5, no. 9, pp.13-20, 2017.

Kratka biografija:



Jovan Topolić rođen je 03.09. 1995. godine u Novom Sadu. Završio je Osnovnu školu "Đorđe Natošević" u Novom Sadu 2010. god. Gimnaziju "Svetozar Marković" završio je 2014. godine i iste upisao Fakultet tehničkih nauka, odsek Računarstvo i automatika. Školske 2016/2017. godine upisao je smer Primenjene računarske nauke i informatika, a u osmom semestru se opredeljuje za modul Internet i elektronsko poslovanje. Nakon toga upisuje master studije školske 2018/2019. godine na Fakultetu tehničkih nauka, smer Primenjeno softversko inženjerstvo, na odseku Računarske nauke i informatike.

**ДИНАМИЧКА ПРОМЈЕНА ПРОФИЛНЕ КОНФИГУРАЦИЈЕ У СИСТЕМУ ЗА
УПРАВЉАЊЕ ПАМЕТНИМ БРОЈИЛИМА****DYNAMIC CHANGE OF PROFILE CONFIGURATION IN SMART METER
MANAGEMENT SYSTEM**

Марина Бобар, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – Овај рад говори о изазовима обраде велике количине догађаја генерисаних на паметним бројилима приликом екстремних временских услова. Описане су предности паметних бројила као и сервиси који су од значаја за пренос информација са паметних бројила до система за управљање паметним бројилима. У раду је предложено софтверско рјешење за унапређење обраде догађаја.

Кључне речи: Паметна мрежа, Напредна мјерна инфраструктура, Паметна бројила, Систем за управљање паметним бројилима

Abstract – This paper addresses the challenges of processing the large amount of events generated on smart meters under extreme weather conditions. The advantages of smart meters and the services that are important for transferring information from smart meters to the smart meter management system are described. This paper proposes a software solution for enhancing event processing.

Keywords: Smart Grid, Advanced Metering Infrastructure, Smart Meter, Smart Meter Management System

1. УВОД

За несметано функционисање индустрије, транспортних, комуникационих, рачунарских система и других инфраструктура неопходно је непрекидно напајање електричном енергијом. Са развојем технологије повећава се и зависност модерног друштва од електричне енергије у свим аспектима живота. Због тога постоји потреба за ефикасним управљањем напајањем и снабдијевањем потрошача висококвалитетном и поузданом електричном енергијом. Прекиди у напајању електричном енергијом имају негативан утицај на квалитет живота, те имају значајне директне или индиректне економске последице. Пружање непрекидног, поузданог и квалитетног напајања електричном енергијом је комплексан задатак и у нормалним условима.

Током последње деценије дошло је до пораста броја елементарних непогода. Глобално загријавање се рефлектује повећаном фреквенцијом великих поплава, урагана, олуја. У зависности од јачине олује и количине начињене штете, прекиди у напајању могу

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Дарко Чапко, ванр. проф.

трајати неколико сати или чак неколико дана. Са дужином трајања прекида повећавају се и економски губици што даје већи значај брзој рестаурацији напајања електричном енергијом.

Паметна бројила помажу у смањењу ризика прије, током и након катастрофе истовремено постављајући темеље за снажнију, отпорнију енергетску будућност. Правилна и брза детекција испада је сложен процес и зависи од великог броја фактора. Један дио тог процеса представља начин на који се догађаји обрађују у систему за управљање паметним бројилима, а то је управо фокус овог мастер рада. У оквиру тог система постоји профилна конфигурација која дефинише на који начин и који догађаји ће се обрађивати. Тематика овог рада јесте реализација софтверског рјешења које ће омогућити аутоматску динамичку промјену профилне конфигурације.

2. ТЕОРИЈСКЕ ОСНОВЕ

Електроенергетски систем је систем у оквиру ког се изводе све електроенергетске трансформације из или у електричну енергију, њен пренос, дистрибуција и непосредна потрошња [1]. Електроенергетска мрежа представља мрежу струјних генератора, далеководна, трансформатора и дистрибутивних система, а чији је циљ квалитетно снабдијевање крајњих корисника потребном количином електричне енергије уз минималне трошкове.

Паметна мрежа је модерна електроенергетска мрежа која пружа побољшану ефикасност, поузданост и сигурност, уз интеграцију обновљивих и алтернативних извора енергије, кроз аутоматизовану контролу и модерне комуникационе технологије [2].

2.1. Паметна бројила

Паметна бројила представљају последњи корак у еволуцији бројила. Повезана су са центром дистрибутивне компаније путем двосмјерне комуникационе везе преко које се могу преносити измјерене вриједности као и контролне поруке. Поред мјерења протока енергије, омогућавају размјењивање информација о потрошњи енергије између дистрибуција и потрошача [3]. Паметна бројила осим прикупљања и слања података омогућавају примање и извршавање команди даљинским путем (као што су укључење и искључење уређаја), као и аутоматско слање даљинских читавања, али такође и слања одговора на захтјев који пристиже са одговарајућег сервиса.

2.2. Напредна мјерна инфраструктура

Напредна мјерна инфраструктура (енг. *Advanced Metering Infrastructure* – AMI) је један од главних аспеката паметне мреже. Састоји се од напредних бројила или паметних бројила који прате, контролишу потрошњу енергије, и комуницирају у циљу оптимизације потрошње електричне енергије, као и система за управљање и складиштење података.

AMI је заправо систем који мјери, прикупља и анализира потрошњу енергије од напредних уређаја као што су паметна бројила, преко различитих комуникационих медија, на захтјев или периодично на основу унапред дефинисаног распореда. Омогућава аутоматско прикупљање велике количине корисничких и мрежних података на сигуран, поуздан и јефтин начин [4].

2.3. Систем за управљање паметним бројилима

Систем за управљање дистрибутивном мрежом надгледа сваку компоненту и извршава операције користећи динамичку процјену стања у реалном времену. Систем за управљање паметним бројилима (енг. *Smart Meter Management System* – SMMS) представља један од подсистема у оквиру система за управљање дистрибуцијом електричне енергије (енг. *Distribution Management Service* – DMS). С обзиром на то да паметна бројила генеришу велики број догађаја које је често тешко обрадити, SMMS има улогу филтрирања тих догађаја и смањења оптерећења осталих сервиса у оквиру DMS-а.

Процјена валидности догађаја обухвата провјеру исправности бројила са ког је догађај стигао, структуре поруке, старости догађаја и профилну валидацију. Довољно је да догађај не испуњава једно од правила да би се сматрао неисправним. Остали сервиси у оквиру DMS система се претплаћују на типове податка који су им од интереса. Приликом промјена SMMS обавјештава претплаћене сервисе о насталој промјени. То омогућава осталим сервисима као и клијентским апликацијама увид у промјену чим промјена настане. Сервиси добијају само информације на које су се преплатили што омогућава избегавање непотребног оптерећивања сервиса.

2.4. Догађаји у оквиру паметне мреже

Догађаји представљају информације које потичу са паметног бројила и индикатори су аномалија у мрежи. Састоји се од неколико атрибута, укључујући информације о извору, односно уређају са ког потиче, уређају који је одговоран за детекцију и комуникацију, врсти и типу догађаја. Врсте догађаја су: аларми, информације, грешке и упозорења. Тип догађаја даје информацију о томе на који процес се догађај односи.

2.5. Профилна конфигурација

Профил се може додијелити цијелој мрежи, региону, једној трансформаторској области или фидеру. Фидер је најмања јединица којој профил може бити додијељен. Фидер „пролази“ поред индивидуалних потрошача који се на њега прикључују, умјесто да се за сваки од њих успоставља сопствена електро-енергетска веза са сабирницама.

Профили у оквиру система за управљање паметним бројилима дефинишу начин на који ће се догађаји обрађивати и филтрирати. Профили се могу креирати и додјељивати одређеним дијеловима мреже. Поред тога, омогућена је и промјена већ постојећих профила, како њихове конфигурације тако и могућности додјељивања другим дијеловима мреже. Профилна конфигурација дефинише који типови догађаја ће се обрађивати, а који занемарити, као и временски период чекања на догађај истог или супротног типа са истог паметног бројила.

3. ФОРМУЛАЦИЈА ПРОБЛЕМА

Са повећањем оптерећења, повећава се и број информација које пристижу са паметних бројила што може узроковати велико кашњење у комуникацији као и губитак података и довести до вишеструких грешака и погрешне координације заштите.

Профилна конфигурација у оквиру SMMS система има два подразумевана профила додијељена дијеловима мреже у зависности од временских услова, нормални и олујни. Нормални профил је подразумевани профил који је додијељен цијелој мрежи, чији се конфигурабилни параметри могу подешавати ручно од стране корисника. Конвенционално рјешење је имплементирано тако да када наступе временске непогоде, корисник ручно из клијентске апликације додјељује олујни профил дијелу мреже који је под утицајем невремена. Постављање активног профила зависи од корисника који управља клијентском апликацијом што може довести до људских грешака и има утицај на правовремену реакцију која може бити кључна у екстремним временским условима.

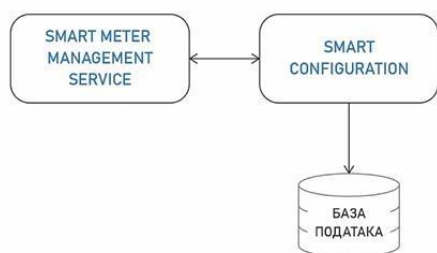
Сви дијелови мреже којима је додијељен олујни профил имају исте параметре по којима се обрађују догађаји пристигли са паметних бројила која се налазе у тим дијеловима. Међутим, поставља се питање да ли је то добра пракса с обзиром на то да различити фидери могу имати различите карактеристике. Фидери погођени олујом не морају бити на истој удаљености од система за управљање догађајима што доводи до тога да догађаји са различитих фидера могу имати различита времена кашњења. Такође, неки фидери имају већи значај од других, док неки генеришу много већи број догађаја од других у зависности од тежине испада.

Када се све претходно наведено узме у обзир долазимо до закључка да би једно од унапређења тренутног рјешења могло бити аутоматско креирање профила засебно за сваки фидер или за групу фидера са истим карактеристикама, које не зависи од корисника апликације. Сходно томе, једно од рјешења би било аутоматизовати систем тако да сам дође до сазнања да је дошло до временских неприлика на основу броја догађаја и времена кашњења, те сам генерише адекватан профил за фидер. На тај начин ће се догађаји са фидера обрађивати на њима одговарајући начин, односно у зависности од њихове фреквенности и кашњења.

Циљ овог рјешења јесте повећање исправности обраде података, смањење оптерећења сервиса који користе обрађене догађаје, као и смањење губитака и грешака приликом обраде. Овакво софтверско рјешење треба да обезбиједи правовремену реакцију на олују и смањи зависност обраде догађаја од корисника апликације.

4. ПРЕДЛОЖЕНО РЈЕШЕЊЕ

Софтверско рјешење описано у овом поглављу је реализовано помоћу *Microsoft Visual Studio 2015* развојног окружења у *C#* програмском језику. Основни циљ *SmartConfiguration* апликације јесте да на основу података пристиглих од система на који је повезана, односно SMMS-а, направи одговарајуће промјене које ће утицати на унапређење обрађивања догађаја са паметних бројила те самим тим довести до бољих перформанси система.



Слика 1. Архитектура предложеног рјешења

На Слици 1 приказана је архитектура рјешења. Комуникација између апликације и SMMS је двосмјерна, гдје систем на сваких 60 секунди шаље информације које су од значаја за рад апликације. У обрнутом смјеру, апликација позива изложене методе SMMS-а које омогућавају манипулације над профилима и фидерима. Приликом креирања или ажурирања профила све настале промјене се смјештају у базу података, тако да увијек постоји могућност увида у постојеће профиле и њихове конфигурабилне параметре.

4.1. Опис рада апликације

Главна намјена апликације јесте да на детектовано „ненормално понашање” паметних бројила (када шаљу велики број догађаја) реагује динамичком промјеном профилне конфигурације у оквиру SMMS-а. Слједи анализа примљених информација и уколико се детектује да је на фидеру дошло до проблема, приступа се бази података и провјерава да ли за тај фидер већ постоји профил. Уколико не постоји, креира се нови профил. У профили се између осталог подешава и који типови догађаја су дозвољени/забрањени чиме се дефинише њихов приоритет и смањује број догађаја.

Параметар профилне конфигурације који је од највећег значаја за апликацију је *stewing* период. Он представља период у ком се обрада паузира и догађај се привремено ставља у *stewing* табелу. Прије покретања *stewing* периода и смјештања у табелу, провјерава се да ли у њој већ постоји догађај супротног типа за исто паметно бројило (за догађај типа *POWER_DOWN* који представља нестанак напајања, супротан је *POWER_UP* односно повратак напајања). Подразумијевана вриједност *stewing* параметра јесте 30 секунди што значи да

уколико у том периоду не стигну два догађаја са истог бројила супротних типова оба се архивирају и њихова обрада се зауставља.

Проблем настаје приликом загушења комуникационих канала. Кашњење информација у току временских непогода, може узроковати да догађај не стигне на сервис за вријеме *stewing* периода иако је генерисан у том периоду што може довести до обрађивања оба догађаја супротног типа са истог бројила те самим тим повећати број обрађених и проследијених догађаја и оптеретити друге сервисе. Информација о просјечном кашњењу догађаја је значајна јер се то вријеме додаје на подразумевану иницијалну вриједност од 30 секунди. На тај начин се параметар *stewing* динамички мијења у зависности од тога колико догађаји касне.

Са повећањем/смањењем кашњења повећава/смањује се и *stewing* период. Овим се обезбјеђује да фидери са различитим временским кашњењем догађаја имају додијељене профиле са различитим конфигурацијама. Ако фидер већ садржи профил, његов *stewing* параметар се ажурира уколико се просјечно кашњење догађаја са тог фидера промијенило и слиједи позив ка SMMS-у за ажурирање промијењеног профила. Након креирања профила позива се метода на SMMS-у којом се профил додјељује одговарајућем фидеру, а затим слиједи још један позив ка сервису којим се мијења активни режим профила над тим фидером, односно умјесто нормалног новокреирани профил постаје активан. Када апликација установи да су непогоде прошле, након пет минута устаљеног нормалног режима фидеру поново се активира нормалан профил.

Развијањем *SmartConfiguration* апликације је обезбјеђена независност система од корисника и аутоматска реакција на промјене у пољу узроковане временским неприликама.

5. ТЕСТИРАЊЕ

Тестирања су вршена на рачунару (виртуелној машини) са процесором Intel i3-3210 CPU @ 3.20 GHz, и меморијом - 32 GB (18000 MB додијељено виртуелној машини). За потребе тестирања сматра се да је позната топологија мреже и узете су слједеће референтне вриједности гдје очекивано је да имамо 10 милиона паметних бројила у моделу мреже што је еквивалентно броју корисника. Нормалан режим рада представља генерисање 8-12 догађаја у секунди, док се у екстремним условима у просјеку генерише 90-120 догађаја.

Прво су приказани резултати симулације праве олује, односно невремена које траје дужи временски период. Сценарио који је тестиран обухвата симулацију 10 минута нормалног режима рада, након чега слиједи 135 минута олујног, а по завршетку олује још 10 минута нормалног режима рада.

На основу графика са Сlike 2, гдје је зеленом бојом представљено оптерећење процесора када апликација није покренута, а плавом када јесте, долази се до закључка да је просјечно оптерећење процесора мање када је имплементирано рјешење покренуто.



Слика 2– Перформансе SMMS-а без и са покренутом апликацијом

Сљедећи сценарио тестирања укључује симулацију „лажних“ екстремних услова у трајању од 5 минута. Тестирано је и након колико времена од детекције абнормалног понашања би било најбоље да апликација реагује на промјене. Посматрано је реаговање апликације одмах и након 3 и 5 минута од детектовања проблема. Резултати су приказани у Табели 2. гдје параметар τ представља вријеме протекло од детекције невремена до реаговања на промјене.

Табела 1. Оптерећење процесора од стране SMMS система без покретања апликације као и за различите периоде реаговања.

τ [min]	avg CPU [%]
Без апликације	15,25
0	13,73
3	13,32
5	18,10

Као што се може видјети у Табели 1 разлике у оптерећењу процесора, са и без утицаја апликације на SMMS су мале и нису од великог значаја. Различити временски периоди дјеловања апликације након детекције проблема такође немају већи утицај на перформансе те се стога као крајње рјешење узима да апликација дјелује одмах по детекцији јер је за вријеме стварних временских непогода сваки минут од значаја.

6. ЗАКЉУЧАК

На основу тестирања перформанси посматраног система долазимо до закључка да имплементирано рјешење побољшава перформансе рада SMMS-а за вријеме стварних или лажних узбуна, а теоријски и осталих сервиса који користе обрађене догађаје, међутим, тестирање перформанси сервиса којима SMMS прослијеђује обрађене и филтриране податке излази из оквира теме овог рада.

Предложено рјешење се даље може проширити тако да се редукује број позива ка SMMS-у, што захтјева модификацију апликације али и самог система. Тренутно су потребна два позива, први за

креирање/ажурирање профила, а други за постављање активног профила што успорава рад и апликације као и SMMS - а. Креирање генеричке апликације, односно апликације која ће моћи да се повеже и са другим сервисима би био значајан напредак који би омогућио ширу примјену апликације. Још једно од унапређења би свакако било додавање нових стратегија за начин обрађивања података као и филтрирања у циљу повећања тачности и брже детекције испада.

7. ЛИТЕРАТУРА

- [1] др Владимир Ц. Стрезоски, „Основи електроенергетике”, *Факултет техничких наука у Новом Саду*, 2019.
- [2] Patrick McDaniel, Stephen McLaughlin, Security and Privacy Challenges in the Smart Grid, *IEEE Security & Privacy*, vol. 7, no. 3, pp. 75 – 77, 2009.
- [3] Qie Sun, Hailong Li, Zhanyu Ma, Chao Wang, Javier Campillo, Qi Zhang, Fredrik Wallin, Jun Guo, A Comprehensive Review of Smart Energy Meters in Intelligent Energy Networks, *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 464 – 479, 2016.
- [4] Enrico Valigi, Eugenio Di Marino, Networks optimization with advanced meter infrastructure and smart meters, *CIREN – 20th International Conference on Electricity Distribution*, 2009.

Кратка биографија:



Марина Бобар рођена је 12.01.1996. године у Власеници, Република Српска. Основну и средњу школу је завршила у Власеници. Факултет техничких наука уписује 2014. године, а основне студије је завршила 2018. и исте године уписује мастер на смјеру Примењено софтверско инжењерство.

IDENTIFIKACIJA ELEMENTARNIH NEPOGODA U SISTEMU ZA UPRAVLJANJE PAMETNIM BROJILIMA**IDENTIFICATION OF NATURAL DISASTERS IN SMART METER MANAGEMENT SYSTEM**

Ivona Jovanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu, predstavljen je servis za upravljanje podacima sa pametnih brojila i opisano je karakteristično ponašanje pametnih brojila tokom elementarnih nepogoda. Analiziran je značaj napredne mernne infrastrukture prilikom ispada i na koje sve načine sistemi pametnih brojila olakšavaju upravljanje ispadima. Predstavljen je problem velike količine i učestalosti poruka tokom oluje koja je opterećujuća za sistem. U radu je opisana implementacija rešenja za problem detekovanja elementarnih nepogoda na osnovu događaja.

Ključne reči: pametna brojila, elementarna nepogoda, sistem za upravljanje pametnim brojilima, napredna merna infrastruktura.

Abstract – In this paper, we present smart meter management service and characteristic behaviour of smart meters during natural disasters. The importance of advanced metering infrastructure during outages is described and in what ways smart meter systems make it easier to manage outages. The problem of large amount and frequency of data during a storm that is burdensome for the system is presented. The paper describes the implementation of a solution to the event-based elemental disaster detection problem.

Keywords: smart meters, natural disaster, SMMS, AMI.

1. UVOD

Industrija električne energije suočava se sa izazovima velikih razmera. Decenijama je fokus bio na pouzdanoj proizvodnji električne energije, kao i da cena proizvodnje bude minimalna. Infrastruktura stari i suočava se sa sposobnošću mreže da ostane pouzdana dok ekonomski i globalni faktori utiču na nestabilnost cene goriva i električne energije.

Uticaj proizvodnje električne energije na životnu sredinu postao je socijalno i političko pitanje i veliki izazov. Svet je fokusiran na razvoj pametne mreže koja će zameniti postojeće mreže električne energije [1]. Prirodne katastrofe i ekstremni vremenski uslovi, naročito uragani rezultuju velikim ekonomskim, socijalnim i fizičkim poremećajima i pričinjavaju velike neprijatnosti stanovnicima u vidu nestanka struje, vode i komunikacija.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Čapko, vanr. prof.

Veliki broj studija iz različitih perspektiva, bavi se ovim problemom kako bi pronašli efikasan način za poboljšanje otpornosti kritičnih sistema. Ekstremni vremenski uslovi predstavljaju veoma velik izazov elektrodistribucijama. Zahteva se veoma brzo donošenje odluka, neophodan je pravovremeni odgovor na trenutno stanje [2].

Zadatak ovog rada je da opiše glavne principe i funkcionalnosti napredne mernne infrastrukture (eng. *Advanced Metering Infrastructure* - AMI) tokom elementarnih katastrofa. Karakteristično ponašanje pametnih brojila za vreme oluje je takvo da je količina generisanih događaja i do 10 puta veća od uobičajene. Takođe pojačana je učestalost generisanja događaja usled čestih ispada. U radu je opisan problem učestalosti i količine podataka koja je opterećujuća za sistem za upravljanje pametnim brojilima. Kao rešenje predstavljen je sistem koji se bavi detekcijom kritičnih stanja i razmatrani su parametri koje je potrebno uzeti u obzir prilikom identifikacije.

Predstavljeni su i predlozi kako je moguće uticati na sistem kako bi se smanjila mogućnost nepotrebne obrade i gubitka podataka. Detaljno su opisani događaji koje šalju pametna brojila i njihov značaj prilikom lociranja i identifikacije ispada. Opisan je sistem za upravljanje pametnim brojilima (eng. *Smart Meter Management System* - SMMS) koja upravlja pametnim brojilima, prihvata događaje generisane od strane pametnih brojila i obrađuje ih. Za potrebe simulacije sistema tokom oluje implementirana je aplikacija koja simulira slanje podataka sa pametnih brojila. Prikazano je testiranje performansi SMMS servisa koristeći implementirani simulator.

U izolaciji je testirano slanje događaja i predstavljani su dobijeni rezultati ovih testova.

2. TEORIJSKA OSNOVA**2.1. Napredna merna infrastruktura**

Napredna merna infrastruktura je integrisan sistem pametnih brojila, komunikacione mreže i sistema za upravljanje podacima koji omogućavaju dvosmernu komunikaciju između distribucije i korisnika. Sistem omogućava brojne bitne funkcije koje ranije nisu bile moguće ili su se obavljale ručno, kao npr. mogućnost da se daljinski i automatski meri potrošnja električne energije, uključanje/isključanje servisa, detekcija treptanja (tampering), identifikacija i izolacija ispada, nadgledanje napona. Kombinovano sa tehnologijama koje su vezane za korisnika (pametne kuće, programibilna regulacija temperature...), AMI omogućava elektrodistribucijama

nove programe pomoću kojih je moguća kontrola potrošnje i cena u realnom vremenu [8]. Zaštita od ispada predstavlja još jedan bitan deo sistema zasnovan na prevenciji i predikciji [3]. AMI omogućava dvosmernu komunikaciju, tako da je moguća komunikacija ili izdavanje komandi od strane elektrodistribucije do pametnog brojila ili uređaja za upravljanje potrošnjom. AMI nije ograničen samo na elektrodistribucije, sistem je takav da pokriva gasne kao i vodovodne mreže [4].

2.2. Pametna brojila

Predstavljaju napredna brojila koja dobijaju informacije sa korisničkih uređaja i mere potrošnju električne energije. Informacije o potrošnji energije korisnika, prosleđuju se dalje kompaniji radi kontrole i naplate utrošene energije. Sastoje se od senzora i kontrolnih uređaja povezanih komunikacionom infrastrukturom. Podaci koje sadrže uključuju: jedinstveno obeležje samog brojila, vremensku odrednicu, količinu utrošene električne energije kao i mnoge druge. Imaju mogućnost prikupljanja dijagnostičkih podataka o distributivnoj mreži kao i sa kućnih uređaja. Njihova glavna funkcionalnost je merenje potrošnje električne energije i prenos tih vrednosti do sistema za naplatu. Komunikacija je dvosmerna tako da nakon što je potrošnja obračunata potrošač dobija informacije o tome kako da optimizuje potrošnju. Pored toga što dvosmerno komuniciraju sa pametnim uređajima i sistemima za naplatu, pametna brojila mogu da komuniciraju i međusobno [5].

Prednosti su mnogobrojne i različite u zavisnosti od korisnika. Sa aspekta distribucije prednosti predstavljaju unapređeni sistem naplate i očitavanje brojila koji dovodi do ravnomernije proizvodnje, što donosi ogromne uštede, mogućnost upravljanja potrošnjom električne energije, posebno u vreme maksimalne potrošnje, efikasnija potrošnja energije i mrežnih resursa, pojava novih tarifnih modela na tržištu. Sa aspekta vlada, investiranjem u mreže pametnih brojila stimuliše se ekonomija, pomaže očuvanju životne sredine smanjivanjem emitovanja CO₂. Podizanjem nivoa svesti korisnika podstiče se smanjivanje potrošnje električne energije.

Omogućavaju bolju prognozu opterećenja u distributivnoj mreži kao i prevenciju većih ispada, bolju efikasnost i pouzdanost servisa. S obzirom na to da se komunikacija odvija u oba smera, signali i komande se šalju direktno do pametnih brojila, lokalno korisnicima ili dalje uređajima za distribuciju. Redovno prikupljanje podataka o potrošnji električne energije svih potrošača omogućava elektrodistribucijama da efikasnije upravljaju potražnjom električne energije, kao i da savetuju potrošače kako da efikasnije koriste svoje električne uređaje. Pametna brojila mogu da prate kvalitet električne energije, odnosno faze, napon, jačinu struje, aktivnu i reaktivnu snagu. Pored toga, pametna brojila pomažu u detekciji neautorizovane potrošnje i krađe električne energije [4].

2.3. Događaji u okviru pametne mreže

Događaji sa pametnih brojila su osnova mernog sistema. Oni predstavljaju promenu stanja u mreži koja je bitna elektrodistribuciji. Mogu biti ključni pri detektovanju kvarova i neuobičajenih stanja u mreži. Postoje i oni obaveštajnog tipa koji se mogu podeliti na više od 150 vrsta. Mogu predstavljati samo status uređaja u realnom

vremenu, ali mogu biti i indikatori kvaliteta isporuke, ukazivati na ispadе ili probleme u komunikaciji. Događaji mogu sadržati sledeće informacije: o uređaju sa kog potiče događaj, stanje mreže, tip događaja, nivo alarmantnosti. Tip događaja sadrži informaciju o tome na koji tip procesa se događaj odnosi.

U okviru posmatranog sistema postoji šest tipova događaja koji predstavljaju podskup događaja definisanih u IEC-61968-9 [6]: POWER DOWN: predstavlja nestanak napajanja električnom energijom na delu električne mreže, POWER UP: predstavlja povratak napajanja električnom energijom na delu električne mreže, UNDERVOLTAGE: predstavlja pad vrednosti napona na delu mreže ispod određene granice, UNDERVOLTAGE RETURN TO NORMAL: predstavlja povratak vrednosti napona na delu mreže iznad određene donje granice, OVERVOLTAGE: predstavlja povećanje vrednosti napona na delu mreže iznad određene granice, OVERVOLTAGE RETURN TO NORMAL: predstavlja povratak vrednosti napona na delu mreže ispod određene gornje granice.

2.4. SMMS

Sistem za uspravljanje pametnim brojilima (eng. *Smart Meter Management System* - SMMS) predstavlja poseban servis u okviru DMS sistema. Zadužen je za procesiranje poruka od pametnih brojila i slanje zahteva za očitavanje upućenih ka njima. Podaci koji stižu u SMMS su nove vrednosti na pametnim brojilima. Proces obrade podataka se konfiguriše putem profila pametnih brojila. Profil je moguće dodeliti različitim delovima mreže. Najmanja jedinica kojoj profil može biti dodeljen je fider. Kada je profil dodeljen fideru, događaji sa svih pametnih brojila dodeljenih tom fideru obrađivače se prema konfiguraciji tog profila. Neki od parametara profila su: koji događaji i merenja su omogućeni, da li su dozvoljena očitavanja, timeout periodi, vreme stabilizacije itd. Promenom pomenutih parametara možemo uticati na to kako će se događaji obrađivati, što je tema ovog rada.

3. IDENTIFIKACIJA VREMENSKIH NEPOGODA

Teški vremenski uslovi kao što su uragani, ledene oluje, grmljavina i ostale elementarne katastrofe mogu da izazovu prekide u snabdevanju korisnika električnom energijom. Jaki vetrovi mogu da sruše drveće na nadzemne dalekovode, munja može da ošteti transformatore, prekidače, dalekovode itd. Dok neki ispadi mogu biti kratkotrajni (npr. par sekundi) mnogi ispadi zahtevaju fizičku popravku ili održavanje prenosnog i distributivnog sistema pre nego što se napajanje može obnoviti. Prvi korak prilikom rešavanja ispada sistema jeste detekcija ispada [7]. Ako je zaštitni uređaj telemetrisan, onda je informacija o ispadu primljena preko SCADA sistema. Ova vrsta prekida se naziva neplanirani SCADA ispad. Ako zaštitni uređaj nije telemetrisan, onda informacija o ispadu dolazi od strane korisnika koji pozivima obaveštavaju da su ostali bez napajanja. Otprilike od 70-75% poziva su oni uzrokovani ispadom servisa, a trećina njih ne zahteva izlazak na teren. Umesto poziva, elektrodistribucije koje imaju AMI su u mogućnosti da prime informaciju o ispadu od događaja koji stižu sa pametnih brojila i poznati su kao „last gasp“ događaji. Ovakvi događaji obezbeđuju bržu

detekciju ispada od korisničkih poziva, ali ne poseduju ih sve elektrodistribucije.

Vreme koje je potrebno da se brojilo očita iz AMI sistema može uštedeti nekoliko sati ekipama, jer bi u suprotnom morali da idu u polje i ručno provere da li je brojilo ispravno [7]. „Last gasp“ poruke su one koje pametno brojilo pošalje nakon što se ispad dogodio. Prednosti takvih poruka su te da elektrodistribucije ne moraju da čekaju pozive korisnika kako bi identifikovali i locirali ispad. I u slučajevima kada kasne, „last gasp“ poruke dolaze brže od poziva korisnika, naročito u noćnim satima kada korisnici spavaju. Za lociranje ispada ključna je integracija sistema za upravljanje ispadima (eng. *Outage Management System* - OMS) i AMI sistema. Mnogi OMS proizvodi zahtevaju pozive 15% korisnika zahvaćenih ispadom, kako bi pronašli tačnu lokaciju kvara i 20% uspešnih „last gasp“ poruka.

4. OPIS APLIKATIVNOG REŠENJA

Zadatak aplikacije je da na osnovu analize učestalosti i količine podataka u zadatom vremenskom opsegu detektuje neuobičajeno stanje i menja način obrade događaja na SMMS komponenti. Komunikacija između aplikacije i SMMS servisa je dvosmerna. Brža detekcija ispada neposredno znači i bržu restauraciju napajanja. Zbog toga se sve više teži ka automatizovanoj detekciji i tu primenu nalazi AMI. Servis SMMS prilikom oluje vrši dodelu tzv. storm profila delovima mreže koji su zahvaćeni elementarnim katastrofama. U slučaju elementarnih nepogoda, mnogo veći broj događaja stiže na sistem. Zbog velike količine podataka koja treba da stigne na servis postoji i mogućnost gubitka istih. Iz tog razloga storm profil, za razliku od normalnog, treba definisati na drugačiji način sa drugim vrednostima parametara.

Prvi korak je identifikacija same nepogode. Podrazumevano je praćenje sistema i uobičajenih vrednosti tokom rada sistema u realnom vremenu. Postoji više načina na koje možemo identifikovati olujni režim u sistemu. Jedan od načina je detektovanje velike količine događaja koji stižu na sistem. Analizom broja pristiglih događaja u odnosu na referentno normalno stanje u sistemu dobijamo informaciju o tome da li je došlo do problema.

Drugi način prepoznavanja je analizom samih događaja. Potrebno je vršiti analizu koliko je događajima sa kog pametnog brojila potrebno da stignu na sistem. Ukoliko je vreme pristizanja duže nego referentno uobičajeno vreme možemo zaključiti da je došlo do problema na pomenutom brojilu. Automatizacijom sistema da sam prepozna kada je u stanju opterećenja velikim brojem događaja je ključna, jer tada sistem sam može da reaguje na odgovarajući način. Aplikacija radi na takav način da se od korisnika zahteva unos referentnih vrednosti za prepoznavanje oluje. Svaka mreža je drugačija, ima različiti broj komponenti pa samim tim i pametnih brojila u sistemu. Različiti broj pametnih brojila znači da se različiti broj događaja očekuje u sistemu i da je svaki sistem jedinstven po tom pitanju. Nakon identifikacije sledi rešavanje problema. Predstavljena su tri načina na koje možemo uticati na obradu događaja u SMMS sistemu i sva tri obuhvataju promenu parametara profilne konfiguracije sistema. Profilna konfiguracija se odnosi na deo mreže, u našem slučaju na fider. Svakom fideru

pripadaju određena pametna brojila tako da se definisana profilna konfiguracija odnosi i na njih. Promenom parametra vremena stabilizacije (*StewTime*) utičemo na vreme čekanja na događaj suprotnog ili istog tipa sa istog brojila. Ukoliko za vreme stabilizacije stigne očekivani događaj, dalja obrada se ne nastavlja. Tokom olujnog režima događaji kasne pa je u skladu sa tim potrebno definisati parametar *StewTime*. Dodeljivanjem većih vrednosti parametru *StewTime* omogućava se duže čekanje na događaj suprotnog ili istog tipa i time je nepotrebna obrada izbegnuta. Filtriranje tipova i onemogućavanje čitanja sa brojila su sledeći parametri kojima se utiče na smanjenje broja događaja za obradu. Kao što je već napomenuto tokom vremenskih nepogoda pametna brojila generišu veliku količinu podataka veoma frekventno tako da čitanja sa njih nisu neophodna.

Promenom parametara dozvoljena/zabranjena čitanja, profilne konfiguracije onemogućavamo čitanja sa pametnog brojila na kog se ona odnosi. Prilikom filtriranja događaja, bitno je da definišemo prioritete tipove događaja čija je obrada ključna. Podešavanjem parametra dozvoljeni/zabranjeni događaji, utičemo na kontrolu obrade i tako što se obrađuju samo oni događaji čiji je tip definisan u listi dozvoljenih događaja.

4.1. Simulacija

Za potrebe simulacije implemenirana je aplikacija koja simulira realan sistem u kome se šalju događaji sa pametnih brojila na SMMS komponentu. Aplikacija je implementirana u C# programskom jeziku, koristeći WPF tehnologiju. Kao razvojni framework, korišćen je .NET framework 4.7.1. Simulacija se vrši na dva načina: ručnim slanjem događaja na servis (korisnik sam unosi željene podatke), automatska simulacija događaja sa polja (sistem sam kreira događaje po nasumičnom principu).

5. REZULTATI I TESTIRANJA

Testiranje je izvršeno pomoću simulatora za slanje događaja. Simulirano je slanje događaja u dva režima olujnom i normalnom režimu. Normalan režim predstavlja slanje 8-13 događaja po sekundi dok olujni režim odlikuje slanje 80-120 događaja po sekundi.

U tabeli 1. prikazane su performanse SMMS sistema za testne slučajeve u kojima je olujni režim trajao 20, 30, 60, 90 i 120 minuta i da pri tome implementirano rešenje nije bilo pokrenuto.

U tabeli 2. prikazane su performanse SMMS sistema za teste slučajeve u kojima je olujni režim trajao 20, 30, 60, 90 i 120 minuta i da je pri tome implementirano rešenje bilo pokrenuto.

Tabela 1. Performanse servisa kada nije pokrenuta aplikacija

Trajanje oluje [min]	max CPU [%]	avg CPU [%]	max Mem [MB]	min Mem [MB]	avg Mem [MB]
20	93,2	35,39	669	511	624
30	87,83	33,96	662	599	636
60	82,34	36,38	684	445	646
90	99,9	43,49	669	452	609
120	81,38	40,03	784	480	697

Tabela 2. Performanse servisa kada je pokrenuta aplikacija

Trajanje oluje [min]	max CPU [%]	avg CPU [%]	max Mem [MB]	min Mem [MB]	avg Mem [MB]
20	80,01	25,51	574	329	528
30	77,12	35,83	637	343	510
60	76,54	31,13	589	358	553
90	91,44	36,39	592	559	575
120	79,66	32,74	669	383	596

Iz tabela možemo zaključiti da su performanse procesora poboljšane u svim slučajima u kojima je razvijena aplikacija imala uticaja na servis, što je i očekivano jer je zadatak aplikacije da rastereti procesor. Kod memorijskih performansi, u slučajima u kojima aplikacija nije bila pokrenuta primećujemo veće prosečne vrednosti memorije u odnosu na iste slučajeve u kojima je aplikacija imala uticaj. Najmanji memorijski skok možemo primetiti u slučaju olujnog režima od 90 minuta, kada implementirano rešenje bilo pokrenuto i on iznosi 592-559=33MB dok je najveći skok bio u slučaju oluje od 120 minuta i iznosi 801-493=308MB u kom implementirano rešenje nije bilo pokrenuto, što je i očekivano jer je olujni režim trajao najduže.

Na skok memorije mogu uticati brojni faktori. Prilikom uspostavljanja komunikacije sa servisom dolazi do većih memorijskih skokova, svaki put kada se aplikacija obrati servisu kako bi promenila konfiguraciju dolazi do većih memorijskih skokova. SMMS je kompleksan servis sa velikim brojem funkcionalnosti tako da ne možemo sa sigurnošću da utvrdimo na koji način se memorija ponaša prilikom obrade događaja.

6. ZAKLJUČAK

U ovom radu opisano je rešenje problema identifikacije elementarnih katastrofa od strane SMMS sistema. Identifikacija i pravovremena reakcija na ispade u mreži su ključni jer elementarne katastrofe mogu prouzrokovati značajnu štetu na elektroenergetskoj mreži i mogu dovesti do rasprostranjenih prekida napajanja električnom energijom. Količina događaja koje pametna brojlara generišu tokom oluje je opterećujuća za sistem tako da pravovremena obrada i odgovor na elementarnu nepogodu nije moguć. Zadatak SMMS sistema je da obradi događaje koji stižu sa pametnih brojila i prosledi ih zainteresovanim stranama. Ukoliko sistem ne realizuje obradu dovoljno brzo to utiče i na ostale servise, najvažniji od njih je sistem za upravljanje ispadima.

Implementirano rešenje omogućava rasterećenje procesora SMMS servisa tako da je moguća brža obrada događaja i pravovremena reakcija na ispade. Kada je u pitanju memorija, prosečne vrednosti memorije su veće u slučaju kada implementirano rešenje nije pokrenuto što je i bio cilj. Kod skokova memorije nije primećeno veće odstupanje ni u jednom od testiranih slučajeva. Nepotrebna obrada je izbegnuta i događaji su filtrirani

tako da je omogućena obrada samo onih validnih i značajnih za detekciju ispada. Dalja unapređenja rešenja su moguća uvođenjem veštačke inteligencije u aplikaciju. Ukoliko bi aplikacija učila na osnovu iskustva, reakcija na događaje bi bila mnogo brža i preciznija. Isključena bi bila analiza mreže i definisanje parametara od strane korisnika.

Posmatranje pristiglih događaja u realnom vremenu omogućava da uvođenje veštačke inteligencije bude sasvim logičan nastavak.

Velika količina podataka u realnom vremenu daju aplikaciji materijal pomoću koga se može utvrditi kako se događaji ponašaju u normalnom režimu i daju prostora da se identifikuje stanje koje se ne uklapa u već viđene obrasce.

7. LITERATURA

- [1] Seth Blumsack, Alisha Fernandez, Ready or not, here comes the smart grid!, *Energy*, vol. 37, no. 1, pp. 61-68, 2012.
- [2] Zohreh S. Hosseini, Mohsen Mahoor, Amin Khodaei, *AMI-Enabled Distribution Network Line Outage Identification via Multi-Label SVM*, *IEEE Transactions on Smart Grid*, vol. 9 no. 5, 2018.
- [3] Yasin Kabalci, *A survey on smart metering and smart grid communication*, *Renewable and Sustainable Energy Reviews*, vol. 57, pp. 302-318, 2016.
- [4] R. R. Mohassel, A. Fung, F. Mohammadi, K. Raahemifar, *A survey on advanced metering infrastructure*, *International Journal of Electrical Power & Energy Systems*, vol. 63, 2014, pp. 473-484, 2014.
- [5] Jixuan Zheng, David Wenzhong Gao, Li Lin, *Smart Meters in Smart Grid: An Overview*, *Green Technologies Conference (IEEE-Green)*, 2013.
- [6] <http://webstore.iec.ch/webstore/webstore.nsf/artnum/-048719!opendocument> (pristupljeno u novembru 2019.)
- [7] Hahn Tram, *Technical and Operation Considerations in Using Smart Metering for Outage Management*, *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, 2008.
- [8] Chen-Ching Liu, *Distribution systems: reliable but not resilient?*, *IEEE Power and Energy Magazine*, vol. 13, no. 3, pp. 93 – 96, 2015.

Kratka biografija:



Ivona Jovanović rođena je u Prištini 1995. god. Fakultet tehničkih nauka upisala je 2014. god. Bečelorski rad iz oblasti Baze podataka odbranila je 2018. god. Master rad iz oblasti Elektrotehničko i računarsko inženjerstvo odbranila je 2019. god.

MASTER/SLAVE KOMUNIKACIONI PROTOKOLI ZA SCADA SISTEME

MASTER/SLAVE COMMUNICATION PROTOCOLS FOR SCADA SYSTEMS

Marko Bubnjević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je prezentovano programsko rešenje kojim simuliramo slave stranu VanComm Master/Slave protokola. Implementirano programsko rešenje može se koristiti za simulaciju komunikacije između master i slave strane, gde master strana šalje zahteve za čitanje podataka ili izvršavanje komandi, a slave strana te zahteve obrađuje i izvršava naređenja dobijena od master strane koristeći VanComm protokol.

Ključne reči: Protokol, SCADA, VanComm Protokol.

Abstract – The document presents a software solution which is used for simulation of slave part of VanComm Master/Slave communication protocol. The implemented software solution may be used for simulation of communication between master and slave sides, where master side sends requests for reading of data or command execution, and slave part process and executes requests which came from master side while using VanComm communication protocol.

Keywords: Protocols, SCADA, VanComm Protocol.

1. UVOD

Komunikacioni protokoli u SCADA sistemima predstavljaju ključnu ulogu prilikom komunikacije između master i slave strane sistema, gde master stranu uglavnom predstavlja aplikacija koja prikuplja podatke sa terena, obrađuje ih i dobijene obrađene podatke prikazuje unutar aplikacije, a slave stranu uglavnom predstavlja RTU koji šalje podatke sa terena, te izvršava komande poslate od strane mastera nad tačkama koje se nalaze „ispod” samog RTU-a.

Posmatraćemo razlike između tri veoma zastupljena protokola (DNP3, IEC-104 i VanComm) i pokušaćemo da izvučemo neke zaključke vezane za svaki protokol uz posebni akcenat na VanComm protokol.

Prilikom implementacije bilo kog protokola važna je mogućnost testiranja protokola pre puštanja u rad kako bi se izbegle neželjene situacije sa potencijalno opasnim ishodom koji mogu da donesu velike gubitke bilo u tehničkom ili ljudskom sektoru usled nekih previda i nepredviđenih situacija. RTU-evi su veoma skupi i teški za konfiguraciju i pristup, a RTU-evi ili programska rešenja koja služe za testiranje protokola često nisu opremljena tako da se mogu ispitati svi mogući slučajevi koji se mogu desiti na terenu.

NAPOMENA:

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

Ovaj problem se može rešiti softverskim putem implementiranjem odgovarajućeg slave-a koji će se ponašati kao RTU sa svim ključnim karakteristikama. Postoje razni protokoli koji na svoj način vrše komunikaciju između komponenti SCADA sistema, te svaki od tih protokola ima svoje prednosti i mane, a samim tim i RTU-evi koji komuniciraju preko tog protokola mogu imati veću ili manju kompleksnost, te implementacija slave-a nosi veću ili manju težinu u smislu korisnosti same aplikacije. U daljim poglavljima opisaćemo implementaciju jednog takvog slave-a koji radi koristeći VanComm protokol i za razliku od testnog RTU-a podržava širi opseg zahteva i komandi, a zbog kompleksnosti samog VanComm protokola donosi i veliku prednost i olakšanje za korisnika.

Postoje već implementirani simulatori koji se ponašaju kao slave za VanComm protokole ali su oni većinom limitirani i uglavnom ne predstavljaju dobro rešenje. Primer jednog takvog simulatora je Perl[1] simulator koji podržava samo jednostavne zahteve i komande kao što su čitanje analognih i digitalnih ulaza i izlaza, ali mu fale kompleksniji zahtevi kao što je to npr. komandovanje preko izlaznih tačaka koji će kao posledicu promeniti ulaznu tačku vezanu za taj izlaz.

Ukratko ćemo opisati redosled poglavlja i njihov sadržaj. U poglavlju „Teorijske osnove“ opisan je SCADA sistem, kao i vrste njegovih protokola i njihove ključne razlike. Poglavlje „Implementirano programsko rešenje“ predstavlja opis implementacije same aplikacije. Poglavlje „Verifikacija programskog rešenja“ prikazuje rezultate testiranja implementiranog rešenja uz poređenje sa Perl simulatorom i njegovim performansama i mogućnostima. Unutar poglavlja „Zaključak“ opisujemo moguća poboljšanja aplikacije, dok u poglavlju „Literatura“ navodimo izvore koji su korišćeni prilikom izrade rada.

2. TEORIJSKE OSNOVE

Usled potrebe industrije za kontrolom procesa sa udaljenih lokacija osmišljen je SCADA [2] sistem. Osnovni problem koji SCADA rešava je kontrola više tačaka sa udaljene lokacije koje takođe mogu biti „razbacane” gledajući njihovu geografsku poziciju. Osnovni delovi svakog SCADA sistema su MTU (Master Terminal Unit) koji predstavlja osnovnu jedinicu tj. „mozak” SCADA sistema, komunikaciona oprema u koju spadaju svi različiti načini komunikacije preko različitih prenosnih medija kao što su npr. internet, radio veza, telefonska veza itd, RTU (Remote Terminal Unit) koji predstavlja jedinicu preko koje master strana dobavlja željene podatke za tačke koje se nalaze „ispod” RTU-a ili

komanduje njima i menja im stanja i vrednosti i istorijska baza podataka u kojoj se čuvaju sve promene sa terena.

Komunikacija između MTU-a i RTU-eva je implementirana koristeći razne protokole koji predstavljaju pravila tj. na koji način će se vršiti komunikacija između komponenti. Svi podaci se moraju preneti preko nekog protokola kako bi znali kako da iz pristigle poruke izvučemo željene podatke koji su pritom tačno interpretirani. Postoje dva načina prenosa poruka preko protokola, prva vrsta je inicirana od strane MTU-a i namenjena je RTU-u, a druga vrsta je inicirana od strane RTU-a i namenjena je MTU-u.

Svi protokoli ne podržavaju mogućnost da se RTU sam „javlja” MTU-u i zbog toga protokole takođe možemo podeliti na dve vrste: balansirane (RTU se može „javiti” sam) i nebalansirane (RTU se ne može „javiti” sam) protokole. Posmatrani su sledeći protokoli: DNP3 [3], IEC-60870-5-104 [4] i VanComm [5], te su opisane ključne razlike između gore navedenih protokola:

Protokol	DNP3	IEC-104	VanComm
Standard organizacija	DNP3 Users group	IEC standard	Ferranti
Adresiranje uređaja	16-bitno adresiranje uređaja	Adresa Master-a je 8 bita, adresa RTU-a i objekta koji se posmatra 16 bita	Adresa RTU-a i Master-a su dužine 8 bita
Mehanizmi za pouzdanost	16-bitni CRC, Start/Stop bit	Kontrolna polja	8-bitna BCH kontrolna suma
Fizički sloj i procedure sloja veze	Nebalansirani mod, Balansirani mod, TCP/IP	Balansirani mod, TCP/IP	Nebalansirani mod, Sinhrona serijska komunikacija
Aplikacioni sloj	Kontrolisanje i obrada više tačaka po poruci; Više tipova podataka po poruci; Ima potvrdu i potvrdu potvrde; Ima potvrdu potvrde; Ima potvrdu potvrde;	Jedan tip podataka po poruci; Kontrolisanje i obrada jedne tačke po poruci;	Kontrolisanje i obrada više tačaka po poruci; Jedan tip podataka po poruci; Potvrda da je master dobio odgovor (Station Check, Status Acknowledge,)
Dominantno tržište	Severna Amerika, Australija, Kina	Evropa	Amerika

3. IMPLEMENTIRANO PROGRAMSKO REŠENJE

U ovom poglavlju prikazana je implementacija programskog rešenja tj. slave-a koji koristi Vancomm protokol, uz opis arhitekture samog rešenja. Slave prima zahteve sa master strane, obrađuje ih, izvršava zahteve i zatim vraća odgovarajuće odgovore nazad ka master-u.

3.1. Arhitektura realizovanog rešenja

Osnovne komponente implementiranog rešenja su:

- VanCommSlaveLibrary
- VanCommSlaveConsole
- CommonVanComm
- SlaveCommonVanComm
- VanCommDataLoader

VanCommSlaveLibrary komponenta predstavlja glavnu komponentu implementiranog rešenja. Ona je zadužena za komunikaciju sa master-om, učitavanje konfiguracije iz NMS-a, obrađivanje i procesiranje zahteva koji su stigli sa master-a, kao i slanje odgovora nakon što je zahtev uspešno ili neuspešno obrađen.

VanCommSlaveConsole komponenta predstavlja konzolnu aplikaciju pomoću koje startujemo slave. Možemo izabrati vrstu konfiguracije koju želimo da učitamo (iz XML fajla ili sa NMS servisa), zatim pomoću VanCommSlaveConsole možemo menjati vrednosti za željene konfigurisane analogne ili diskretne tačke, takođe možemo menjati vrednost UnitStatus-a za odgovarajući RTU i kad završimo sa korišćenjem istog prekinuti rad slave-a.

CommonVanComm komponenta je biblioteka koja sadrži sve zajedničke podatke i metode važne za rad celog sistema kao što su na primer: Enums, ByteParityCalculator, SecurityCoding itd.

SlaveCommonVanComm je biblioteka koja formira odgovore za odgovarajuće zahteve. to znači da unutar SlaveCommonVanComm komponente formiramo frejmove, zaglavlja i poruke koje se kasnije koriste kao odgovor na zahtev.

VanCommDataLoader je komponenta zadužena za popunjavanje memorije odgovarajućim podacima iz XML fajlova.

3.2. Implementacija realizovanog rešenja

Opisaćemo na koji način je implementirano programsko rešenje. Podelićemo poglavlje na više celina kako bi lakše ispratili sve funkcionalnosti slave-a kao i na koji način je izvršena njihova implementacija. Programsko rešenje implementirano je u C# programskom jeziku.

3.2.1. Komunikacija sa Master-om

Komunikacija između master-a i slave-a se uspostavlja preko TCP konekcije koristeći socket-e. Glavna klasa koja je zadužena za sve aspekte komunikacije se naziva CommunicationEngine. Pre nego što master pokuša da uspostavi konekciju sa slave-om potrebno je kreirati i pokrenuti TCPServer-e koje iščitavamo iz konfiguracije.

Klasa TCPServer je klasa koja je zadužena za kreiranje soketa, njihovo održavanje, pravovremeno gašenje ili stopiranje ako je to potrebno. TCPServer takođe prihvata i uspostavlja konekciju sa master-om, ili gasi konekciju ukoliko dođe do neke greške prilikom komunikacije. Svaki TCPServer je pokrenut na jedinstvenoj kombinaciji IP adrese i porta.

ConnectionState je pomoćna klasa služi za kontrolu stanja trenutne konekcije između master-a i pokrenutih TCPServer-a na slave-u.

3.2.2. Konfiguracija podataka

Svi podaci koje slave koristi za svoj rad se iščitavaju iz konfiguracije, kako bi te podatke mogli predstaviti što jednostavnije i bolje. Bilo je potrebno modelovati podatke i kreirati odgovarajuće klase koje sadrže odgovarajuće delove podataka koji zajedno čine jednu veliku celinu.

3.2.3. Parsiranje i procesiranje pristiglih zahteva

Klasa Manager je klasa koja kontroliše ceo rad slave-a. Poziva metode za čitanje podataka u konfiguraciju, startuje CommunicationEngine i startuje RequestReceiver koji je zadužen za parsiranje i obradu pristiglih zahteva. Manager takođe stopira servise, ali glavna uloga Manager-a je da uz pomoć event-ova pokreće parsiranje pristiglog zahteva i šalje nazad odgovore koji su formirani od strane slave-a.

Klasa RequestReceiver je glavna klasa za parsiranje pristiglih zahteva od strane master-a. Svaki tip zahteva ima odgovarajući procesor koji kasnije ukoliko do takvog zahteva dođe parsira zahtev i nakon toga formira odgovor. Procesori se mapiraju pomoću dictionary-ja requestProcessorsMap, dok se linkovi između master-a i slave-a mapiraju pomoću dictionary-ja ipPortToLinkMap. Popunjavanje samih dictionary-ja se vrši u metodama ConfigureLinksMap i PopulateRequestProcessorMap.

Svaki zahtev koji pristigne sa master-a na slave se procesira u metodi ProcessRequest. Metoda prvo parsira sirove podatke tako da popuni sva potrebna bolja i konvertuje sirove podatke u čitljiv zahtev. Zatim iz zahteva izvlačimo informacije koje su nam potrebne kako bi formirali odgovor. Pomoću polja StationAddress proveravamo da li imamo konfigurisan RTU pod tom adresom, proveravamo OperationCode i iz dictionary-ja procesora pozivamo odgovarajući procesor koji je zadužen za tu vrstu zahteva. Procesor dalje obrađuje zahtev i formira odgovor, vraća odgovor nazad, te se odgovor šalje master strani koja je poslala zahtev.

3.2.4. Formiranje odgovora za odgovarajući zahtev

Osnovnu metodu predstavlja metoda Process koja kao parametre prima formiran zahtev od strane master-a i RtuSlaveModel koji predstavlja RTU nad kojim se izvršava neka akcija ili čitanje podataka. Svaki procesor zadužen je za tačno jedan tip zahteva, tj. za formiranje odgovarajućeg odgovora.

3.2.5. Konzolna aplikacija

Konzolna aplikacija nam služi za pokretanje samog slave-a, kao i za još neke dodatne funkcionalnosti koje ćemo objasniti u ovom podpoglavljju. Prilikom pokretanja prvo je potrebno da izaberemo odakle želimo da učitamo konfiguraciju koja će napuniti našu memoriju podacima.

Pored učitavanja konfiguracije preko konzole imamo mogućnosti da setujemo analogni ulaz, analogni izlaz, UnitStatus bajt, kao i da ugasimo slave. Ove opcije su omogućene kako bi preko konzole mogli da simuliramo spontane promene koje će kasnije master strana prepoznati i obraditi.

4. VERIFIKACIJA PROGRAMSKOG REŠENJA

VanComm slave služi za simulaciju velikih industrijskih SCADA sistema sa više RTU-eva i više tačaka nakačenih na te RTU-eve. U ovom poglavljju ćemo testirati skalabilnost implementirane aplikacije tj. maksimalni broj RTU-eva i tačaka gde će naše programsko rešenje raditi bez problema i zadovoljiti standarde, takođe izvršićemo poređenje sa već postojećim VanComm slave simulatorom „Strawberry Perl” i pokazati zašto je naše implementirano rešenje bolje, tj. koja funkcionalna

proširenja smo dobili našim rešenjem koja ranije nisu bila podržana.

Kapacitet implementiranog rešenja u mnogome zavisi od kapaciteta računara na kojem smo pokrenuli VanComm slave, kao i od sposobnosti mreže preko koje se izvršava komunikacija master-a i slave-a, zbog toga bi bilo bespotrebno testirati performanse u obliku merenja brzine formiranja odgovora, ili brzine slanja svih odgovora na sve zahteve pristiglih u nekom vremenskom periodu. Jedina „jedinica” kojom možemo dati težinu našem radu je kao što smo već rekli broj RTU-eva i tačaka gde će VanComm slave i dalje raditi bez problema. U tabeli prikazaćemo scenarija za koje je slave testiran uz napomenu da se za svaki RTU vršilo periodično skeniranje svih tačaka na 30 sekundi i za svaki zahtev je stigao odgovarajući odgovor, a slave je radio u periodu od 10 minuta. Memorija koju zauzima slave prilikom inicijalizovanja podataka za virtuelne RTU-eve je zanemarljiva i nećemo je posmatrati kao faktor rada slave-a. Testiranje smo vršili na računaru sa 24.3 GB RAM memorije i Inter Core i3-3210 procesoru 3.20 Ghz.

Broj RTU-eva	Ukupan broj analognih i digitalnih tačaka	Prosek obrađenih zahteva u periodu od 30 sekundi	Slave pravilno radi
10	2000	133,44	Da
20	3000	241,61	Da
30	4000	273,42	Da
40	5000	386,17	Da
50	6000	317,28	NE

Prilikom testiranja sa 50 RTU-eva i 6000 tačaka došlo je do grešaka prilikom odgovora na zahteve i za taj slučaj možemo reći da VanComm slave nije kompetentan. VanComm slave je radio najbrže sa konfiguracijom od 40 RTU-eva i 5000 analognih i digitalnih tačaka gde je u proseku na svakih 30 sekundi obrađivao 386,17 različitih zahteva.

Strawberry Perl je aplikacija koja predstavlja slave stranu za više različitih protokola kao što su DNP3, Series5, Modbus i VanComm, dok ćemo se mi fokusirati samo na deo vezan za VanComm protokol. Aplikacija sama po sebi ima dosta nedostataka koji su rešeni implemenitranjem našeg VanComm slave-a.

Strawberry Perl podržava terminal server-e ali samo sa jednim RTU-em nakačenim „ispod” tog terminal server-a, ili drugačije rečeno samo jedan RTU ukoliko ne postoje nikakvi terminal server-i, dok naša aplikacija nema to ograničenje i moguće je simuliranje više RTU-eva odednom i konektovanje master strane na njih uz omogućeno paralelno izvršavanje zahteva i vraćanje odgovora na zahteve. Perl podržava promenu i iščitavanje analognih tačaka u potpunosti, ali prilikom promene ili čitanja digitalnih tačaka moguće je promeniti ili iščitati samo nultu grupu, tj. ne postoji ni jedna više grupa sem nulte grupe što predstavlja veliki nedostatak koji je rešen implementacijom našeg slave-a. Perl simulator pored čitanja analognih i statusnih tačaka podržava report by

exception, read internal status, raise lower immediate execute, control select, control execute i control immediate execute zahteve, ali ne podržava još dosta važnih zahteva kao što su prvenstveno setpoint select, setpoint execute i setpoint immediate execute koji programski izvršavaju promene na analognim izlazima i predstavljaju pandan control zahtevima koji se kao što je već rečeno koriste prilikom promene digitalnih izlaza i tačaka vezanih za te izlaze.

Starberry Perl simulator takođe ne podržava zahtev za restartovanje podataka unutar RTU-a (RTU restart zahtev), zahtev za brisanje internih bajtova (Clear internal statuses) i zahtev za proveru stanja samog RTU-a, tj. komunikacije između komponenti (Station Check). Jasno je posle upoređivanja našeg rešenja i Perl simulatora da je naše rešenje mnogo kompleksnije i upotrebljivije za korisnika, te da predstavlja unapređenje u odnosu na prethodna rešenja koja se mogu naći.

5. ZAKLJUČAK

Protokoli predstavljaju ključni deo sistema komunikacije unutar SCADA sistema te je njihova implementacija i unapređivanje uvek aktuelna tema. Uporedili smo tri veoma zastupljena protokola i videli na koji način su implementirani, te koje su njihove prednosti i mane, kao i gde se najviše koriste posmatrajući geografsko područje i infrastrukturu koju ti protokoli koriste.

Cilj ovog rada je bio razvoj aplikacije koja će uspešno simulirati rad više RTU-eva koji primaju zahteve od master strane, obrađuju te zahteve i izvršavaju ih uz vidnu promenu „na terenu” koja se može ispratiti iščitavanjem podataka „sa terena”. Važna stavka same aplikacije je takođe i obrada i odgovor na zahteve u neko prihvatljivo vreme koje bi nalikovalo na realni sistem, kao i sama skalabilnost aplikacije kako bi taj „realni sistem” mogao biti što veći kako bi mogli testirati scenarije koji bi se mogli desiti u realnom životu. Aplikacija predstavlja unapređenje u odnosu na prethodno implementirana rešenja koja imaju velike nedostatke i kod kojih VanComm protokol nije u potpunosti iskorišćen, tj. nisu iskorišćene sve njegove funkcionalnosti i karakteristike koje ga čine pogodnim za upotrebu.

Moguća unapređenja našeg programskog rešenja su vezana za zahteve koji nisu obrađeni u trenutnoj verziji

VanComm slave-a, a to su: 16-bit Accumulator Request, 32-bit Accumulator Request, Clear Accumulators, Defrost Accumulators, Freeze Accumulators i Freeze and Clear Accumulators, gde su akumulatori memorije koje možemo oblikovati po sopstvenim potrebama, npr. ako nam treba podatak koji zauzima 64 bita možemo spojiti dva akumulatora i tako iščitati podatak.

Zahtevi koji takođe nisu podržani od strane našeg slave-a su i SOE Dump, SOE Capture Line, SOE Line Delay, SOE Acknowledge, SOE Purge Events i RTU Clock Set, gde SOE predstavlja Sequence of Events, tj. skup nekih događaja i ovim zahtevima manipulišemo tim događajima, a RTU Clock Set postavlja tačno vreme na strani RTU-a ukoliko je potrebno sinhronizovati vremena na master-u i slave-u. Još jedno od mogućih unapređenja je i menjanje konzolne aplikacije sa nekom modernijom vrstom korisničkog interfejsa koji se može implementirati koristeći npr. WPF tehnologiju.

6. LITERATURA

- [1] Perl simulator. <https://www.perl.org/> (Accessed 23.09.2019.)
- [2] What is SCADA?
<https://inductiveautomation.com/resources/article/what-is-scada> . (Accessed 24.09.2019.)
- [3] DNP3 SPECIFICATION, Version 2.01, DNP Users Group, 2007.
- [4] P.Matoušek, *Description and analysis of IEC 104 Protocol*, Brno University of Technology, 2017.
- [5] *Appendix B, Ferranti International Controls, Message Standard: Van-Comm*, Ferranti International Controls, 1986.

Kratka biografija:

Marko Bubnjević rođen je 31.05.1994. godine u Novom Sadu. Završio je osnovnu školu „Dušan Radović” 2009. godine, a Elektrotehničku školu „Mihajlo Pupin” u Novom Sadu 2013. godine. Osnovne akademske studije na Fakultetu Tehničkih Nauka u Novom Sadu završio je 2018 godine, a iste godine upisao master akademske studije na Fakultetu Tehničkih Nauka, smer Primenjeno softversko inženjerstvo. Ispunio je sve obaveze i položio je sve ispite predviđene studentskim programom sa prosečnom ocenom od 8.83.

Kontakt: marko.bubnjovic94@hotmail.com

**SISTEM ZA PRIKUPLJANJE I ANALIZU FOTOGRAFIJA SA ULIČNIH KAMERA
UPOTREBOM MAŠINSKOG UČENJA****SYSTEM FOR COLLECTING AND ANALYZING PHOTOS FROM STREET CAMERAS
USING MACHINE LEARNING**

Aleksandar Nikolić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Rad opisuje distribuiranu arhitekturu i implementaciju sistema za prikupljanje i analizu fotografija sa uličnih kamera upotrebom modela mašinskog učenja za detekciju objekata baziranog na konvolucionim neuronskim mrežama.

Ključne reči: *Elasticsearch, mašinsko učenje, mikroservisi, detekcija objekata*

Abstract – This paper describes the distributed architecture and implementation of the system for collecting and analyzing photos from street cameras using machine learning model for object detection based on convolutional neural networks.

Keywords: *Elasticsearch, machine learning, microservices, object detection*

1. UVOD

Sa rastom računarskih sistema, količina podataka koje ti sistemi prikupljaju takođe raste. Jedan deo ovih podataka čine fotografije o čijoj semantici nije moglo mnogo da se zaključi na osnovu samih fotografija, osim podataka kao što su anotacije fotografija koje su ljudi manuelno anotirali. Pojava robusnih modela mašinskog učenja i dovoljne količine podatka za njihovo obučavanje omogućava automatizovanu ekstrakciju informacija iz fotografija. U kontekstu DIKW piramide [1], fotografije predstavljaju sirove podatke, čijom transformacijom u korisne informacije je moguće doći do znanja. Sistem opisan u ovom radu ima zadatak da fotografije prikupljanje sa uličnih kamera širom sveta analizira uz pomoć modela mašinskog učenja, koji služi za detekciju objekata, analizira i time ove podatke pretvori u korisne informacije. Ove informacije je zatim moguće staviti u geo-prostorni kontekst koji daje uvid u dešavanja na ulicama, kao što je kretanje ljudi i vozila.

2. DETEKCIJA OBJEKATA

Detekcija objekata je oblast računarske vizije koja se bavi identifikovanjem i lociranjem objekata određenih klasa na slikama. Probleme koje rešava detekcija je moguće predstaviti pomoću četiri kategorije [2]:

Semantička segmentacija: Ovo je oblast koja se bavi problemom klasifikacije svakog pojedinačnog piksela u određenu kategoriju.

Klasifikacija i lokalizacija: Proces koji se, pored uspešne klasifikacije slike u određenu kategoriju, sastoji iz određivanja graničnog okvira oko klasifikovanog objekta. Pretpostavka je da se na slici nalazi fiksni broj objekata (uglavnom tačno jedan).

Detekcija objekata (u užem smislu): Ovo je generalniji slučaj klasifikacije i lokalizacije u kojem broj objekata nije fiksni, već se na slikama može nalaziti više objekata različitih kategorija.

Segmentacija instance: Umesto određivanja graničnog okvira određuje se koji pikseli pripadaju detektovanom objektu. Razlikuje se od semantičke segmentacije po tome što bi u slučaju da na slici postoje dve instance iste klase semantička segmentacija detektovala obe instance kao jednu instancu.

Klasifikacija i lokalizacija posmatra problem detekcije kao regresiju. Cilj je klasifikovati objekat na slici i detektovati njegov granični okvir. Pošto je izlaz ovog procesa fiksni broj parametara, tj. tačno četiri (klasa detektovanog objekta, x i y koordinate gornjeg levog ugla graničnog okvira, širina i visina graničnog okvira) moguće je model obučavati na principu regresije gde se računaju dve greške: *Softmax* za klasifikaciju i L2 udaljenost za granični okvir.

Kada je u pitanju detekcija proizvoljnog broja fotografija, javlja se problem predlaganja regija fotografije koje su verovatne da sadrže objekte. Na osnovu načina na koji se ove regije predlažu postoji nekoliko načina za rešavanje ovog problema.

R-CNN model, koji je opisan u radu [3], za predlaganje regija koristi eksterni algoritam pod nazivom Selektivna pretraga. Selektivna pretraga je algoritam koji kao osnovu koristi segmentaciju slike baziranu na grafovima čije segmente zatim iterativno spaja po sličnosti (boja, tekstura, veličine i oblik). U svakoj iteraciji manji segmenti se grupišu u veće segmente koji se dodaju u listu predloženih regija. Kao rezultat ovog hijerarhijskog procesa dobija se oko 2000 predloženih regija.

Fast R-CNN, koji je opisan u radu [4], takođe koristi selektivnu pretragu. Jedina razlika je što se ne propusta svaka regija kroz CNN, već se regije direktno projektuju na mapu obeležja koja je rezultat konvolucije cele slike. Ovo znatno ubrzava algoritam.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivanović, vanr. prof.

Faster R-CNN, koji je opisan u radu [5], u potpunosti zamenjuje selektivnu pretragu sa Mrežom sa predlaganje regija (eng. *Region proposal network*) koju je moguće obučavati zajedno sa ostatkom modela. Na ovaj način je deo algoritma za predlaganje regija znatno ubrzan u odnosu na ostale modele.

3. KORIŠĆENE TEHNOLOGIJE

Spring je radni okvir koji predstavlja rešenje za brz i jednostavan razvoj poslovnih aplikacija. Zamišljen je kao lagan i neinvazivan okvir koji teži tome da se aplikacija što manje prilagođava *Spring* API-u. Sadrži veb modul koji pruža podršku za razvijanje veb aplikacija po MVC modelu. *Spring Boot* je radni okvir koji olakšava konfiguraciju *Spring* aplikacija.

RabbitMQ je softver otvorenog koda koji ima ulogu brokera poruka. Originalno, *RabbitMQ* je implementirao *Advanced Message Queuing Protocol* (AMQP), ali vremenom je proširen sa još protokola (STOMP, MQTT). AMQP je protokol koji omogućava komunikaciju između klijentskih aplikacija i brokera poruka koji implementiraju protokol. Brokeri poruka primaju poruke od izdavača poruka (eng. *publishers*) i rutiraju ih do potrošača (eng. *consumers*) odnosno aplikacija koje procesiraju poruke.

Elasticsearch je softver otvorenog koda za pretraživanje i analizu koji se zasniva na *Lucene* biblioteci i razvijen u Java programskom jeziku. Omogućava pristup informacijama u realnom vremenu preko REST API-a u JSON formatu. Takođe poseduje mnoštvo klijentskih biblioteka u svim trenutno popularnim programskim jezicima.

Kibana je platforma za analizu i vizualizaciju koja je dizajnirana za rad sa Elasticsearch-om. Pomoću Kibane je moguće pretraživati, pregledati i interagovati sa *Elasticsearch* indeksima. Kibana pruža korisniku mogućnost da lako izvršava napredne analize i vizualizuje podatke. Poseduje funkcionalnosti kao što su histogrami, linijski grafikoni, pita grafikoni, toplotne mape, kao i ugrađenu podršku za geo-prostornu vizualizaciju.

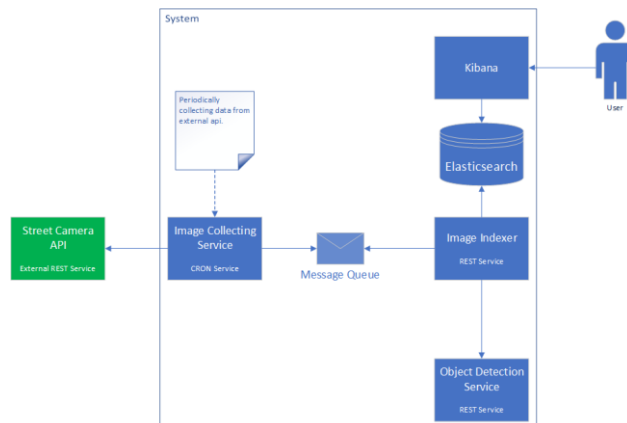
Flask je radni okvir za veb napisan u *Python* programskom jeziku. Klasifikuje se kao mikro radni okvir, jer ne zahteva nikakve dodatne alate i biblioteke. Ne sadrži sloj za apstrakciju baze podataka, validaciju formi i ostale funkcionalnosti za koje već postoje biblioteke koje ih pružaju. Zasniva se na dva projekta, a to su *Werkzeug* i *Jinja*.

Docker je je skup „platforma kao servis“ (eng. *platform as a service*) (PaaS) alata koji koriste virtualizaciju na nivou operativnog sistema da obezbede korisnicima softver koji je upakovan u pakete pod nazivom kontejneri. Kontejneri su izolovani jedni od drugih i sadrže sve potrebne resurse kao što su biblioteke i konfiguracioni fajlovi potrebni za uspešno izvršavanje željenog softvera unutar kontejnera. Na ovaj način programeri mogu da se pouzda u to da će njihove aplikacije raditi isto na svim mašinama koje podržavaju izvršavanje *Docker* kontejnera.

Azure je Majkrosoftova platforma za računarstvo u oblaku (eng. *cloud computing*) koja omogućava pravljenje, upravljanje i raspoređivanje aplikacija korišćenjem raznovrsnih alata, programskih jezika i radnih okvira.

4. ARHITEKTURA SISTEMA

Sistem opisan u ovom radu se sastoji iz više mikroservisa gde je svaki servis zadužen za izvršavanje određenog zadatka. Arhitektura sistema je predstavljena na drugom nivou C4 modela za vizualizaciju [6] i prikazana na slici 1.



Slika 1 Arhitektura sistema prikazana na drugom nivou C4 modela

Servis za detekciju objekata je veb aplikacija realizovana kao REST API koja omogućava detekciju objekata na fotografijama dobijenih od strane eksternog servisa.

Servis za prikupljanje fotografija je veb aplikacija koja ima zadatak da periodično šalje zahteve eksternom servisu i na taj način prikuplja fotografije sa uličnih kamera. Nakon toga fotografije preko brokera poruka šalje u red sa kojeg dalje servis za indeksiranje pristupa fotografijama.

Servis za indeksiranje fotografija je zadužen da preuzima poruke iz reda čije podatke dalje šalje na obradu servisu za detekciju objekata. Nakon što servis dobije odgovor od servisa za detekciju, podatke indeksira u *Elasticsearch* pretraživač.

5. IMPLEMENTACIJA SISTEMA

Servis za detekciju objekata je implementiran kao veb aplikacija po REST principima u *Flask* radnom okviru koristeći *Python* programski jezik. API sadrži *endpoint* za detekciju koji prima *POST* HTTP zahteve. Telo *POST* zahteva je JSON koji se sastoji iz jednog polja pod nazivom *data*. Ovo polje sadrži fotografiju u *base64* formatu radi jednostavnosti samog API-a. Nakon što funkcija koja je mapirana na rutu za detekciju primi zahtev, njegov sadržaj zatim isparsira i konvertuje u rečnik objekat. Iz ovog rečnika je potrebno iščitati polje *data* koje sadrži fotografiju u *base64* formatu. Fotografija se dalje šalje komponenti za detekciju koja je implementirana pomoću *GluonCV* skupa alata za duboko učenje u kompjuterskoj viziji. *GluonCV* obezbeđuje *Faster R-CNN* model koji je obučen na *Comon objects in context* (COCO) [7] skupu podata. Ovaj skup podataka sadrži 80 klasa među kojima su ljudi, automobili, itd. Model se koristi za detekciju objekata na fotografijama koje prima servis. Odgovor servisa je JSON objekat koji sadrži polja koja odgovaraju svakoj klasi COCO skupa

podataka i čije vrednosti predstavljaju broj instanci objekta odgovarajuće klase.

Servis za prikupljanje fotografija je implementiran kao veb servis pomoću *Spring Boot* radnog okvira. Realizovan je kao servis koji periodično automatski prikuplja fotografije sa eksternog REST API-a pod nazivom *Webcams.travel* [8]. *Webcams.travel* omogućava besplatan pristup kamerama širom sveta preko svog API-a. Trenutno sadrži više od 75 hiljada kamera čiji broj raste svakog dana. Za periodično preuzimanje fotografija sa API-a koristi se *Scheduled* anotacija kojom se specificira kada da se pokreće anotirana metoda. U ovom slučaju podešeno je da se metoda poziva svakih 30 minuta.

Da bi preuzimanje fotografija moglo da se distribuira na više računara potrebno je specificirati redni broj kamere od koje počinje preuzimanje i broj kamera. Na taj način moguće je pokrenuti dva ili više servisa od kojih će svaki preuzimati fotografije sa različitih kamera.

Nakon što se prikupe svi potrebni podaci, za svaku fotografiju se kreira poruka koja se šalje na *RabbitMQ* razmenu koja dalje rutira poruku na odgovarajući red. *RabbitMQ* je konfigurisan kroz *Spring* konfiguracioni fajl gde su definisani svi potrebni parametri.

Servis za indeksiranje fotografija je implementiran kao *Spring Boot* veb servis koji komunicira sa *RabbitMQ* brokerom sa čijeg reda preuzima poruke, šalje ih servisu za detekciju objekata i rezultate indeksira u *Elasticsearch*. Sastoji se iz komponente za primanje poruka, komponente za detekciju i komponente za indeksiranje.

Komponenta za primanje poruka je implementirana kao *Spring Boot* komponenta koja poseduje metodu anotiranu sa *RabbitListener* anotacijom. Ova anotacija govori *Spring*-u sa kog reda da čitaju poruke. Komponenta za detekciju komunicira sa servisom za detekciju pomoću *Spring RestTemplate* klase, i omogućava slanje fotografije na analizu. Komponenta za indeksiranje indeksira podatke u *Elasticsearch*. Za komunikaciju sa *Elasticsearch*-om se koristi *Spring Data* repozitorijum interfejs za *Elasticsearch* pod nazivom *ElasticsearchRepository*.

Ovaj interfejs zahteva da se specificira klasa koja predstavlja jedinicu indeksiranja. Da bi aplikacija znala da komunicira sa *Elasticsearch* instancom potrebno je u *Spring* konfiguracionom fajlu obezbediti potrebne vrednosti kao što su *host* i *port* na kom se nalazi *Elasticsearch*, kao i naziv klastera.

Sistem je konfigurisan tako da ga je moguće pokrenuti preko *Docker*-a. Svaki servis poseduje sopstveni *Dockerfile* koji definiše korake za pokretanje servisa. Da bi se svi servisi pokrenuli odjednom koristi se *Docker Compose*. U ovom fajlu su specificirani svi ostali servisi koji su potrebni da bi sistem radio kako treba, kao što su *Elasticsearch*, *RabbitMQ* i *Kibana*. Ovo znatno olakšava ceo proces pokretanja sistema pošto sve što je potrebno da bude instalirano na računaru je *Docker*.

U ovoj konfiguraciji su svakom servisu prosleđeni svi potrebni parametri kao što u promenljive okruženja, portovi koje koriste, kao i URL na koji skripta koja čeka da se servis pokrene može da proveri status servisa koji čeka. U slučaju servisa za detekciju kreiran je poseban *endpoint* koji vraća HTTP status 200 kada servis počne da radi. Takođe kreiran je volumen koji skladišti *Elasticsearch*

indekse tako da u slučaju ponovnog pokretanja sistema prethodno prikupljeni podaci ostaju sačuvani.

Jedan od servisa koji pruža *Azure* je pokretanje aplikacija u *Docker* kontejnerima. Ova funkcionalnost omogućava da se prethodno kreirani *Docker Compose* fajl iskoristi da se ceo sistem pokrene na *Azure* platformi. *Azure* pruža dva načina da se koristi platforma. Prvi način je kroz njihov *online* portal koji daje grafički interfejs za upravljanje svim resursima na platformi. Drugi način je kroz *Azure CLI* alat za komandnu liniju. Ovo omogućava kreiranje skripte koja automatizuje ceo proces pokretanja sistema na *Azure* platformi.

6. REZULTATI I DISKUSIJA

U ovom poglavlju su prikazani rezultati sistema nakon dva dana rada, nedelja 24.11.2019. i ponedeljak 25.11.2019. radi poređenja stanja tokom radnih i neradnih dana. Sistem je pokrenut na računaru sa 16 GB rama, *Intel Core i5-4670K CPU @ 3.40GHz* i *NVIDIA GeForce GTX 1060 6GB*.

Prikupljeni su podaci sa 1000 najpoznatijih kamera na *Webcams.travel* servisu. Fotografije su preuzimane svakih pola sata i analizirane. Ukupno je prikupljeno 91617 indeksiranih fotografija.

U Kibani su kreirane dve komandne table, jedna na kojoj su prikazani podaci vezani za detektovane ljude na fotografijama i druga na kojoj su prikazani podaci o broju automobila.

Kontrolna tabla sa podacima o broju ljudi se sastoji od tri vizualizuje. Na jednoj je prikazana mapa koordinata gde se vidi suma ljudi na svakoj koordinati. Druge dve vizualizacije su histogram broja ljudi na dve lokacije. Prva lokacija je trg u Novom Sadu u Srbiji, a druga lokacija je skijalište u Švajcarskoj. Kontrolna tabla se može videti na slici 2.



Slika 2 Kibana kontrolna tabla sa podacima o broju ljudi

Ovde se može uočiti šablon kretanja ljudi na datim lokacijama. U Novom Sadu se vidi da najviše ljudi ima u posle podnevnim časovima oko 16 časova, što je moguće primetiti na fotografiji prikazanoj na slici. Kada je u pitanju ponedeljak broj ljudi se povećava tek u kasnim posle podnevnim časovima oko 19 časova, što se može objasniti radnim vremenom većine ljudi koji je od 9 do 17 časova. Skijalište u Švajcarskoj je aktivnije nedeljom.

Na drugoj kontrolnoj tabli gde su prikazani podaci o broju automobila mogu se videti tri vizualizacije. Prva je mapa koordinata sa sumom automobila, dok su druge dve histogrami sa sumom automobila sa dve lokacije: luka u gradu Palma, Majorka i grad Pilsen u Češkoj. Ova tabla je prikazana na slici 3.



Slika 3 Kibana kontrolna tabla sa podacima o broju automobila

Na ove dve lokacijama je moguće primetiti časove gde je broj detektovanih automobila manji, što može da se protumače kao periodi kada je manja gužva na putevima.

7. ZAKLJUČAK

U ovom radu predstavljena je arhitektura sistema za prikupljanje i analizu fotografija sa uličnih kamera upotrebom mašinskog učenja. Opisane su svi servisi koji čine ovaj sistem, što podrazumeva komponente od kojih su sačinjeni, kao i komunikaciju između pojedinačnih servisa.

Predstavljeni su osnovni koncepti detekcije objekata na kojima se zasniva ovaj sistem kao i poređenje različitih modela za detekciju objekata. Takođe su opisane tehnologije koju su korišćeni za implementaciju sistema i prikazani su detalji implementacije svakog servisa. Pored toga, predstavljena je konfiguracija koja omogućava jednostavno pokretanje čitavog sistema.

Na kraju su predstavljeni rezultati u obliku vizualizacija podataka prikupljenih od strane sistema. Ove vizualizacije daju uvid u način kretanja ljudi i vozila koji omogućavaju donošenje zaključaka koji mogu biti od značaja.

Može se primetiti da servis za detekciju na većini primera gde postoji veliki broj instanci objekata nije precizan, pogotovo ako su ti objekti udaljeni zato što su same fotografije veoma male (400 x 224 px). Ovo je moguće lako unaprediti korišćenjem premijum pretplate na *Webcams.travel* API koja obezbeđuje fotografije visoke rezolucije. Servis za prikupljanje je moguće optimizovati tako što će se iskoristiti informacija o intervalu osvežavanja kamera i na taj način uz pomoć redova prioriteta optimalnije prikupljati fotografije.

Proces donošenja zaključaka je takođe moguće automatizovati kroz različite modele mašinskog učenja. Na osnovu podatka može se predvideti broj ljudi na određenim lokacijama, broj automobila na važnim raskrsnicama i na osnovu toga bolje planirati javni prevoz, odnosno procenu kada stiže naredni autobus, taksi, itd. Takođe moguće je iskoristiti ove podatke pri organizaciji raznoraznih promocija gde je veći broj ljudi poželjan. Modeli koji su dobri za rad sa istorijskim

podacima su rekurentne neuronske mreže (RNN) koje mogu služiti za predviđanje budućih podataka.

8. LITERATURA

- [1] J. Rowley, „The wisdom hierarchy: representations of the DIKW hierarchy,“ *Journal of Information Science*, 2007.
- [2] „Object Detection: An End to End Theoretical Perspective,“ [Na mreži]. Available: <https://towardsdatascience.com/object-detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-4ca27eee8a9a>. [Poslednji pristup Novembar 2019].
- [3] R. Girshick, „Rich feature hierarchies for accurate object detection and semantic segmentation,“ 2014.
- [4] R. Girshick, „Fast R-CNN,“ 2015.
- [5] S. Ren, „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,“ 2016.
- [6] „The C4 model for visualising software architecture,“ [Na mreži]. Available: <https://c4model.com/>. [Poslednji pristup Novembar 2019].
- [7] „Common objects in context,“ [Na mreži]. Available: <http://cocodataset.org/>. [Poslednji pristup Novembar 2019].
- [8] „Webcams.travel,“ [Na mreži]. Available: <https://www.webcams.travel/>. [Poslednji pristup Novembar 2019].

Kratka biografija:



Aleksandar Nikolić je rođen 14.03.1995. godine u Novom Sadu. Osnovnu školu „Jožef Atila“ u Novom Sadu završio je 2010. godine kao učenik generacije. Gimnaziju „Jovan Jovanović Zmaj“ u Novom Sadu završio je 2014. godine. Iste godine upisao se na Fakultet tehničkih nauka, smer Softversko inženjerstvo i informacione tehnologije i diplomira 2018. godine sa prosekom 9,85. Iste godine upisuje master studije, smer Softversko inženjerstvo i informacione tehnologije, modul Elektronsko poslovanje. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo i informacione tehnologije odbranio je 2019. godine.

INTEROPERABILNI ADAPTER ZA RAD SA LDAP SISTEMIMA**INTEROPERABLE ADAPTER FOR MANAGING LDAP SYSTEMS**Aleksandar Maričić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu istražen je način korišćenja dva različita sistema informacione-bezbednosti – Microsoft Active Directory zasnovan na Windows platformi i OpenLDAP zasnovanog na Linux platformi, uočene su njihove sličnosti i razlike, razvijen je, implementiran i verifikovan interoperabilni adapter za rad sa ova dva sistema informacione-bezbednosti koji omogućuje korišćenje administratorske i korisničke funkcionalnosti.

Ključne reči: Informaciona-bezbednost, Interoperabilnost, Adapter, LDAP

Abstract – In this paper the way of using two different information security system is researched – Microsoft Active Directory based on Windows platform and OpenLDAP based on Linux platform, their similarities and differences are identified, the interoperable adapter for those two directory services was developed, implemented and verified with supporting administration and user-related functionalities.

Keywords: Information Security, Interoperability, Adapter, LDAP

1 UVOD

Informaciona-bezbednost jedna od ključnih komponenti za uspešan razvoj i poslovanje kompanija. U savremenim informaciono-tehnološkim rešenjima zahteva se sve veća standardizacija, integracija i interoperabilnost rešenja koja omogućuju fleksibilan i ekonomski vođen odabir informaciono-tehnoloških (*eng. Information Technologies-IT*) komponenti kao i mogućnost što jednostavnije promene IT komponenti tokom eksploatacionog perioda. Informaciono-bezbednosni sistem upravlja podacima o korisnicima, njihovim ličnim podacima, grupama kojima korisnici pripadaju, pravima pristupa i entitetima koji čuvaju podatke. Postoje različite informaciono-tehnološke platforme i nad njima zasnovana rešenja informacione-bezbednosti. U ovom istraživanju fokus je na dva različita informaciono-bezbednosna sistema – Microsoft Active Directory (*eng. Active Directory-AD*), zasnovan na Windows operativnom sistemu (*eng. OS*) i OpenLDAP, zasnovan na Linux OS - u. Ovi bezbednosni sistemi se oslanjaju na – Lightweight Directory Access Protocol (*eng. LDAP*), ali i pored toga imaju brojne razlike koje se ogledaju u različitim tipovima entiteta, atributima, načinima povezivanja entiteta i u interfejsima za pristup njihovim bazama podataka.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Nemanja Popović, docent.

Prilikom migracije sa jedne platforme na drugu, ove razlike zahtevaju promenu svih linija koda koje su specifične za jednu od platformi. Što je informacioni sistem veći, to je broj specifičnih linija veći i promena je složenija. Za jednostavnu migraciju sa jednog LDAP sistema na drugi, potrebno je razviti konfigurabilni interoperabilni adapter koji će lokalizovati sve pozive funkcija ka sistemima informacione-bezbednosti (uključujući i one specifične) i obezbediti pristup ka njima kroz generički sloj, dok bi se LDAP sistem odredio postavkom konfiguracije na jednom mestu.

U ovom radu istraženi su principi funkcionisanja sistema informacione-bezbednosti Microsoft Active Directory, zasnovan na Windows platformi i OpenLDAP, zasnovan na Linux platformi, njihove postavke, biblioteke za pristup i za rad sa navedenim platformama. Identifikovane su funkcije i strukture podataka za rad sa navedenim sistemima, njihove sličnosti i razlike. Dizajniran je i implementiran konfigurabilni interoperabilni adapter koji omogućuje rad sa obe bezbednosne platforme kroz generički sloj, dok se odabir platforme i bezbednosnog sloja konfiguriše na jednom mestu. Definisano i postavljeno testno okruženje koje obuhvata AD i OpenLDAP server i razvijena je testna aplikacija, kojom je verifikovan rad adaptera.

Ovaj rad je organizovan na sledeći način: u prvom poglavlju dat je uvod. Drugo poglavlje daje kratak pregled teorijskih osnova informacione-bezbednosti. U trećem poglavlju predstavljeno je predloženo rešenje interoperabilnog adaptera. Verifikacija rešenja data je u poglavlju četiri. U petom poglavlju dat je zaključak. Literatura je navedena u šestom poglavlju

2 TEORIJSKE OSNOVE

Informaciona-bezbednost predstavlja vrlo široku i multidimenzionalnu oblast jedne organizacije koja obuhvata ljude, procese i tehnologije. Osnovni cilj informacione-bezbednosti je da zaštiti karakteristike informacija koje imaju vrednost za organizaciju [1]: **a)** Poverljivost – zaštita pristupa podacima od neautorizovanih korisnika, **b)** Integritet – zaštita od neautorizovanih izmena i brisanja podataka, **c)** Dostupnost – mogućnost korisnika da neometano pristupa za koje ima ovlašćenja.

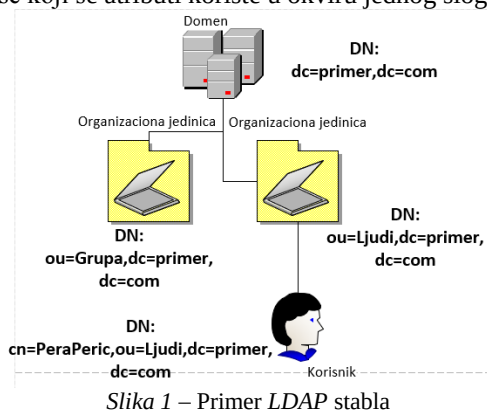
2.1 Kontrola pristupa

Kontrola pristupa oslanja se na pojmove subjekata, objekata, operacija i dozvola da se operacija izvrši. Subjekt je entitet koji ima aktivnu ulogu u sistemu bezbednosti, dokazuje svoj identitet, pristupa objektima i inicira prenos informacija do drugih objekata ili subjekata. Subjekt može da bude: autorizovani ili neautorizovani korisnik,

aplikacija, sistem ili mreža. Subjektu na nekoliko različitih načina može da se ograniči pristup objektu, a neka od tih ograničenja su: **a)** vreme pristupanja, **b)** lokacija autentifikacije subjekta, **c)** pristup van lokalne mreže, **d)** specijalna dodeljena prava pristupa. Objekat je entitet koji nema aktivnu ulogu kao subjekat. On prima ili čuva podatke, a objekat može da bude: aplikacija, mreža, fizički prostor za čuvanje memorije, a i sam podatak može da ima ulogu objekta [2]. Operacija predstavlja vrstu obrade informacije, dok dozvole predstavljaju pravila po kojima subjekat pristupa objektu. Svaki subjekat ima listu dozvola i ta lista je definisana u listi kontrole pristupa [2]. Kontrola pristupa definiše kojim objektima subjekat sme da pristupa, kakva su njegova prava i šta korisnik sme da radi sa tim objektima. Da bi se uspostavila potpuna kontrola pristupa, subjekat mora da prođe kroz tri faze: identifikacija, autentifikacija i autorizacija. Identifikacija subjekta je njegovo predstavljanje. Autentifikacija je dokazivanje identiteta korisnika. Postoje tri načina autentifikacije subjekta [2]: **a)** Nešto što subjekat zna (npr. lozinka), **b)** nešto što poseduje (npr. hardverski uređaj), **c)** nešto što jeste (npr. biometrijski otisak prsta). Autorizacija korisnika je mehanizam kojim se određuju privilegije subjekta i njegovi nivoi pristupa objektima. Autorizacija je moguća tek kada je subjekat identifikovan i autentifikovan. Evidentiranjem bezbednosno-informacionih događaja se prate, zapisuju i čuvaju aktivnosti na mreži u sistemu [2].

2.2 Lightweight Directory Access Protocol

LDAP je industrijski protokol standard zasnovan na Internet Protokolu za pristup i izmenu distribuiranih podataka u bazi informacionih podataka koja čuva specifične i uređene informacije o objektima [3]. **Slog** predstavlja sve podatke o jednom entitetu. Svaki slog se sastoji od tri glavne komponente: jedinstveno ime (*eng. Distinguished Name - DN*), objektnih klasa i atributa. Slogovi su organizovani u strukturu stabla. Primer *LDAP* stabla sa jednim domenom, dve organizacione jedinice i jednim korisnikom se nalazi na Slici 1. Više stabala koji su sačinjeni od slogova čine šumu. **Jedinstveno ime** je atribut kojim se na jedinstven način identifikuje svaki slog. *DN* se sastoji od relativnog jedinstvenog imena – *Relative Distinguished Name (eng. RDN)* i putanje u stablu, gde se slog nalazi. **Atributi** predstavljaju osobine subjekata i objekata, imaju svoj tip i vrednost ili listu vrednosti. Slogovi se sastoje od atributa. **Klasa objekata** je specijalan atribut zajednički za slogove u bazi, jedinstveno identifikovan objektnim identifikatorom, definiše koji se atributi koriste u okviru jednog sloga [3].



2.3 Microsoft Active Directory

Predstavlja implementaciju *LDAP* protokola kompanije *Microsoft*. Prvi put je implementiran na *Windows 2000* operativnom sistemu, a danas se koristi na *Windows Server* operativnim sistemima (poslednja verzija je *Windows Server 2019*). *AD* pruža bazu informacione-bezbednosti i tehnološko rešenje kontrole pristupa koje se koriste u ovom radu. Ono omogućuje upravljanje korisnicima, grupama, organizacionim jedinicama, štampačima, aplikacijama i servisima [4]. Pored toga *AD* ima i razne druge funkcionalnosti koje prevazilaze obim ovog istraživanja.

2.4 OpenLDAP

Predstavlja implementaciju bezbednosnog servisa na *Linux* operativnom sistemu. *OpenLDAP* pruža bazu informacione-bezbednosti, koja može da upravlja korisnicima, grupama i organizacionim jedinicama. Osim rada sa ovim entitetima, koji spadaju u osnovnu šemu baze podataka, *OpenLDAP* pruža mogućnost proširenja šeme i samim tim se otvara mogućnost za proširenje funkcionalnosti. Prilikom instalacije *OpenLDAP* sistema, inicijalno su uključene tri šeme koje obezbeđuju osnovne definicije za najčešće korišćene ekstenzije[5]:

- a) *Core.schema*,
- b) *Cosine.schema*,
- c) *Inetorgperson.schema*,

2.5 Interoperabilnost

U današnjem svetu raste potreba za složenim sistemima, koji su sačinjeni od više funkcionalnih celina. Svakom funkcionalnom celinom upravlja po jedan ili više servisa. Da bi sistem funkcionisao, a da korisnik ima osećaj kao da ga uslužuje jedan sistem, manji servisi moraju da funkcionišu skladno. Usklađenost ovih sistema je interoperabilnost. Postoje dve vrste interoperabilnosti [6]:

- a) Sintaksna interoperabilnost – sistemi komuniciraju i razmenjuju podatke, čak i ako im interfejsi ili programski jezici nisu isti.
- b) Semantička interoperabilnost – razmenjeni tipovi podataka su razumljivi u oba sistema.

3 PREDLOŽENO REŠENJE INTEROPERABILNOG ADAPTERA

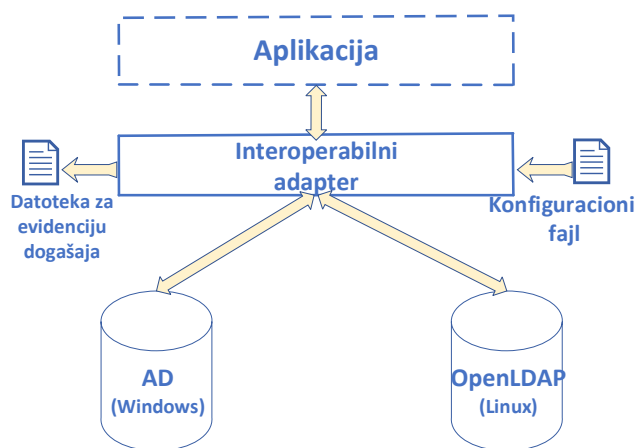
U ovom radu razvijen je interoperabilni adapter zasnovan na generičkom sloju, funkcionalan sa *AD* sistemom, na *Windows* i *OpenLDAP* sistemom na *Linux OS* platformi. Identifikovane su i istražene biblioteke za rad sa bezbednosno-informacionim sistemima funkcionalnostima koje nude i koje strukture podataka se koriste za razvijanje adaptera. Prilikom pokretanja adaptera konfiguriše se informaciono-bezbednosni sistem sa kojim adapter komunicira, a izmenom na samo jednom mestu se menja komunikacija ka drugom bezbednosnom sistemu, bez dodatnih izmena u postojećem kodu. Dizajn rešenja zasnovan je na Fabričkim dizajn paternom, koji od korisnika sakriva način pristupanja *AD* i *OpenLDAP* bezbednosnim sistemima.

3.1 Arhitektura rešenja

Arhitektura sistema, prikazana na Slici 2 se sastoji od sledećih komponenti: **a)** *AD* i *OpenLdap* sistemi,

b) aplikacija, c) konfiguracioni fajl, d) interoperabilni adapter, e) datoteka za evidenciju događaja.

AD i OpenLDAP su na slici prikazani kao dve baze podataka, zato što čuvaju informaciono-bezbednosne podatke. Implementirana je korisnička **aplikacija**, koja je povezana sa interoperabilnim adapterom i nudi korisnicima popunjavanje podataka za kreiranje novih, izlistavanje i modifikaciju postojećih slogova, pozivajući funkcije sa interoperabilnog adaptera. **Konfiguracioni fajl** sadrži neophodne podatke za pokretanje sistema, a to su: a) izbor bezbednosnog servisa, b) naziv ciljanog domena (npr. *primer.com*), c) IP adresa i port bezbednosnog sistema. **Interoperabilan adapter** je smešten na generičkom sloju i predstavlja centralnu komponentu sistema, zato što je povezan sa svim ostalim komponentama. Prilikom pokretanja sistema, iščitavaju se podaci iz konfiguracionog fajla i adapter uspostavlja konekciju sa željenim informaciono-bezbednosnim servisom. Kada se uspostavi konekcija, komande se šalju sa aplikacije, do adaptera, koji ih prosleđuje do bezbednosnog servisa, koji šalje odgovor nazad do adaptera. **Datoteka za evidenciju događaja** čuva zapis o poslednjem ulogovanom klijentu i vremenu kada se ulogovao.



Slika 2 – Arhitektura predloženog rešenja

3.2 Generički sloj

Interoperabilan adapter je zasnovan na generičkom sloju, funkcionalnom na različitim platformama OS - a. Da bi se postigla nezavisnost od platforme na koju se oslanja sistem, rešenje je implementirano na .NET Core radnom okruženju koje se sastoji od procesne virtualne mašine i kolekcije klasa i sam je nezavisan je od platforme OS - a na koju se oslanja.

3.3 Interoperabilni adapter

Za razvijanje interoperabilnog adaptera, prvo su istražene karakteristike bezbednosnih sistema: a) interfejsi za povezivanje sa AD i OpenLDAP bezbednosnim sistemima, b) tipovi podataka, c) način povezivanja podataka u informacionim bazama podataka, d) sličnosti i razlike tipova podataka i njihovih atributa između dva različita sistema. U sledećem koraku su istražene sličnosti i razlike dva bezbednosna sistema, razvijen je interoperabilan adapter koji je funkcionalan sa oba sistema. Promenom konfiguracije na samo jednom mestu može da se pristupi drugom sistemu.

3.4 Funkcionalnosti

AD u svojoj bazi podataka čuva informacije o korisnicima, grupama, računarima, kompjuterima, štampačima, aplikacijama i servisima. Za razliku od AD, OpenLDAP u svojoj osnovnoj šemi čuva podatke samo o korisnicima i grupama, pa su se prilikom implementacije rešenja realizovale funkcije za rad samo sa tim entitetima. Iako se imena entiteta na dva različita sistema poklapaju, imena tipova atributa su različita. Zato je izvršeno mapiranje atributa entiteta sa generičkog sloja, sa semantički jednakim atributima entiteta sa AD – a i OpenLDAP – a. Mapiranje je prikazano na Tabeli 1 i Tabeli 2. U Tabeli 1 se vidi mapiranje atributa korisnika, a u Tabeli 2 mapiranje atributa grupe.

Tabela 1 - Mapiranje korisničkih atributa generičkog sloja na atribute na AD i OpenLDAP atribute korisnika

Generički korisnik	AD korisnik	OpenLDAP korisnik
username	userPrincipalName	cn
name	givenName	givenName
surname	surname	sn
password	password	userPassword
phoneNumber	voiceTelephoneNumber	mobile
emailAddress	emailAddress	mail
description	description	description
	Distinguished-Name	Distinguished-Name

Tabela 2 - Mapiranje grupnih atributa generičkog sloja na atribute na AD i OpenLDAP atribute grupe

Generička grupa	AD grupa	OpenLdap grupa
groupName	name	cn
Members<string>	Members<Principal Collection>	Members<string>
description	description	description
	Distinguished-Name	Distinguished-Name

Za prikazane entitete razvijene su dve grupe funkcija: administratorske i korisničke.

Administratorske funkcije (Tabele 3 i 4) obuhvataju dodavanje, modifikovanje i brisanje iz baze podataka. Korisničke funkcije logovanje i proveru dozvole pristupa (Tabela 5).

Tabela 3 – Administratorske funkcije za manipulisanje korisnicima i grupama

Naziv funkcije	Opis
CreateUser	Kreiranje korisnika
DeleteUser	Brisanje korisnika
ChangeUserUsername	Izmena jedinstvenog imena korisnika
ChangeUserName	Izmena imena korisnika
ChangeUserSurname	Izmena prezimena korisnika
ChangeUserPassword	Izmena lozinke korisnika
ChangeUserEmail	Izmena adrese elektronske pošte korisnika
ChangeUserPhoneNumber	Izmena broja telefona korisnika
ChangeUserDescription	Izmena opisa korisnika
CreateGroup	Kreiranje grupe
ChangeGroupName	Izmena jedinstvenog imena grupe
DeleteGroup	Brisanje grupe

Tabela 4 – Administratorske funkcije

Naziv funkcije	Opis
AddUserToGroup	Dodavanje korisnika u grupu
AddGroupToAnotherGroup	Dodavanje grupe u drugu grupu
IsMemberOf	Proveravanje članstva
GetAllGroupMembers	Dobavljanje članova
CreateCustomPermission	Kreiranje nove dozvole
AssignCustomPermission	Dodeljivanje dozvole
ListAllUsers	Izlistavanje korisnika
ListAllGroups	Izlistavanje grupe

Tabela 5 – Korisničke funkcije

Naziv funkcije	Opis
Login	Logovanje
CheckPermission	Proveravanje dozvole

4 VERIFIKACIJA REŠENJA

4.1 Testno okruženje

Za testno okruženje u ovom istraživanju postavljeni su *Windows Server 2019* i *Linux* operativni sistem *Ubuntu 18.04*. Bezbednosni sistem *AD* je postavljen na *Windows Server OS*. Drugi sistem na kojem je testirano rešenje je *OpenLDAP*, verzija 2.4.48.

Korišćene su i sledeće verzije programskih biblioteka:

- a) *Novell.Directory.Ldap* 3.1.0,
 - b) *System.DirectoryServices* 4.6.0,
 - c) *System.DirectoryServices.AccountManagement* 4.6.0.
- Za verifikaciju korišćen je *Microsoft.NETCore* 2.1.

4.2 Testiranje

Za verifikaciju rešenja, razvijena je korisnička aplikacija na programskom jeziku *C#* koja omogućuje unos korisničkih i grupnih podataka za kreiranje novih entiteta, pretragu korisnika za modifikovanje i brisanje entiteta. Implementirana je i testirana funkcionalnost za pristup i izmenu informaciono-bezbednosnog sistema. Korisnička aplikacija poziva funkcije sa generičkog sloja i prikazuje stanje *AD* i *OpenLDAP* baze informacionih podataka.

Za verifikaciju generičkog sloja testni slučaj je obuhvatio:

- a) Kreiranje 50 novih korisnika i 5 novih grupa,
- b) Učlanjenje korisnika u grupe,
- c) Izmena atributa kod korisnika i grupa,
- d) Logovanje korisnika,
- e) Kreiranje i dodeljivanje dozvola pristupa,
- f) Provera dozvole pristupa.

Uspešno su testirane sve funkcionalnosti na oba bezbednosna sistema, a rezultat je verifikovan izlistavanjem svih korisnika i grupa, proverom članstva u grupama, a za funkcije logovanja korisnika su verifikovani fajlovi za praćenje logovanje korisnika.

5 ZAKLJUČAK

U današnje vreme bezbednost predstavlja jedan od najbitnijih faktora za uspešan rad softvera i poslovanje *IT* kompanija. U ovom radu istraženi su *LDAP* bezbednosni sistemi za upravljanje informacionim podacima o korisnicima, grupama i njihovim vezama. Za razumevanje informaciono-bezbednosnih sistema, potrebno je bilo da se definišu sledeći koncepti bezbednosti: kontrola pristupa, pojmovi subjekta i objekta i osnovni pojmovi i

uloga *LDAP* sistema. Za ovaj rad su izabrane dve različite implementacije prokola *LDAP*, *AD* zasnovan na *Windows* platformi i *OpenLDAP*, zasnovan na *Linux* platformi. Istraženi su interfejsi za pristupanje *LDAP* sistemima, dizajniran je i implementiran interoperabilan adapter koji je kompatibilan sa obe *LDAP* implementacije bezbednosnih sistema. Adapter se nalazi na generičkom sloju, funkcionalnom na oba *OS* – a. Kompanijama je omogućen izbor između platforme za rad sa informaciono-bezbednosnim sistemima i jednostavna promena platforme, sa kojom će funkcionalnosti za upravljanje informacionim podacima ostati nepromenjene.

Generički sloj se može proširiti i unaprediti na nekoliko načina. Jedan pravac daljeg istraživanje jeste da se omogućiti pristup još jednom informaciono-bezbednosnom sistemu. Dodatni sistem bi proširio opseg mogućih korisnika i usluga ovog rešenja. Drugi pravac bi bio da se omogućiti proširenje postojeće šeme na *AD* – u i *OpenLDAP* – u, čime bi novi tipovi entiteta kreirali u toku rada adaptera.

6 LITERATURA

- [1] M. E. Whitman and H. J. Mattord, *Principles of Information Security Fourth Edition*, no. January 2015., Course Technology, 20 Channel Center Boston, MA 02210 USA, 2011.
- [2] E. Harold F. Tripton, *Official (ISC) Guide To The SSCP CBK*. 1385., CRC Press-Taylor and Francis Group, 2011.
- [3] S. Tuttle et al., *Understanding LDAP Design and Implementation*, IBM RedBooks, 2006.
- [4] B. Desmond, J. Richards, R. Allen, and A. Lowe-Norris, *Active Directory, 5th Edition*, vol. 91, no. 5. 2012, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, April 2013.
- [5] M. Butcher, *Mastering OpenLDAP - Configuring, Securing and Integrating Directory Services.*, Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK, 2007.
- [6] Techopedia, "Interoperability.". Accessed on: Oct. 22, 2019. [Online]. Available: <https://www.techopedia.com/definition/631/interoperability>.

Kratka biografija:



Aleksandar Maričić rođen je 30.5.1995. godine u Zrenjaninu. Završio je Gimnaziju u Zrenjaninu 2014. godine. Fakultet tehničkih nauka u Novom Sadu je upisao 2014. godine, a Osnovne akademske studije završio je 2018. godine. Ispunio je sve obaveze i položio je sve ispite predviđene studijskim programom.

ARHITEKTURA MJERNO-INFORMACIONOG SISTEMA U TELEMEDICINI**ARCHITECTURE OF MEASUREMENT AND INFORMATION SYSTEM IN TELEMEDICINE**

Jelena Ećimović, *Fakultet tehničkih nauka, Novi Sad*

Oblast- ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj- Ovaj rad se bavi arhitekturom mjerno-informacionog sistema u telemedicini. Prvi dio rada se odnosi na telemedicinu, njenu isoriju, podjelu i razvoj. Drugi dio rada se odnosi na primjenu telemedicine na već postojećim zdravstvenim sistemima. Treći dio rada obuhvata arhitekturu mjerno-informacionog sistema u telemedicini i njeno testiranje. Završni dio rada se odnosi na podatke, medicinsku etiku i na kraju rada su data završna razmatranja.

Ključne reči: Telemedicina, Mjerno-informacioni sistem, Arhitektura sistema.

Abstract- This paper is about the architecture of measurement and information system in telemedicine. The first part of the paper is about the history of telemedicine. The second part is about the application of measurement and information system in telemedicine to already existing healthcare systems. The third part of the paper contains the architecture and testing of the measurement and information system in telemedicine. The final part of the paper is data and medical ethics, final considerations and a list of used literature.

Keywords: Telemedicine, measurement and information system, system architecture.

1. UVOD

Nedavni razvoj digitalnih tehnologija otvara put sve većoj upotrebi informacione tehnologije i sistema zasnovanih na podacima u medicinskoj i zdravstvenoj praksi. Ovaj rad opisuje kako informacione i komunikacione tehnologije, internet, bežične tehnologije i bežične mreže, baze podataka i telemetrija omogućavaju prenos informacija i kontrolu informacija unutar medicinskog centra i između medicinskih centara. Na primjer, jednostavna udaljena bazna jedinica može kontinualno da prikuplja i lokalno integriše mnoge dolazne signale, kao što su elektrokardiografija, zasićenost kiseonikom, brzina otkucaja srca, neinvazivni krvni pritisak, temperatura i disanje i pruži informacije potrebne za otkrivanje mogućih hitnih slučajeva za pacijenti. Medicinski centar tada može da primi sve komplementarne i glomazne sisteme, uključujući telemedicinsku opremu kao što su jedinice intenzivne njege, inteligentni analizatori i automatski snimači i profesionalno vođen sistem baza podataka koji podržava profesionalni provajder [5].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Platon Sovilj.

2. UOPŠTENO O TELEMEDICINI**2.1 Podjela telemedicine**

Današnja tehnologija omogućava da se klinički procesi odvijaju na daljinu. Tehnologija je sredstvo koje to omogućava, ali sama po sebi tehnologija nije telemedicina. Telemedicina se može smatrati zadatkom koji medicinsko osoblje ili ljekar obavlja (kao što je posmatranje, konsultacije, tumačenje i pružanje mišljenja) uz pomoć informacionih i komunikacionih tehnologija u okolnostima kada su učesnici fizički razdvojeni [5-7].

Podjela telemedicine se vrši na osnovu načina kako se ona može koristiti tako da se može govoriti o sledećim telemedicinskim servisima:

- telekonsultacije (obezbeđenje daljinskog pristupa bilo medicinskim stručnjacima bilo informacijama smeštenim u elektronskim bazama podataka).
- teledijagnostika (postavljanje dijagnoze pacijentu od strane lekara koji nema direktan kontakt sa pacijentom, a uz pomoć medicinskih informacija (nalaza, slika, video snimaka).
- telemonitoring (daljinsko praćenje fizioloških parametara pacijenata, najčešće kod hroničnih pacijenata koji ne zahtevaju stalan bolnički nadzor).
- telenega (terapija pacijenata van fizičkih okvira zdravstvene ustanove).
- teledukacija (obuka i vježba medicinskog osoblja van ustanove koja ih provodi, daljinski pristup bazama podataka preko telekomunikacionih kanala (Internet) [6].

Ako medicinu gledamo kao na specijalnosti onda primenom telemedicine na te specijalnosti dobijamo sledeću podjelu, ali bitno je napomenuti da zapravo ne postoje ograničenja da se telemedicina primjeni u svim oblastima medicine:

- teleradiologija (dijagnostika, edukacija...).
- telepatologija (postavljanje dijagnoze, edukacija...)
- telehirurgija (robotska hirurgija upravljana daljinski od strane hirurga).

2.2. Istorija telemedicine

Istorija moderne telemedicine povezana je sa istorijom telekomunikacija i informatike. Značajni momenti u razvoju telekomunikacija i informatike, koji su istovremeno značajni i za razvoj telemedicine, su pronalasci i uvođenje: električne telegrafije (sredinom 19-og veka), telefona i radio prenosa (krajem 19-og veka), televizije (prva polovina 20-og veka), satelitskih telekomunikacija (druga polovina 20-og veka), uvođenje digitalnih umesto ana-

lognih telekomunikacija (druga polovina 20-og vijeka), Interneta i mobilne telefonija (krajem 20-og vijeka).

Jedan od događaja koji je povezan sa stvaranjem moderne telemedicine odigrao se sa Belovim otkrićem telefona. Kada je njegov telefon prvi put proradio, Bel je vikao u ovu spravu: „Votson, dođi ovamo! Trebaš mi.“ Bel je pokušavao da dozove u pomoć svog pomoćnika Votsona, nakon što iz oborene baterije prosuo sumpornu kiselinu po svojoj ruci. Početkom dvadesetog vijeka putem telefonske linije izvršen je prenos elektrokardiograma (EKG) i elektroencefalograma (EEG). Telefon je i danas ostao najčešće korišćeno tehnološko sredstvo za telemedicinske aktivnosti kao što je npr. pozivanje hitne pomoći ili dobijanje savjeta od zdravstvenog radnika. Prvi medicinski savjet pomorskim putnicima, korišćenjem radio talasa, ostvaren je 1920. godine [5].

2.3. Razvoj telemedicine

Uvođenje Interneta u obična domaćinstva i razvoj usluge kratkih poruka (SMS) 1990-ih proizveli su najranije alate za telehealth. Medicinski i inženjerski stručnjaci su nastojali da integrišu dostupnu tehnologiju sa zdravstvenom zaštitom kako bi pružili bolje usluge. Kako su dostupnost i brzina Interneta poboljšani, stvorene su različite web aplikacije dostupne PC računarima kako bi se pružila podrška pacijentima kod kuće da upravljaju svojim vlastitim uslovima. Do sredine 2000-ih mobilni telefoni su uključivali mogućnost pristupa internetu, a istovremeno su i laptop računari kao i lični digitalni asistenti (PDA) postali široko dostupni. Mobilni telefoni i PDA bili su izbor mnogih studija telemedicine zbog njihove mobilnosti i mogućnosti pristupa internetu [5].

3. PRIPREMA MEDICINSKIH SISTEMA ZA TELEHEALTH

3.1. Tehnološki zahtjevi za mjerno-informacione sisteme u telemedicini

Da bi se analizirali razvojni aspekti pružanja zdravstvenih usluga na daljinu pomoću novih tehnologija, proces je raspoređen u tri paralelna komplementarna dijela kao što su eHealth tehnologija, TLM tehnologija i TLH tehnologija. Proces tehnološkog razvoja omogućen TLH-om je servisna platforma kombinovanih medicinskih alata i sistema, standarda komunikacijskih usluga, sistemskih mogućnosti i raspoloživosti usluga koji mogu garantovati pouzdan, fleksibilan i nesmetan protok usluge potrebnom brzinom i ugovorenim kvalitetom.

Zbog toga mjerno-informacione usluge vode računa o medicinskim i profesionalnim zdravstvenim aplikacijama za udaljene klinike. Proces tehnološkog razvoja omogućen telehealthnim zdravljem osigurava da su sve usluge dizajnirane za finansijski održive scenarije aplikacije. Na osnovu tipa interakcije između zdravstvenih radnika/medicinskog osoblja i korisnika zdravstvene zaštite, telemedicina može biti:

- Sinhrona (interaktivna, u realnom vremenu)- zasnovana na komunikaciji zdravstvenog radnika i pacijenta ili komunikaciji dva zdravstvena radnika u relanom vremenu, npr. putem direktne dvosmjerne audio komunikacije ili videokonferencije.
- Asinhrona (sačuvaj-i-proslijedi)- zasnovana na čuvanju i prenosu (sa jedne lokacije na drugu) zapisa raz-

ličitog formata kao što su tekst, slike (npr. radiografija), audio ili video zapisa (video klipovi). Ne zahtijeva neposrednu komunikaciju zdravstvenih radnika i pacijenata. Najčešće se radi o prenosu radiografskih slika ili slika patohistoloških preparata.

Što se tiče infrastrukturnih zahtjeva za pružanje zdravstvenih i pridruženih medicinskih usluga, kategorički se može razmotriti dvosmjerna razmjena informacija radi pružanja bilo koje aplikacije utemeljene na scenariju sa dva osnovna dijela:

- Telekomunikacione tehnologije za pouzdan pristup koje uklanjaju sve moguće brige povezane sa udaljenošću i
- kolekcijom interaktivnih medicinskih uređaja i sistema (senzor, aktuator i akcelerometar).

3.2. Sadašnji izazovi i prednosti

Mnoga dostignuća u telekomunikacijama i zdravstvu se uglavnom temelje na korištenju digitalnog medija za zamjenu tradicionalnih fizičkih načina zapisivanja. Neki od dobro poznatih primjera uključuju rukovanje rendgenskim snimcima, koji su se nekada prikazivali fotografskim filmom, a sada su prikazani digitalno. Digitalno čitanje smanjuju potrebu za papirom i istovremeno pomažu u osiguravanju čitljivosti nalaza i smanjenju grešaka koje nastaju zbog loše čitljivosti medicinske dokumentacije. Modeli razmjene podataka koji se temelje čuvanju podataka u Cloud-u često su spomenuti u istraživanjima i od strane različitih IT rješenja u medicinskom sektoru. Međutim, istovremeno, ovaj novi model interakcije stvara puno novog posla i podataka u procesu. Postojeća fragmentacija u postupcima snimanja između različitih zdravstvenih ustanova i dalje je najveća prepreka koju treba riješiti prije nego što pacijenti zaista iskoriste njene pogodnosti [5].

3.3 Problemi sa primjenom

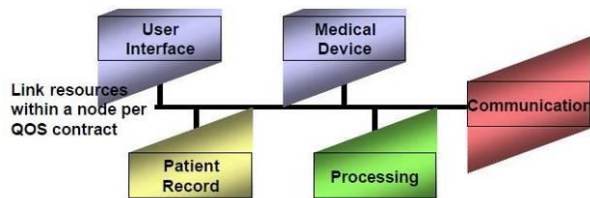
Pored jasne koristi koje nudi telemedicina, ona istovremeno ima i određene nedostatke:

- Teško dostupna telekomunikaciona oprema visokog kapaciteta upravo tamo gdje je telemedicina najpotrebnija, npr. u ruralnim predjelima. Ovo može značajno povećati inicijalne troškove razvoja mjerno-informacionog sistema. Jedno od rješenja je upotreba bežične infrastrukture u tim područjima.
- Nedovoljno razrađeni pravni aspekti rada u uslovima telemedicine. Između ostalog, problemi sa licencama za rad ljekara ako se pacijent nalazi u jednoj državi, a ljekar u drugoj državi.
- Nemogućnost da ljekar obavi pregled pacijenta kao u uobičajenim uslovima. Korišćenje telefonske linije omogućava prenos samo audio informacija. Dvosmjerna audio-vizuelna veza omogućava prenos i audio i vizuelnih informacija, ali ne omogućava perkusiju, što može biti od ključnog značaja za postavljanje nekih dijagnoza ili stanja, npr. akutnog abdomena. Nije mogućen ni prenos mirisa. Ljekar takođe nije u mogućnosti da izvodi određene testove ili manipulacije sa pacijentom, npr. ispitivanje mišićne snage, ili ispitivanje refleksa. Sve ovo može umanjiti pouzdanost dijagnoza u telemedicini u odnosu na uobičajeni način rada u medicini.

- Nedovoljna spremnost jednog dijela zdravstvenih radnika da primjenjuju telemedicinu zbog potrebe za dodatnom edukacijom, male materijalne naknade ili straha od gubitka pacijenata.
- Održavanje zdravstvene dokumentacije je komplikovanije u telemedicini [3].

3.3 Tehnički izazovi i prolagođavanje mjerno-informacione usluge sistema na već postojećem servisu

Hostovanoj platformi, obično pametnom telefonu ili računaru, često nedostaje poželjni bežični komunikacijski hardver koji bi komunicirao sa uređajem. To ograničava mogućnosti bežične komunikacije: ili samo korisnici koji imaju uređaj sa odgovarajućim bežičnim prijemnikom mogu koristiti uslugu ili usluga mora isporučiti odgovarajući adapter. Pored toga, postojeće mrežne mogućnosti mnogih bolnica i klinika nisu spremne pacijentima omogućiti pristup njihovim podacima, posebno ne potencijalno velikom broju korisnika koji pokušavaju pristupiti podacima i omogućiti pristup podacima iz spoljašnje mreže kao i da bi zaštitili podatke (slika 1).



Slika 1. Kako bi trebalo da izgleda mrežna sposobnost bolnice

Nekoliko specifikacija o kojima bi trebalo razgovarati između medicinskih i inženjerskih timova kako bi se osigurala koncepcija dobre telekomedicinske usluge. Prvo je pitanje napajanja. Ako usluga uključuje elektroniku u bil Jelena Ećimović o kojem obliku, to je neizbježno. Napajanja imaju ograničeno punjenje, te težine i prostorne troškove. Za medicinsko osoblje koje je na terenu, glomazni uređaj je nepoželjan dodatak već brojnim alatima kojima moraju upravljati.

Alternativno, napajanje može imati kraće vrijeme zauzvrat, kako bi se smanjila težina i prostor, mada to eventualno znači da rezervne ćelije moraju biti pripremljene. Ovdje je ključno uravnotežiti te faktore, na osnovu postupka koji će medicinsko osoblje izvoditi koristeći telemedicinsku uslugu ili uređaj.

Bez mogućnosti prenošenja podataka u nekom obliku, pružena usluga/uređaj bila bi samo tradicionalna medicinska aplikacija. Izbor komunikacionog pristupa je presudan jer se direktno vezuje za izbor napajanja o kome se upravo raspravljalo.

4. STRUKTURA MJERNO-INFORMACIONOG SISTEMA U TELEMEDICINI

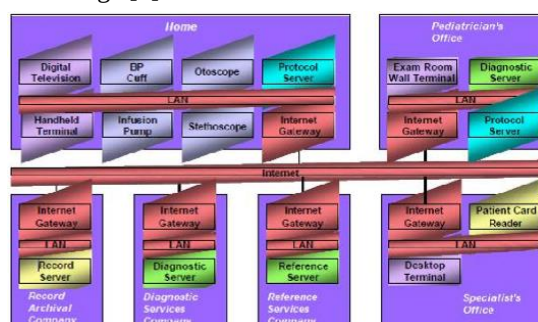
4.1. Arhitektura mjerno-informacionog sistema

Arhitektura mjerno-informacionog sistema u telemedicini (telekomunikacione i informacione tehnologije) je pored telemedicinskih servisa druga osnovna komponenta telemedicinske tehnologije.

Prije izbora opreme neophodno je razmotriti prirodu informacija koje treba premjestiti sa jednog mjesta na

drugo, kao i količinu informacija i bezbjednost informacija. Oprema mora imati tehničke karakteristike koje će omogućiti prenos onog tipa informacija koji je predviđen planom mjerno-informacionog sistema. Neophodno je da oprema, kao i svaki drugi medicinski uređaj, ima visoku pouzdanost u svom radu. Oprema je često, dijelom ili kompletno, već prisutna prije planiranja mjerno-informacionog sistema (npr. telefonska linija). Poželjno je inicijalnu nabavku opreme uskladiti sa unaprijed postavljenim zdravstvenim ciljevima, i na taj način izbjeći nepotrebne troškove.

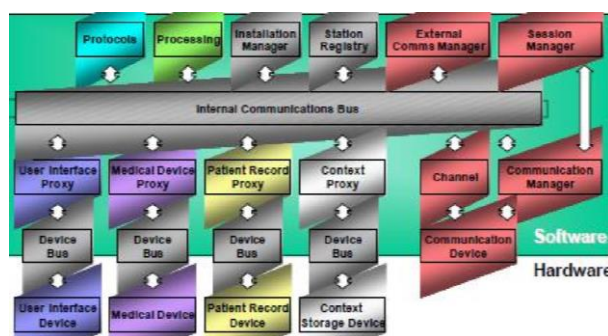
Umjesto zatvorene mreže u kojoj se nalaze sve usluge unutar stanica mreže, ovaj sistem se sastoji od skupa usluga koje su distribuisane na čitav niz uređaja i lokacija (slika 2). Baš kao što uređaji unutar stanice moraju moći da se lociraju i međusobno udružuju. Stanice unutar mreže bi takođe trebale biti u mogućnosti da se udružuju međusobno radi obavljanja kliničkih funkcija. Da bi se to postiglo, mjerno-informacioni sistemi moraju da podržavaju: lokacije resursa, sticanje resursa, konferencije, kvalitet usluge [4].



Slika 2. Arhitektura mjerno-informacionog sistema

4.2 Arhitektura mjerno-informacione stanice

Na slici 3 je prikazana arhitektura mjerno-informacione stanice koja podržava plug and play režim za komponente iz kojih je sastavljena. Glavni cilj ovog dizajna je omogućavanje ad hoc sastavljanja sistema. U središtu ovog dizajna su interna komunikaciona magistrala (internal communications bus) i registar stanice (station registry). Interna komunikaciona magistrala predstavlja one mehanizme unutar stanica koji se koriste za razmjenu informacija između komponenti [4].



Slika 3 Logička arhitektura mjerno-informacione stanice

5. TESTIRANJE SPECIFIKACIJA SISTEMA

Da bi se pokazala prava funkcionalnost mjerno-informacionog sistema neophodno je izvršiti reviziju i adekvatno testiranje sistema. Samo testiranje se može podijeliti u devet faza. Svaka faza testiranja se sastoji od neko-

liko etapa koje se obuhvataju svrhu, pristup i rezultat testiranja. Faze koje su potrebne za testiranje sistema su:

- Faza 1- obuhvata provjeru rada na onim dijelovima arhitekture koji se odnose na lokalne medicinske uređaje.
- Faza 2- podrazumjeva rad na osnovnim operacijama od stanice do stanice.
- Faza 3- rad na korisničkom interfejsu i obrada aspekata arhitekture.
- Faza 4- testiranje rada kopponenti implementacije stanice.
- Faza 5- dodavanje mogućnosti snimanja podataka pacijenata na stanici.
- Faza 6- inkorporacija sigurnosnih mehanizama u stanici.
- Faza 7- rad na naprednim mogućnostima konfiguracije stanice.
- Faza 8- dodavanje funkcije administracije stanice u sistem.
- Faza 9- ova faza rada bavi se operacijama u kojima je uključeno više od dvije stanice u datoj sesiji [4].

6. PODACI

6.1 Elektronski zdravstveni karton pacijenta

Elektronski zdravstveni karton pacijenta (electronic health record-EHR) je sistematsko prikupljanje i čuvanje podataka o zdravlju u digitalnom formatu. Ovi zapisi se mogu koristiti u različitim oblicima zdravstvene zaštite. Zapisi se dijele putem povezanih informacionih sistema koji su povezani sa mrežom ili drugim informacionim sistemima. EHR-ovi mogu obuhvatati niz podataka, uključujući demografiju, istoriju bolesti, lijekove i alergije, status imunizacije, rezultate laboratorijskih ispitivanja, radiološke slike, vitalne funkcije, lične podatke kao što su starost, težina, i podatke o zdravstvenom osiguranju [1].

6.2 Sistemi rada i protoka rada u arhitekturi Clouda

Preduzecima poput medicinskih ustanova i bolnica, kao i naučnim, industrijskim i istraživačkim organizacijama su obično potrebna velika skladišta za održavanje arhiva, koje imaju velike količine podataka. Stoga je izvodljivo da se ova aktivnost može prenijeti na centre podataka. U mnogim slučajevima hijerarhijsko naređivanje zasnovano na upotrebi zajedno sa inteligentnim uslužnim programima za pretraživanje sa podrškom za keširanje, poboljšava dostupnost podataka [2]. U Cloud okruženju programer iznajmljuje uređaje za centre podataka, štedeći vreme za istraživanje i inovativni razvoj od rukovanja rutinskim poslovima [3].

7. MEDICINSKA ETIKA

Etički kodeks za medicinske i zdravstvene informatičare (MZI) mora biti jasan, nesumnjiv i lako primjenjiv u praksi. Štaviše, s obzirom na to da se informatičko područje konstantno mjenja, kodeks mora da bude fleksibilan kako bi se prilagodio promjenama bez žrtvovanja primjenjivosti osnovnih principa. Zato je neprimjereno da se etički kodeks za MZI-e bavi specifičnostima svake pojedine situacije koja se može dogoditi.

To bi kodeks učinilo previše uskim, previše rigidnim i zavisnim od postojećeg statusa u informatici. Umjesto toga kodeks mora da se usredsredi na etičke pozicije MZI-a kao stručnjaka, i na odnos između MZI-a i drugih strana s kojima su u međuo odnosu tokom svog profesionalnog rada [7].

8. ZAKLJUČAK

Često se sreće usluga ili uređaje za mjerno-informacione svrhe u telemedicini koje su vrlo dobre namjene, ali tada obično potrebna ključna infrastruktura za uslugu nije prisutna. Namjera da se domet medicinske zaštite proširi na veće i ruralnije regione često podleže činjenici da internetska i komunikaciona infrastruktura ne postoji u ciljnoj regiji ili da su signali nestabilni, čineći da preko komunikacione veze nije moguće pokrenuti uslugu.

Kao i sva nova tehnološka dešavanja, početna primjena se gotovo uvijek susreće sa skepticizmom. Sa sve većim pokazateljima efikasnosti, sve više ljudi prihvata upotrebu mjerno-informacionih usluga i sistema u telemedicini. Koristeći pametne telefone i tablete kao platformu domaćina, mnoge bolnice su rasporedile svoje konsultantske usluge izvan granica ustanove. U budućnosti se očekuje još veći razvoj i primjena mjerno-informacionih sistema u telemedicini.

9. LITERATURA

- [1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," 17th Int. Workshop Quality Service IWQoS, 2009.
- [2] K. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," RSA Laboratories, Cambridge, Massachusetts 2013
- [3] P. Feresten, "Storage multi-tenancy for cloud computing," SNIA, Mar. 2010.
- [4] Rick Craft. "Telemedicine System Interoperability Architecture- Concept Description and Architecture Overview" Sandia National Labs, PO Box 5800, Albuquerque, NM, 87185-0839, USA, 2009
- [5] Halit Eren and Jhon G. Webster. "Telemedicine and electronic medicine" 2018
- [6] Institut za onkologiju Vojvodine, dostupno na <http://www.onk.ns.ac.rs/cir-telemedicina.htm> datuma: 10.11.2019
- [7] Prvi telemedicinski sajt u Srbiji, dostupno na <http://www.telemed.rs/> datuma: 10.11.2019

Kratka biografija:



Jelena Ećimović rođena je u Sokocu 1994. godine. Diplomirala je na Fakultetu tehničkih nauka u Novom Sadu, na smjeru Biomedicinsko inženjerstvo 2017. godine. Master studije upisuje 2018. godine na smjerovima Biomedicinsko inženjerstvo i Merni sistemi (Energetika, elektronika i telekomunikacije). Masterirala je 2019. godine, na smjeru Biomedicinsko inženjerstvo. Stipendista je firme "Schneider Electric DMS NS".

**STUDIJA PODEŠENJA ELEKTRIČNIH ZAŠTITA TRANSFORMATORSKE STANICE
33/220kV VETROELEKTRANE KOVAČICA****STUDY OF RELAY PROTECTION SETTINGS FOR POWER SUBSTATION
33/220kV KOVAČICA WIND FARM**

David Ivanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je rađena studija podešenja relejne zaštite na jednom savremenom projektu od velike važnosti za elektroenergetski sistem Srbije – vetroelektrana „Kovačica“. Detaljno je odrađen proračun podešenja zaštitnih funkcija vetroelektrane u 220kV postrojenju na zaštiti energetskih transformatora. Zatim zaštita u 33kV postrojenju na transformatorskim čelijama, izvodnim čelijama ka vetrogeneratorima i spojnim čelijama su prikazana. Svi neophodni tehnički podaci kao i kratak opis vetroelektrane su dati. Odrađen je i opisan proračun podešenja a na osnovu tehničkih preporuka, tipskih podešenja, odgovarajućih tehničkih upštava dobijeni su krajnji rezultati koji su tabelarno predstavljeni.

Ključne reči: *Relejna zaštita, Zaštitne funkcije, Parametri*

Abstract – This paper deals with the study of relay protection settings on a modern project of great importance for the power system of Serbia – wind farm „Kovačica“. The calculation of the settings of wind power protection functions in a 220kV switchyard on the protection of power transformers has been done in detail. Subsequently, protection in 33kV switchgears on transformer switchgears, feeder switchgears to wind turbine generators and connection switchgears are shown. All the necessary technical data as well as a brief description of the wind farm are given. The calculation of the settings was made and described, and based on the technical recommendations, type settings, corresponding technical instructions and final results were presented in tables.

Keywords: *Relay protection, Protection functions, Parameters*

1. UVOD

Elektroenergetski sistem je skupa investicija koja treba da se dizajnira, sprovede i upravlja sa velikom oprežnošću tako da pruži pouzdano i sigurno rešenje. Transformatorske stanice i razvodna postrojenja su jedna od komponenti investicija koja zahtevaju detaljan inženjerski rad od faze dizajniranja do faze implementacije. Ako odgovarajući dizajn sistem zaštite ne sledi, struje kvara abnormalnih stanja u par sekundi mogu da oštete komponente tih velikih investicija. Prema tome, tokom dizajniranja potrebno je da se izvrši detaljna analiza sistema.

NAPOMENA:

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

Na osnovu rezultata analiza, sistem zaštite, oprema i tehnika treba ispravno da se definiše pre faze implementiranja. To uključuje selekciju odgovarajućih prekidača, osigurača, izolatore, merne transformatore, zaštitne releje i tako dalje.

Štaviše, pravilna koordinacija među tom opremom je presudan zadatak koji treba da izvrše inženjeri zaštite i upravljanja. U ovom radu izvršen je detaljan proračun i implementacija relejne zaštite na konkretnoj savremenoj transformatorskoj stanici vetroparka. Selekcija odgovarajuće zaštitne opreme je jedna od ključnih koraka za ispravno funkcionisanje relejne zaštite. U drugoj glavi opisane su zaštitne funkcije glavne zaštite, uređaj za automatsku regulaciju napona i rezervne zaštite u 220kV postrojenju. U trećoj glavi opisane su funkcije zaštite na 33kV postrojenju na transformatorskim, izvodnim i spojnim čelijama. Izvršen je proračun i određeni su parametri podešenja prikazani tabelarno na kraju rada. U petoj glavi data je korišćena literatura.

**2. ZAŠTITNI UREĐAJI I PRORAČUN PODEŠENJA
TRANSFORMATORSKIH POLJA U 220 kV
POSTROJENJU**

Ovim delom obuhvaćeni su sledeće dve zaštite i njihove zaštitne upravljačke funkcije od kojih će samo neke biti opisane u nastavku u ovom radu:

Glavna zaštita (F302) transformatora je tipa ABB RET670 i sadrži sledeće zaštitne funkcije:

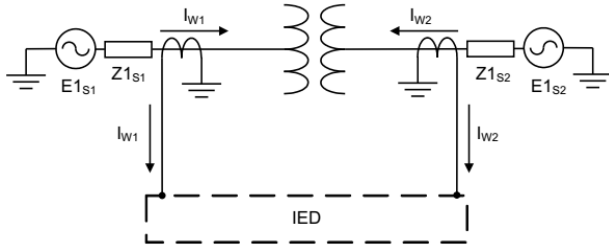
- 87T Diferencijalna zaštita transformatora.
- 87N Ograničena zemljospojna zaštita (Visokonaponska strana u daljem VN strana).
- 50/51 Prekostrujna i kratkospojna zaštita (VN strana).
- 46 Zaštita od nesimetričnog opterećenja.
- 51N Zemljospojna zaštita (VN strana).

Uređaj za automatsku regulaciju napona i rezervna zaštita (A202) je tipa ABB RET650 i sadrži sledeće zaštitne i upravljačke funkcije:

- 87T Diferencijalna zaštita transformatora.
- 87N Ograničena zemljospojna zaštita (Niskonaponska strana u daljem NN strana).
- 51 Prekostrujna zaštita (NN strana).
- 51 Prekostrujna zaštita neutrale 33kV.
- 90; 84 Automatska regulacija napona; Upravljanje i nadzor regulacione preklopke.

Diferencijalna zaštita (87T)

Diferencijalna zaštita, uz Buholc i ograničenu zemljospojnu zaštitu, predstavlja osnovnu zaštitu energetskog transformatora 220/33kV – slika 1.



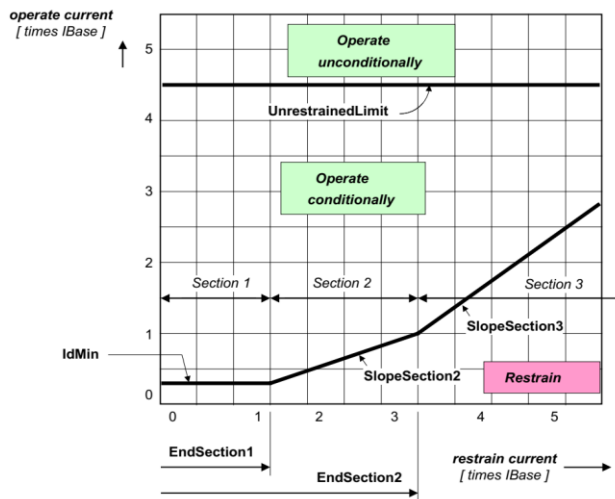
Slika 1 – Shema diferencijalne zaštite transformatora

Algoritam diferencijalne zaštite obavlja normalizaciju prenosnog odnosa strujnih mernih transformatora, kompenzaciju faznog pomeraja, kao i stabilizaciju uticaja viših harmonika struje magnetizacije. Algoritam sadrži i napredni diskriminator spoljnih i unutrašnjih kvarova na osnovu merenja struje inverzne komponente.

Diferencijalna zaštita ima dva stepena (slika 2):

stabilisani (niži stepen, restrained) – $I_d >$,

nestabilisani (brzi, viši stepen, unrestrained) – $I_d \gg$.



Slika 2 – Karakteristika diferencijalne zaštite transformatora

Podešavanje diferencijalne struje prorade definiše se kao procenat bazne struje (I_{Base} , definiše se zajedno sa baznim vrednostima napona i prividne snage, V_{Base} i S_{Base} , u okviru grupe podešenja Global base values). Za vrednost bazne struje se usvaja nominalna struja transformatora (na VN strani). Stabilisani stepen ima takvu karakteristiku (slika 2, donja karakteristika) da je diferencijalna struja prorade (I_d) zavisna od stabilizacione struje (I_{bias}), pri čemu se vrednosti ovih struja određuju na sledeći način:

$$I_d = |I_1 + I_2| \quad (2.1)$$

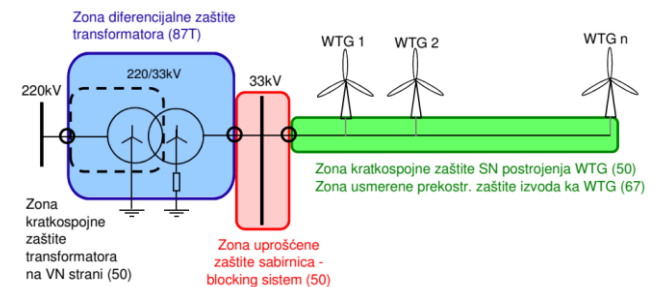
$$I_{bias} = \max(|I_1|, |I_2|) \quad (2.2)$$

gde I_1 i I_2 predstavljaju struje pojedinačnih namotaja transformatora. Karakteristika se sastoji iz tri sekcije, jedne

horizontalne (I_{dMin}) i dve pod nagibom (SlopeSection2, SlopeSection3) i definiše se uz pomoć parametara I_{dMin} , EndSection1, EndSection2, SlopeSection2, SlopeSection3.

Kod podešenja minimalne proradne struje I_{dMin} uzima se u obzir greška koju unose strujni merni transformatori (koja je relativno mala u slučaju male vrednosti diferencijalne struje u početnom delu karakteristike) kao i greška koju unosi regulaciona sklopka (ukoliko se informacija o položaju sklopke ne dovodi u zaštitni uređaj, što je ovde slučaj). Podešenje minimalne proradne struje zavisi od tipa energetskog transformatora, u ovom slučaju transformator sa regulacijom pod opterećenjem, preporuka proizvođača zaštitnog uređaja za podešenje parametra I_{dMin} a i EMS-ova dosadašnja praksa za transformatore sa regulacijom pod opterećenjem je $\Delta I=30\%$.

U okviru sekcije 2, vrednosti struja transformatora su veće od nominalnih i greška koju unose strujni merni transformatori je srazmerno veća, te je i proradna vrednost diferencijalne struje veća i povećava se sa porastom stabilizacione struje. U okviru sekcije 3, ova greška je još više izražena jer dolazi do potencijalnog zasićenja strujnih transformatora, pa je i nagib karakteristike veći nego u drugoj sekciji. Za podešenja parametara duge i treće sekcije karakteristike stabilisanog stepena, usvajaju se preporuke proizvođača zaštitnog uređaja. Delovanje diferencijalne zaštite transformatora je trenutno. Na slici 3 je prikazana zona delovanja diferencijalne zaštite transformatora, kao i zone delovanja ostalih zaštita od međufaznih kvarova u postrojenjima 220kV i 33kV.



Slika 3 – Zone delovanja diferencijalne zaštite i zaštita od međufaznih kvarova u postrojenjima 220kV i 33kV

Karakteristika nestabilisanog stepena ($I_d \gg$) diferencijalne zaštite je horizontalna i definiše se kao procenat bazne struje (I_{Base}), bez uticaja stabilizacione struje. Ova karakteristika se koristi za veoma visoke vrednosti diferencijalne struje, za koje se sa sigurnošću može tvrditi da su posledica unutrašnjeg kvara transformatora.

Podešenje ovog stepena treba izabrati tako da bude ispunjen uslov:

$$(I_d \gg) < I'_{3KS} = \frac{c \cdot U_L}{\sqrt{3} \cdot (Z_{m220} + Z_{TR220})} \quad (2.3)$$

gde su:

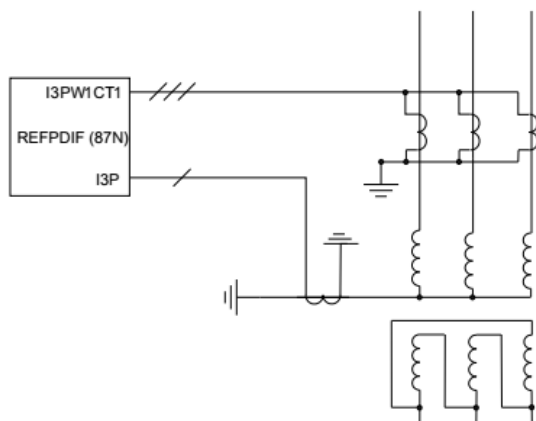
I'_{3KS} - struja trofaznog kratkog spoja na VN strani transformatora,

Z_{m220} - ekvivalentna impedansa mreže 220kV,

Z_{TR220} - impedansa transformatora na 220kV strani.

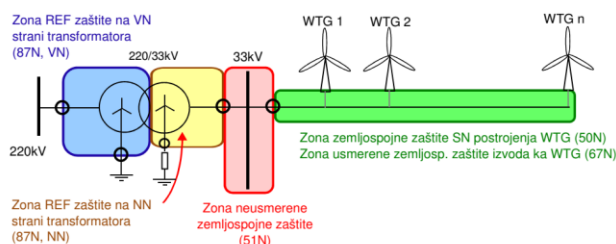
Ograničena zemljospojna zaštita (87N)

Ograničena zemljospojna zaštita (engl. Restricted Earth Fault protection – REF) deluje u slučaju zemljospoja na šticeu namotaju transformatora. Naziva se i diferencijalna zemljospojna zaštita a diferencijalna zona njenog delovanja jeste između strujnog transformatora u zvezdištu šticeu namotaja i tri strujna transformatora u fazama datog namotaja energetskog transformatora. Može se primeniti i u slučaju direktno uzemljene neutralne transformatora (slika 4), ali i u slučaju uzemljenja preko otpornika za ograničenje struje zemljospoja. Tako se ova zaštita može primeniti za zaštitu od zemljospoja unutar namotaja transformatora i na VN i na NN strani energetskog transformatora 220/33kV u TS Kovačica.



Slika 4 – Primena REF zaštite u slučaju direktno uzemljene neutralne transformatora

Na slici 5 je prikazana zona delovanja REF zaštite transformatora, kao i zone delovanja ostalih zaštita od zemljospojeva u postrojenjima 220kV i 33kV.



Slika 5 – Zone delovanja zaštita od zemljospojeva u postrojenjima 220kV i 33kV

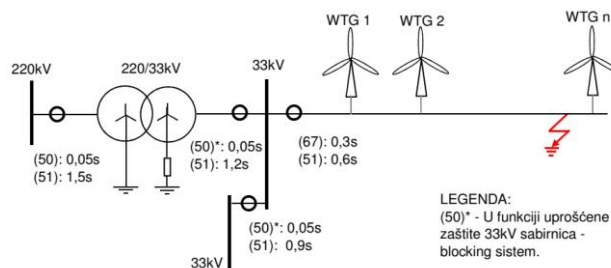
Prekostrujna zaštita (51)

Prekostrujna zaštita se koristi kao rezervna zaštita za diferencijalnu zaštitu i obezbeđuje zaštitu od kvarova u mreži. Trebalo bi je podestiti tako da obezbedi privremeno preopterećenje transformatora tokom ograničenog vremenskog perioda. Uzimajući u obzir da je dozvoljeni faktor preopterećenja prema IEC standardu 20%, za proračun vrednost prekostrujne zaštite se uzima:

$$I >= 1,2 \cdot I_{nTR} \quad (2.4)$$

Vreme zatezanja se podešava na vrednost veću za jedan vremenski interval ($t=0,3-0,4s$) od vremena isključenja

prekostrujne zaštite NN strane. Na slici 6 je prikazan koncept podešavanja vremena isključenja prekostrujnih zaštita u postrojenjima 220kV i 33kV.



Slika 6 – Podešavanja vremena isključenja prekostrujnih zaštita u postrojenjima 220kV i 33kV

3. ZAŠTITNI UREĐAJI I PRORAČUN PODEŠENJA U 33 kV POSTROJENJU

Uređaj zaštite i upravljanja transformatorskih ćelija je tipa ABB REF620 i sadrži sledeće zaštitne funkcije:

- 50/51 Prekostrujna i kratkospojna zaštita,
- 51N Zemljospojna zaštita,
- 50BF Zaštita od otkaza prekidača.

Kratkospojna zaštita (50)

Prema tehničkom uputstvu za zaštite transformatora (Elektroistok, TU-ZA-03/2) za gornju granicu podešavanja struje kratkospojne zaštite na NN strani energetskih transformatora, koristi se sledeća relacija:

$$I \gg_{max} \leq k_r \cdot I''_{3KS} \quad (3.1)$$

$$I''_{3KS} = \frac{c \cdot U_L}{\sqrt{3} \cdot Z''_{ekv}} = \frac{c \cdot U_L}{\sqrt{3} \cdot (Z_{m33} + Z_{TR33})} \quad (3.2)$$

gde su k_r – koeficijent rezerve, za koji se uzima vrednost $k_r = 0,6$, I''_{3KS} – struja trofaznog kratkog spoja na NN strani transformatora, Z_{m33} – ekvivalentna impedansa mreže na 33kV sabirnicama, Z_{TR33} – impedansa transformatora na 33kV strani.

Vrednost podešavanja treba da zadovolji dva kriterijuma:

- 1) da je podešena struja veća od najveće vrednosti podešenih struja reagovanja kratkospojne odn. usmerene prekostrujne zaštite na izvodima 33kV.
- 2) da je podešena vrednost veća od trostruke vrednosti nominalne struje odgovarajućeg namotaja energetskog transformatora.

Vreme zatezanja se podešava na vrednost $t_{t} \gg 0,05s$, s tim što je kratkospojna zaštita blokirana od pobude usmerene prekostrujne zaštite u svim izvodnim ćelijama u 33kV postrojenju, u okviru iste sekcije.

Izvodne ćelije ka vetrogeneratorima

Srednje naponska mreža vetroelektrane je izvedena kao radialna struktura s povezivanjem grupe od pet do sedam vetrogeneratora preko jednog zajedničkog kabla na srednjenaponsko postrojenje 33kV u okviru TS 33/220kV Kovačica. Takav način povezivanja smanjuje

raspoloživost pogona vetroelektrane u slučaju kvara kabla, ali je ukupna dužina SN mreže manja.

U okviru transformatorske ćelije 33kV SN postrojenja na samom vetrogeneratoru, realizovane su sledeće zaštitne funkcije:

Podešenja zaštita na vetrogeneratorima:

- Prekostrujna zaštita (51) $I >$, IDMT,
- Kratkospojna zaštita (50) $I >>$,
- Zemljospojna zaštita (51N) $I0 >$.

Uređaj zaštite i upravljanja na izvodnim ćelijama je tipa ABB REF620 i sadrži sledeće zaštitne funkcije:

- 51 Prekostrujna zaštita,
- 67 Usmerena prekostrujna zaštita,
- 67N Usmerena zemljospojna zaštita,
- 50BF Zaštita od otkaza prekidača.

Proračun podešenja

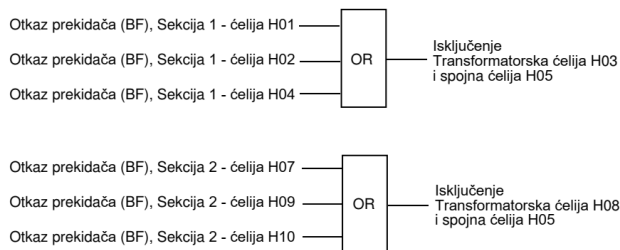
Za izvođenje predmetnih deonica kablovskih vodova ka vetrogeneratorima odabrani su jednožilni vodonepropusni kablovi tipa XHE 49-A ((N)A2XS(FL)2Y) 20,8/36/42kV. Kablovi su preseka od 150mm² do 500mm², zavisno od mesta deonice polaganja. Na osnovu podataka vezanih za impedanse kablova i na osnovu poznatih dužina kablovskih trasa po strujnim krugovima kao i vrednosti ekvivalentne impedanse mreže na sabirnicama 33kV, dobijaju se struje kratkih spojeva po strujnim krugovima, na mestu svakog pojedinog vetrogeneratora, prema izrazu:

$$I_{ks} = \frac{c \cdot U_L}{\sqrt{3} \cdot \sqrt{(R_{m33} + R_{TR33} + R_d)^2 + (X_{m33} + X_{TR33} + X_d)^2}} \quad (3.3)$$

gde su R_d i X_d rezistansa i reaktansa ukupnih kablovskih deonica do mesta kvara tj. do svakog pojedinačnog vetrogeneratora, a za naponski koeficijent se uzima $c = 1$, pošto nam je potrebna minimalna vrednost struje kratkog spoja. Kratkospojna zaštita u izvodnoj ćeliji 33kV ne bi trebalo da reaguje u slučaju kratkog spoja na 6kV strani blok transformatora na vetrogeneratoru. Za proračun ove struje kratkog spoja (I_{ks} , 6kV), u obzir se uzima i impedansa kratkog spoja blok transformatora 33/6/0,69kV i dodaje u gornji izraz (u deo za ukupnu reaktansu). Za vrednost naponskog faktora se uzima $c = 1,1$, s obzirom da nam je potrebna maksimalna vrednost struje kratkog spoja.

Zaštita od otkaza prekidača (50BF)

Zaštita od otkaza prekidača radi na principu merenja struje kroz zaštitni uređaj i ako je ta struja iznad nekog podešenog praga posle isteka određenog vremena nakon izdavanja naloga za isključenje od neke zaštitne funkcije, generiše se signal otkaza prekidača. Takođe, radi sigurnosti, uređaj bi trebalo da proverava i status kontakata prekidača nakon izdavanja signala za isključenje. Svaka zaštitna funkcija koja šalje signal za isključenje bi trebalo da startuje i zaštitu od otkaza prekidača. Na slici 7 ispod se može videti logički dijagram.



Slika 7 – Logički dijagram zaštite od otkaza prekidača

Svaka zaštitna funkcija koja šalje signal za isključenje bi trebalo da startuje i zaštitu od otkaza prekidača.

Zaštita spojne ćelije i dodatak spojne ćelije

Uređaj zaštite i upravljanja je tipa ABB REF620 i sadrži sledeće zaštitne funkcije:

- 50/51 Prekostrujna i kratkospojna zaštita,
- 51N Zemljospojna zaštita,
- 50BF Zaštita od otkaza prekidača,

čiji je princip podešenja sličan prethodnim primerima.

4. ZAKLJUČAK

U radu je realizovana detaljna studija podešenja relejne zaštite transformatorske stanice 33/220kV vetroelektrane Kovačica. Transformatorska stanica i razvodna postrojenja su jedna od velikih investicija čiji elementi zahtevaju sigurnu i pouzdanu zaštitu. Detaljno je prikaza postupak testiranja i puštanja relejne zaštite nakon čega je moguće vetropark uspešno priključiti kao aktivan na elektroenergetski sistem.

5. LITERATURA

- [1] IEE d.o.o: Sveska 4.1- Projekat elektroenergetskih instalacija Transformacija 33/220kV I RP 33kV, Projektna sveska, Novi Sad, 2018.
- [2] IEE d.o.o: Sveska 4.5 Sistem zaštite i upravljanja, Projektna sveska, Novi Sad, 2018.
- [3] IEE d.o.o: Sveska 4.6-Projekat elektroenergetskih instalacija-Upravna zgrada, Projektna sveska, Novi Sad, 2018.
- [4] Log Design d.o.o.: Sveska 4.7 Projekat EE i TK kablova za povezivanje vetroturbina, Projektna sveska, Beograd, 2018.
- [5] Javno Preduzeće Elektromreža Srbije; Studija priključenja VE Kovačica na prenosni sistem Srbije – sistemske analize, Studija, Beograd 2015

Kratka biografija:



David Ivanović, rođen je u Zrenjaninu 1993. godine, završio je osnovne studije na Fakultetu tehničkih nauka 2017. godine iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi. Master rad na istom fakultetu smer Elektroenergetika – Elektroenergetski sistemi odbranio je 2019. godine.

PREDIKCIJA ZARADE FILMOVA TOKOM VIKENDA**PREDICTING WEEKEND FILM BOX OFFICE**Miloš Ostojić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je predstavljeno istraživanje o predviđanju zarade filmova tokom vikenda. Podaci su prikupljeni sa najrelevantnijih sajtova koji se bave ocenjivanjem i analitikom zarade filmova i iz jednog naučnog rada. Zarada filma je predviđana na osnovu više atributa, jedan od njih je i koeficijent zainteresovanosti publike po polu za film na osnovu opisa zapleta filma. Za predviđanje koeficijenta zainteresovanosti publike korišćene su NLP tehnike, logistička regresija, SVM i k-NN. Problemu predikcije zarade filma se prišlo pomoću regresionih i klasifikacionih modela.

Ključne reči: Predviđanje zarade filma, regresija, klasifikacija

Abstract – This paper presents a study about predicting weekend film box office. The dataset that was made and described in the paper was taken from various relevant websites and from one scientific paper. Box office was predicted based on many attributes, one of which is an attribute that speaks of interest of public for a film based on their gender. That attribute was made when the synopsis of the film was run through NLP techniques, logistic regression, SVM and k-NN. The problem of predicting box office was approached with regression and classification models.

Keywords: Predicting film box office, regression, classification

1. UVOD

Filmska industrija je tokom 2018. godine zaradila 41.5 milijardi američkih dolara. Najbitnije filmsko tržište na svetu za holivudske studije je tržište Sjedinjenih Američkih Država i Kanade. To tržište uzima najveći udeo ukupne zarade, 11.5 milijardi dolara. Kinesko filmsko tržište je u porastu (prošle godine zarada od 8.9 milijardi dolara) i preči da će u skorijoj budućnosti preći po zaradi američko tržište. Holivudski studiji će i kada se to dogodi i dalje smatrati američko tržište najbitnijim, jer tu uzimaju najveći procenat od prodatih karti za filmove. Iz tog razloga najveći broj analitika i predviđanja zarada se vrši nad podacima vezanim za američko tržište. U ovom radu biće predstavljeno jedno rešenje za predikciju vikend zarada filmova na teritoriji SAD. Rešenje je realizovano pomoću više modela za predikciju, na osnovu dostupnih podataka. Rađeno je i predviđanje zainteresova-

nosti publike po polu za film na osnovu opisa zapleta filma.

Postojalo je mnogo izazova pri realizaciji. Podatke je teško prikupiti. Mnoge organizacije nisu voljne da dele svoje podatke, što je otežalo ili onemogućilo dobijanje željenih podataka. Izazov je predstavljalo i spajanje podataka iz različitih izvora.

Kod različitih organizacija se znalo desiti da je isti film zaveden pod različitim imenima ili pod različitim godinama izlaska.

Opisi zapleta filma mogu biti previše kratki ili neodređeni, što je otežalo predikciju žanra i zainteresovanosti publike.

Detaljniji opis podataka i metodologija je izložen u ostatku rada. Drugo poglavlje se bavi srodnim istraživanjima na ovu temu. U trećem poglavlju je opisan skup podataka, način pripreme podataka za obučavanje modela i predstavljeni su modeli koji su korišćeni. U četvrtom poglavlju su prikazani i prokomentarisani rezultati primene modela. Poslednje poglavlje je sumiralo sadržaj ovog rada i u njemu su iznete za buduće unapređenje istraživanja teme rada.

2. PRETHODNA REŠENJA

Quader i drugi su u svom radu [1] predviđali uspeh filma (da li je profitabilan ili ne) koristeći podatke dobijene sa Box Office Mojo-a, Metacritic-a i drugih. Predviđali su uspešnost pomoću tehnike potpornog vektora (eng. *Support vector machine*, skr. SVM), neuronskih mreža i NLP-a (eng. *Natural language processing*).

Došli su do zaključka da budžet, broj IMDb ocena i broj bioskopa u kojima se film prikazuje najviše utiču na finansijski uspeh filma.

Sharda i Delen u svom radu [2] su koristili neuronske mreže i logističku regresiju za predviđanje zarada filmova. Podelili su filmove na 9 različitih kategorija po zaradi i predviđali su pripadnost tim kategorijama.

Cook i drugi su napisali rad [3] koji predviđa uspeh na osnovu na osnovu da li je zaradio 110% svog budžeta. Od klasifikatora su koristili i naivni Bajes. Imali su 65% uspešnosti predviđanja klase zarade.

Hoang je u svom radu [4] koristio naivni Bajes, Word2Vec+XHBoost i rekurentne neuronske mreže za tekstualnu klasifikaciju. Za označavanje žanra je koristio k-binarnu transformaciju, rang metodu i probablističku klasifikaciju nad 250,000 filmova. Postigao je F meru od 0.56 i 80.5% uspešnih pogađanja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Jelena Slivka, docent.

3. METODOLOGIJA I ALATI

3.1 Skup podataka

Inicijalni skup podataka je preuzet sa *Cinemascore*-ove internet stranice. On sadrži ocene filma od strane publike koja ga je gledala tokom prvog vikenda tog filma u bioskopima na teritoriji SAD-a. Potom je skup podataka obogaćen sa skupom podataka dobijenim pomoću OMDb API-a, koji preuzima podatke sa *rottentomatoes.com*, *imdb.com* i *metacritic.com* sajtova. Tom skupu podataka je potom dodat skup podataka preuzetih sa *boxofficemojo.com*, koji sadrži detaljne podatke o zaradi filmova. Preuzeti su podaci o filmovima koji su bili u bioskopima u periodu od 2000. do 2018. godine.

Konačni skup podataka je sadržao, posle izbacivanja podataka sa nepoznatim vrednostima, 15419 redova i sledeće podatke o filmovima: naziv filma, godina izlaska (eng. *wide release*, srp. pušten u većini bioskopa u SAD), tačan datum izlaska u SAD, opis zapleta filma (OMDb API ga preuzima sa *imdb.com*), *Cinemascore* ocena, ocena korisnika sajta *imdb.com*, *Metacritic* ocena, *Rotten Tomatoes* ocena kritike, žanrovi kojima film pripada, vreme trajanja filma u minutima, MPAA rejting film (sistem klasifikacije filmova u SAD na osnovu pogodnosti za uzrast), produkcijska kuća, režiser filma, glavni glumci u filmu, vikend u godini (u kom je prikazan film), godina u kojoj je posmatrani vikend za koji će se vršiti analiza, broj bioskopa u kojima je film bio prikazan za taj vikend, zarada filma za posmatrani vikend u američkim dolarima, ukupna zarada filma do posmatranog vikenda u američkim dolarima i produkcijski budžet u milionima američkih dolara.

Napravljen je atribut koji je govorio o rednom broju vikenda filma u bioskopima. Na osnovu navedenih atributa napravljeni je još nekoliko atributa koji su korišćeni za predikciju zarade filmova:

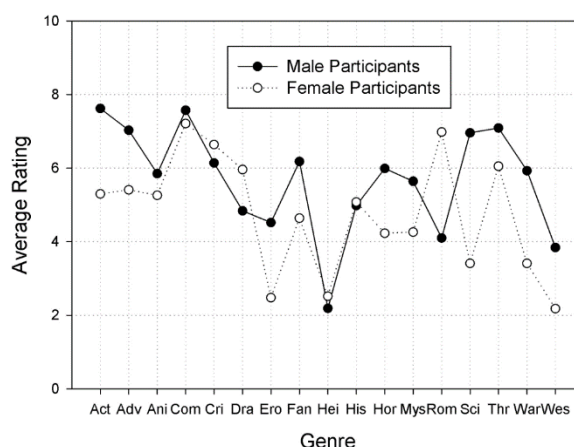
- Zarada filmskog studija: atribut je napravljen sabirajući zarade filmskih studija na filmovima u skupu podataka na teritoriji SAD.
- Zarada režisera: atribut je napravljen sabirajući zarade filmova koje je režirao režiser.
- Zarada glumaca: atribut je napravljen tako što su prvo sabirane zarade filmova koje je glumio glumac. Potom su za određeni film sabirane zarade glumaca koji su navedeni kao zvezde filma.
- Interesovanje publike za film: atribut je napravljen kombinacijom informacija o predviđenom žanru na osnovu kratkog opisa zapleta (detaljnije u potpoglavlju 3.2).
- Težina vikenda: atribut je dobijen na osnovu postavljenih upita o vikendu u kom se film prikazuje u bioskopima. Ako film u vikendu ima bar jedan film koji ima zaradu veću od 60 miliona dolara i sličnu zainteresovanost publike po polu i ako ima bar jedan film sa zaradom većom do 20 miliona dolara i sličnom zainteresovanšću po polu onda je težina vikenda kategorisana kao teška (vrednost 2). Ako ima više filmova koji su imali zaradu veću od 20 miliona dolara i sličnu zainteresovanost publike onda je težina vikenda kategorisana kao srednja (vrednost 1). U suprotnom je težina vikenda klasifikovana kao laka (vrednost 0).

Iz napravljenog skupa podataka izdvojeni su podaci koji govore o zaradi filmova tokom prvog vikenda u bioskopima. Pripremajući podatke za klasifikaciju dodato je jedno obeležje koja govore o klasi zarade. To obeležje predstavlja klasifikaciju na 8 klasa. Opsezi za klase po zaradi su: 1. do milion dolara, 2. od milion do 5 miliona dolara, 3. od 5 miliona do 10 miliona dolara, 4. od 10 miliona do 15 miliona dolara, 5. od 15 miliona do 20 miliona dolara, 6. od 20 miliona do 30 miliona dolara, 7. od 30 miliona do 50 miliona dolara i 8. od 50 miliona dolara.

3.2 Metode određivanja zainteresovanosti publike za film na osnovu opisa zapleta i žanra filma

Rad [6] je izvršio ispitivanje o zainteresovanosti za neki žanr filma u odnosu na pol i dobijeni rezultati se porede sa stereotipima o muškim i ženskim filmovima.

U ispitivanju su učestvovali 80 muškaraca i žena, koji su ocenjivali 17 žanrova ocenom od 1 do 10 (1-najmanje, 10-najviše). Kao rezultat ispitivanja dobijena je tabela, na osnovu koje je pravljena predikcija prihvaćenosti filma u odnosu na pol. Te ocene su ušle kao ocene žanra u podatke.



Slika 1. Ocene privlačnosti žanrova po polu [6]

Ova studija nije obuhvatila sve žanrove u setu podataka, pa su njihovi koeficijenti određeni na osnovu ličnog iskustva. Koeficijenti i žanrovi koji su dobijeni na ovaj način su *Biography* (za muškarce 6, za žene 6), *Documentary* (5, 5), *Family* (5, 6.3), *Music* (4, 5.5), *Musical* (4, 5.5) i *Sport* (5.1, 5).

Predviđanje žanra na osnovu opisa zapleta filma spada u probleme analize sentimenta. Korišćeni su sledeći modeli:

- Logistička regresija
- *k*-NN
- SVM

Performanse modela su unapređene još sa tehnikom relativne frekvencije izraza (eng. *term frequency-inverse document frequency*, *tf-idf*), prilikom obrade opisa zapleta filma.

Prvi korak za svaku od navedenih tehnologija predstavlja pretprocesiranje opisa filma, koje je zasnovano na tehnici zbrajanja reči (eng. *bag of words*). Za svaki film opis se svodi na mala slova, određuje se set zaustavnih reči, koje nemaju kontekstnu vrednost, da bi se na kraju ostavile sve ostale reči koje nose značenje.

Problem određivanja liste žanrova može se svesti na predviđanje pojedinačnih žanrova na osnovu opisa radnje (eng. *one-versus-all*). Ovaj pristup je korišćen kod svih modela.

Na osnovu udela žanrova se može videti da su podaci nebalansirani. Ovo je ipak ostavljeno prilikom obuke, jer bi na ovaj način mreža trebala da odbaci žanr, osim ako ne postoje velike indicije za njega..

U svakom od prethodnih modela urađena je unakrsna validacija metodom *k-fold cross validation* kako bi se dobila predikcija za ceo skup podataka i kako bi parametri preciznosti bili pouzdani.

Labela prihvaćenosti filma u odnosu na pol nalazi se u opsegu od -1 do 1, gde -1 označava veće interesovanje ženske publike dok 1 označava veće interesovanje muške publike.

3.3 Metode predviđanja zarada filmova

Postojala su 2 pristupa prilikom predikcije zarade filmova. Prvi pristup je bio da se problemu predikcije pristupi pomoću regresionih modela. Korišćeni su regresioni modeli linearne regresije i *Random forest*-a. Drugi pristup je bio da se problem predikcije reši pomoću klasifikacionih modela. Korišćeni klasifikacioni modeli u ovom radu su bili modeli SVM-a, naivnog Bajesa, veštačke neuronske mreže, *AdaBoost*-a, *Random forest*-a, *Gradient boosting*-a, *k-NN*-a i linearne regresije, čiji su rezultati regresionog modela iskorišćeni za klasifikaciju. Korišćena je *Scikit* biblioteka u *Python*-u za primenu modela.

4. REZULTATI

4.1 Rezultati prepoznavanja žanra i zainteresovanosti publike za film na osnovu njegovog opisa zapleta

Određivanje naklonosti publike u odnosu na žanrove ima R^2 meru 0.93. Prosečno odstupanje dobijenog rezultata od labele je 0.065, dok je medijan odstupanja 0.08.

Kao metrika uspešnosti predviđanja korišćeni su preciznost (eng. *precision*), opoziv (eng. *recall*) i F1 mera, kako bi se prikazala realna slika klasifikatora, imajući u vidu veliku nebalansiranost skupa podataka. Tačnost u ovom slučaju je velika za svaki žanr, ali ne oslikava stvarnu sliku, zato je većina tačnih predviđanja *true-false*, čime se predviđa da žanr nije prisutan. Kako bi bilo jasniji odnos prisustva nekog žanra u filmu, pored svakog žanra dat je procenat prisustva u celom skupu podataka.

U ovom slučaju komedije i drame imaju daleko veći udeo i balansiranoost nego vesterni i dokumentarni filmovi, zbog čega rezultati njihove klasifikacije daju mnogo bolje rezultate. Ovo se takođe oslikava i u rezultatima opoziva. Mali opoziv je posledica mnogo većeg broja negativnih odgovora u skupu podataka, čime se model obučava da odbacuje žanrove sa većom verovatnoćom. Sa druge strane preciznost u nekim slučajevima ima veliku vrednost usled nedostatka podataka i nebalansiranosti podataka.

Na osnovu rezultata može se zaključiti da modeli logističke regresije i SVM-a podjednako dobro klasifikuju, dok model *k-NN*-a daje značajnije lošije rezultate. U žanrovima sa većom zastupljenošću tehnika *tf-idf* u velikom broju slučajeva pozitivno deluje na predviđanje, što je bilo očekivano.

4.2 Rezultati predikcije zarade filmova pomoću regresionih modela

Korišćeni su modeli linearne regresije i *Random forest*-a. Skup podataka je podeljen u razmeri 4:1 na skup podataka za trening i test. Za određivanje uspešnosti predikcije korišćena je R^2 metrika. Takođe je korišćena i unakrsna validacija, da bi se videlo da li postoji određeno odstupanje ako bi se drugačije napravio skup trening i test podataka.

Tabela 1. Rezultati regresionih modela

Model	R^2	10-fold cross validation
Linearna regresija	0.59	0.58 (+/- 0.17)
<i>Random forest</i>	0.76	0.72 (+/- 0.27)

Na osnovu tabele se vidi da *Random forest* pravi bolju predikciju zarade filmova od linearne regresije. Rad [5] koji se bavio predviđanjem ukupne zarade filmova za južnokorejsko tržište je dobio sličnu vrednost za R^2 za model linearne regresije, 0.588

4.3 Rezultati predikcije zarade filmova pomoću klasifikacionih modela

Rezultati klasifikatora su prikazani u tabeli 2. Korišćena je 10-fold cross validation i ukupan procenat pogodne klase. Takođe su posmatrani preciznost, opoziv, F1 mera i konfuzione matrice, radi boljeg uvida u rezultate klasifikacije.

Tabela 2. Rezultati klasifikacionih modela

Modeli	Procenat pogodnih klasa	10-fold cross validation
SVM	40.00%	0.39 (+/- 0.10)
Naivni Bajes	38.26%	0.38 (+/- 0.06)
<i>k-NN</i>	33.04%	0.35 (+/- 0.10)
Linearna regresija	33.91%	0.58 (+/- 0.17)
<i>AdaBoost</i>	40.87%	0.43 (+/- 0.09)
<i>Random forest</i>	42.61%	0.45 (+/- 0.10)
<i>Gradient boosting</i>	40.43%	0.42 (+/- 0.07)
Veštačka neuronska mreža	44.35%	0.42 (+/- 0.09)

Najuspešnijim od svih klasifikatora po procentu pogodnosti pripadnosti klasi zarade za dati skup podataka je model veštačke neuronske mreže, sa 44.35% uspešnosti pogađanja klase za skup podataka. Model veštačke neuronske mreže se pokazao boljim od sledećeg najuspešnijeg modela, *Random forest* modela, za 1.74%. Najmanje uspešni modeli su bili *k-NN* sa 33.04% pogodnih klasa i linearna regresija sa 32.61% pogodnih klasa.

Poredivši F1 mere za svaku pojedinačnu klasu predviđanja najuspešnijih klasifikatorskih modela za skup podataka za prvi vikend može se zaključiti da model veštačke neuronske mreže uspešnije prepoznaje dve „ekstremne” (1. i 8. klasa), kao i u slučaju najbrojnije, 3.

klase. Predviđajući 3. klasu model veštačke neuronske mreže je oslabio F1 mere za predviđanje 4. i 5. klase. Model *Random forest* je konstantniji u predviđanju klasa, tj. nema izuzetno nisku F1 meru prilikom predviđanja klasa (za razliku od veštačke neuronske mreže). Zanimljivo da je kod oba modela 5. klasa (zarade od 15 miliona dolara do 20 miliona dolara) imala najmanju F1 meru, kod veštačke neuronske mreže je imala vrednost 0.08, kod *Random forest*-a je imala 0.23. Kod većine modela predviđanje 5. klase rezultovalo najmanjom F1 merom, sa izuzetkom naivnog Bajesa, kome je predviđanje 4. klase bilo najveći problem, i SVM-a, koji je imao vrednost 0 za F1 meru za 2. i 5. klasu, svrstavaju ih u 3. i 4. klasu, respektivno.

5. ZAKLJUČAK

U ovom radu prikazani su modeli za predviđanje zarade filma u bioskopima na teritoriji SAD tokom prvog vikenda. Predviđana zarada su vršena na osnovu više obeležja. Obeležja koja nisu korišćena ili su drugačije konstruisana u radovi su obeležja zarade glumca, režisera i produkcijske kuće, težina vikenda i zainteresovanost publike po polu u odnosu na žanr filma. Podaci su prikupljeni sa vodećih sajtova za ocenjivanje i analitiku filmova. Prvo je rađena predikcija žanra u odnosu na opis zapleta filma. Ideja je bila da se zaključi zainteresovanost publike po polu za određeni film na osnovu par rečenica o filmu. Potom su primenjeni regresioni i klasifikatorski modeli za predikciju zarade filma.

Za predikciju žanra filma u odnosu na opis zapleta filma korišćeni su modeli logističke regresije, *k*-NN i SVM. Modeli logističke regresije i SVM-a pokazali su slične performanse dok je model *k*-NN bio značajno lošiji. U svakom modelu korišćenje tehnike *tf-idf* dovelo je do značajnog poboljšanja preciznosti. Kako postoji korelacija između žanrova, odsustvo prepoznavanja žanra u nekim slučajevima nije značajno uticalo na predikciju preferencija filma u odnosu na pol.

Od regresionih modela korišćena je linearna regresija i *Random forest* ansambl. Model linearne regresije je nad skupom podataka za prvi vikend imao R^2 vrednost od 0.59. Model *Random forest*-a je ostvario veću R^2 vrednost modela linearne regresije sa vrednošću od 0.76 za prvi vikend.

Od klasifikatorskih modela korišćeni su SVM, naivni Bajes, veštačka neuronska mreža, linearna regresija, *AdaBoost*, *Random forest*, *Gradient boosting* i *k*-NN. Zarade filmova su klasifikovane na 8 klasa..

Za skup podataka za prvi vikend kada je rađena klasifikacija na 8 klasa najuspešnijim u predviđanju se pokazao model veštačke neuronske mreže sa 44.35% tačnih predviđanja klasa. Najslabiji klasifikatori prilikom

klasifikacije na 8 klasa su bili modeli *k*-NN-a i linearne regresije.

Ovaj rad bi se mogao unaprediti dodavanjem atributa koji bi govorili o aktivnosti na društvenim mrežama, npr. broj pregleda trejlera filma na Youtube-u u prvih 24 sata, ili aktivnost na Twitter-u u vreme objave trejlera i u vreme objave kritika na internetu. Time bi se dobili podatke koji bi govorili o zainteresovanosti publike za film, i moglo bi se pratiti koliko se ti komentari preneli na zaradu u bioskopu. Takođe bi se mogao pratiti uticaj pozitivnih i negativnih komentara na performans filma u bioskopima. Veliko proširenje ovom radu bi bilo kada bi bili dostupni podaci MPAA-a. Ta organizacija je jedan od glavnih autoriteta za podatke o zaradama filmova, ali nema ih dostupne za javnost. Svakog vikenda puštaju deo podataka koji su prikupili tokom njega, u kome imaju podatke o publici koja je videla film, počevši od broja prodatih karata, do podataka o rasnom, starosnom i polnom sastavu publike.

6. LITERATURA

- [1] Quader, N., Gani, M. O., Chaki, D., & Ali, M. H. (2017, December). A machine learning approach to predict movie box-office success. In *2017 20th International Conference of Computer and Information Technology (ICCIT)* (pp. 1-7). IEEE.
- [2] Sharda, R., & Delen, D. (2006). Predicting box-office success of motion pictures with neural networks. *Expert Systems with Applications*, 30(2), 243-254.
- [3] Cook, C., Cunningham, B., Reading, E., Sedgewick, M., & Tilcock, K. Predicting Blockbuster Success.
- [4] Hoang, Q. (2018). Predicting movie genres based on plot summaries. *arXiv preprint arXiv:1801.04813*.
- [5] Song, J., & Han, S. (2013). Predicting gross box office revenue for domestic films. *Communications for Statistical Applications and Methods*, 20(4), 301-309.
- [6] Wühr, P., Lange, B. P., & Schwarz, S. (2017). Tears or fears? Comparing gender stereotypes about movie preferences to actual preferences. *Frontiers in psychology*, 8, 428.

Kratka biografija:



Miloš Ostojić rođen je u Novom Sadu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2019. godine.
kontakt: milos.ostojic@uns.ac.rs

POREĐENJE NIRS UREĐAJA PRIMENJENIH U SPORTU**COMPARISON OF NIRS DEVICES USED IN SPORTS**Goran Maksimović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu su prikazane osnove NIRS uređaja, zasnovanog na upotrebi blisko-infracrvene spektroskopije. Opisana je njegova upotreba u sportu i izvršeno je poređenje različitih NIRS uređaja primenjenih u sportu.

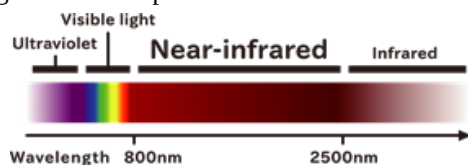
Ključne reči: Blisko-infracrvena spektroskopija, NIRS, SmO_2 , FTP, LT, PortaMon, Moksi Monitor, Humon Heks, BSX senzor

Abstract - The paper presents the basics of a NIRS device, based on the use of near-infrared spectroscopy. Its use in sport is described and a comparison of different NIRS devices used in sport is made.

Keywords: Near-infrared spectroscopy, NIRS, SmO_2 , FTP, LT, PortaMon, Moxxy Monitor, Humon Hex, BSX insight

1. UVOD

Blisko-infracrvena spektroskopija ili skraćeno NIRS, je spektroskopska metoda koja koristi blisko-infracrvenu oblast elektromagnetnog spektra (od oko 800nm do 2500nm). Standardna primena je u medicinskoj i fiziološkoj dijagnostici i istraživanjima, rehabilitaciji, sportskoj medicini, pulsnoj oksimetriji, neonatalnim istraživanjima, treningu vrhunskih sportista.



Slika 1. Talasne dužine blisko-infracrvene svetlosti

Postoji veliki broj različitih NIRS uređaja koji se primenjuju radi poboljšanja treninga sportista. U ovom radu će biti opisane njihove sličnosti i razlike na osnovu dimenzija, funkcija, mesta i načina upotrebe i mnogih drugih karakteristika.

2. OSNOVE NIRS-a**2.1. Osnovni principi**

Sve ćelije u svim organima tela imaju konstantnu, ali promenljivu potrebu za kiseonikom. Međutim, mogućnost

tela da skladišti kiseonik je minimalna. Neophodno je konstantno i adekvatno snabdevanje tkiva kiseonikom putem cirkulacije. Svako smanjenje oksigenisanosti tkiva će vrlo brzo dovesti do nepovratnog oštećenja.

Optička oksimetrija i NIRS su tehnike za procenu stanja oksigenisanosti i hemodinamike različitih organa, na primer mišića ili mozga na neinvazivan način. Oslanjaju se uglavnom na dve karakteristike ljudskog tkiva. Prva je, relativna transparentnost tkiva za svetlost u blisko-infracrvenom opsegu, a druga je, apsorpcija hemoglobina u zavisnosti od oksigenisanosti [1,2]. Koristeći više različitih talasnih dužina, relativne promene koncentracije hemoglobina se mogu prikazivati kontinualno.

Definisanje algoritma koji koristi NIRS, zahteva spektralne koeficijente ekstincije različitih hromofora. Spektri dva glavna hromofora su oksihemoglobin (O_2Hb) i dezoksi-hemoglobin (HHb).

Suma O_2Hb i HHb je mera ukupnog volumena hemoglobina (tHb) u tkivu [1,3]. Mišićno tkivo sadrži još dva hromofora: oksimioglobin (O_2Mb) i dezoksio-mioglobin (HMb). Da bi razlikovali hemoglobin i mioglobin u mišićnom tkivu, spektri moraju biti dovoljno različiti. Nažalost, ovo nije slučaj u blisko-infracrvenoj oblasti spektra. To znači da NIRS ne može razlikovati da li izmerenu koncentraciju kiseonika prenosi hemoglobin ili mioglobin. Talasne dužine koje mogu odrediti razlike između hemoglobina i mioglobina ne mogu prodrati dovoljno duboko u tkivo.

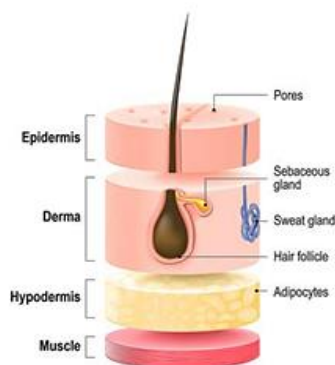
2.2. Upotreba NIRS uređaja u sportu

Vidljiva i infracrvena svetlost može da putuje kroz biološka tkiva, a crvena i blisko-infracrvena područja su korisna za instrumentaciju. Dok se svetlost kreće kroz mišiće, masti i kožu, ona se raspršuje i identifikuje oksigenisane i dezoksigenisane krvne ćelije [4]. Poput vena i arterija koje imaju iluzorne boje plave i crvene, svetlost može pomoći u prepoznavanju preciznog fiziološkog stanja površinskih mišića. Izvlačenjem tih svetlosnih podataka, treneri zaista mogu dobiti uvid u ono što se događa tokom treninga. Međutim, kao i u svemu, potrebna je edukacija i dobar plan za korišćenje podataka nakon što budu prikupljeni.

Čitanja zahtevaju postupak izrade algoritama zbog znoja, količine telesne masti i drugih prirodnih faktora. Prikupljanje validnih i tačnih podataka iz mišića ispod slojeva tkiva je veoma komplikovan proces. NIRS sistemi su kalibrisani za uzorkovanje mišićnih grupa sa visokim nivoom preciznosti i ako se uloži malo više vremena, prikupljene informacije predstavljaju šta se dešava ispod površine kože [5].

NAPOMENA:

Ovaj rad proistekao je iz master rada, čiji mentor je bio prof. dr Platon Sovilj.



Slika 2. Prikaz različitih slojeva kože

Treneri bi trebali da budu veoma zainteresovani za NIRS, jer se sve češće prelazi sa merenja otkucaja srca i laktata na neinvazivne metode. Testiranje laktata ili praćenje rada srca nije neprikladno i ne zastareva, ali samo oksigenisanost mišića ima nezavisnu vrednost. NIRS sistem može da odgovori na pitanja koje većina trenera ne bi ni pretpostavljala. Konkretno, NIRS odgovara na sledeća pitanja:

Medicina: Programi za povratak u igru mogu reći da li je trening primeren ili agresivan, bez rizika da se sportisti ponovo povrede.

Performanse: Treneri mogu da primele kondicione trendove lokalnih mišićnih grupa umesto sistemskih globalnih promena. Što je još važnije, treneri mogu otkriti koji fiziološki sistem sprečava njihove sportiste da se poboljšaju.

Ishrana: Istraživači i nutricionisti mogu da istraže kako ishrana utiče na vazodilataciju, kao i ostale uticaje hrane i suplemenata na zdravlje i performanse.

Oporavak: Profesionalci, bilo da su treneri ili čak sportski naučnici, mogu videti kako rehabilitacija i treninzi olakšavaju oporavak.

Postoje i drugi razlozi zbog kojih bi sportsko osoblje koristilo NIRS sa sportistima, ali u najvećem delu ova četiri zahteva u četiri kategorije, sjajni su primeri koji ilustruju potencijal tehnologije [5].

2.3. Potreba za merenjem lokalne oksigenisanosti mišića

Sasvim je razumno držati se štoperice i plana vežbi, bez bilo kakvog merenja ili nadgledanja treninga. Ljudsko telo je veoma kompatibilno i čudesno je u prilagođavanju različitim programima treninga. Nažalost, sportista nije besmrtni, a neki treninzi mogu prouzrokovati povrede ili uništiti karijeru. Ako želite direktne odgovore na treningu, morate da procenite za šta je sportista sposoban.

Cilj NIRS-a je da pomogne u rešavanju misterija kondicioniranja. Nije da ne znamo kako telo funkcioniše - imamo dobru ideju - ali ne znamo tačno, kako se svaki sportista može poboljšati kada se približi svom genetskom plafonu.

Presek biomehanike i fiziologije vežbanja je u tome gde leži vrednost NIRS-a. Na osnovu povratnih informacija o sportu i iskustva naučnika koji su koristili NIRS, njegova snaga je merenje oksigenisanosti mišića. Pored kardiorespiratornih treninga ili treninga radi smanjenja otkucaja srca, NIRS direktno i ekološki cilja i specifične mišićne grupe i različite oblike pokreta.

Otkucaji srca su odlični i testiranje razmene gasova je korisno za sportiste koji se bave ekstremnom izdržljivošću, ali izvođenje VO_2 maksimalnog testa za košarkaša nije od velike pomoći. Testiranje laktata je vrlo relevantno i korisno, ali razlika između uzorkovanja krvi i lokalnog testiranja mišića je u tome što je prvo globalna reprezentacija, a drugo lokalna pojava [5].



Slika 3. Treneri imaju tendenciju da žele postupke koji su jednostavni za razumevanje, a NIRS testiranje cilja na dosledan napredak.

3. NIRS UREĐAJI PRIMENJENI U SPORTU

Nauka konstantno napreduje i sve se više upotrebljava u svakodnevnom životu. Pored korišćenja standardnih uređaja kao što su televizor, računar i mobilni telefon, ljudi postepeno ubacuju tehniku i u segment fizičke aktivnosti. U te svrhe, dosad su se uglavnom koristili samo pametni satovi koji mere rad srca, odnosno puls. Međutim, u poslednje vreme, razvojem NIRS uređaja akcenat je stavljen i na praćenje rada mišića na osnovu njihove zasićenosti kiseonikom. Neki od njih će biti opisani detaljnije u nastavku rada.

3.1. PortaMon

PortaMon je prenosni wireless NIRS uređaj specijalno dizajniran za istraživanje mišića. On koristi bluetooth mrežu, koja čak omogućava trenutno snimanje dok je osoba u pokretu. Na većim razdaljinama PortaMon takođe može sačuvati svoje podatke, spremne za korišćenje kada se povežemo na računar [6].



Slika 4. PortaMon uređaj

PortaMon se uglavnom koristi za istraživanja u sportu, ali se takođe može koristiti i u rehabilitaciji.

Primena:

- Merenje oksihemoglobina, dezoksihemoglobina i ukupnog hemoglobina u mišićnom tkivu
- Omogućava uvid u lokalnu potrošnju kiseonika, volumen krvi i protok krvi
- Merenje zasićenosti kiseonika u tkivu

3.2. Moksi Monitor

Moksi Monitor je NIRS uređaj koji meri nivo kiseonika u mišićima tokom treninga. Moksi jedinica je nosiva, bežično se povezuje na pametni uređaj kao što je tablet i uključuje softver za analizu. Pored izvornog softvera koji je kompanija napravila, drugi programeri su proširili izbor softvera za analizu dodatnih vrednosti. Nekoliko potrošačkih proizvoda je dolazilo i nestajalo, ali Moksi je uspeo jer je verodostojan za iskusnog trenera, a ipak prilagođen za individualne sportiste. Može da se poveže jedan senzor, ali mnogi korisnici povezuju više senzora kako bi uporedili povređene mišiće sa zdravima, različite mišiće koji su pod stresom ili promene na mišićnim grupama koje se ne koriste mnogo tokom vežbe [5].

Moksi senzor je srce Moksi Monitor sistema. Njegove diode i foto detektori zajedno stvaraju spektrometar koji omogućava merenje kiseonika u mišićima. Potpuno funkcionalan i samostalan, Moksi senzor ima bateriju za napajanje, mikroprocesor za rad spektrometra i memorijski čip za skladištenje podataka; takođe ima radio predajnik za slanje podataka uživo na druge uređaje [7].

Moksi pruža fiziološke povratne informacije u realnom vremenu za bilo koji sport: dovoljno je jak da izdrži poteškoće pri vožnji motokrosa i u ragbiju; dovoljno je siguran da funkcioniše i u bazenu; dovoljno je prenosiv za nošenje na fudbalskom terenu. Moksi prenosi fiziološki nadzor iz laboratorije i u stvarni svet sporta.



Slika 5. Moksi Monitor uređaj

3.3. Humon Heks

Humon Heks senzor za mišićni kiseonik je upravo to - senzor za kiseonik u mišićima, meri se zasićenost mišića kiseonikom (SmO_2). On je prvi klinički potvrđen senzor za merenje kiseonika u mišićima [8]. Najčešće se koristi za neinvazivno merenje laktatnog praga (LT).



Slika 6. Humon Heks uređaj

Humonovi napredni pokazatelji performansi se koriste za:

- Dobijanje povratnih informacijama u realnom vremenu
- Smanjenje povreda nadgledanjem zagrevanja i oporavka
- Praćenje napretka pomoću laboratorijskih procena

Kako se koristi

- Podaci su beskorisni bez velikog zagrevanja. Proći će dosta vremena pre nego što se vaši mišići potpuno oksigenuju i zagreju, 10 ili 15 minuta zagrevanja to neće učiniti. Umesto toga, zagrevanje zaista traje najmanje 25-30 minuta. Zagrevanje uključuje dva procesa - povećanje protoka krvi (i mišićnog kiseonika) i potpuno proširenje

krvnih sudova [9]. Isti trening može izgledati potpuno drugačije na uređaju sa ili bez dobrog zagrevanja - i da, osetićete i tu razliku (tj. vežbanje uz pravilno zagrevanje će ići mnogo bolje).

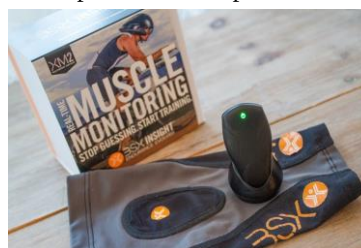
- Ne očekujte savršenstvo - biološki sistemi nisu mehanički sistemi. Za razliku od fizičkih sistema koji se ponašaju na savršeno predvidivi način, živi organizmi nisu u potpunosti predvidivi. Jednostavno je previše faktora koji čine vaše telo teško predvidivim. Iako možete reći da su vaše vrednosti FTP testa (određivanje funkcionalnog praga snage) prilično predvidive, to je posledica zato što ulazite u svoje FTP testove sa dobrim očekivanjem o tome koji će biti vaši ciljevi i na taj način ga činite predvidivim. Ako ste sve svoje FTP testove "slepili" za vaše trenutne performanse, pretpostavka je da će vaše testne vrednosti biti prilično raznolike. Suprotno tome, sa očitavanjem mišićnog kiseonika, ne možete iskriviti rezultate - oni su jednostavno takvi kakvi jesu [9,10].

3.4. BSX senzor

BSX senzor je nosiv uređaj koji se koristi za merenja nivoa kiseonika u mišićima u realnom vremenu. Želite trenirati pametnije? Treba vam uvid u to šta vam telo govori. Profesionalni sportisti dugo su zavisili od ispitivanja izdržljivosti u laboratorijama da bi stekli prednost nad konkurentima [11].

Senzor vam daje iste rezultate kao laboratorijski test, ali koristi LED svetla umesto igala da bi ušao u mišić. Koristeći već patentiranu tehnologiju laktatnog praga, meri i aerobnu i anaerobnu kondiciju. Time daje kompletan pregled atletskih sposobnosti koje omogućavaju sportisti da optimizuje trening na različite energetske nivoe i praćenje poboljšanja tokom vremena. To znači da nema težih terenskih testova ili bolnih poseta laboratorijama.

Mnogi veruju da ove informacije predstavljaju buduću metriku za nivo napora, umora, oporavka i kondicije.



Slika 7. BSX senzor

Uvid u BSX senzor

Iako se laboratorijski rezultati ne prenose uvek na navike treniranja, a rutinsko vađenje krvi nije praktično, BSX senzor daje trenutne podatke koji daju snažne rezultate. Vodeća tehnologija i inovacije zasnovane na sportistima znače da su se svakodnevni treninzi upravo pretvorili u zlatni standardni trening laktatnih pragova.

BSX senzor koristi LED svetla da "pogleda" u mišić lista noge. Dok svetlosni niz prolazi kroz mišićno tkivo, on se oblikuje hromatoforima (ili karakteristikama markera) tkiva koje su jedinstvene. To stvara biološki signal koji se snima i obrađuje BSX algoritmima kako bi se stvorio jedinstveni profil. Analizom hiljada karakteristika signala stvara se vrednost laktatnog praga jednako tačno kao i metode zasnovane na krvi [11-13].

4. ZAKLJUČAK

U ovom radu su opisane osnove NIRS tehnologije, detaljno je opisana upotreba NIRS uređaja u sportu. Takođe su detaljno opisani različiti NIRS uređaji i izvršeno je njihovo međusobno poređenje.

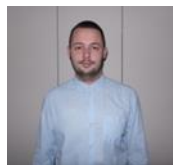
U današnje vreme, tehnologija se razvija velikom brzinom i sve je zastupljenija kako u medicini tako i u drugim sferama života ljudi. Neki od glavnih ciljeva medicine su minimizacija aparata i neinvazivnost i u tom smeru se razvija moderna tehnologija. Kao primer te tehnologije su nastali NIRS uređaji. Najčešće se koriste za dijagnostiku bolesti ili povrede, a u poslednje vreme postaju zastupljeniji i u sportu.

U profesionalnom sportu najvišeg nivoa, treneri sve više koriste NIRS uređaje. Uz pomoć određenih merenja, formiraju plan treninga na osnovu kojeg će njihovi sportisti moći najbolje da napreduju uz optimalan utrošak energije. Najčešće se vrše ispitivanja pri trčanju i vožnji bicikla, međutim uz dobru izolaciju i zaštitu tako da uređaj bude vodootporan, moguće je izvoditi merenja i tokom plivanja. U oblasti oporavka i rehabilitacije, ova tehnika može da omogućiti kako sagledati efekte oporavka pri potapanju u ledenu vodu i kakav to uticaj ima na protok krvi i nivo oksigenisanosti u mišićima. S obzirom na sve navedeno, NIRS uređaji imaju potencijal da postanu nezostavni deo treninga vrhunskog sportiste u budućnosti.

5. LITERATURA

- [1] Chris J. McManus, Jay Collison, Chris E. Cooper: "Performanse comparison of the MOXY and PortaMon near-infrared spectroscopy muscle oximeters at rest and during exercise", Journal of Biomedical Optics 23(1), 015007 (2018)
- [2] Moxy Monitor: "The science behind Moxy", dostupno na www.moxymonitor.com, datuma: 06.12.2019.
- [3] Humon: "Interval training with muscle oxygenation", dostupno na www.humon.io, od datuma: 20.06.2017.
- [4] Moxy Monitor: "Introduction to muscle oxygen monitoring with Moxy", dostupno na www.moxymonitor.com, datuma: 06.12.2019.
- [5] Carl Valle: "How to use the Moxy monitor with athletes", dostupno na www.simplifaster.com/articles/moxy-monitor-athletes-uses/, od datuma: 18.10.2018.
- [6] PortaMon leaflet, dostupno na www.artinis.com/portamon datum: 06.12.2019.
- [7] Moxy Specifications sheet, dostupno na www.moxymonitor.com, datuma: 06.12.2019
- [8] Humon Muscle Oxygen Sensor, dostupno na www.humon.io, datuma: 06.12.2019.
- [9] Ken Nakata: "Using the Humon muscle oxygen sensor", dostupno na www.athletictimemachine.com/2018/03/29/humon-muscle-oxygen-sensor, od datuma: 29.03.2018.
- [10] Humon, Dr. Pamela Anderson, PhD: "The limitations of training with heart rate and the crucial information muscle oxygenation can offer you", dostupno na www.humon.io, od datuma: 05.05.2017.
- [11] BSX Insight technology, dostupno na www.bsxinsight.com/technology, datuma: 06.12.2019.
- [12] BSX Athletics: "What is BSX Insight and how does it work?", dostupno na support.bsxinsight.com/hc/en-us/articles/204394525-What-is-BSXinsight-and-how-does-it-work, datuma: 06.12.2019.
- [13] Ray Maker: "BSX Insight Gen2 in-depth review", dostupno na www.dcrainmaker.com/2016/01/insight-depth-review.html, od datuma: 20.01.2016.

Kratka biografija:



Goran Maksimović rođen je u Novom Sadu 05.05.1994. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu, na smeru Biomedicinsko inženjerstvo 2018. godine. Nakon završetka osnovnih akademskih studija, upisao je master studije na Fakultetu tehničkih nauka, smer Energetika, elektronika i telekomunikacije.

TEHNIČKI ASPEKTI INSTALACIJA INDUSTRIJSKIH POGONA

TECHNICAL ASPECTS OF INDUSTRIAL INSTALLATION

Mladen Vuković, Dejan Jerkan, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Osnovni cilj ovog rada je planiranje izvođenja elektroenergetske instalacije pogonskih postrojenja, sa sagledavanjem svih tehničkih aspekata neophodnih za efikasnu i nesmetanu funkciju postrojenja.

Ključne reči: Električne instalacije, elektromotorni pogoni.

Abstract–The main objective of this paper is design of the electrical installation, with consideration of all technical aspects necessary for the efficient and undisturbed function of the plant.

Keywords: Electrical installations, electrical drives.

1. UVOD

Pod pojmom projektovanja električne instalacije u pogonu podrazumeva se pre svega odabir parametara opreme koja se instalira unutar postrojenja. Pod ovom opremom smatraju se: vodovi, rastavljači, prekidači, pogonski elementi, itd.

Izradom ovog rada težilo se planiranju električne instalacije industrijskog postrojenja sa planiranom kompenzacijom reaktivne energije. Kroz ovaj rad ispraćeni su svi propisani normativi za izvođene električne instalacije, sagledan je proces kompenzacije i ispraćena moguća neželjena pojava rezonancije u instalaciji, kao i neželjene posledice do kojih ona može dovesti.

2. OBRAČUN ELEKTRIČNE ENERGIJE

Pod terminom tarifnog sistema podrazumeva se sistem obračuna isporučene električne energije kao i skup stručnih usluga sprovedenih od strane elektrodistributivnog preduzeća ka potrošaču. Tarifnim sistemom za odračun električne energije određuju se tarifni elementi i tarifni stavovi čijom se primenom dolazi do formiranja cene isporučene energije ka kupcu. Formiranje tarifnog sistema na području republike Srbije sprovodi preduzeće „Elektroprivreda Srbije“ i odeljenje za distribuciju električne energije. Tarifnim sistemom i njegovim pravnicima elektrodistribucija teži podsticaju potrošača na što manju potrošnju reaktivne energije.

Sagledavanjem najvažnijih karakteristika potrošačkog konzuma i praćenjem njihovog uticaja na ukupne troškove unutar tarifnog sistema, elektrodistribucija sprovodi niz mera i olakšica kojima potrošače podstiče na racionalno trošenje energije.

NAPOMENA:

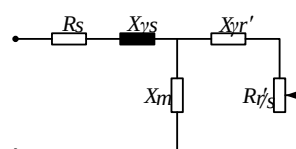
Ovaj rad proistekao je iz master rada čiji je mentor bio dr Dejan Jerkan.

Sa gledišta reaktivne energije tarifni sistem u Srbiji je takav da potrošači plaćaju svu reaktivnu energiju preuzetu iz distributivne mreže, dok količinu prekomerne reaktivne energije plaćaju po višestruko većim cenama. Upravo u ovome vidi se težnja distribucije i podsticaj potrošača na instalisanju kompenzacionih sklopova reaktivne energije.

3. ASINHRONI MOTORI

Asinhroni motori kao najčešće korišćeni elementi elektromotornih pogona poznati su i pod nazivom indukcioni. Ovaj naziv baziran je na činjenici da se energija sa statora na rotor prenosi isključivo elektromagnetnom indukcijom. Asinhroni motori kao izrazito induktivni potrošači smatraju se istaknutim potrošačima reaktivne energije. Procentualno gledano asinhroni motori veći su potrošači reaktivne energije od transformatora, budući da je pored magnećenja magnetnog jezgra jedan deo ove energije potrebno izdvojiti za magnećenje vazdušnog zazora između statora i rotora.

Na sledećoj slici prikazana je pofazna šema asinhronog motora pomoću koje će biti izveden izraz potrošnje reaktivne energije :



Slika 1. Pofazna ekvivalentna šema asinhronog motora

R_s – Statorska otpornost

X_{ys} – Reaktansa rasipanja statora,

X_m – Reaktansa magnećenja,

$X_{yr'}$ – reaktansa rasipanja rotora svedena na stator,

R_r'/s – Rotorska otpornost zavisna od klizanja mašine svedena na stator.

Reaktivna snaga praznog hoda asinhronog motora sa zanemarenim gubicima:

$$Q_m = \sqrt{S_0^2 - P_{fe}^2} \quad (1)$$

Kako je potrošnja snage asinhronog motora u direktnoj vezi sa njegovim opterećenjem, neophodno je u izraz za snagu uvesti K_{opt} faktor kojim će se dosledno predstaviti opterećenje:

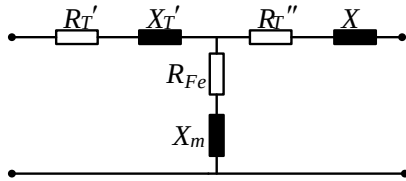
$$S_m = 3 \cdot (K_{opt} \cdot I_{nf})^2 \cdot |Z_{mot}| = 3 \cdot K_{opt}^2 \cdot I_{nf}^2 \cdot \frac{U_n}{I_{nf}} = K_{opt}^2 \cdot S_n \quad (2)$$

Izraz za proračunavanje kompenzacione baterije asinhronog motora izvodi se sledeći izraz:

$$Q_{baterije} = \sqrt{\left(S_n \cdot \frac{i_0[\%]}{100}\right)^2 - P_{fe}^2} + K_{opt} \sqrt{S_n^2 - P_{cu}^2} \approx S_n \cdot \frac{i_0[\%]}{100} + K_{opt}^2 \cdot S_n \quad (3)$$

4. TRANSFORMATORI

Prevashodna uloga transformatora u elektroenergetskim sistemima je da posredstvom elektromagnetne indukcije vrše pretvaranje vrednosti napona i struje definisane prenosnim odnosom transformatora, pri nepromenjenoj frekvenciji. Upravo njihov princip rada zasnovan na magnetnoj indukciji, uz asinhronne motore, svrstava ih u najznačajnije potrošače reaktivne snage. Potrošnja reaktivne snage transformatora praćena je prikazom pofazne ekvivalentne šeme na slici 2.



Slika 2. Pofazna ekvivalentna šema transformatora

Kao polazna tačka za izvedbu potrošnje reaktivne snage transformatora uzima se uključanje neopterećenog transformatora na mrežu opisano izrazom za prividnu snagu:

$$S_{ph} = \sqrt{3} \cdot U_0 \cdot I_0 = \sqrt{3} \cdot U_n \cdot \left(I_n \cdot \frac{i_0[\%]}{100} \right) = S_n \cdot \frac{i_0[\%]}{100} \quad (4)$$

Prateći prethodni izraz i poznavajući gubitke praznog hoda moguće je izračunati potraživanu reaktivnu snagu transformatora:

$$Q_m = \sqrt{S_{ph}^2 - P_{fe}^2} = \sqrt{\left(S_n \cdot \frac{i_0[\%]}{100} \right)^2 - P_{fe}^2} \quad (5)$$

Na pofaznoj šemi transformatora X_T' i X_T'' modeluju redne reaktanse rasipanja magnetnog fluksa u namotajima. Njima je opisan deo reaktivne energije potreban za zatvaranje fluksa između pojedinih namotaja i neferomagnetne sredine. Uzimajući u obzir da se ovim reaktansama opisuje priroda samih namotaja lako se zaključuje da je potrošnja reaktivne energije ovih elemenata zavisn od struje kroz ovaj namotaj tj. od opterećenja transformatora. Kako bi se opterećenje transformatora dosledno iskazalo u izrazu za potrošnju reaktivne energije ovih reaktansi uvodi se faktor opterećenja K_{opt} , iskazan kao relativna vrednost opterećenja u odnosu na naznačenu.

Kako je vrednost struje otočne grane u odnosu na struju redne grane relativno mala, moguće je smatrati da je struja kroz obe redne grane iste vrednosti.

Gubici prividne snage na rednoj grani usled opterećenja transformatora sa predstavljanim faktorom K_{opt} износи:

$$S_k = 3 \cdot (K_{opt} \cdot I_{nf})^2 \cdot |Z_t| = 3 \cdot K_{opt}^2 \cdot I_{nf}^2 \cdot \frac{u_k \cdot U_n^2}{100 \cdot S_n} = K_{opt}^2 \cdot \frac{u_k}{100} \cdot S_n \quad (6)$$

Kvadratni stepen struje u izrazu za Džulove gubitke prenosi se i na sledeći izraz:

$$Q_k = \sqrt{S_t^2 - P_{cu}^2} = \sqrt{\left(K_{opt}^2 \cdot \frac{u_k[\%]}{100} \cdot S_n \right)^2 - (K_{opt}^2 \cdot P_{cu})^2} \\ = K_{opt}^2 \sqrt{\left(\frac{u_k[\%]}{100} \cdot S_n \right)^2 - P_{cu}^2} \quad (7)$$

Kao veoma tačan izraz za izračunavanje ukupne reaktivne energije transformatora pokazao se zbir izraza reaktivnih energija rednih i otočne grane transformatora, spram koga je moguće odrediti vrednost kondenzatorskih baterija za kompenzaciju:

$$Q_{kB} \approx S_n \cdot \frac{i_0[\%]}{100} + K_{opt}^2 \cdot \frac{u_k[\%]}{100} \cdot S_n \quad (8)$$

5. KOMPENZACIJA REAKTIVNE ENERGIJE

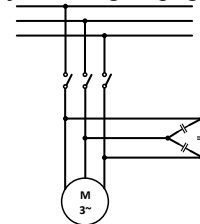
Trend razvoja elektroenergetike i privrede u svetu, ogleda se u što konkurentnijem i efikasnijem načinu poslovanja industrijskih postrojenja.

U cilju što pouzdanijeg i ekonomičnijeg snabdevanja, kao i u cilju popravke kvaliteta električne energije, poslednjih godina sve prisutnija su ulaganja u osavremenjavanje, odnosno automatizaciju procesa proizvodnje, kao i popravku njihovih tehničko-ekonomskih karakteristika.

Industrijski potrošači ostvaruju značajan uticaj u potrošačkom konzumu reaktivne energije elektrodistributivnog sistema, te je kompenzaciju reaktivne energije potrebno koncentrisati unutar postrojenja. Unutar postrojenja bez primenjene kompenzacije, faktor snage pokazuje tendenciju pogoršanja što rezultira ukupnim pogoršanjem stanja u distributivnoj mreži.

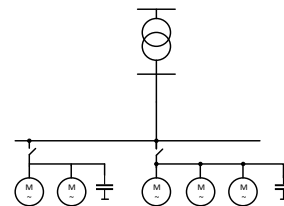
U praksi se kompenzacija reaktivne energije izvodi kroz četiri načina vezivanja kondenzatorskih baterija:

Pojedinačna kompenzacija podrazumeva da se kondenzatorske baterije priključuju direktno na priključke potrošača i da se njihovo uključivanje/isključivanje odvija vremenski istovetno sa radom potrošača. Prema empirijskim zaključcima pojeđinana kompenzacija izvodi se kod potrošača najveće snage u pogonu.



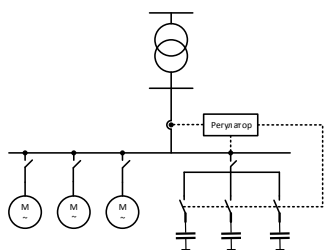
Slika 3. Šematski prikaz pojedinačne kompenzacije

Način **grupne kompenzacije** primenjuje se na mestima gde se istim vodom iz kondenzatorskih jedinica reaktivnom energijom napaja grupa induktivnih potrošača.



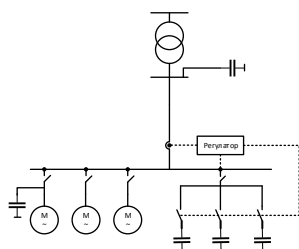
Slika 4. Šematski prikaz grupne kompenzacije

Pri **centralnoj kompenzaciji** reaktivna energija citavog postrojenja nadoknađuje se iz jednog kompenzacionog sklopa. Prednost ovog načina kompenzacije ogleda se u jednostavnom instalisanju opreme. Razvoj programabilnih komponenata automatizacije uvideo je svoju ulogu i pri procesu kompenzacije, tako da u slučaju promenljivog opterećenja postrojenja i centralne(grupne) izvedbe kompenzacije potrošnju reaktivne energije baterija mozemo kontrolisati regulatorom. Ovim uređajem postiže se ravnomerno trošenje kompenzacionog sklopa, cime se radni vek baterija značajno produžava.



Slika 5. Šematski prikaz centralne kompenzacije

Kao najčešći i najefikasniji način izvedbe izdvaja se **mešovita kompenzacija**. Pri ovoj izvedbi uviđaju se prednosti svih prethodnih načina. Kako bi se ovaj vid kompenzacije izveo što efikasnije potrebno je sagledati čitav sistem i njegovu funkcionalnost i prema tome predvideti najpogodniju topologiju kondenzatorskih baterija.



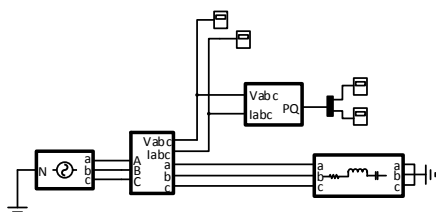
Slika 6. Šematski prikaz mešovite kompenzacije

6. HARMONIJSKA ANALIZA REZONANTNOG KOLA

Baza ovog rada bila je projektovanje električne instalacije pogona konstruisanih sa asinhronim motorima kao dominantnim potrošačima i planiranom kompenzacijom reaktivne snage, te je neophodno bilo razmotriti moguću neželjenu pojavu rezonantnog kola. Pojava rezonance prevashodno je istraživana iz aspekta kvaliteta električne energije i harmonijske analize napojne mreže postrojenja. Radi sagledavanja neželjenih uticaja po postrojenje uticajem rezonantnog kola izrađeno je simulacioni Matlab model.

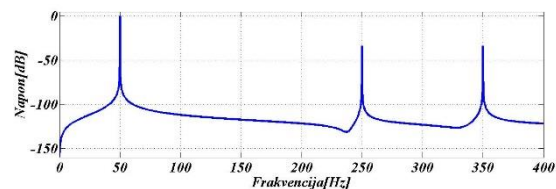
Jednostavan model izrađen od naizmeničnog izvora 400 V, 50 Hz, kojim se simulira napajanje omsko-induktivnog potrošača. Omske, odnosno induktivne vrednosti potrošača izračunate su tako da prilikom dodavanja kondenzatorske jedinice u kolo, pojava rezonance pokaže svoje kritične uticaje na frekvenciji od 250 Hz tj. na petom harmoniku.

Kroz ovaj rad uticaj viših harmonika sagledan je kroz njihovu štetnost po kondenzatorske baterije, prikazanu kroz višestruko veće struje kroz kondenzator. Pojava visokih struja izazvala bi visoka termička naprezanja i na taj način u najkritičnijem slučaju izazvala potpuno uništenje baterije.



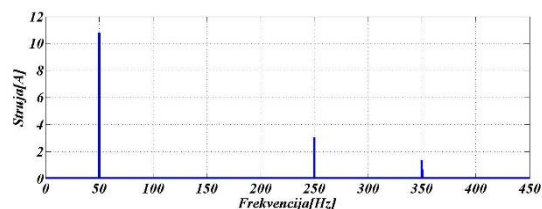
Slika 7. Matlab model simulacije rezonancije

Činjenicom da se napojnom mrežom pored razmatranih postrojenja napaja niz drugih potrošačkih područja, neminovno je bilo procentualno generisati udeo mreže u višim harmonicima sistema. Procentni udeo mreže u generisanju viših harmonika ograničio se na 2 %, koja su se spram negativnih uticaja rezonantnog kola mogla smatrati zanemarljivim.



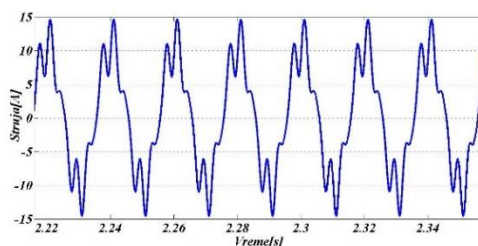
Slika 8. Harmonijski prikaz napona rezonantnog kola

Prethodna slika predstavlja grafički prikaz harmonijskog izobličenja napona izazvanog priključenjem rezonantnog kola u sistem. Procentualno posmatrano harmonijska distorzija napona iznosi 2,89 % što prema pravilnicima kvaliteta električne energije predstavlja dozvoljenu vrednost.



Slika 9. Harmonijski prikaz struje rezonantnog kola

Kritična očekivanja jasno su oslikana u prikazu harmonika struje. Posmatrajući prethodni harmonijski prikaz uviđa se kritično učešće petog harmonika u ukupnoj harmonijskoj distorziji. Procentualno ovaj harmonik prikazan je kao 30,9 % osnovnog harmonika što izlazi iz granica dozvoljenog. Iako je preciznim računom želja bila ostvariti negativan uticaj petog harmonika, u ukupnoj distorziji uviđa se uticaj i sedmog harmonika.



Slika 10. Talasni oblik struje rezonantnog kola

Kao konačni prikaz harmonijskog izobličenja dat je talasni oblik struje mreže, na kom se uviđaju značajna odstupanja od prihvatljive.

Prema svemu do sad izloženom u ovom poglavlju zaključuje se da je pri instalisanju kompenzacije opreme neophodno voditi računa o mogućoj pojavi rezonance i prema potrebi u kondenzatorska kola uključivati filterske prigušnice. Kao opisni faktor filterskih prigušnica usvojen je faktor **p**, iskazan odnosom osnovne(50Hz) i rezonantne frekvencije:

$$p[\%] = 100 \cdot \left(\frac{f_{50}}{f_{\text{rezonantno}}} \right)^2 \quad (9)$$

Gde je rezonantnu frekvenciju moguće odrediti poznavanjem induktivnosti i kapacitivnosti sklopa:

$$f_{\text{rezonantno}} = \frac{1}{2\pi\sqrt{L \cdot C}} \quad (10)$$

Pored prethodnih izraza pri određivanju filtarskih prigušnica neophodno je ispratiti pravne propise i standarde kvaliteta električne energije u kojima su navedene dozvoljene vrednosti pojedinih harmonika. Neretko je u praksi pri podešavanju frekvencije filtra potrebno pribegavati kompromisima po pitanju suzbijanja viših harmonika, smanjenju talasnosti upravljačkih veličina i pojavi rezonancije.

7. REALIZACIJA PRAKTIČNOG DELA

Završni deo rada predstavlja detaljan proračun vodova i prateće opreme neophodnih za bezbedno i nesmetano izvođenje elektroenergetske instalacije područja elektromotornih postrojenja. Početak projektovane trase predstavljen je distributivnom transformatorskom stanicom i proces proračuna završava se na krajnjem potrošaču.

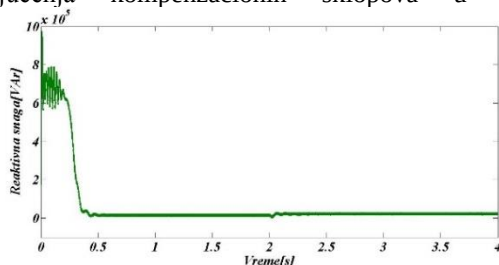
Sledeći tabelarni prikaz predstavlja konačni rezultat izvedenih proračuna sprovedenih za Pogon A. U tabelu su uvršteni strujni krugovi sa početkom od razvodne table pogona.

Kao dodatni segment praktičnog dela izrađen je simulacioni Matlab model sa identičnim potrošačima i planski proračunatom kompenzacijom reaktivne snage.

Strujni krug	Potrošnja [kW]	Presek voda [mm ²]	Trajno dozvoljena struja [A]	Automatski osigurač [A]	Dužina strujnog kruga [m]	Kompenzacione baterije [μF]
1.	75000	50	179	125	4,55	303,363
2.	15000	4	40	25	8,35	Grupna: 171,327
3.	15000	4	40	25	5,5	
4.	37000	16	96	63	10,6	
5.	7500	4	40	25	11,95	Grupa 124,506
6.	7500	4	40	25	9,98	
7.	7500	4	40	25	13,1	
8.	4000	1	17	10	17,7	
9.	4000	1	17	10	20,55	

Tabela 1. Tabelarni prikaz Pogona A.

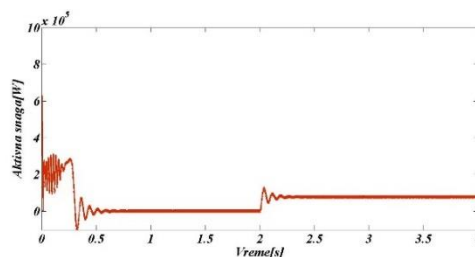
Proračun kondenzatorskih baterija izveden je prema izmerenom faktoru snage određenih segmenata pogona i baterije su izabrane kako bi ovi faktori doveli na željene vrednosti. Validnosti proračuna jednostavno je proprađena signalima reaktivne snage mreže nakon uključenja kompenzacionih sklopova u pogonu.



Slika 11. Signal reaktivne snage mreže nakon početka kompenzacije

Kompenzacioni sklop i potrošačka grupa početak svog rada započinju istovremeno sa početkom procesa simulacije. Posmatrajući signal reaktivne snage zaključuje se da na samom početku simulacije reaktivna snaga ima kratkotrajno visoku vrednost, nakon čega se signal smiruje i ustaljuje na vrednosti veoma bliskoj nuli.

Tokom daljeg toka simulacije ne uviđaju se značajnija odstupanja od ove vrednosti, sem kratkotrajni zanemarljivi pikovi u trenucima promene opterećenja. Kako bi se jasnije sagledali trenuci promene opterećenja na slici 12. priložen je i signal aktivne snage pogona.



Slika 12. Signal aktivne snage koju potrošačka grupa preuzima iz mreže

8. ZAKLJUČAK

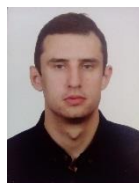
Izradom projekta elektroenergetske instalacije simuliranih motornih postrojenja i težnji za njihovim što efikasnijim funkcionisanjima neophodno je bilo ispratiti mnoge segmente energetske stabilnosti. Kao veoma važan deo električne instalacije pogonskog postrojenja smatra se deo za kompenzaciju. Pored ostvarenih benefita primenom kompenzacije reaktivne energije, neophodno je bilo sagledati i moguća neželjena stanja do kojih može doći pri instalisanju kondenzatorskih jedinica u mrežu pretežno opterećenom asinhronim motorim.

Kroz rad razmotrena je upotreba mogernih regulatora reaktivne energije i njihovo pridruživanje kompenzacionim sklopovima. Izneseni zaključak je da je njihova upotreba efikasna i ekonomski isplativa isključivo u većim pogonskim postrojenjima, sa čestom promenom opterećenja.

9. LITERATURA

- [1] Strahil J. Gušavac: Osnovni principi projektovanja u mrežama srednjeg i niskog napona, Fakultet tehničkih nauka, Novi Sad, 2014.
- [2] Gojko Dotlić: Elektroenergetika kroz standarde, pravilnike i tehničke preporuke, Savez mašinskih i elektrotehničkih inženjera i tehničara Srbije, Beograd, 2004.
- [3] Duško Bekut: Relejna zaštita, Fakultet tehničkih nauka, Novi Sad, 2009.
- [4] Miomir Kostić: Teorija i praksa projektovanja električnih instalacija, Akademski misao.
- [5] Vladan Vučković: Električni pogoni, Akademski misao, Beograd.

Kratka biografija:



Mladen Vuković rođen je u Novom Sadu 1994. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio 2018. godine. kontakt: mladenvukovic1994@gmail.com



Dejan Jerkan je docent na Fakultetu tehničkih nauka u Novom Sadu, na Katedri za Energetsku elektroniku i pretvarače. Oblast interesovanja su mu modelovanje i dijagnostika električnih mašina, kao i metoda konačnih elemenata.

IDENTIFIKACIJA I UPRAVLJANJE CNG SISTEMOM OPISANOG IRACIONALNOM FUNKCIJOM PRENOSA**IDENTIFICATION AND CONTROL OF CNG SYSTEM DESCRIBED BY IRRATIONAL TRANSFER FUNCTION**Filip Stojanović, *Fakultet Tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu prikazan je jedan pristup modelovanja složenog dinamičkog proces koji obuhvata identifikaciju strukture modela kao i estimaciju parametara modela. Opisani pristup biće prikazan na primeru modelovanja dinamičkog pritiska u CNG (Compressed Natural Gas) sistemu. Identifikovani model je potom korišćen za projektovanje optimalnog regulatora za upravljanje datim sistemom.

Ključne reči: Identifikacija, PID kontroler, CNG, common rail

Abstract - The purpose of this article presents a modeling approach to a complex dynamic process which includes model structure identification as well as model parameter estimation. The described approach will be exemplified by the modeling of dynamic compression in the CNG (Compressed Natural Gas) system.

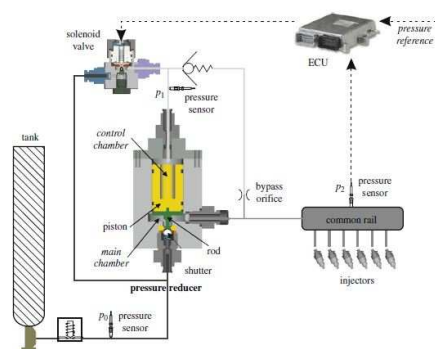
Keywords: Identification, PID controller, Common rail

1. UVOD

Rastuća potražnja za motorima sa smanjenim troškovima goriva ili sistemima podsticaja za smanjeni porez i mobilnost, koji se primenjuju u mnogim zemljama, podstakla je proizvođače automobila da razviju sisteme koji rade na alternativnim gorivima. Benzinski motori sa unutrašnjim sagorevanjem koji rade na komprimovanom prirodnom gasu bazirani su na proverenim tehnologijama i obično imaju nižu emisiju izduvnih gasova i niže troškove goriva nego vozila na benzin i dizel. Očekuje se da će broj vozila sa pogonom na CNG (engl. *Compressed Natural Gas*) nastaviti da raste u idućim godinama. Mnogo je razloga za povećanu upotrebu alternativne tehnologije pogonskog sklopa CNG kao goriva. U motorima sa unutrašnjim sagorevanjem, koji su opremljeni sistemom ubrizgavanja *common rail injection* sistem, tačnost merenja vazduha i goriva striktno zavisi od regulacije pritiska. Tačnost u merenju je teško postići posebno za sisteme sa komprimovanim prirodnim gasom, jer kompresibilnost gasa čini proces isporuke goriva složenijim. Takođe, on sagoreva praktično bez čestica. U radu se govori o identifikaciji sistema nelinearnog modela (zbog složenosti nelinearnog sistema radi se sa linearnim sistemom, tako da se opisuje

ponašanje sistema oko neke radne tačke), njegovih parametara, kao i o modelovanju tog sistema.

Ovaj rad je posebno usmeren na definisanje dovoljno jednostavnog modela, koji je potom korišćen za projektovanje kontrolera za upravljanje opisanom sistemom. Sve pretpostavke su jasno specificirane u svakoj fazi postupka modelovanja. Konačni rezultati su potvrđeni upoređivanjem sa eksperimentalnim skupom podataka. Izgled CNG sistema koji je bio predmet istraživanja dat je na slici 1.



Slika 1: Common rail sistem. (Slika preuzeta iz [1])

2. MOTORI SA CNG SISTEMOM

Prirodni gas, kao alternativno gorivo sa povoljnim ekološkim karakteristikama, je veoma atraktivan za primenu u današnje vreme i u budućnosti. Kao motorno gorivo, on se koristi u sabijenom obliku, gde se čuva u rezervoarima i pod pritiskom od 300 bar-a. Da bi se koristio u sklopu automobila potrebno je izvršiti redukciju pritiska i ugraditi *common rail* sistem za ubrizgavanje. Kako bi se postigla precizna regulacija pritiska u *common rail*-u, zapremina u reduktoru pritiska se deli na gornju „kontrolnu (upravljačku) komoru” i donju „glavnu komoru”. Komore su razdvojene pokretnim klipom i obe se napajaju gorivom iz rezervoara. Vertikalno pomeranje tog klipa zavisi od upravljanja i pritiska u glavnoj komori. Dotok pritiska kontrolne komore podešava se elektro-hidrauličnim solenoidnim ventilom, dok se dotok u glavnu komoru reguliše pomeranjem poklopca koji se nalazi na samom klipu. Protok u upravljačkoj komori će dovesti do porasta pritiska gasa na gornjoj površini klipa i time će se pomeriti ka dole. Time se i poklopac pomera te omogućava veći dotok u glavnu komoru čime povećava pritisak. Nasuprot tome,

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Mirna Kapetina, docent.

zatvaranje ventila smanjuje pritisak na gornjoj površini klipa koji se kreće na gore, poklopac se zatvara preko opruge i pritisak se smanjuje. Stoga, pošto se *common rail* puni gorivom koje dolazi iz upravljačke i glavne komore, elektronska upravljačka jedinica (ECU) može regulisati pritisak tako što pravilno upravlja servo ventilom. Precizna regulacija pritiska za ubrizgavanje je složena zbog sabijenosti stanja gasa, zbog čega se kao rešenje nameće projektovanje upravljanja zasnovano na modelima. Modelovanje CNG sistema sa ubrizgavanjem je korisno kako za analizu tako i za projektovanje upravljanja.

3. IDENTIFIKACIJA CNG SISTEMA

Model CNG sistema za ubrizgavanja razmatra dve zapremine kojima se upravlja, tj. upravljačku komoru i *common rail* krug (uključujući glavnu komoru i *common rail*).

Pretpostavimo da je pritisak u upravljačkoj komori p_1 , pritisak u glavnoj komori p_2 i da su x_1 i x_2 njihova promenljiva stanja, respektivno. Signali solenoidnog ventila ET_{sv} i injektora ET_{inj} kao ulazne promenljive u_1 i u_2 , takođe respektivno. Tj. $x_1 = p_1$, $x_2 = p_2$, $u_1 = ET_{sv}$ i $u_2 = ET_{inj}$.

U literaturi postoje različiti modeli ovakvog sistema [1]. Oni su po svojoj strukturi složeni i veoma nelinearni. Kako je većina metoda za projektovanje regulatora namenjena za linearne sistema i za takve pokazuje dobre performanse, cilj je modelovati proces linearnim sistemom u okolini radne tačke.

Ukoliko prostorna komponenta dinamike procesa nije od interesa, i želimo da uzmemo samo vezu ulaza i izlaza, onda je pogodan način za opisivanje takvih sistema opisivanje uz pomoć raznih oblika iracionalnih funkcija prenosa [2]. Kada govorimo o frakcionim modelima, njihova osnovna karakteristika je da dopuštaju amplitudsku karakteristiku koja je proizvoljnog negativnog nagiba. Kao posledica toga, logaritamska amplitudska frekvencijska (Bodeova) karakteristika kod frakcionih sistema može da sadrži segmentne proizvoljnog nagiba.

Obično se to zapisuje kao $\alpha \times 20\text{dB/dec}$, gde α može imati različite vrednosti. Ako to uporedimo sa sistemima celobrojnog reda, videćemo da se kod njih javljaju segmenti čiji su nagibi isključivo jednaki celobrojnog umnošku 20 dB/dec. Još jedna od osobina frakcionih sistema jeste da oni imaju fazno kašnjenje koje je konstantno. Promenom reda ta vrednost se može kontinualno podešavati. Kao posledica toga, asimptotsko fazno kašnjenje kod sistema frakcionog reda na visokim učestanostima može imati proizvoljnu vrednost. Kod klasičnih sistema to fazno kašnjenje može imati samo vrednosti celobrojnih umnožaka pravog ugla ($n\frac{\pi}{2}$) [2]. Postoje nekoliko skupova podataka koji su dobijeni eksperimentalnim putem iz FIAT-ovog istraživačkog centra (*Valenzano, Italy*) [1]. Jedan od tih skupova podataka korišćen je u ovom radu najpre za identifikaciju strukture modela nad kojim će kasnije biti vršeno upravljanje tj. za projektovanje optimalnog regulatora za upravljanje datim sistemom. Kako bi bilo intuitivnije, u tabeli 1 su napomenute i date osnovne veličine tj. ulazi i izlazi iz postojećeg skupa podataka. To su veličine koje figurišu u modelu (i samom sistemu, naravno) i koje su korišćene pri identifikaciji.

Tabela 1: Veličine u sistemu.

Veličine	
Simboli	Signal
$ET_{ev} (u_1)$	solenoidnog ventila [A]
$ET_{inj} (u_2)$	injektora (jedan injektor) [A]
$p_1 (x_1)$	pritisak upravljanja (engl. <i>control pressure</i>) [bar]
$p_2 (x_2)$	pritisak u šini (engl. <i>common rail pressure</i>) [bar]

Identifikacija sistema vršena je iz dva dela. Prvi deo sistema činio je deo sistema od u_1 do p_1 , dok je drugi deo sistema nad kojim je vršena identifikacija bio deo sistema od p_1 do p_2 . Metod identifikacije sastoji se od sledećih koraka:

- formiranje diskretnog ARX modela visokog reda na osnovu raspoloživih skupova podataka,
- analiza frekvencijske karakteristike dobijenog modela i identifikacija adekvatne strukture frakcionog sistema na osnovu Bodeove karakteristike,
- estimacija parametara modela primenom PSO algoritma [3].

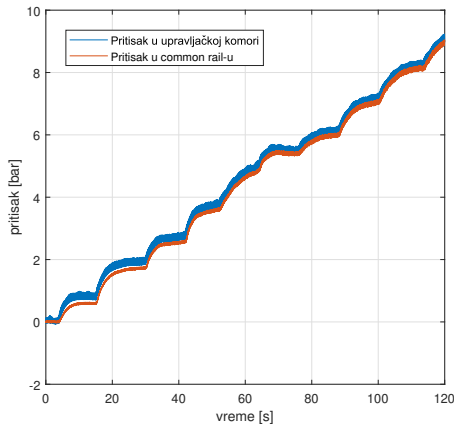
3.1 Identifikacija frekvencijskih karakteristika

U naučnoj literaturi postoji veći broj metoda za identifikaciju modela frakcionog reda [5]. Različiti pristupi za modelovanje uslovljeni su različitim podacima koji su raspoloživi i nameni modela. U sklopu ovog rada na raspolaganju su bili podaci u vremenskom domenu. Takve podatke je bilo moguće iskoristiti da bi se formirao autoregresioni model sa egzogenom (spoljašnjom) pobudom (eng. *Auto-Regressive with eXogenous input model*), tj. ARX model visokog reda. Dalja analiza frekvencijskih karakteristika tog modela omogućila je procenu strukture frakcionog modela nižeg reda koji će dovoljno dobro aproksimirati frekvencijske karakteristike dobijenog ARX modela. Ovi postupci će biti opisani u nastavku.

Kako je već navedeno, predloženo je da se skupovi podataka u vremenskom domenu iskoriste za formiranje ARX modela visokog reda. Strukturu ovog modela možemo posmatrati kao vremenski diskretan proces opisan diferencnom jednačinom:

$$A(q)y(k) = B(q)u(k-1) + e(k), \quad (1)$$

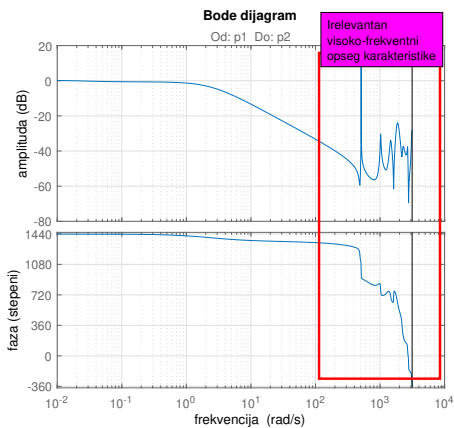
gde y predstavlja izlaz iz procesa, u je zadati ulaz, a e je poremećaj. $A(q)$ i $B(q)$ su polinomi gde promenljiva q predstavlja operator jediničnog prednjačenja. Ti polinomi su reda n i $n-1$, respektivno. Kažemo da red modela n mora biti dovoljno veliki kako bi se mogla opisati sva dinamika sistema. Pre bilo čega, tj. bilo koje identifikacije nekog linearnog modela, neophodan je izbor radne tačke. Radne tačke u ovom slučaju jesu 4.596 i 3.688. Ulazni pritisci sa varijacijom signala oko radne tačke dati su na slici 2.



Slika 2: Varijacije signala oko radne tačke, pri čemu se signali koriste kao ulazni i izlazni podaci u procesu identifikacije. (Perioda uzrokovanja je 1ms, a Nikvistova frekvencija je 500Hz)

3.2 Formiranje diskretnog ARX modela

Empirijskim putem, nakon nekoliko numeričkih testova, određeno je da je odgovarajući ARX model 20-og reda, jer modeli nižeg reda nisu mogli adekvatno da predstavljaju frekventnijske karakteristike u opsegu koji je bio od interesa. Na slici 3 prikazana je frekventnijska karakteristika modela koji je dobijen. Lako se može uočiti da je amplitudska frekventnijska karakteristika približno linearna po segmentu, međutim, takođe sa nagibom koji nije celobrojni umnožak od 20 dB/dec.



Slika 3: Bodeova karakteristika ARX modela.

3.3 Identifikacija frakcionog modela

Ukoliko analiziramo amplitudsko frekventnijsku karakteristiku ARX modela sa slike 3 dolazimo do modela sledeće strukture

$$G(s) = \frac{k}{\left(\frac{s}{\omega_1} + 1\right)^{\alpha_1}} \quad (2)$$

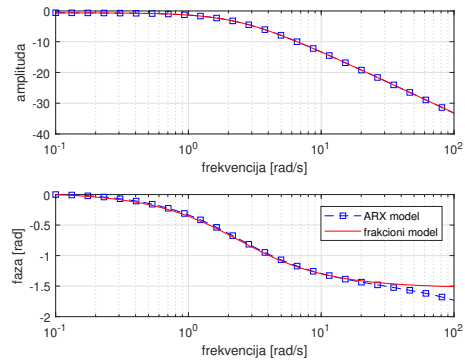
Parametri u jednačini su pojačanje k , prelomna frekvencija ω_1 i koeficijent necelog reda α_1 . Vrednosti parametara modela određene su minimizacijom greške amplitudskih odziva dobijenog ARX modela i frakcionog modela, odnosno, minimizacijom kriterijuma optimalnosti koji je naveden

$$J = \sum_{k=1}^N \left| 20 \log |G(j\xi_k)| - 20 \log \left| \frac{B(e^{j\xi_k T})}{A(e^{j\xi_k T})} \right| \right|, \quad (3)$$

gde $\xi_k, k \in \{1, \dots, N\}$ predstavljaju ugaone frekvencije uniformno raspoređene u opsegu od $\omega_{min} = 0.1$ rad/s do $\omega_{max} = 100$ rad/s, a $N = 100$.

4. NUMERIČKI REZULTATI SIMULACIJE

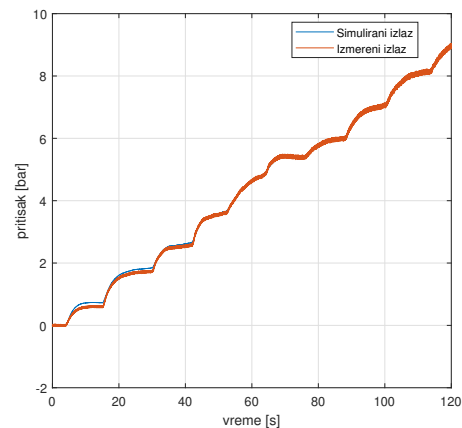
Nakon optimizacije frakcionog modela (2) PSO algoritmom u cilju dobijanja parametara frakcionog modela, ti parametri prikazani su u $k = 0.929843$, $\omega_1 = 2.385601$ i $\alpha_1 = 1.002118$. Na osnovu dobijenih parametara optimizacije, sa slike 4 možemo videti slaganje frakcionog i ARX modela u frekventnijskom domenu.



Slika 4: Frekventnijske karakteristike dobijenog ARX modela i optimalnog frakcionog modela prikazane u relevantnom.

4.1 Validacija frakcionog modela

Na slici 5 prikazani su izlazi iz ARX modela i podataka koji su eksperimentalno dobijeni u vremenskom domenu.



Slika 5: Prikaz validacije modela: poređenje rezultata simulacije dobijenog ARX modela sa eksperimentalnim rezultatima.

Ovakav postupak identifikacije formiran je za modelovanje sistema od signala u_1 do x_1 , kao i od x_1 do x_2 za više skupova podataka. Kao rezultat toga dobijene su dve funkcije prenosa G_1 i G_2 koje u proizvodu čine funkciju prenosa procesa koja opisuje uticaj ulaza ventila u_1 na pritisak p_2 , odnosno promenljivu x_2 . S obzirom da se radi, o kako je već rečeno, frakcionim funkcijama, kako bi se obavila simulacija, one su aproksimirane racionalnim funkcijama nekog višeg reda. Pošto govorimo o rednoj vezi ove dve funkcije, onda će proizvod novoformiranih funkcija $G_1(s)$ i $G_2(s)$ činiti funkciju procesa celokupnog sistema $G_p(s)$. Funkcija prenosa celokupnog procesa data je u jednačini. Takva

funkcija prenosa (4) je dalje poslužila za formiranje optimalnog regulatora

$$G_p(s) = \frac{1201 \cdot 10^4 s^4 + \dots + 1665 \cdot 10^8}{s^6 + 33870 s^5 + \dots + 1118 \cdot 10^7 s + 2864 \cdot 10^6} \quad (4)$$

5. OPTIMALNI PID REGULATOR

U ovom poglavlju biće reči o optimalnom PID regulatoru koji je korišćen za sistem koji je prethodno modelovan.

5.1 PID regulator za CNG sistem

Najpre su definisane su mere za ocenu kvaliteta regulacije koje su primenjene da bi se pronašli optimalni parametri za PID regulator kojim se vrši upravljanje. Kriterijum optimalnosti i odgovarajuća ograničenja koja su u izvedbi predstavljena kao kaznene funkcije treba da garantuju kvalitet regulacije. Optimizacijom su dobijeni parametri PID regulatora koji su se pokazali kao dobri. Zahvaljujući tim merama performansi sistema projektovan je odgovarajući PID regulator za upravljanje dinamičkim pritiskom CNG sistema.

Funkcija prenosa sistema navedena je u prethodnom delu rada dok regulator treba da bude u formi

$$G_c(s) = \frac{K_p + \frac{K_i}{s} + K_d s}{s T_f + 1} \quad (5)$$

Kriterijum optimalnosti je integral apsolutne greške koji se definiše kao:

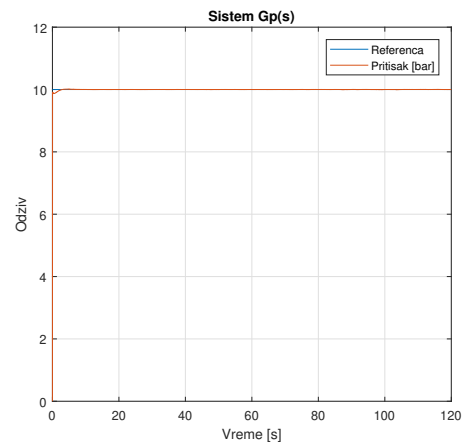
$$IAE = \int_0^\infty |e(t)| dt \quad (6)$$

Ograničenja su data kao:

$$\begin{aligned} M_s^{min} &\leq \max_{\omega \geq 0} \left| \frac{1}{1 + G_c(j\omega)G_p(j\omega)} \right| \leq M_s^{max}, \\ Q &= \max_{\omega \geq 0} \left| \frac{k_i \frac{G_p(j\omega)}{j\omega}}{1 + G_c(j\omega)G_p(j\omega)} \right| \leq Q^{max}, \\ M_p^{min} &\leq \max_{\omega \geq 0} \left| \frac{G_c(j\omega)G_p(j\omega)}{1 + G_p(j\omega)G_c(j\omega)} \right| \leq M_p^{max}, \end{aligned}$$

Parametri M_s^{min} i M_s^{max} su minimalna i maksimalna vrednost vrednosti M_s , a njihove vrednosti su redom 1.7 i 2. Parametar Q^{max} predstavlja ograničenje na rezonantni vrh frekvencijske karakteristike, a već je pokazano da je ta vrednost 1.01. Parametri M_p^{min} i M_p^{max} predstavljaju minimalnu i maksimalnu dozvoljenu vrednost M_p i te vrednosti su 1.3 i 1.6, respektivno. Parametar M_n^{max} predstavlja maksimalnu dozvoljenu osetljivost na šum i ima vrednost 20. Navedena ograničenja se dodaju u kriterijum optimalnosti uz pomoć kaznenih funkcija, te se dobija konačni kriterijum optimalnosti koji predstavlja funkciju prilagođenosti za PSO algoritam.

Posle optimizacije PSO algoritmom dobijaju se parametri PID regulatora $K_p = 2.35$, $K_i = 0.95$, $K_d = 1.85$ i $T_f = 0.1$. Na slici 6 prikazan je odziv CNG sistema koji je modelovan i na koji deluje izlazni poremećaj u vidu signala injektora, kada se njime upravlja PID regulatorom sa gore navedenim parametrima. Sa slike 6 se takođe može videti da regulator uspešno može da se izbori sa ovom vrstom poremećaja i da eliminiše njegovo dejstvo u pristojnom vremenskom intervalu.



Slika 6: Odziv CNG sistema upravljen PID regulatorom.

Stoga, govorimo o sistemu koji je stabilan i koji zadovoljava kriterijume performansi i robusnosti.

6. ZAKLJUČAK

Ovaj rad se bavi problemom identifikacije modela i upravljanjem CNG sistemom. Predstavljena je ideja da se fenomen nelinearnog procesa modeluju frakcionim modelom u frekvencijskom domenu, a verifikacija je izvršena poređenjem dobijenih rezultata simulacije sa eksperimentalnim podacima. Dobijeni model je potom poslužio za projektovanje optimalnog regulatora koji je pokazao zadovoljavajuće performanse u rezultatima simulacije.

7. LITERATURA

- [1] P. Lino, B. Maione and A. Rizzo, "Nonlinear modelling and control of a common rail injection system for diesel engines.", *Applied Mathematical Modelling*, 9(31), 1770–1784, 2007.
- [2] M. N. Kapetina, "Adaptivna estimacija parametara sistema opisanih iracionalnim funkcijama prenosa.", Novi Sad, 2017.
- [3] Ž. Kanović, M. R. Rapaić and Z. D. Jeličić, "Generalized particle swarm optimization algorithm - theoretical and empirical analysis with application in fault detection.", *Applied Mathematics and Computation*, 217(24):10175–10186, 2011.
- [4] M. N. Kapetina, P. Lino, G. Maione and M. R. Rapaić, "Estimation of non-integer order models to represent the pressure dynamics in commonrail natural gas engines.", *In 20th IFAC World Congress Toulouse, France.*, 15116–15121, 2017.
- [5] S. Victor, R. Malti, H. Garnier and A. Oustaloup, "Parameter and differentiation order estimation in fractional models.", *Automatica* 49(4):926 – 935, 2013.

Kratka biografija:



Filip Stojanović rođen je u Vranju 1995. god. Nakon završetka gimnazije, 2014. god. upisuje Fakultet tehničkih nauka smer Elektrotehnika i računarstvo čije osnovne studije završava 2018. Kontakt: filipicastojanovic@gmail.com

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2019. godine učestvovali su sledeći recenzenti:

Aco Antić	Đorđe Lađinović	Milan Trivunić	Staniša Dautović
Aleksandar Erdeljan	Đorđe Obradović	Milan Vidaković	Stevan Gostojić
Aleksandar Ristić	Đorđe Vukelić	Milena Krklješ	Stevan Milisavljević
Bato Kamberović	Đula Fabian	Milica Kostreš	Stevan Stankovski
Biljana Njegovan	Đura Oros	Milica Miličić	Strahil Gušavac
Bogdan Kuzmanović	Đurđica Stojanović	Mijodrag Milošević	Svetlana Nikoličić
Bojan Batinić	Filip Kulić	Milovan Lazarević	Tanja Kočetov
Bojan Lalić	Goran Sladić	Miodrag Hadžistević	Tatjana Lončar -
Bojan Tepavčević	Goran Švenda	Miodrag Zuković	Turukalo
Bojana Beronja	Gordana	Mirjana Damnjanović	Uroš Nedeljković
Branislav Atlagić	Milosavljević	Mirjana Malešev	Valentina Basarić
Branislav Nerandžić	Gordana Ostojić	Mirjana Radeka	Velimir Čongradec
Branislava	Igor Budak	Mirko Borisov	Veran Vasić
Novaković	Igor Dejanović	Miro Govedarica	Veselin Perović
Branka Nakomčić	Igor Karlović	Miroslav	Vladimir Katić
Branko Milosavljević	Ivan Beker	Hajduković	Vladimir Strezoski
Branko Škorić	Ivana Katić	Miroslav Popović	Vlado Delić
Damir Đaković	Ivana Kovačić	Mitar Jocanović	Vlastimir Radonjanin
Danijela Lalić	Ivana Miškeljin	Mladen Kovačević	Vuk Bogdanović
Darko Čapko	Jasmina Dražić	Mladen Radišić	Zdravko Tešić
Darko Marčetić	Jelena Atanacković	Nemanja	Zoran Anišić
Darko Reba	Jeličić	Stanisavljević	Zoran Brujić
Dejan Ubavin	Jelena Borocki	Nemanja Sremčev	Zoran Jeličić
Dejana Nedučin	Jelena Kiurski	Nikola Đurić	Zoran Mitrović
Dragan Ivanović	Jelena Radonić	Nikola Jorgovanović	Zoran Papić
Dragan Ivetić	Jovan Petrović	Nikola Radaković	Željko Trpovski
Dragan Jovanović	Jovanka Pantović	Ninoslav Zuber	Željko Jakšić
Dragan Kukolj	Ksenija Hiel	Ognjen Lužanin	
Dragan Mrkšić	Laslo Nađ	Pavel Kovač	
Dragan Pejić	Lazar Kovačević	Peđa Atanasković	
Dragan Šešlija	Leposava Grubić	Petar Malešev	
Dragana Bajić	Nešić	Predrag Šiđanin	
Dragana	Livija Cvetičanin	Radivoje Dinulović	
Konstantinović	Ljiljana Vukajlov	Radovan Štulić	
Dragana Šarac	Ljiljana Cvetković	Relja Strezoski	
Dragana Štrbac	Ljubica Duđak	Slavica Mitrović	
Dragoljub Šević	Maja Turk Sekulić	Slavko Đurić	
Dubravka Bojanić	Marko Todorov	Slobodan Dudić	
Dušan Dobromirov	Marko Vekić	Slobodan Krnjetin	
Dušan Gvozdenac	Maša Bukurov	Slobodan Morača	
Dušan Kovačević	Matija Stipić	Sonja Ristić	
Dušan Uzelac	Milan Rapajić	Srđan Kolaković	
Duško Bekut	Milan Simeunović	Srđan Popov	
Đorđe Ćosić	Milan Trifković	Srđan Vukmirović	