



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXXV

Број: 1/2020

Нови Сад

Едиција: „Техничке науке – Зборници“

Година: XXXV

Свеска: 1

Издавач: Факултет техничких наука Нови Сад

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

Уредништво:

Проф. др Раде Дорословачки

Проф. др Драгиша Вилотић

Проф. др Срђан Колаковић

Проф. др Владимир Катић

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 прва овогодишња свеска часописа „Зборник радова Факултета техничких наука“.

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

Друштво у коме живимо базирано је на знању. Оно претпоставља реорганизацију наставног процеса и увођење читавог низа нових струка, као и квалитетну организацију научног рада. Значајне промене у структури високог образовања, везане за имплементацију Болоњске декларације, усвајање нове и активне улоге студената у процесу образовања и њихово све шире укључивање у стручне и истраживачке пројекте, као и покретање нових 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 30.09.2019. до 31.10.2019. год., а који се промовишу 27.01.2020. 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 две свеске.

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

- машинства и
- електротехнике и рачунарства.

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

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

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

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

„Високо место у друштву најбољих“

Уредништво

SADRŽAJ

	STRANA
Radovi iz oblasti: Mašinstvo	
1. Milan Lončar, Branko Štrbac, ANALIZA TAČNOSTI KOORDINATNIH MERNIH SISTEMA PRI ODREĐIVANJU GREŠKE RAVNOSTI	1-4
2. Radmila Miljanović, Damir Đaković, POSTROJENJE SUŠARE ZA ŠLJIVE NA BIOMASU	5-8
3. Nikola Oluški, ISTRAŽIVANJE STRUJANJA JONSKE TEČNOSTI PRIMENOM RAČUNARSKE DINAMIKE FLUIDA	9-12
4. Александар Рацић, АНАЛИЗА СТРУЈАЊА У ЦЕНТРИФУГАЛНИМ ПУМПАМА ПРИМЕНОМ МЕТОДЕ КОНАЧНИХ ЗАПРЕМИНА	13-16
5. Čedomir Ždrnja, Borislav Savković, MODELIRANJE SILA REZANJA PRI ČEONOM GLODANJU VISOKO-HROMIRANOG LIVENOG GVOŽĐA	17-20
6. Радивојка Вучинић, ДЕТЕКЦИЈА КАВИТАЦИЈСКИХ РЕЖИМА У ЦЕНТРИФУГАЛНИМ ПУМПАМА ПРИМЕНОМ ВИБРОДИЈАГНОСТИЧКИХ МЕТОДА	21-24
7. Siniša Kraljević, Ninoslav Zuber, Dejan Reljić, VIBRODIJAGNOSTIKA STANJA ELEKTROMOTORA SA OŠTEĆENIM ROTOROM	25-28
Radovi iz oblasti: Elektrotehnika i računarstvo	
1. Mladen Vučković, Vladimir Popović, IDENTIFIKACIJA PARAMETARA EKVIVALENTNE ŠEME ASINHRONE MAŠINE METODOM DVA TEST SIGNALA	29-32
2. Stefan Bratić, PRIMENA SPRING CLOUD PROGRAMSKOG OKVIRA ZA RAZVOJ MIKROSERVISNIH APLIKACIJA	33-36
3. Nevena Đerić, SISTEM ZA PERSONALIZACIJU VIDEO TUTORIJALA ZASNOVAN NA MAŠINSKOM UČENJU	37-40

4. Novica Šarenac, PRIMENA UČENJA USLOVLJAVANJEM NA OBUČAVANJE AGENTA ZA AUTONOMNO KRETANJE U OKRUŽENJU CarRacing-v0	41-44
5. Predrag Njegovanović, NAVIGACIJA U TRODIMENZIONALNOM OKRUŽENJU Obstacle Tower UPOTREBOM UČENJA USLOVLJAVANJEM	45-48
6. Boris Knežević, ULAZNI STEPEN EDUKATIVNOG ELEKTROFIZIOLOŠKOG AKVIZICIONOG UREĐAJA	49-52
7. Igor Trpovski, AUTOMATSKA DETEKCIJA STAVKI MENIJA UNUTAR TEKSTOVA RECENZIJA RESTORANA...	53-56
8. Milorad Maksimović, PRIMENA MAŠINSKOG UČENJA U ANALIZI RELEJNE ZAŠTITE PRI STARTU ASINHRONIH MOTORA	57-60
9. Viktor Trifunjagić, KOORDINACIJA PREKOSTRUJE ZAŠTITE SA INVERZNOM KARAKTERISTIKOM REAGOVANJA U DISTRIBUTIVNOJ MREŽI	61-64
10. Marija Sarić, ZAŠTITA KABLOVA	65-68
11. Marija Jovanović, Vladimir Katić, KROVNE VETROELEKTRANE - MODELOVANJE I PREGLED SOFVERSКИH ALATA	69-72
12. Марија Ковачевић, СИСТЕМ ЗА УПРАВЉАЊЕ РАЊИВОСТИМА У СОФТВЕРУ	73-76
13. Nevena Lazarević, SISTEM ZA UPRAVLJANJE BEZBIJEDNOŠĆU RAČUNARSKIH RESURSA	77-80
14. Dragan Erić, NAPREDNA KONTROLA PRISTUPA U PAMETNIM MREŽAMA	81-84
15. Stevan Srećkov, PROFIBUS INDUSTRIJSKA PERIFERIJA	85-88
16. Miloš Vorkapić, Vladimir Katić, IZBOR ELEKTRIČNIH MOTORA ZA POGON ELEKTRIČNIH VOZILA	89-92
17. Marija Kukić, RAZVOJ GRAFIČKOG NAMENSKOG JEZIKA ZA ARHIVIRANJE I MIGRACIJU BAZA PODATAKA	93-96
18. Žaklina Stojšin, GRAFIČKI INTERFEJS EDUKATIVNOG ELEKTROFIZIOLOŠKOG AKVIZICIONOG UREĐAJA	97-100
19. Горан Мамлић, Веран Васић, Ђура Орос, ЕЛЕКТРОМАГНЕТНА КОМПАТИБИЛНОСТ ПРЕТВАРАЧА ЕНЕРГЕТСКЕ ЕЛЕКТРОНИКЕ У ИНДУСТРИЈИ	101-104
20. Darian Pudić, Predrag Teodorović, CONNECTOR ANDROID APLIKACIJA	105-108
21. Zvezdana Klačnić, PROGRAMSKO REŠENJE ZA PRAĆENJE MREŽNOG SAOBRAĆAJA SCADA PROTOKOLA	109-112
22. Зоран Лулеџија, РЕАКТИВНО ПРОГРАМИРАЊЕ У ПРОГРАМСКОМ ЈЕЗИКУ JAVA	113-116
23. Bojana Samardžić, ANALIZA BEZBEDNOSNIH RIZIKA I MERA ZAŠTITE VEB APLIKACIJA	117-120
24. Milena Počuča, Srđan Popov, PREDLOG ARHITEKTURE CI/CD PIPELINE-A KORIŠĆENJEM JENKINS I DOCKER ALATA	121-124
25. Milan Stanković, ANALIZA ALGORITAMA ZA VEKTORSKU REPREZENTACIJU TEKSTA	125-128
26. Ilija Grbić, PROCES RAZVOJA GENERIČKE POSLOVNE APLIKACIJE	129-132

	STRANA
27. Olivera Hrnjaković, ANALIZA I OBRADA TEKSTA POMOĆU RAZLIČITIH MODELA TEMA	133-136
28. Ivana Tešanović, VEB APLIKACIJA BAZE RETKIH BOLESTI NA SRPSKOM JEZIKU	137-139
29. Dejan Stojkić, PRIMENA KVIZA DOMENSKOG ZNANJA UZ POMOĆ SPHERO ROBOTA U OBRAZOVANJU	140-143
30. Miloš Vulin, VIŠEKRITERIJUMSKO ODLUČIVANJE U PLANIRANJU AUTOMATIZACIJE ELEKTRODISTRIBUTIVNIH MREŽA	144-147
31. Daniela Kotur, Željko Trpovski, OSNOVI CLOUD TEHNOLOGIJE SA PRIMERIMA PRIMENE	148-151
32. Стефан Петковић, ПРОШИРЕЊЕ MONGODB БАЗЕ ПОДАТАКА ОТВОРЕНОГ КОДА ЗА ИНДЕКСИРАЊЕ И ПРЕТРАЖИВАЊЕ СРПСКИХ ТЕКСТОВА	152-155
33. Nina Maljković, Nikša Jakovljević, DETEKCIJA STRESA KOD PILIĆA TOVLJENIKA NA OSNOVU ANALIZE ZVUKA	156-159
34. Smiljana Tedić, RAZVOJ I ANALIZA ANDROID APLIKACIJE PET SHOP	160-163
35. Saša Maksimović, REGULACIJA TEMPERATURE KGH SISTEMA UPOTREBOM PI-LIKE FUZZY REGULATORA I GA	164-167
36. Ivan Jančić, Srđan Popov, SMART CITY – NOVI SAD MODEL PRAĆENJA JAVNOG SAOBRAĆAJA	168-172
37. Ђорђе Марјановић, PRIMENA DUBOKOG UČEŃA USLOVLJAVANJEM NA SAMOVOZEŃI AUTOMOBIL U SIMULIRANOM OKRUŽENJU TORCS	173-176
38. Miloš Arbanas, SISTEM ZA LOKALIZACIJU OBJEKATA U ROBOTICI	177-180
39. Kristina Nikolić, PROJEKTOVANJE AKTIVNOG NISKOPROPUSNOG FILTRA U 0,18 μm CMOS TEHNOLOGIJI	181-184
40. Vlada Đurđević, AUTOMATSKO ODREĐIVANJE TEMA KNJIGA POMOĆU TEHNIKA ZA PROCESIRANJE PRIRODNOG JEZIKA	185-188
41. Nina Miladinović, GENERISANJE FORMI ZA POSLOVNE PROCESSE KORIŠĆENJEM MAGRITTE RAZVOJNOG OKVIRA	189-192
42. Aleksandra Mišić, STEGOSPLOIT U SISTEMIMA ZA PRENOS I PRIKAZ DIGITALNIH SLIKA	193-196
43. Rade Radišić, Srđan Popov, IMPLEMENTACIJA VIZUALIZATORA STRUKTURA PODATAKA KORIŠĆENJEM WEBASSEMBLY PLATFORME	197-200

ANALIZA TAČNOSTI KOORDINATNIH MERNIH SISTEMA PRI ODREĐIVANJU GREŠKE RAVNOSTI**ACCURACY ANALYSIS OF COORDINATE MEASURING SYSTEMS TO DETERMINE THE FLATNESS ERROR**

Milan Lončar, Branko Štrbac, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Razvojem proizvodnje ka složenim radnim predmetima sa uskim tolerancijama izrade dovelo je koordinatne merne sisteme (KMS) na leadersku poziciju u procesima verifikacije. Postoji niz različitih KMS koje se danas koriste a karakteriše ih način uzorkovanja, tačnost, brzina, fleksibilnost. Mnoge KMS nemaju utvrđene procedure za proveru tačnosti. Ova rad je imao za cilj da utvrdi tačnost različit KMS na primeru merenja etalona ravnosti sa poznatim kalibracionim vrednostima. Rezultati su pokazali visoke performanse koordinatnih mernih mašina za obe vrste senzora a loše performanse za industrijski CT.

Ključne reči: Koordinatni merni sistemi, greška ravnosti, rotacija kroz jednu tačku

Abstract – With the development of manufacturing towards complex workpieces with narrow tolerances, it has brought coordinate measurement systems (CMS) to a leading position in the verification processes. There are a number of different CMS used today that are characterized by sampling mode, accuracy, speed, flexibility. Many CMS do not have established procedures for verifying accuracy. This paper aimed to determine the accuracy of different CMS on the example of measuring the standards of flatness with known calibration values. The results showed high performance coordinate measuring machines for both probe and poor performance for industrial CT.

Keywords: *Coordinaty measurement systems, flatness error, Bundle of plains through one point*

1. UVOD

Sa početnim razvojem koordinatnih mernih sistema (KMS) 1970-tih, trodimenzionalna (3D) metrologija je postala osnova u industriji. Koordinatni merni sistemi su postigli široku popularnost kao fleksibilno rešenje u bezbroj mernih potreba. Sa porastom korišćenja geometrijskog dimenzionisanja i tolerisanja, KMS su dominantne mašine dimenzionalnih metrologa. KMS, sa pogodnom opremom, se mogu koristiti za merenje praktično bilo kog parametra koji je označen na tehničkoj dokumentaciji, pošto u osnovi uočavaju jedino koordinate tačaka koje se nalaze na površini radnog predmeta [1].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Branko Štrbac.

Tačnost u detektovanju položaja mernih tačaka zavisi pre svega od tipa koordinatnog mernog sistema (hadverska struktura), sistema uzorkovanja (merni senzor), načina uzorkovanja tačaka, itd. Koordinatni merni sistemi se mogu podeliti u dve grupe: sistemi koji uzorkuju tačke na diskretan način i sistemi koji prikupljaju tačke u vidu skeniranja. Preciznost i tačnost ovih različitih mernih sistema se oslikava u ispravnosti detektovanja koordinata tačaka. Nakon toga, u nezavisnim softverskim analizama, na bazi izmerenih tačaka, procenjuju se karakteristike kvaliteta proizvoda. Industrija zajedno sa metrološkim institutima razvila procedure za testiranje koje omogućavaju proizvođačima ovih metroloških sistema kao i krajnjim korisnicima da procene performanse različitih tipova KMS. Specifične procedure prihvatanja i reverifikacije su uspostavljene u mnogim nacionalnim i međunarodnim smernicama. Međutim, uspostavljanje sledljivosti na različitim KMS moguće je ostvariti samo korišćenjem dobro poznatih metoda supstitucije zasnovane na kalibrisanim radnim predmetima i korišćenjem naprednih virtualnih mernih mašina [2].

U ovom radu analizirana je greška ravnosti merena na različitim koordinatnim mernim sistemima a procena greške je izvršena u istom programskom sistemu. Sledljivost je uspostavljena korišćenjem etalona ravnosti – plan paralelne merke.

2. PREDMET ISTRAŽIVANJA

Kako je već rečeno, princip rada većine koordinatnih mernih sistema se odvija u dve nezavisne faze: fizičko uzorkovanje podataka tačaka sa realne geometrije radnog predmeta i nezavisna softvera analiza uzorkovanih podataka [3]. Druga faza je kod svih KMS identična tj. koriste se kompatibilni softveri za određeni KMS ili se koriste univerzali softveri ako su izvedeni podaci o realnoj geometriji radnog predmeta čitljivi. U prvoj fazi su prisutne razlike između određenih KMS. Naime, tačnost određenog mernog sistema se ispoljava u tačnosti uzorkovanja koordinate uzorkovane tačke. Na tačnost uzorkovane tačke utiče mnogo faktora, pre svega tu je 21 geometrijska greška ako se radi o konvencionalnim KMM i greške senzora. Za ocenu tačnosti uzorkovane tačke na KMM vrši se merenje male precizne sfere u 49 tačaka i (najmanji kvadrati) najbolje fitovana sfera se izračunava [4]. Radijalni reziduali svake merne tačke od best fit radijusa su određeni, i njihova standardna devijacija se koristi za pojedinačni parametar da okarakterise nesigurnost tačke KMM. Iako je ova procedura

sprovedena na maloj sferi, može se zamisliti da se njegov prečnik smanjuje do nule, i rezultat 49 merenja iste (fizičke) tačke, sa standardnom devijacijom ovih 49 vrednosti dovodi do karakteristika nesigurnosti tačke. Prednost ove metode jeste da mnogi KMM korisnici znaju vrednost za tačka po tačka testa performanse senzora i stoga i lako mogu izvršiti test za procenu ove vrednosti. Ovo pokazuje da svaka izmerena tačka ima verovatnosti opseg koji se naziva merna nesigurnost uzorkovane tačke i na ovaj način se samo definiše netačnost senzora. Međutim, uzorkovana tačka u drugom položaju u KMM zampemini će imati drugačiji verovatnosti opseg pojavljivanja jer geometrijske greške nisu jednake duž mernog prostora mašine.

Stoga, ne može se uvesti pojednostavljenje da je greška tačke uzorkovanja ravnomerna u raspoloživom mernom prostoru KMM. Isto tako, pri oceni supstitutivne geometrije, potrebno je analizirati određeni skup tačaka koja svaka ima različit stepen tačnosti. Ovde je jako bitno i koja matematička metoda za procenu supstitutivne geometrije se koristi. Posebno je ovo se interesantno ispitati ako se procenjuje neka geometrijska karakteristika kao što je na primer greške oblika određenog geometrijskog primitiva.

Isto tumačenje se može izvesti za ostale KMS stim da postoji mnogo više faktora koji će uticati na tačnost uzorkovanja koordinate tačke sa radnog predmeta.

Ovaj rad ima za cilj da analizira tačnost različitih KMS tako što se meri etalon ravnosti (plan paralelna granična merka) poznatih kalibracionih vrednosti. Izlaz iz svakog analiziranog KMS je skup koordinata tačaka (x, y, z) koji se analiziraju u nezavisnoj softverskoj analizi. Procenjuje se greška ravnosti u nekomercionalnom softveru koji se bazira na metodi - Rotacija kroz jednu tačku [5].

Koordinate uzorkovanih tačaka će sadržati sve nesavršenosti tog mernog sistema i procenjena greška merenja će se porediti sa kalibrisanom vrednošću greške ravnosti i na taj način će se sprovesti uporedna analiza tačnosti različitih koordinatnih mernih sistema za merni zadatak greške ravnosti

2.1. Predmet merenja

U cilju procene tačnosti različitih koordinatnih mernih sistema merena je plan paralelna merka koja se može smatrati etalom ravnosti. Plan-paralelne merke izrađuju se od optičkog stakla velike tačnosti. U ovoj studiji slučaja mereno je plan paralelna granična merka prikazano na slici 1 čije su karakteristike date u tabeli 1. Kalibracione vrednosti su dobijene koristeći laserski interferometar.

Tabela 1. Podaci o predmetu merenja

Proizvođač	Mitutoyo
Debljina	$t=12 \text{ mm}$
Prečnik	$d=30 \text{ mm}$
Greška ravnosti iz kalibracionog sertifikata	$\delta=0,03 \times 10^{-3} \text{ mm}$
Nesigurnost kalibracije	$U=20 \times 10^{-6} \text{ mm}$



Slika 1. Predmet merenja

2.3. Ocena greške ravnosti

Geometrijski element ravan je jedan od osnovnih geometrijskih primitiva koji je osnova mnogih mehaničkih proizvoda za postizanje namenjene funkcionalnosti. Ravnost površine je stanje u kom svi elementi površine pripadaju jednoj ravni što odgovara idealnoj ravni. Procena greške ravnosti se obično vrši uz pomoć računara zasnovano na algoritmima koji bi trebalo da budu robusni, efikasni i na bazi proverenih matematičkih principa. Takođe, ovi programi bi trebalo da slede smernice postavljene u standardima. ANSI standard dimenzionisanja i tolerisanja Y14.5 definiše, procenjuje tolerancije oblika na komponentama u odnosu na idealne geometrijske oblika. Za evoluciju oblika, ISO standardi navode da idealni geometrijski elementi moraju biti uspostavljeni iz stvarnog merenja, tako da odstupanje između idealnih i stvarnih geometrijskih elemenata je najmanja moguća vrednost. Podaci dobijeni pomoću KMM su analizirani koristeći tehnike zasnovane na fitovanju. Dve najčešće tehnike fitovanja su metoda najmanjih kvadrata (LS) i metod minimalne zone (MZ) koji spada u ekstremne algoritme fitovanja. LS je superiorniji u odnosu na MZ sa stanovišta jednostavnosti i vremena računanja. LS generalno procenjuje grešku oblika i samim tim može dovesti do odbacivanja dobrih delova gde MZ teži da loše proceni grešku oblika i veoma je osetljiv na pikove koji mogu, koje ako se ne prime, dovesti do loših rezultata. LS ne prati pažljivo standarde i ne može da garantuje da je rešenje minimalne zone specifikovano u standardu. U ovom master radu za procenu greške ravnosti korišćena je metoda rotacija kroz jednu tačku bazirana na MZ metodologiji i razvijena je na Fakultetu tehničkih nauka [5].

Metoda rotacija kroz jednu tačku se bazira na osobinama referentne ravni koja predstavlja osnovu za procenu greške ravnosti. Iz analitičke geometrije je poznato da su tri koordinate normalnog vektora ravni jedini parametri koji utiču na vrednost greške ravnosti. Pritom položaj stvarne ravni nije značajan iz razloga što njegove ortogonalne transformacije ne utiču na ravnost. To znači da u jednačini referentne ravni date preko formule (1)

$$z = -\frac{Ax + By + D}{Cz} \quad (1)$$

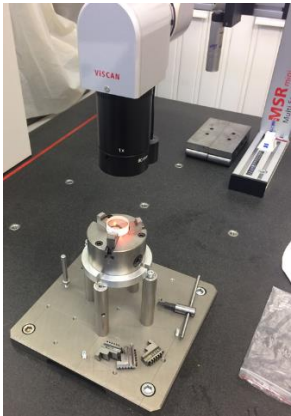
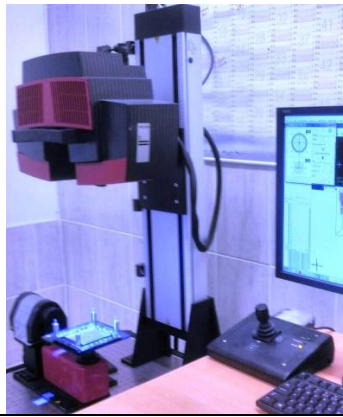
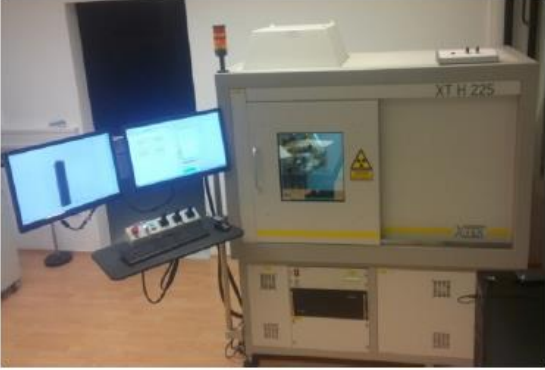
A, B i C predstavljaju koordinate vektora normale ravni a D je konstanta koja je definisana sa položajem ravni. Parametar D je beznačajan i nema uticaja na procenu

greške ravnosti. Ovo navodi na zaključaj da ako se uzme bilo koja ravan sa specifičnim vektorom normale definisanim preko parametra A, B i C rezultat procene greške ravnosti će biti isti. Drugim rečima, položaj ravni određen preko rotacije vektora normale i činjenice da tačka A pripada ravni, definiše vrednost parametra D koji se koristi za izračunavanje ortogonalnih rastojana od generisanih ravni i procenu greške ravnosti. Referentna ravan može biti smeštena u bilo koji tačku koja pripada setu uzorkovanih tačaka na koordinatnom mernom sistemu. Položaj ravni u prostoru određen uglovima ξ i θ i parametri referentne jednačine A, B i C se određuju preko iterativnog postupka. Naime, ravan rotira oko slučajno izabranih tačaka u prostoru i minimalno rastojanje između uzorkovanih tačaka se određuje prema definiciji tolerancijske zone ravnosti. Koraci rotacije oko x i z ose su definisani kao odnos maksimalnih uglova ξ and θ i odgovarajućih brojeva definisanih koraka. Broj koraka je potrebno definisati pre softverskog proračuna. Svaki put kada referentna ravan zauzme određeni položaj u prostoru definisan uglovima ξ and θ softver će registrovati parametre vektora normale referentne ravni i grešku ravnosti. Postupak će se ponavljati za sve definisane

korake i softver će registrovati onaj vector normale referentne ravni čiji položaj određuje najmanju grešku ravnosti za posmatrani skup uzorkovanih tačaka.

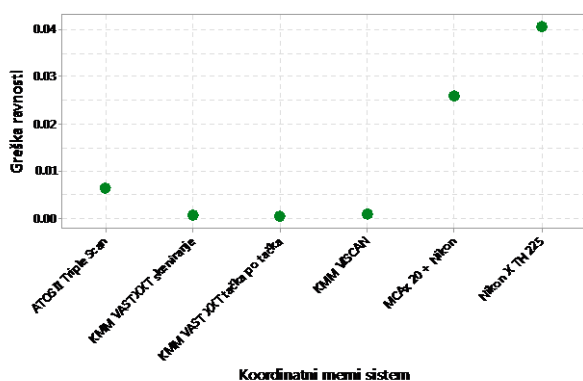
3. EKSPERIMENTALNA ISTRAŽIVANJA

Plan – paralelna merka je merena na nekoliko različitih koordinatnih mernih sistema i različitih senzora. Operateri su uzimali slične strategije merenje na taj način uzorkovane tačke obuhvata celu ispitivanu površinu (kod nekih mernih instrumenata nije bilo izvodljivo primeniti sličnu strategiju merenja). Nakon uzorkovanja, podaci uzorkovanih tačaka (x,y,z koordinate) u izveze u programski sistem koji je izgrađen na bazi MZ rešenja – rotacija kroz jednu tačku. Na ovaj način su isključene eventualne različitosti između softvera koji podržavaju rad određenog KMS i u rezultatu merenja su sadržane netačnosti koji potiču od hadvera određenog KMS. Na slici 3 su prikazani KMS i merni senzori, kao i njihove karakteristike, koji su se koristili u ovom istraživanju. Više detalja o analizi greške ravnosti, različitim mernim instrumentima, proceni njihove tačnosti kao i vrednosti koordinata uzorkovanih tačaka može se pronaći u master radu iz kog je proistekao ovaj rad.

		
<i>Koordinatna merna ruka MCAx 20 + Nikon</i> Tačnost: ± 0.033	<i>Carl Zeiss Contura g2 RDS</i> Mod tačka po tačka: $MPE_p = 1.8 \mu m$ Mod skeniranje: $MPE_{THP} = 3.5 \mu m$ Beskontaktno merenje: $U = 0.4 \mu m$	
		
<i>ATOS II Triple Scan</i> Tačnost: 0.003	<i>Industrijski CT – Nikon XT H 225</i> Parametar tačnosti: -	

Slika 1. Koordinatni merni sistemi korišćeni u studiji

Na slici 4. data je uporedna analiza grešaka ravnosti procenjenih na različitim koordinatnim mernim sistemima. Sa slike se mogu uočiti visoke performanse KMM pri merenju i sa kontaktnim i beskontaktnim senzorom što pokazuje da na grešku merenja više utiču geometrijske greške mašine od samih grešaka senzora. Ovo što je potvrdio ovaj eksperiment je to da u modu tačka po tačka merni sistem ima bolje performanse u rado od moda skeniranje. Skener ATOS je pokazao veoma dobre performanse i njegovi rezultati su neznatno manji od merenja na KMM. Veoma loše rezultate su pokazali merna ruka i industrijski CT. Merna ruka koja je korišćena u ovom radu spada nema visoke metrološke performanse dok se od industrijskog CT-a, čije merenje ima najveću grešku, očekivali mnogo bolji rezultati. Međutim, treba još jednom pomenuti da je predmet merenja bio proziran što predstavlja velik problem da beskontakne metode merenja među kojima je i CT.



Slika 4. Analiza tačnosti različitih KMM pri merenju etalona ravnosti

4. ZAKLJUČAK

Provera usaglašenosti dimenzionalnih i geometrijskih tolerancija sa specifikacijskim zahtevima skoro da je nezamisliva bez upotrebe nekog tipa koordinatnog mernog sistema. To su merni instrumenti koji prikupljaju podatke o realnog geometriji preko određenog tipa mernog senzora. Danas postoji mnoštvo različitih tipova KMS i mernih senzora. Opšta podela je na kontaktne i beskontakne merne sisteme, gde određeni sistem, uz odgovarajuće senzore, može biti i jedno i drugu. Uglavnom su kontaktni sistemi tačniji ali im treba mnogo više vremena da uzorkuju (digitalizuju) predmet merenja dok beskontaktni mogu sa veoma kratak vremenski period da prikume i do nekog hiljada tačaka.

Ovaj rad se bavio tačnosti nekoliko različitih KMS pri merenju etalona ravnosti – plan paralelna merka. Etalon je meren sličnom strategijom, koliko je to bilo moguće, a greška merenja je procenjena na isto nekomercionalnom softveru. Eksperimentalna istraživanja su pokazala visok stepen tačnosti KMM pri korišćenju oba moda kontaktnog senzora kao i beskontaktnog senzora. Takođe, utvrčena je velika netačnost najsavremenijeg KMS – industrijskog CT-a.

Imajući u vidu da je radni predmet proziran ova činjenica je dosta nepogodna za upotrebu bilo koje vrste beskontaktnog senzora. Iz tog razloga, istraživanja bi trebalo proširiti i na druge referentne predmete kao što su etalon prstenovi, granične merke, referenti konusi, ball bar, ploča sa otvorima, itd.

5. LITERATURA

- [1] B. Štrbac, V. Radlovački, V. Spasić-Jokić, M. Delić, M. Hadžistević, „The difference between GUM and ISO/TC 15530-3 method to evaluate the measurement uncertainty of flatness by a CMM“ *MAPAN*, Vol. 32, pp. 251–257, 2017.
- [2] B. Acko, M. McCarthy, F. Haertig, B. Buchmeister, „Standards for testing freeform measurement capability of optical and tactile coordinate measuring machines“, *Measurement Science and Technology*, , Vol 23, 2012.
- [3] X-L. Wen, X-C. Zhu, B-Y. Zhao, D-X. Wang, F-L Wang, Flatness error evolution and verification based on new generation geometrical product specification (GPS), Vol. 36, pp. 70-76, 2012.
- [4] S. D., Phillips, B. Borchardt, W.T. Estler, and J. Buttress, The estimation of measurement uncertainty of small circular features measured by coordinate measuring machine, *Precision Engineering*, Vol. 22(3), pp. 87-97, 1998.
- [5] V. Radlovački, M. Hadžistević, B. Štrbac, M. Delić, M, B. Kamberović, „Evaluating minimum zone flatness error using new method - bundle of plains through one point“, *Precision Engineering*, Vol. 43, pp. 554-562, 2016.

Kratka biografija:



Milan Lončar rođen je u Novom Sadu 1974. god. Master rad na Fakultetu tehničkih nauka iz oblasti Proizvodno mašinstvo odbranio je 2019.god.
kontakt: milan.loncar@unimet.rs



Dr Branko Štrbac rođen je u Novom Sadu 1983. Doktorirao je na Fakultetu tehničkih nauka 2017. god. i od tada je u zvanju docent. Oblast interesovanja su proizvodna metrologija.

POSTROJENJE SUŠARE ZA ŠLJIVE NA BIOMASU**PLUM DRYER PLANT FUELED BY BIOMASS**

Radmila Miljanović, Damir Đaković, Fakultet tehničkih nauka, Novi Sad

Oblast – MAŠINSTVO

Kratak sadržaj – U okviru rada prikazan je primer proračuna i konstrukcije jedne šaržne sušare sa lesama za sušenje šljiva. Cilj rada je da se napravi sušara na biomasu sa upotrebom cirkulacione pumpe i toplovodnog kotla. Na osnovu definisanog sistema, imamo odgovarajuću sušaru.

Ključne reči: Sušara, šljive, cirkulaciona pumpa, toplovodni kotao

Abstract – The paper presents an example of the calculation and construction of a batch dryer for plums. The aim of this work is to make a biomass dryer using a circulation pump and a hot water boiler. Based on the defined system, we have a suitable dryer.

Keywords: Dryer, plums, circulation pump, hot water boiler

1. UVOD [1]

Sušenje je veoma star proces kao i sve ostale osnovne delatnosti vezane za opstanak ljudskog roda. Sušenjem se smanjuje vlažnost poljoprivrednih proizvoda, sa ciljem da se što manje utiče na njihov kvalitet. Sušenje može biti prirodno i veštačko. Prirodno se obavlja pod dejstvom energije sunca, a veštačko u specijalno projektovanim uređajima-sušarama. Pored navedenih sušenja imamo i sušenje u rerni, pod nastrešicom, na tavanu.

Sušenje šljiva kod nas je dosta zastupljeno. Postoji oko 2000 sorti šljiva. Generalno se svrstavaju u šest kategorija: japanske, američke, „damson“, orijentalne, „divlje“ i evropske-baštenske. Kod nas su zastupljene najviše evropske-baštenske šljive. U našoj zemlji su uglavnom zastupljene sledeće sorte šljiva:

Stenlej
Požegača
Aženka
Čačanska rana

Plodovi šljiva sazrevaju, u zavisnosti od sorte i podloge, podnevlja i vremenskih prilika, osobine zemljišta, nadmorske visine, položaja i načina nege u zasadu, od sredine juna do polovine septembra. Plodovi koje sušimo moraju biti zreli i zdravi, odgovarajuće veličine bez mehaničkih oštećenja. Branje šljiva se obavlja u nekoliko segmenata, pošto svi plodovi na stablu ne sazrevaju u isto vreme. Sve faze sušenja, dorade i pakovanja suvih šljiva imaju važan uticaj na organoleptička svojstva finalnog proizvoda, ali odgovarajuću ulogu ima kvalitet sirovine. U tabeli 1. prikazane su fizičke osobine plodova nekih šljiva.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Damir Đaković, vanr.prof.

Tabela 1. Fizičke osobine plodova nekih šljiva za sušenje [4]

Sorta	Plod			Koštica		
	Visina	Širina	Debljina	Visina	Širina	Debljina
	mm	mm	mm	mm	mm	mm
Stenlej	48,2	35,1	33,9	29,3	15	9
Čač.rodna	41,2	31,7	28,6	23	13,2	7,3
Aženka	39,9	30,2	30,4	23,1	13,8	7,4

2. SUŠENJE ŠLJIVA-STENLEJ

Sušenje šljiva je sezonski posao i neophodno je da nema zastoja tokom sezone berbe. Da bi sušenje teklo bez prekida treba izvršiti sve potrebne pripreme. Šljiva stenlej ima dobar kvalitet ploda i zadovoljavajuću krupnoću, ali plodovi moraju da budu ujednačeni, da se sušenjem ne bi dobio veliki procenar sitnih plodova. Berba mora biti obavljena blagovremeno, gde se selektivno biraju plodovi odgovarajuće zrelosti.

Za sušenje šljive se beru u punoj tehnološkoj zrelosti bez petiljki. Prilikom transporta šljiva vodi se računa da nema oštećenja plodova i da sve bude čisto i potpuno sigurno. Posle branja neophodno je pristupiti daljem postupku obrade. Do samog procesa sušenja šljive se skladište u hladnim i suvim skladištima.

2.1. Prerada šljive [2]

Sušenje šljive obuhvata dve faze: sušenje i završnu obradu. Prva faza obuhvata sledeće operacije: inspekciju-pranje- inspekciju-klasiranje- stavljanje na lese- redanje lesa na kolica- sušenje.

Inspekcija. Vršiti se analiza kvaliteta plodova pre početka istovara. Nakon obavljenih kontrola i merenja prispelih šljiva za sušenje vrši se istovar.

Pranje. Pranjem se otklanjaju mehaničke nečistoći i izvesna količina prisutnih mikroorganizama. Posle pranja vrši se ponovo inspekcija, odnosno kontrola već čistih proizvoda koji idu na dalju obradu.

Klasiranje. Proces u kom se vrši razdvajanje plodova po veličini sa ciljem da se slažu na lese plodovi približno iste veličine.

Stavljanje na lese. Stavljanje šljiva na lese može se obavljati ručno ili automatski prko specijalno konstruisanih uređaja.

Redanje lesa na kolica. Lese se postavljaju na kolica sa pokretnim točkovima, na koje se razastiru plodovi.

Sušenje. Sušenje je proces kojim se iz šljiva uklanja veća količina vlage i time se omogućava njihovo dugotrajno čuvanje.

Hlađenje. Pre unošenja u skladište šljiva mora da se ohladi. Koriste se čiste, promajne prostorije, specijalno pripremljene za skaldštenje suvih šljiva.

Klasiranje. Po obavljenom kondicioniranju (skladištenje u cilju izjednačavanja vlage), šljiva se ponovo klasira u procesu završene obrada, kako bi proizvod bio što ujednacenijeg kvaliteta.

Etiviranje. Prilikom etiviranja plodovi se kratko potapaju u vrelu vodu a onda vade i pakuju. Upakovana šljiva se stavlja u komore, koje se zagrevaju indirektnim putem preko ugrađenih kalorifera.

Postupak završne obrade obuhvata: inspekcija- klasiranje- pranje- pasterizaciju- odstranjivanje površinski zadržane vode- dodavanje konzervansa- pakovanje. Završna obrada može se obaviti na dva načina: suvim načinom ili sterilizacijom i vlažnim postupkom.

Pasterizacija. Oprana šljiva pada u pasterizator u kom je ugrađen kofičasti transporter, pomoću kojeg se proizvod kreće kroz vodu. Voda se zagreva parom.

Odstranjivanje površinski zadržane vlage. Posle pasterizacije na vibrirajućem uređaju ostranjuje se površinski zadržana vlaga. Zati ide još jedna inspekcija.

Dodavanje konzervansa. Pred samo punjenje u ambalažu dodaje se konzervans, prskanjem rastvorom određene koncentracije preko atomizera.

Pakovanje. Za pakovanje suvih šljiva se uglavnom se koriste plastične kese, kao i kartonske kutije.

2.2. Režim sušenja šljiva

Pod pojmom režim sušenja šljiva podrazumeva se intenzitet osnovnih parametara agensa sušenja, koja se ostvaruju tokom trajanja samog procesa. Te osnovne parametar čine: temperature, vlažnost i brzina stujanja.

Sadržaj vlage u odnosu na vreme sušenja sa toplim vazduhom sušenja normalnih i tretiranih uzoraka šljiva kako pod uticajem različitih temperature vazduha, navešćemo preko primera. Početni sadržaj vlage je oko 78,1% i on se smanjuje sa vremenom sušenja. Vreme sušenja potrebno da se redukuje vlaga iz početnog sadržaja na željenu vrednost kod normalnih uzoraka je 37,29 i 23h na temperature vazduha od 50,60 i 70 °C, dok je kod tretiranih uzoraka vreme sušenja 27, 22 i 18 h na temperaturi vazduha od 50, 60 i 70 °C pri jednakoj brzini vazduha od 0,8 m/s [3].

Vlažnost zagrejanog vazduha, kao agensa sušenja, može se posmatrati kroz apsolutnu i relativnu vlažnost.

Apsolutna vlažnost vazduha se ne menja jer je određena meteorološkim uslovima spoljnog vazduha, dok je relativna vlažnost parametar koji možemo menjati uz zagrevanje na različite vrednosti temperature.

3. PODELA SUŠARA I NAČINI PROVOĐENJA TOPLOTE

Tradicionalno direktno sušenje je stari način sušenja šljiva, direktno na suncu. Ima prednosti u uštedi energije, ali ima i dosta nedostataka jer tako nije zaštićeno od prašine i ptica. Pasivni solarni sistem sušenja je sličan staklenicima, orijentisane staklene površine okrenute prema jugu. Ulaganje je veliko i neophodna je veća površina da bi se to napravilo. Industrijski način sušenja

kod nas je počeo da se primenjuje prvo izgradnjom tunelske sušare, a kasnije se to sve više širilo. Sušare predstavljaju objekte u kojima se pod određenim kontrolisanim uslovima odvodi voda-vlaga iz šljiva. Postoji veliki broj tipova sušara, a za sušenje šljiva najviše se primenjuju komorne i tunelske sušare, a zajednička osobina im je da rade pri atmosferskom pritisku. Osnovni pokazatelji dobrog tehničkog rešenja sušare za šljive: jednostavna izrada, lako opsluživanje, minimalno anganžovanje ljudskog roda, pouzdranost u radu, itd. U zavisnosti od načina dovodenja toplote sušenje može biti:

Konvektivno sušenje sa atmosferskim pritiskom radnog fluida

Konvektivno sušenje u vakuumu

Konduktivno sušenje

Sušenja zračenjem

Mikrotalasno sušenje

Osmotsko sušenje

Sušenje zamrzavanjem ili liofalizacija

4. PRORAČUN SUŠARE I SKLOPNI CRTEŽ

4.1. Proračun sušare

U pitanju je šaržna sušara na čvrsto gorivo. Ulazni podaci u šaržu su dati u tabeli 2.

Tabela 2. Ulazni podaci u šaržu

Veličina	Oznaka	Jedinica	Komora	Ukupno
Sirovina	G	Kg	1500	3000
Voda	G_{vp}	Kg	1200	2400
Suva materija	G_{sm}	Kg	300	600
Masa lese	G_l	Kg	25	
Broj lesa			60	120

Poznati podaci su: ulazna temperatura $t_0=20$ °C, temperatura na izlazu iz grejača $t_1=75$ °C, temperatura na izlazu iz sušare $t_2=40$ °C. Relativna vlažnost vazduha na ulazu u sušaru $\phi_0=80$ %, relativna vlažnost vazduha na izlazu iz grejača $\phi_1=48$ %, relativna vlažnost vazduha na izlazu iz sušare $\phi_2=64$ %. Apsolutna vlažnost na ulazu u sušaru $x_0=8,8$ g/kg, apsolutna vlažnost na izlazu iz grejača $x_1=8,8$ g/kg, apsolutna vlažnost vazduha na izlazu iz sušare $x_2=22,7$ g/kg.

Entalpija na ulazu u sušaru $h_0=90,8$ kJ/kg, entalpija na izlazu iz grejača $h_1=146,7$ kJ/kg, entalpija na izlazu iz sušare $h_2=223,4$ kJ/kg.

G_1 - količina vode u 1 kg presnih šljiva

G_{sm} - količina suve materije u 1 kg

G_1 - količina vode u 1 kg suve šljive

W - količina vode odvedena sušenjem.

$$G_1 + G_{sm} = G_2 + G_{sm} + W$$

$$W = G_1 - G_2$$

$$W = 0,750 \frac{\text{kg}}{\text{kg}} \times 1500 \text{ kg} = 1125 \text{ kg}$$

$$W = 0,750 \frac{\text{kg}}{\text{kg}} \times 3000 \text{ kg} = 2250 \text{ kg}$$

Vreme sušenja je 24h.

Sadržaj vlage:

$$\Delta x = x_2 - x_0 = 22,7 \frac{\text{gr}}{\text{kg}} - 8,8 \frac{\text{gr}}{\text{kg}} = 13,9 \text{g / kg}$$

$$m_v = \frac{1125 \text{kg}}{24 \text{h}} = 46,875 \text{kg / h}$$

$$m_v = \frac{2250 \text{kg}}{24 \text{h}} = 93,75 \text{kg / h}$$

Količina vazduha:

$$L = \frac{W}{\Delta x} = \frac{1125 \frac{\text{kg}}{\text{h}}}{0,0139 \frac{\text{kg}}{\text{kg}}} = 80,935 \text{kg / šarži}$$

$$L = \frac{W}{\Delta x} = \frac{2250 \frac{\text{kg}}{\text{h}}}{0,0139 \frac{\text{kg}}{\text{kg}}} = 161,87 \text{kg / šarži}$$

Maseni protok:

$$m_{sv} = \frac{m_v}{\Delta x} = \frac{46,875 \frac{\text{kg}}{\text{h}}}{22,7 \frac{\text{gr}}{\text{kg}} - 8,8 \frac{\text{gr}}{\text{kg}}} = \frac{46,875 \text{kg}}{13,9 \frac{\text{gr}}{\text{kg}}} = \frac{46,875 \text{kg}}{0,0139 \frac{\text{kg}}{\text{kg}}} = 3372 \text{kg / h}$$

$$m_{sv} = \frac{m_v}{\Delta x} = \frac{93,75 \frac{\text{kg}}{\text{h}}}{22,7 \frac{\text{gr}}{\text{kg}} - 8,8 \frac{\text{gr}}{\text{kg}}} = \frac{93,75 \text{kg}}{13,9 \frac{\text{gr}}{\text{kg}}} = \frac{93,75 \text{kg}}{0,0139 \frac{\text{kg}}{\text{kg}}} = 6744 \text{kg / h}$$

$$m_{sv} = \frac{6744 \frac{\text{kg}}{\text{h}}}{3600 \text{s}} = 1,87 \text{kg / s}$$

Zapreminski protok:

$$\dot{V} = \frac{m_{sv}}{\rho} = \frac{3372 \frac{\text{kg}}{\text{h}}}{1,2 \frac{\text{kg}}{\text{m}^3}} = 2810 \text{m}^3 / \text{h}$$

$$\dot{V} = \frac{m_{sv}}{\rho} = \frac{6744 \frac{\text{kg}}{\text{h}}}{1,2 \frac{\text{kg}}{\text{m}^3}} = 5620 \text{m}^3 / \text{h}$$

Potrebna snaga grejača:

$$Q = m_{sv} \times c_p \times \Delta t = 1,87 \times 1 \times 55 = 102,85 \text{kW}$$

Koeficijent prolaza toplote:

$$k = \frac{1}{\frac{1}{\alpha_{up}} + \sum \frac{\delta}{\lambda} + \frac{1}{\alpha_{sp}}} = 0,322 \text{W / (m}^2 \text{K)}$$

$$\dot{Q}_{gub} = A \times k \times \Delta t = 22 \times 0,322 \times 37,5 = 265,65 \text{W}$$

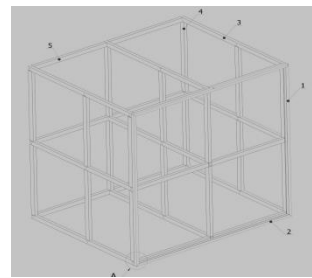
Gubici su zanemarljivo mali i sigurno postoji rezerva na grejaču koja će to pokriti.

4.2 Sklopni crtež

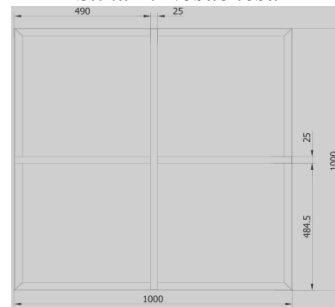
U sklopnom crtežu imamo prikazane pokedinačno delove sušare. Lese su pravljenе od materijala koji je dozvoljen u prehrambenoj proizvodnji.

U pitanju su prohromske lese. Sušara je urađena od čelične konstrukcije, obložena panelima. Svi metalni delovi u unutrašnjosti sušare su pocinkovani.

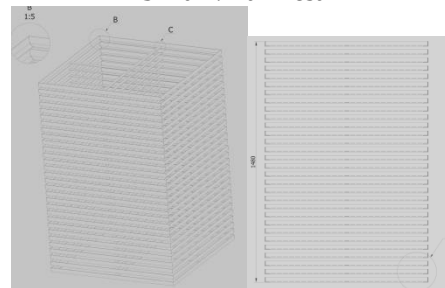
U jednu šaržu se ubacuje 60 lesa, 30 lesa u visinu, imamo dva reda u jednoj šarži. Na lesu se stavlja 25 kg šljiva. Razmak između lesa je 2 cm. Na slici 1. prikazan je nosač konstrukcije, na slici 2. ram lesa a na slici 3. šarže.



Slika 1. Nosač lesa



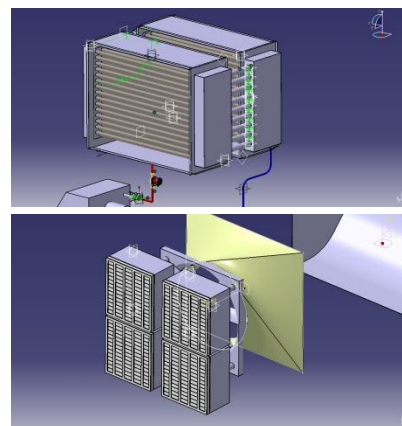
Slika 2. Ram lesa



Slika 3. Šarža

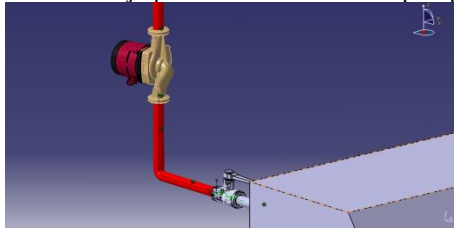
Da bi sušara radila, odnosno da bi bio obavljen proces sušenja šljiva usvojen je razmenjivač toplote, ventilator za transport vazduha, cirkulacionu pumpu i toplovodni kotao. Razmenjivač toplote je uređaj kod kog se toplota prenosi sa jednog fluida na drugi. U razmenjivačima toplote se najčešće javlja kombinacija provođenja i prelaza toplote. Razmenjivač toplote koji sam usvojila je kompaktni lamelni Cu-Al razmenjivač toplote.

Cu-Al razmenjivač toplote kao grejna i rashladna tela preko čijih površina se vrši razmena toplote izrađeni su od bakarnih bešavnih cevi na koje su navučene aluminijumske lamele. Ovaj tip razmenjivača toplote ima široku peimenu. Na slici 4. je prikazan lameli razmenjivač toplote i ventilator za transport vazduha za sušenje sa filterskom sekcijom.



Slika 4. Razmenjivač toplote i ventilator[5]

Alfa plus-cirkulaciona pumpa može biti ugrađena bilo gde. Efektivno zamenjuje više od 400 tradicionalnih modela cirkulacionih pumpi. Pumpa automatski reguliše protok i pritisak prema uslovima i time eliminiše buku u sistemu. Na slici 5. je prikazana cirkulaciona pumpa.



Slika 5. Cirkulaciona pumpa toplovodnog grejača[5]

Usvojeni kotao je toplovodni kotao TKK3 do max 120kW, zapremina vode u kotlu je 230L. Namenjen je za ručno loženje mrkog uglja, uz mogućnost korišćenja drveta, kriketa i biomase kao ogrev. Deklarisana toplotna snaga važi za ogreve čija je specifična vrednost sagorevanja od 15000 kJ/kg.

Kotao je čelične varene konstrukcije. Step en iskorišćenja kotla je iznad 76 %. Potrošnja goriva pri max opterećenju od 120 kW:

$$B = \frac{Q}{Q_r \times \eta_k} \times 100 = \frac{120}{15000 \times 76} \times 100 = 0,01053 \text{ kg/s}$$

Potrošnja goriva pri opterećenju od 102,85 kW:

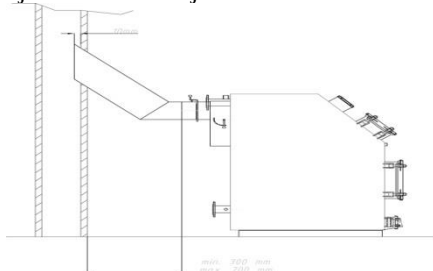
$$B = \frac{Q}{Q_r \times \eta_k} \times 100 = \frac{102,85}{15000 \times 76} \times 100 = 0,009021 \text{ kg/s}$$

Na slici 6. prikazan je toplovodni kotao na čvrsto gorivo.



Slika 6. Toplovodni kotao na čvrsto gorivo[6]

Prilikom povezivanja kotla na dimnjak luk treba da ima blagi uspon od kotla ka dimnjaku. Ako nije moguće povezivanje sa blagim usponom i jednim lukom, može se izvršiti povezivanje sa dva luka. Dimni kanal je neophodno izolovati. Na slici 7. imamo prikazano povezivanje kotla na dimnjak.



Slika 7. Povezivanje kotla na dimnjak[7]

5. ZAKLJUČAK

Tema ovog rada bila je proračun i konstrukcija sušare za sušenje šljiva uz korišćenje kotla koji koristi usitnjenu biomasu. U radu je obuhvaćen i izbor materijala od kog je napravljena sušara. Osnova je da imamo ulaznu temperaturu vazduha, temperaturu na izlazu iz grejača i temperaturu na izlazu iz sušare. Sušara ima dvoje kolica sa lesama gde se može sušiti od 1500 kg do 3000 kg šljive. Sama konstrukcija sušare je takva da imamo što manje gubitaka toplote i što efikasnije sušenje.

Izbor ventilatora, cirkulacione pumpe, razmenjivača toplote i kotla su rađeni u skladu sa vrstom sušare i kao mali potrošači. Prilikom priprema šljiva neophodno je obaviti niz pregleda sušare. Za rad na ovoj sušari neophodno je obezbediti i radnike koji će raditi po smenama.

Nedostatak ovog tipa sušenja je što se sve radi ručno i što pre svakog sušenja mora biti obavljen detaljan pregled kako lesa tako i kotla i ostalih uređaja.

6. LITERATURA

- [1] Mirković V., "Tehnologija proizvodnje suvih šljiva bez koštica", Loznica, 2009
- [2] Babić Lj., Babić M., "Sušenje i skladištenje", Poljoprivredni fakultet, Novi Sad, 2000
- [3] Sacilik K., "Dryer kinetics of uryani plum in a convective hot-air dryer", Journal of Food Engineering 76, 2015, pp. 362-368
- [4] www.tehnologijahrane.com (psistup januar 2015.)
- [5] www.etaznogrejanje.com (pristup maj 2016.)
- [6] www.termotehnika.com (pristup maj 2016.)

Kratka biografija:



Radmila Miljanović rođena je u Ključu, BiH, 1983.god. Upisala je osnovne akademske studije na Fakultetu tehničkih nauka, oblast Mašinstvo-Energetika i procesna tehnika. Trenutno zaposlena u Gradskoj upravi. Kontakt: ljrada@hotmail.com

ISTRAŽIVANJE STRUJANJA JONSKE TEČNOSTI PRIMENOM RAČUNARSKE DINAMIKE FLUIDA

RESEARCH OF IONIC LIQUID FLOW USING COMPUTATIONAL FLUID DYNAMICS

Nikola Oluški, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Predmet rada jeste strujanje jonske tečnosti *N*-butil *N,N,N*-trimetilamonijum bis(trihlormetilsulfonil)imid, skraćeno $[N_{4111}][NTf_2]$, kroz pravu cev. Cilj rada jeste istraživanje prinudne konvekcije i hidrauličkih karakteristika pri strujanju jonske tečnosti kroz pravu cevnu deonicu. Istraživanje je urađeno primenom računarske dinamike fluida pomoću programskog paketa StarCCM+. Pri analizi termičkih i hidrauličkih karakteristika zaključeno je da predmetna jonska tečnost ne predstavlja najbolje rešenje kao medijum za prenos toplote za date uslove.

Ključne reči: Računarska dinamika fluida, jonska tečnost

Abstract – The subject of the paper is the flow of an ionic liquid *N*-butyl-*N,N,N*-trimethylammonium bis(trifluoromethylsulfonyl)imide $[N_{4111}][NTf_2]$ through a straight tube. The aim of this paper is to investigate forced convection and hydraulic characteristics in a straight pipe section. The research was performed using computational fluid dynamics, in the StarCCM+ software package. Analyzing the thermal and hydraulic characteristics, it was concluded that this ionic liquid does not represent the best solution as a medium for heat transfer for the given conditions.

Keywords: Computational fluid dynamics, ionic liquids

1. UVOD

Jonske tečnosti su mešavine suprotno naelektrisanih jona povezanih kovalentnom vezom, tačnije rastvarači koji se u potpunosti sastoje od jona. Kako je u hemiji so postala opšti izraz za jonska jedinjenja, onda se još može reći da su to soli koje se nalaze u tečnom stanju na sobnoj temperaturi.

Jonske tečnosti, kao još uvek nedovoljno istraženi fluidi, mogu imati veliki uticaj u budućnosti kada je u pitanju prenos i akumulacija toplotne energije. Poseduju takve osobine da se mogu koristiti za prenos toplote od niskih (hemijaska postrojenja, rashladni sistemi) do visokih temperatura (sakupljanje i skladištenje solarne energije).

Zbog navedenih karakteristika početkom 21. veka počinje detaljnije istraživanje ovih fluida, kao medijuma za sisteme toplotnog skladištenja i prenos toplote. 2000. g., grupa autora, ispituje termička svojstva nekoliko imidazolijumskih soli. Zapažaju da imaju niže tačke topljenja

i termički su stabilnije od litijum jona, a termička stabilnost raste sa alkilnom supstitucijom [1]. Naučnici 2008. godine otkrivaju da toplotna provodljivost jonske tečnosti ima veoma malu zavisnost od molske frakcije CO₂, kada je u pitanju jonska tečnosti na bazi imidazolijuma [2]. Grupa naučnika radila je eksperimentalno istraživanje termo-fizičkih osobina jonske tečnosti $[N_{4111}][NTf_2]$, za njenu potencijalnu primenu u centralnim solarnim kolektorima [3]. Iako je koeficijent prenosa toplote niži nego kod destilovane vode, njena toplotna stabilnost i druga svojstva čine je kandidatom za korišćenje u solarnim elektranama.

Cilj rada jeste da se istraže prinudna konvekcija i hidrauličke karakteristike jonske tečnosti kroz pravu cevnu deonicu. Istraživanje prinudne konvekcije urađuje se analizom Nuseltovog broja *Nu* i koeficijenta prelaza toplote α , dok će se istraživanje hidrauličkih karakteristika uraditi analizom pada pritiska Δp i koeficijenta trenja λ .

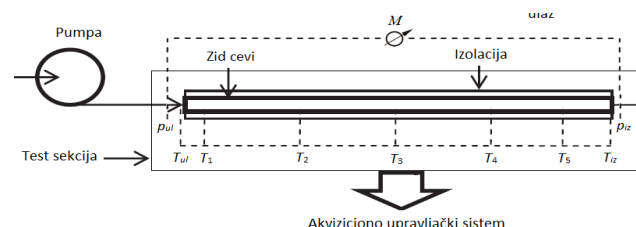
2. MATERIJAL I METOD

2.1. Uvod

Primenom računarske dinamike fluida ponovio se istraživanje strujanja i prenosa toplote jonske tečnosti koje je predhodno eksperimentalno urađeno. Fizička svojstva dobijena eksperimentalnim istraživanjem koristiće se za podešavanje numeričke simulacije.

2.2. Eksperimentalno postrojenje

Za eksperimentalno ispitivanje jonske tečnosti napravljeno je cirkulaciono postrojenje, čiji je deo šematski prikazan na slici 1. Postrojenje čine rezervoar, pumpa, ispitna sekcija, toplotni izmenjivač, rezervoari za skladištenje tečnosti i trokraki ventil.



Slika 1 Test sekcija eksperimentalnog postrojenja [3]

2.3. Verifikacija podataka

Razmatrano je stacionarno, potpuno razvijeno, laminarno strujanje nestišljivog fluida kroz pravu cev. Za potvrdu tačnosti rezultata numeričke simulacije vrši se verifikacija poznatim podacima iz literatura i rezultatima prethodnih

NAPOMENA:

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

eksperimentalnih istraživanja. Eksperimentalni podaci dobijeni predhodnim istraživanjem [3] pokazuju da se jonska tečnost, kao i voda, ponaša kao Njutnovski fluid. Takođe jonska tečnost ima iste trednove promene fizičkih svojstava sa temperaturom kao i voda. Zbog toga je voda izabrana kao fluid za utvrđivanje tačnosti.

Korišćeni su izrazi za određivanje pada pritiska, Nuseltovog broja i koeficijenta trenja:

$$\Delta p = \lambda \frac{L}{D} \cdot \frac{\rho \cdot v^2}{2} \quad (1)$$

$$Nu = \frac{\alpha \cdot D}{k_f} \quad (2)$$

$$\lambda = \frac{2 \cdot \Delta p \cdot D}{\rho \cdot L \cdot v^2} \quad (3)$$

gde su: Δp pad pritiska [Pa], L dužina cevi [m], D unutrašnji prečnik cevi [m], ρ gustina fluida [kg/m³], v brzina strujanja tečnosti [m/s], λ koeficijent trenja [-], α koeficijent prelaza toplote i k_f toplotna provodljivost [W/mK].

Relativna greška ε za ocenu odstupanja stvarne od izmerene vrednosti glasi.

$$\varepsilon = \frac{\Delta p_{sim} - \Delta p_{D-W}}{\Delta p_{D-W}} \cdot 100 \quad (4)$$

gde su: ε relativna greška [%], Δp_{sim} pad pritiska određen pomoću numeričke simulacije [Pa] i Δp_{D-W} pad pritiska određen pomoću Darsi-Vajsbahove formule [Pa].

Pored klasične formule za određivanje Nu koriste se i empirijske kriterijalne jednačine poput Šahove jednačine:

$$Nu(x) = \begin{cases} 1,953 \left(RePr \frac{D}{x} \right)^{\frac{1}{3}} & \left(RePr \frac{D}{x} \right) \geq 33,3 \\ 4,364 + 0,0722 RePr \frac{D}{x} & \left(RePr \frac{D}{x} \right) \leq 33,3 \end{cases} \quad (5)$$

gde su: x aksijalno rastojanje od ulaza cevi [m], Pr Prantlov broj [-] i Rejnoldsov broj Re [-].

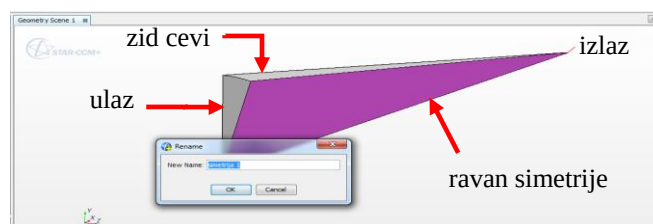
Kao parametar za evaluaciju performansi prenosa toplote koristi se Mourcefov broj Mo [-]:

$$\frac{Mo_{nf}}{Mo_{bf}} = \frac{k_{nf}}{k_{bf}} \quad (6)$$

gde su k_{nf} i k_{bf} toplotne provodljivosti nanofluida i bazne tečnosti [W/mK]. U slučaju potpuno razvijenog laminarnog strujanja Mourcefov Mo za nanofluid u odnosu na vodu jednak je odnosu odgovarajućih toplotnih provodljivosti

2.4. Podešavanje numeričke simulacije

Za pravljenje geometrijskog modela korišćić se grafički editor programskog paketa *StarCCM+* pod nazivom



Slika 2. Izgled geometrijskog modela

3D-CAD. Kako bi se simulacija odvijala što brže, uzdužno je iz cevi izdvojen klin od 15°, slika 2. Na ovaj način znatno će se smanjiti broj ćelija i ubrzati izvršenje numeričke simulacije. Treba napomenuti da su numeričke simulacije podešene uz određene aproksimacije:

- jonska tečnost je posmatrana kao jednofazni fluid;
- fizička svojstva jonske tečnosti podešena su za konstantnu vrednost temperature koja odgovara temperaturi tečnosti na ulazu u region i
- toplotni fluks se dovodi na unutrašnji zid cevi i usvojena je vrednost toplotnog fluksa iz kataloga proizvođača;

Na ulazu u region podesiće se brzina i temperatura jonske tečnosti, na zidu će se zadati konstantna vrednost toplotnog fluksa od 13300 W/m², dok će se na izlazu izabrati opcija **Field Function**. Na zidu je podešen uslov **No-Slip** što znači da fluid na čvrstoj granici ima nultu brzinu u odnosu na tu granicu, odnosno nema klizanja između tečnosti i zida. Generisani broj ćelija mrežnog kontinuuma iznosio oko 143000.

Dva osnovna kriterijuma konvergenције su korišćena za zaustavljanje iterativnog postupka kada rezultat postane dovoljno tačan:

- relativni ostaci pri rešavanju jednačina ($<10^{-4}$) i
- vrednost pada pritiska kao fizičke veličine od interesa između dve poslednje iteracije ($<1\%$).

Prilikom simulacije, veća važnost je data drugom kriterijumu konvergenције, tj. fizičkoj veličini od interesa, a to je bio pad pritiska.

3. REZULTATI I DISKUSIJA

Pre iznošenja rezultata dobijenih za jonsku tečnost prikazaće se rezultati verifikacije numeričkog modela.

3.1. Verifikacija rezultata

Pad pritiska tokom nerazvijenog strujanja je veći od pada pritiska na istoj dužini cevi sa razvijenim strujanjem. Ovo je posledica konverzije energije pritiska u kinetičku energiju i većeg trenja. Izvršavanjem numeričke simulacije dobija se pad pritiska od $\Delta p_{sim}=433,6$ Pa. Korišćenjem formule (1) dobija se vrednost pada pritiska od $\Delta p_{D-W}=430,4$ Pa.

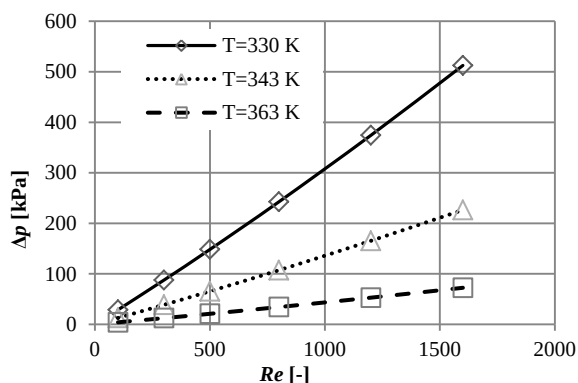
Izračunavanjem relativne razlike između simulacije i formule (1) dobija se vrednost $\varepsilon=0,7\%$ što se može smatrati prihvatljivim odstupanjem izmerene od stvarne vrednosti.

Ovim je potvrđen numerički model koji će se koristiti za numeričke simulacije jonske tečnosti [N₄₁₁][NTf₂].

3.2. Pad pritiska

Jonska tečnost na sobnoj temperaturi je i dalje tečljiva. Međutim, gustina i dinamička viskoznost imaju znatno

veću vrednost od vode, što otežava transport ovih fluida kroz cevi malog prečnika. Iz tog razloga razmatrane su veće temperature za istraživanje fizičkih osobina predmetne jonske tečnosti, slika 3.

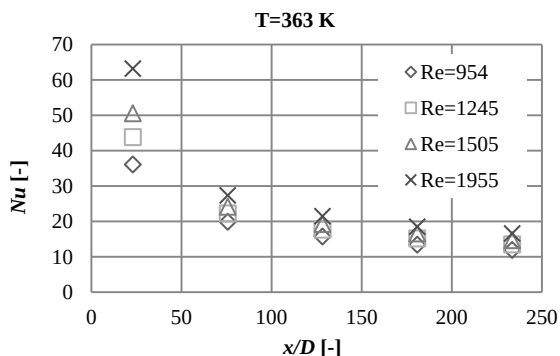


Slika 3 Zavisnost pada pritiska od Reynoldsovog broja za tri različite temperature jonske tečnosti na ulazu u cev

Moguće je primetiti da sa porastom temperature dolazi do značajnog smanjenja vrednosti pada pritiska. Uzrok toga je značajno smanjenje dinamičke viskoznosti sa povećanjem temperature jonske tečnosti. Međutim, ove vrednosti su i dalje daleko veće nego kod vode, što ne predstavlja povoljnu karakteristiku.

3.3. Nuseltov broj

Na slici 4 prikazana su rešenja numeričke simulacije pri istim Re kao u eksperimentu. Trebalo bi napomenuti da autor eksperimenta nije definisao tačnu temperaturu pri kojoj je vršio merenja, već samo opseg, pa su zbog toga ali i ostalih aproksimacija u odnosu na fizički model moguća mala odstupanja u rezultatima.



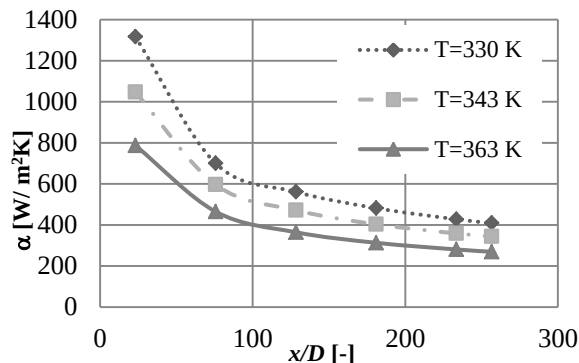
Slika 4 Zavisnost Nuseltovog broja Nu od bezdimenzionog rastojanja za različite vrednosti Reynoldsovog broja

Ukoliko je laminarno strujanje sa konstantnim toplotnim fluksom i ukoliko je potpuno razvijeno strujanje, Nuseltov broj će biti konstantan i iznosiće 4,36 i neće zavisiti od Re i Pr [4]. Sa slike 4 vidi se da Nuseltov broj opada sa povećanjem aksijalne udaljenosti. Razlog naglog pada je ulazni efekat, tačnije nerazvijeni termički sloj. Na samom ulazu dominira konvekcija, jer se toplota predaje sa zida cevi direktno u tečnost.

Iz razloga što nije dobijeno poklapanje vrednosti Nuseltovih brojeva Nu na istom aksijalnom rastojanju može se sa sigurnošću potvrditi da termički granični sloj nije potpuno razvijen.

3.4. Koeficijent prelaza toplote

Koeficijent prelaza toplote najčešće zavisi od brzine strujanja, gustine, dinamičke viskoznosti tečnosti, kao i geometrije cevi. Kako se geometrija nije menjala, primećuje se da koeficijent prelaza toplote opada sa povećanjem temperature što je uobičajeno za većinu tečnosti, slika 5.

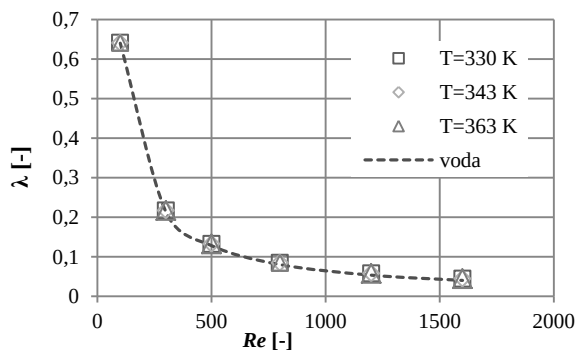


Slika 5 Zavisnost koeficijenta prelaza toplote od bezdimenzionog rastojanja za različite temperature jonske tečnosti na ulazu u cev

Do pada dolazi iz razloga što se temperatura fluida približava temperaturi zida tokom strujanja. Kako je razlika između temperatura sve manja i koeficijent prelaza toplote postaje manji.

3.5. Koeficijent trenja

Koeficijent trenja je određen primenom računarske dinamike fluida i jednačine (3) na tri različite temperature, uz variranje Reynoldsovog broja, slika 6. Na dijagramu se primećuje kako λ opada sa povećanjem Re , što predstavlja povoljnu karakteristiku ove jonske tečnosti. Pri manjim vrednostima Reynoldsovog broja Re dešava se značajnija promena, dok sa povećanjem Reynoldsovog broja Re koeficijent trenja počinje da se ustaljuje.



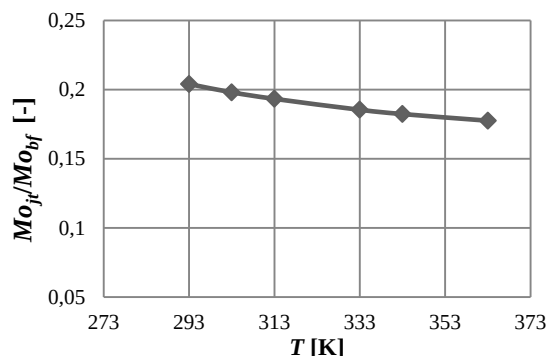
Slika 6 Zavisnost koeficijenta trenja od Reynoldsovog broja Re za različite vrednosti temperature na ulazu u cev

Na slici 6 vidi se kako koeficijent trenja ima istu vrednost za sve tri razmatrane temperature jonske tečnosti. Radi se o svojevrsnoj verifikaciji rezultata numeričke simulacije, jer za laminarno strujanje Njutnovskog fluida koeficijent trenja mora da se povinuje Stoksovoj jednačini.

3.6. Mourcefov broj

Za kraj je ostavljeno da se uradi sveukupna procena jonske tečnosti, kako bi se ustanovilo da li je njena

primena smisljena u procesima prenosa toplote. Procena je izvršena preko Mourcefovog broja, tačnije poređenjem toplotne provodljivosti jonske tečnosti i toplotne provodljivosti vode, slika 7.



Slika 7 Zavisnost odnosa Mourcefovih brojeva jonske tečnosti i vode od temperature

Sa slike se primećuje da odnos Mourcefovih brojeva jonske tečnosti i vode, formula (6), kontinualno opada sa povećanjem temperature. Kako je dobijena vrednost znatno manja od 1, a opšte prihvaćen uslov kaže da su samo vrednosti veće od 1 vredne razmatranja [5], zaključuje se da ova tečnost i nije baš najbolje rešenje kada je u pitanju primena kod solarnih kolektora. Treba napomenuti da je određivanje ovog broja uslovno iz razloga što se on primenjuje za procenu nano fluida.

4. ZAKLJUČAK

Predmet master rada bilo je strujanje jonske tečnosti kroz pravu cev, kružnog poprečnog preseka unutrašnjeg prečnika $D=3,86$ mm i dužine $L=990,6$ mm, pri laminarnom režimu i stacionarnim uslovima strujanja.

Cilj rada bio je da se istraži prinudna konvekcije i hidrauličke karakteristike pri strujanju jonske tečnosti kroz pravu cevnu deonicu. Analizom Nuseltovog broja Nu , koeficijenta prelaza toplote α , pada pritiska Δp , koeficijenta trenja λ i Mourcefovog broja Mo dobijena je jasnija slika o prinudnoj konvekciji i strujanju jonske tečnosti kroz pravu cev.

Istraživanje je urađeno primenom računarske dinamike fluida, pomoću programskog paketa *StarCCM+*. Mreža je većim delom struktuirana sačinjena od tetraedarskih i heksaedarskih ćelija, sa ukupnim brojem ćelija fizičkog modela od oko 143000. Kao kriterijum zaustavljanja podešen je maksimalan broj iteracija, a praćene su dve relevantne fizičke veličine od interesa: relativni ostaci pri rešavanju sistema jednačina i vrednost pada pritiska. Pad pritiska je veličina koja je služila za verifikaciju rezultata numeričke simulacije.

Pri analizi podataka uočeno je da pad pritiska znatno opada sa porastom temperature. Ova pojava može se pripisati velikoj zavisnosti viskoznosti od temperature, kao kod većine tečnosti. U poređenju sa rezultatima simulacije dobijenim za vodu može se primetiti da pad pritiska kod jonske tečnosti ima i do 1.000 puta veću vrednost pri istim vrednostima Rejnoldsovih brojeva Re zbog velike vrednosti dinamičke viskoznosti. Usled toga predlaže se da se u daljem istraživanju ovako visoka vrednost pada pritiska predmetne jonske tečnosti potvrdi eksperimentalnim putem.

Nuseltov broj Nu uglavnom varira u granicama od 1 do 10. U poređenju sa vodom, jonska tečnost ima veće vrednosti Nuseltovog broja. Ovo ukazuje da je konvekcija izraženija kod jonske tečnosti nego kod vode. Razvijanjem profila brzine (povećanjem aksijalnog rastojanja) Nuseltov broj Nu približava se vrednosti od 4,36, koja važi za potpuno razvijeni termički granični sloj pri konstantnom toplotnom fluksu.

Kod koeficijenta prelaza toplote sasvim očekivano dobija se ista tendencija kao i kod Nuseltovog broja. Sa porastom temperature i udaljavanjem od ulaza cevi koeficijent prelaza toplote kontinualno opada, a vrednosti su približne vrednostima za vodu.

Na kraju je urađeno izračunavanje Mourcefovog broja Mo , procenitelja da li je upotreba ove tečnosti interesantna kao medijum za prenosa toplote. Kao granična vrednost određen je broj 1. Ukoliko je dobijena vrednost manja od granične smatra se da nije smisleno korišćenje, što je slučaj kod predmetne jonske tečnosti. Treba napomenuti da je određivanje ovog broja uslovno iz razloga što se on primenjuje za procenu nano fluida.

5. LITERATURA

- [1] McEwen A.B., Ngo H.L., LeCompte K., Hargens L., (2000). *Thermal properties of imidazolium ionic liquids*, Thermochimica Acta, 357-358 (2000), 97-102.
- [2] Tomida D., Kenmochi S., Tsukada T., Qiao K., Yokoyama C., (2010). *Thermal Conductivities of Imidazolium-Based Ionic Liquid + CO₂ Mixtures*, Int J Thermophys 31 (10), 1888-1895.
- [3] Paul T.C., Morshed A.K.M.M., Fox E.B., Visser A.E., Bridges N.J., Khan J.A., (2014). *Thermal performance of ionic liquids for solar thermal applications*, Experimental Thermal and Fluid Science 59 (2014), 88-95.
- [4] Incropera F., DeWitt D., Bergman T., Lavine A., *Fundamentals of Heat and Mass Transfer, sixth edition*, John Wiley & Sons, Inc (2007).
- [5] Zyla G., Fal J., (2017). *Viscosity, thermal and electrical conductivity of silicon dioxide-ethylene glycol transparent nanofluids: An experimental studies*, Thermochimica Acta 650 (2017), 106-113.

Kratka biografija:



Nikola Oluški rođen je u Novom Sadu 1994. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mašinstva – Energetika i procesna tehnika odbranio je 2019.god.
kontakt: oluski.nikola@gmail.com

АНАЛИЗА СТРУЈАЊА У ЦЕНТРИФУГАЛНИМ ПУМПАМА ПРИМЕНОМ МЕТОДЕ КОНАЧНИХ ЗАПРЕМИНА**FLOW ANALYSIS IN CENTRIFUGAL PUMPS USING THE FINITE VOLUME METHOD**

Александар Рацић, Факултет техничких наука, Нови Сад

Област – МАШИНСТВО

Кратак садржај – У раду је изведена 3D нумеричка симулација рада једне центрифугалне пумпе коришћењем методе коначних запремина. Струјање у центрифугалним пумпама описано је помоћу Рејнолдсових осредњених Навије Стоксових једначина. У циљу постизања нумеричке стабилности прорачуна извршена је оптимизација нумеричке мреже. Симулација је изведена за случај нестационарног струјања при различитим протоцима, а у циљу анализирања струјања у радном колу и проточним деловима центрифугалне пумпе. Као резултат прорачуна добијена су поља брзина и притисака за различите радне режиме пумпе.

Кључне речи: Рачунарска динамика флуида, метод коначних запремина, центрифугалне пумпе.

Abstract – Within the paper a 3D numerical simulation of a centrifugal pump, using the finite volume method, was performed. The flow in centrifugal pumps is described using Reynolds averaged Navies-Stokes equations. In order to achieve numerical stability of calculation, optimization of numerical mesh was performed. The simulation was performed in case of unsteady flow for different flow rates, in order to analyze phenomena in the impeller and volute. such as pressure and velocity distribution.. As a result of calculation, fields of pressure and velocities for different operating points are obtained

Keywords: CFD, finite volume method, centrifugal pump.

1. УВОД

Предмет овог рада је анализа струјања у радном колу и проточним деловима центрифугалне пумпе применом методе коначних запремина, као једне од најпопуларнијих дискретизационих метода рачунарске динамике флуида (РДФ). РДФ представља један од алата при пројектовању и анализи свих типова турбомашина. Даје могућност обављања симулација и испитивања карактеристика пумпи без потребе за скупим експериментима, те уколико је коришћена на прави начин може да доведе до значајне уштеде времена.

Кориснички програми који се користе за нумеричке симулације базирају се на Навије-Стоксовим једначи-

нама и једначинама турбулентних модела којима је струјање флуида описано. Решавањем датих једначина добија се низ скаларних величина, чије се вредности додељују централним тачкама ћелија које чине нумеричку мрежу. У раду је изведена нумеричка симулација рада једностепене центрифугалне пумпе у корисничком програму STAR CCM+.

Симулације су урађене за случај нестационарног струјања течности кроз центрифугалну пумпу, у циљу одређивања радних карактеристика пумпе и анализирању поља притисака и брзина при константној брзини обртања пумпе.

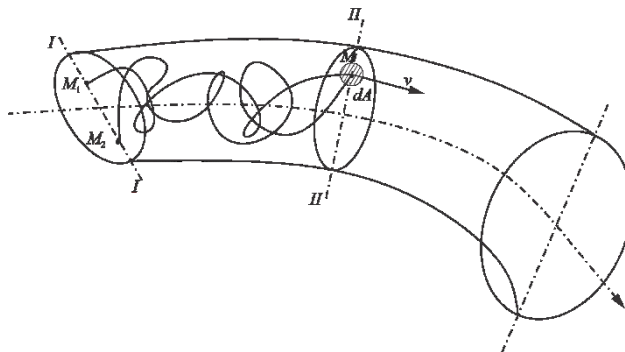
Коришћена пумпа је названа ПУМПА 1. За ПУМПУ 1 је позната геометрија која одговара номиналној (оптималној) вредности протока $Q = 210 \text{ m}^3/\text{h}$ и напора $H = 50 \text{ m}$ при брзини обртања радног кола пумпе од 1470 min^{-1} .

2. ПАРЦИЈАЛНЕ ДИФЕРЕНЦИЈАЛНЕ ЈЕДНАЧИНЕ И МАТЕМАТИЧКИ МОДЕЛ

Почетни корак ка разумевању рачунарске динамике флуида представља сагледавање улоге парцијалних диференцијалних једначина у процесу нумеричких симулација. Проблеми у области механике флуида, описују се парцијалним диференцијалним једначинама. Спознаја ПДЈ представља неизоставан сегмент приликом решавања проблематике рачунарске динамике флуида у оквиру турбомашина.

2.1 Турбулентно струјање

Струјање унутар пумпе је у потпуности турбулентно па све струјне карактеристике као што су брзина, притисак и тангенцијални напон имају пулзирајући карактер који је приказан на слици 1.



Слика 1. Турбулентне трајекторије флуидних делића и пулзација брзине

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Слободан Ташин, доцент.

Услед таквог пулзирајућег карактера споменуте величине морају бити изражене преко просечних величина и величина тренутног одступања [2]:

$$p = \bar{p} + p'; \tau = \bar{\tau} + \tau'; v = \bar{v} + v', \quad (1)$$

где су:

p', v', τ' – тренутна одступања притиска, брзине и тангенцијалних напона,

$\bar{p}, \bar{v}, \bar{\tau}$ – просечне вредности притиска, брзине и тангенцијалних напона.

Једначине којима се описује турбулентно струјање унутар центрифугалне пумпе су Рејнолдсове осредњене Навије-Стоксове једначине и једначина континуитета, које записане у функцији од општих координата (x_i) имају следећи облик [1]:

$$\frac{\partial \bar{v}_i}{\partial t} + \frac{\partial}{\partial x_j} (\bar{v}_i \bar{v}_j) = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\nu \frac{\partial \bar{v}_i}{\partial x_j} - \bar{v}_i' v_j' \right), \quad (2)$$

$$\frac{\partial \bar{v}_i}{\partial x_i} + \frac{\partial \bar{v}_j}{\partial x_j} = 0. \quad (3)$$

2.2 Моделирање турбуленције

Решавањем система једначина (2)-(3) добила би се струјна слика унутар пумпе. Међутим, број непознатих у овим једначинама је већи од броја једначина, па је неопходно дефинисати допунске услове, што се постиже различитим моделима турбуленције [2]. Моделирање турбуленције подразумева увођење одређених претпоставки о турбулентним напонима $\bar{v}_i' v_j'$ које резултирају допунским једначинама. У раду је коришћен $k-\omega$ модел са 2 допунске једначине:

- једначина за турбулентну кинетичку енергију k

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho \bar{v}_j k)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\eta + \frac{\eta_t}{\sigma_k^*} \right) \frac{\partial \omega}{\partial x_j} \right] + P_k - \rho \beta^* \omega k \quad (4)$$

и једначина за турбулентну фреквенцију ω

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho \bar{v}_j \omega)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\eta + \frac{\eta_t}{\sigma_\omega^*} \right) \frac{\partial \omega}{\partial x_j} \right] + \alpha \frac{\omega}{k} P_k - \rho \beta^* \omega^2. \quad (5)$$

Са једначинама (4) и (5) и одговарајућим граничним условима систем једначина (2)-(3) је решив (затворен), односно број непознатих је исти као и број доступних једначина.

3. МЕТОД КОНАЧНИХ ЗАПРЕМИНА

Метод коначних запремина је данас један од најкоришћенијих нумеричких метода. У већини проблема где је геометрија комплексна и где се мора формирати неструктурирана нумеричка мрежа овај метод је једино решење. Изузетно је лак за разумевање, јер сви чланови имају физички смисао.

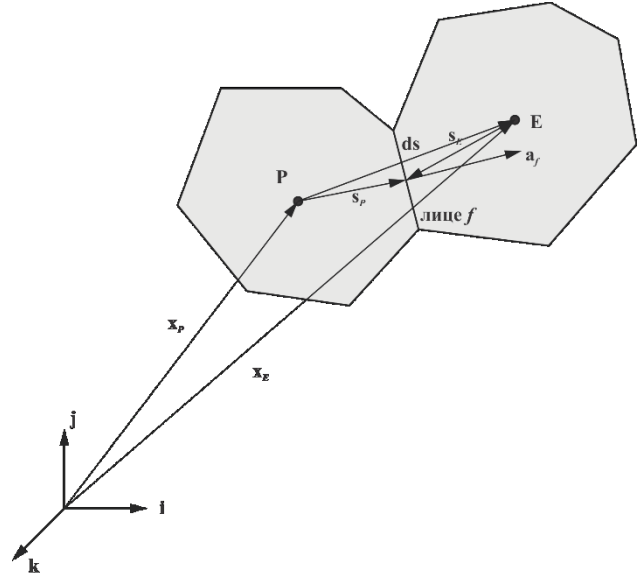
Најједноставније речено, суштина методе коначних запремина састоји се у томе да се флуидни домен најпре издели на велики али коначан број, међусобно повезаних контролних запремина енгл. Control volume (CV). Интеграњем задатог система ПДЈ по целој

контролној запремини добија се вредност тзв. генералисане скаларне променљиве ϕ , која се додељује централној тачки контролне запремине. Све наведене једначине могу се представити преко генерализоване транспортне једначине:

$$\frac{d}{dt} \int_V \rho \phi dV + \int_A \rho \phi \vec{v} \cdot \vec{n} dA = \int_A \Gamma \nabla \phi \cdot \vec{n} dA + \int_V q \phi dV \quad (6)$$

где чланови редом представљају:

- члан нестационарности,
- конвекциони члан,
- дифузиони члан,
- извор/понор.



Слика 2. Две полиедарске контролне запремине

Интеграњем једначина (6) по контролној запремини полиедарске ћелије (слика 2.) добија се вредност генералисане скаларне променљиве ϕ_P у централној тачки, која зависи од вредности генералисаних скаларних променљивих из суседних мрежних тачака ϕ_n . На овај начин формира се низ алгебарских једначина следећег облика:

$$\frac{a_P}{\omega} \Delta \phi_P + \sum_n a_n \Delta \phi_n = r_o. \quad (7)$$

где су:

a_P – коефицијент генералисане скаларне величине у централној тачки P ,

a_n – коефицијент генералисане скаларне величине у било којој централној тачки,

r_o – нумерички остатак.

Сукцесивним решавањем система алгебарских једначина (7) долази се до вредности генералисаних скаларних величина у свим централним тачкама а самим тим и до карактеристика струјног поља.

4. ГЕОМЕТРИЈА И НУМЕРИЧКИ ДОМЕН

Можда и најбитнији корак у нумеричким симулацијама представља правилно дефинисање геометрије проблема и креирање нумеричког домена тј. нумеричке мреже [3].

4.1 Геометрија пумпе

У Табели 1. приказане су димензије радног кола и спирале.

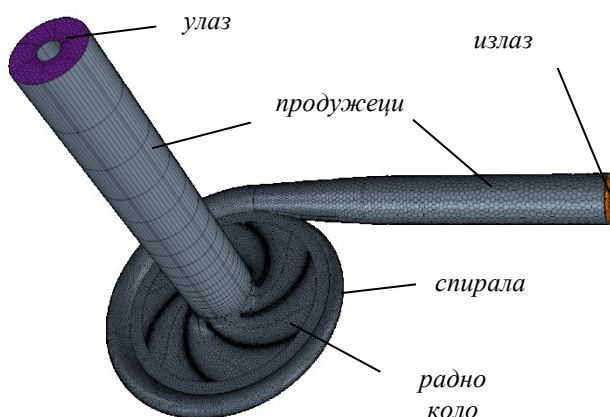
Табела 1. Димензије пумпе

Параметар	Вредност	Опис
Радно коло		
d_0	145 mm	пречник усисног грла радног кола
d_1	142 mm	пречник радног кола на улазу
b_1	36,3 mm	ширина радног кола на улазу
β_{1L}	24°	лопатични угао на улазу у радно коло
d_2	394 mm	пречник радног кола на излазу
b_2	17 mm	ширина радног кола на излазу
β_2	28,3°	лопатични угао на излазу из радног кола
z	6	број лопатица
δ	4 mm	дебљина лопатица
n_{sq}	18,88	специфична брзина обртања пумпе
Спирала		
R_3	205 mm	полупречник базе спирале
b_3	34 mm	ширина спирале
d_H	100 mm	спољашњи пречник дифузора

4.2 Нумерички домен

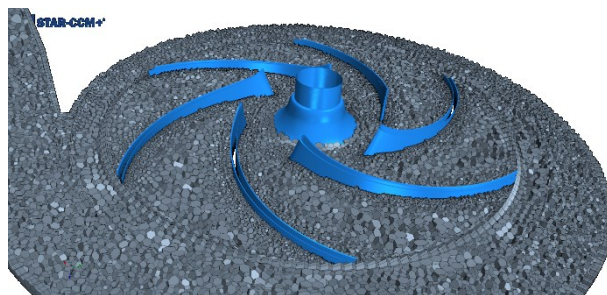
На основу података из табеле 1. пројектована је геометрија пумпе, из које је користећи болеан (Boolean) операције, креиран флуидни домен. Да би се од флуидног домена добио нумерички, потребно је флуидни домен изделити на велики али коначан број контролних запремина (ћелија), по којима се интеграле парцијалне диференцијалне једначине. На слици 3. се може видети нумерички домен (мрежа) пумпе. На улазу у радно коло и на излазу из дифузора спирале налазе се продужеци домена.

Дати продужеци изводе се ради спречавања рецикулације течности на улазу и излазу. Препорука је да износе 5-7 дужина пречника на улазу/излазу.



Слика 3. Нумеричка мрежа

Нумеричка мрежа оформљена за потребе симулације је неструктурирана и састављена од полиедарских ћелија. Поступак формирања мреже у случају центрифугалних пумпи, није једноставан. Потребна су подешавања мреже на местима где постоји интензиван транспорт флуидних величина.



Слика 4. Мрежа у радијалном пресеку радног кола

Са слике 4. се може видети да је густина мреже већа око лопатица радног кола, као и на излазу. То се постигло постављањем додатних призматичних слојева на тим местима. Посебна пажња је посвећена површини на којој долази до клизања ротора и статора, као и површинама које ограничавају флуидни домен, како би утицај зидова спирале био узет у обзир. Осцилације у нумеричким решењима могу упућивати на лошу обраду мреже у овим деловима. Адекватним пројектовањем геометрије и формирањем квалитетне нумеричке мреже остварује се директан утицај на тачност резултата симулације.

У табели 2. се може видети бројно стање ћелија по регијама. Очигледно, највећи број ћелија је у радном колу услед најучесталијег транспорта флуидних величина.

Табела 2. Број ћелија по регијама пумпе

Регија	Број ћелија
продужетак на улазу	8720
продужетак на излазу	6936
спирала	59937
радно коло	355150
УКУПНО	430743

5. РЕЗУЛТАТИ

Симулације су изведене за пет различитих протока $1,4Q$, $1,2Q$, Q , $0,8Q$, $0,5Q$, где је $Q = 210 \text{ m}^3/\text{h}$, при константној брзини обртања радног кола од 1470 min^{-1} .

Табела 3. Гранични услови

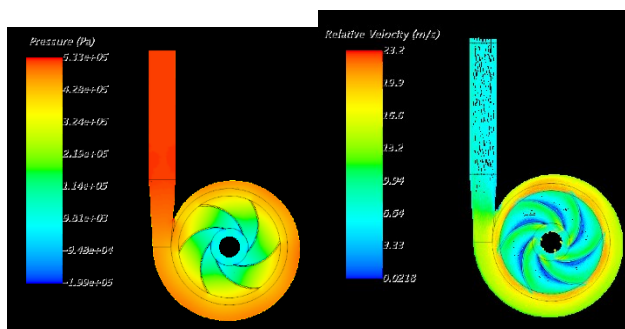
Граница	Гранични услов	Опис
Улаз	Stagnation inlet	$p_t = p_d = \frac{1}{2} \rho v^2$,
Излаз	Pressure outlet	$\dot{m}$ [kg/s] (Mass Flow Rate)
Зидови спирале и зидови радног кола	Wall	Не клизајућа површина (No Slip).
Зидови на продужецима на улазу и излазу	Wall	Клизајућа површина (Slip).
RSI	Interface	Mixing plane (стационарно струјање), In place (нестационарно струјање).

5.1 Нестационарно струјање

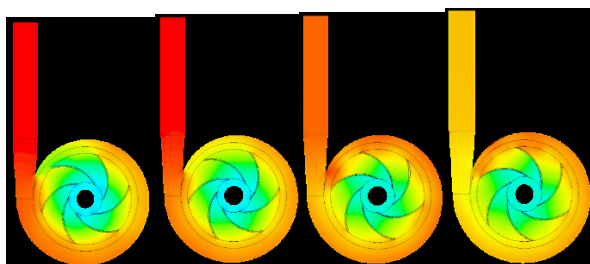
Симулације нестационарног струјања захтевају дефинисање временског корака. За његово дефинисање не постоји тачна формула али постоје препоруке којих се треба држати. Препорука је да временски корак буде време за које би се лопатица померила $0,5^\circ$ до 1° степена у кружном правцу. Усвојен временски корак је $\Delta t = 10^{-4}$.

При нестационарном струјању као решења добијена су поља притиска и поља брзина која су приказана на сликама 5, 6 и 7. Такође, исцртани су дијаграми радних карактеристика пумпе $H=f(Q)$, и $\eta=f(Q)$.

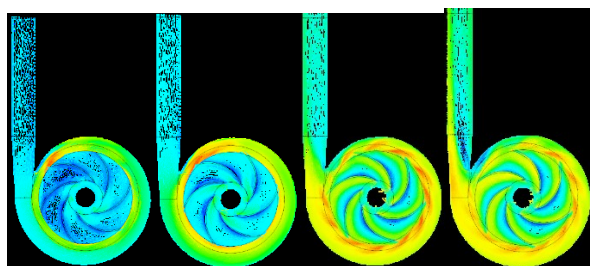
5.1.1 Скаларна поља притиска и векторска поља брзина



Слика 5. Скаларно поље притисака и векторско поље брзина при Q



Слика 6. Скаларно Поље притисака при $0,5Q_{opt}$, $0,8Q$, $1,2Q$ и $1,4Q$



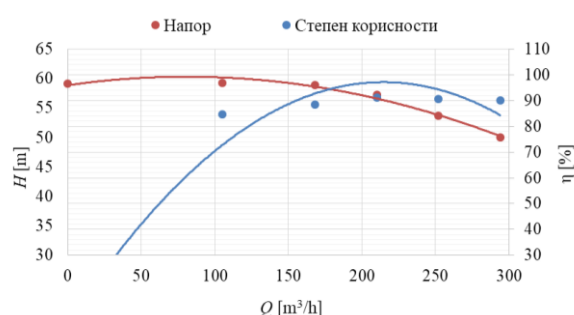
Слика 7. Распоред вектора брзина при $0,5Q$, $0,8Q$, $1,2Q$ и $1,4Q$

Са слике 5. се види да је распоред притисака у радном колу и спирали униформан са повољним градијентима. При протоцима нижим и вишим од оптималних (слика 6.) распоред притисака постаје све неуниформнији што доводи до појаве радијалних сила у радном колу, које негативно утичу на животног век пумпе. При изузетно ниским протоцима при проласку лопатице поред језичка долази до наглог пада притиска, где може доћи до појаве кавитације. Векторско поље брзина при оптималном протоку указује на очекиван

распоред брзина. На леђној страни лопатица долази до гомилања струјница те је ту локално већа брзина, док је на грудној мања, самим тим је условљен и распоред притисака на леђној односно грудној страни лопатице. Унутар радног кола у лопатичном каналу примећује се појава вртлога и одлепљивања флуидне струје што указује на могућ лош одабир облика и димензија лопатица. При изузетно ниским протоцима примећена је појава вртложења на самом улазу течности у радно коло, док је при високим протоцима примећено одлепљивање флуидне струје при уласку у дифузор спирале (слика 7.).

5.1.2 Радне карактеристике пумпе

Дијаграм 1. Радне карактеристике пумпе при $n=1470 \text{ min}^{-1}$



Са дијаграма 1. се потврђује да је највећа ефикасност управо при оптималном протоку $Q = 210 \text{ m}^3/\text{h}$.

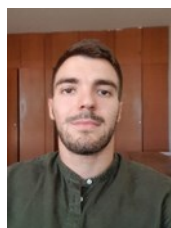
6. ЗАКЉУЧАК

Изведена је анализа струјања кроз центрифугалну пумпу користећи STAR CCM+ кориснички програм. На распоред притисака и векторског поља брзина директан утицај има величина протока. Закључено је да се при оптималним условима постиже највиши степен корисности пумпе. Забележени су феномени унутар лопатичног канала као што су појаве вртлога услед одлепљивања струје флуида и појаве вртлога на улазу у радно коло.

7. ЛИТЕРАТУРА

- [1] Букуров Ж., „Механика Флуида“, Факултет техничких наука, Нови Сад, (1967), 321-360.
- [2] „The Third Summer School of CFD“, Faculty of Technical Sciences, University of Novi Sad, Serbia, (2019), 20-24.
- [3] Best Practices for Modeling Centrifugal Pumps, <https://theesteveportal.plm.automation.siemens.com>, (2018).

Кратка биографија:



Александар Рацић рођен је у Новом Саду 1994. год. Дипломирао је 2017 год., на Факултету техничких наука у Новом Саду студијски програм Енергетика и процесна техника, а мастер рад на истом програму из области Механика флуида одбранио је 2019 године.

Контакт: aleksandar.racic.ns@gmail.com

MODELIRANJE SILA REZANJA PRI ČEONOM GLODANJU VISOKO-HROMIRANOG LIVENOG GVOŽĐA

MODELING OF CUTTING FORCES IN FACE MILLING OF HIGH-CHROME CAST IRON

Čedomir Ždrnja, Borislav Savković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – U okviru ovog rada prikazani su eksperimentalno dobijeni rezultati pri obradi čeonom glodanju visoko-hromiranog livenog gvožđa. Rezultati su predstavljani u vidu dijagrama i tabela i izvršena je njihova analiza. Merenje i prikaz ovih rezultata izvršen je putem merno akvizicionog sistema. U radu je dat opis ovog sistema i način njegovog funkcionisanja.

Ključne reči: Čeono glodanje, visoko-hromirano liveno gvošđe, sile rezanja, akvizicija podataka.

Abstract – Within this paper experimental results are obtained in machining with the main milling of high - chromium iron. The results were presented as diagrams and tables and their analysis was performed. Measurement and presentation of these results was carried out through a measurable acquisition system. The paper describes this system and how it works.

Key words: Face milling, high-chromium cast iron, cutting forces, data acquisition.

1. UVOD

Proučavanju sila rezanja koje nastaju u procesu glodanja materijala posvećeno je veliki broj radova. Postoji mnogo tehnika za izučavanje sila rezanja tokom glodanja u zavisnosti od tehnoloških parametara obrade. Trenutno se empirijske, analitičke i grafoanalitičke zavisnosti i metode široko koriste za izučavanje sila rezanja za oblikovanje glodanjem.

Najznačajniji uticaj u tehnologiji mašingradnje imaju sile i momenti pri rezanju metala. Sile i momenti pri rezanju su jedna od osnovnih karakteristika stanja procesa rezanja [1,2].

Kada se odrede funkcije sila rezanja i njihovih momenata, one omogućuju da se:

- Proračuna snaga koja se troši u datom zahvatu ili operaciji i izabere odgovarajući motor mašine alatke;
- Izaberu i proračunaju glavni i pomoćni sistemi prenosnih mehanizama od izvora do reznog alata;
- Izvrši proračun čvrstoće i dimenzionišu elementi i jedinice mašine;
- Odrede dimenzije i izvrši odgovarajući proračun pomoćnih pribora u projektovanoj tehnologiji;

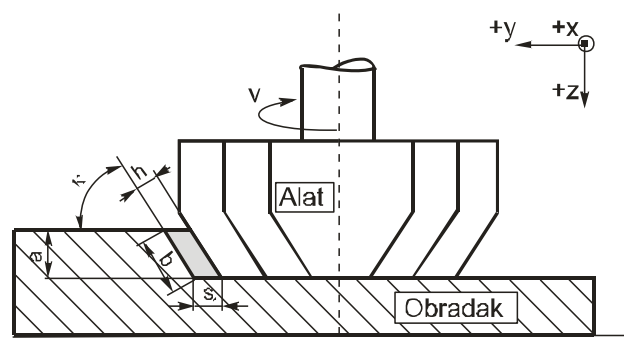
NAPOMENA:

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

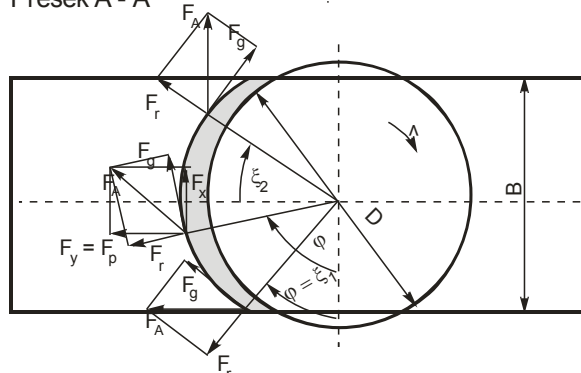
- Odaberu tipovi i dimenzije reznog alata i proveriti stabilnost alata u celini;
- Odrede elementi režima i uslovi obrade, pri projektovanju ekonomičnih i produktivnih varijanti tehnološkog procesa obrade;
- Izvrši proračun tačnosti i provera mogućnosti obrade nekog mašinskog dela sa stanovišta propisanog kvaliteta.

Na ovaj način se u proizvodnim uslovima, na osnovu poznavanja i korišćenja funkcija sila, obezbeđuje racionalna konstrukcija i ekonomična eksploatacija proizvodnih sredstava, optimizacija procesa obrade, kao i razvoj određenih koncepata upravljanih obradnih sistema sa graničnim regulisanjem.

Pri analiziranju sila rezanja kod čeonog glodanja doazi se do toga da se sila meri u tri osnovna pravca x , y i z . Šema sila rezanja i pomenuta tri pravca prikazana su na slici 1. Kod rezanja razlikuju se sile po zubu i sile koje deluju na istovremeno na sve zube alata. Kada se vrši proračun sila rezanja prvo se računa sila po jednom zubu, a potom se u proračun uključuju i ostali zubi reznog alata.



Presek A - A



Slika 1. Šema sila rezanja pri čeonom glodanju [3,4]

U zavisnosti od ugla φ , može se izračunati trenutna vrednost glavne sile rezanja F_g koja deluje tangencijalno, radijalne sile rezanja F_r , sile prodiranja F_p i aksijalne sile F_a koja deluje paralelno sa osom vretena [5].

U nastavku su prikazane jednačine koje opisuju sile rezanja na jednom zubu glodala. Posmatrana su dva položaja zuba glodala u zahvatu. Prvo su definisane sile za položaj I, za ugao $\varphi < \pi/2$, dok je drugi položaj II za ugao $\varphi > \pi/2$.

Položaj I:

$$F_x = -F_g \cdot \sin \varphi + F_r \cdot \cos \varphi \quad (1)$$

$$F_y = F_g \cdot \cos \varphi + F_r \cdot \sin \varphi \quad (2)$$

$$F_z = F_a \quad (3)$$

Rešavajući sistem jednačina (1), (2) i (3) dobija se:

$$F_g = -F_x \cdot \sin \varphi + F_y \cdot \cos \varphi \quad (4)$$

$$F_r = F_x \cdot \cos \varphi + F_y \cdot \sin \varphi \quad (5)$$

$$F_a = F_z \quad (6)$$

Sila u pravcu pomoćnog kretanja F_p u svakom trenutku jednaka je sili F_y , odnosno:

$$F_p = F_g \cdot \cos \varphi + F_r \cdot \sin \varphi = F_y \quad (7)$$

Položaj II:

$$F_x = -F_g \cdot \cos \left(\varphi - \frac{\pi}{2} \right) - F_r \cdot \sin \left(\varphi - \frac{\pi}{2} \right) \quad (8)$$

$$F_y = -F_g \cdot \sin \left(\varphi - \frac{\pi}{2} \right) + F_r \cdot \cos \left(\varphi - \frac{\pi}{2} \right) \quad (9)$$

$$F_z = F_a \quad (10)$$

Uzimajući u obzir adicione formule:

$$\sin \left(\varphi - \frac{\pi}{2} \right) = \sin \varphi \cdot \cos \frac{\pi}{2} - \cos \varphi \cdot \sin \frac{\pi}{2} = -\cos \varphi \quad (11)$$

$$\cos \left(\varphi - \frac{\pi}{2} \right) = \cos \varphi \cdot \cos \frac{\pi}{2} + \sin \varphi \cdot \sin \frac{\pi}{2} = \sin \varphi \quad (12)$$

Jednačine (8), (9) i (10) se svode na jednačine (4), (5) i (6) čije je rešavanje već pokazano.

2. CILJ ISPITIVANJA

U ovom radu je pomoću razvijenog sistema za praćenje, akviziciju i merenje sila pri obradi čeonim glodanjem, izvršeno merenje sila. Cilj zadatka je bio izvršiti analizu uticaja elemenata režima obrade na vrednosti sila rezanja. Takođe je trebalo izvršiti proračunavanje izmerenih komponenti sila rezanja u glavnu silu rezanja, silu prodiranja i silu pomoćnog kretanja u vremenskom domenu trajanja jednog obrtaja alata, a zatim doneti odgovarajuće zaključke. Eksperimenta je sproveden po centralno kompozicionom planu, gde su promenjivi ulani parametri varirani na pet nivoa.

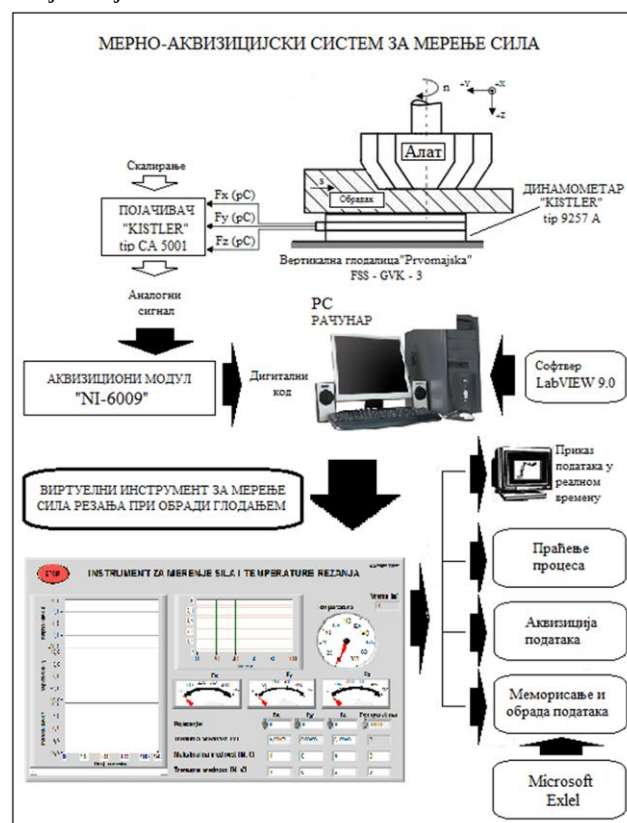
3. MERNO-AKVIZICIONI SISTEM ZA MERENJE SILA PRI ČEONOM GLODANJU

3.1. Karakteristike korišćenog akvizicionog sistema

Pojava digitalnih računara šezdesetih godina dvadesetog veka dovela je do pojave novih mogućnosti u oblasti nadzora i upravljanja tehnološkim procesima i sistemima. Razvojem računara razvile su se merno-akvizicione tehnologije koje svoju primenu baziraju na mikroračunarskoj tehnici.

Akvizicija (Acquisition) označava pojam sticanja, prikupljanja ili akumuliranja nečega. Akvizicija podataka - DA (Data Acquisition) predstavlja proces prikupljanja informacija iz sredine koja se razmatra u cilju njenog upoznavanja.

Šematski prikaz merno-akvizicionog sistema za merenje sila pri čeonom glodanju primenom virtuelne instrumentacije dat je na slici 2.



Slika 2. Model merno-akvizicionog sistema za merenje sila rezanja pri čeonom glodanju [4]

Sa slike 2 se vidi da se sistem sastoji od sledećih komponenti:

- glodalica – „Prvomajska“ FSS-GVK-3
- dinamikometar – „KISTLER“ tip 9257A
- pojačalo – „KISTLER“ tip CA 5001
- NI USB-6009 – akvizicioni modul
- Računar sa instalisanim LabVIEW programskim paketom.

4. EKSPERIMENTALNA ISPITIVANJA

Primenom prikazanog sistema (slika 2.) izvršena su merenja i akvizicija ortogonalnih sila rezanja pri čeonom glodanju.

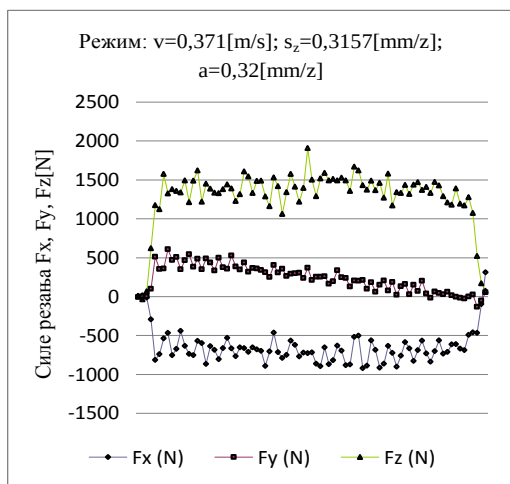
Prilikom eksperimentalnih ispitivanja merene su sile F_x , F_y , F_z , a na osnovu njih su računskim putem dobijene sile F_g , F_r , F_p i F_a .

Pri eksperimentu korišćeno je glodalo prečnika $D=100\text{mm}$, sa mehanički pričvršćenim reznim pločicama. Kao rezne pločice korišćene su pločice koje se mogu okretati u slučaju loma odnosno habanja, sa prevlakama od tvrdog metala, dobijene postupkom CVD (hemijsko taloženje pare). Materijal obratka koji korišćen pri izvođenju eksperimenta u ovom radu je visoko-hromirano liveno gvožđe sa tvrdoćom $44\div 46\text{ HRC}$ i hemijskim sastavom koji je datu tabeli 1.

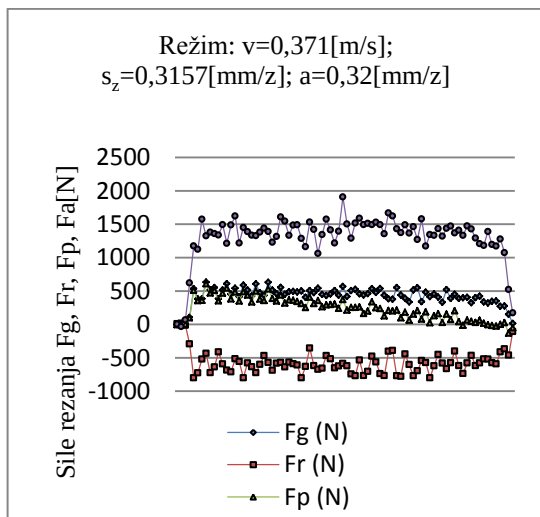
Tabela 1. Hemijski sastav visoko-hromiranog livenog gvožđa

Oznaka materijala	C	Si	Mn	Cr	Mo	Ni
17680	3,95	0,5	0,98	21,25	1,7	5,7

Na slici 3 prikazan je dijagram promene sila rezanja u izvornom merenom obliku sa komponentama F_x , F_y , F_z za režim obrade: $V=0,371\text{ [m/s]}$, $s_z=0,3157\text{ [mm/z]}$, $a=0,32\text{ [mm]}$. Nakon primene jednačina (4), (5), (6) i (7) dobija se dijagram sa komponentama F_g , F_r , F_p i F_a , slika 4.



Slika 3. Promena sila rezanja za komponente F_x , F_y , F_z



Slika 4. Promena sila rezanja za komponente F_g , F_r , F_p i F_a

Na slikama 3 i 4 prikazana je promena sila rezanja za srednji nivo režima po planu eksperimenta prikazanog u radu.

Poseban slučaj se javlja prilikom analize grafika za glavnu silu rezanja i radijalnu silu. Prema literaturnim podacima [2] radijalna sila bi trebala da ima istu orijentaciju kao i glavna sila i da bude nešto manje vrednosti. Međutim, prilikom obrade rezultata pokazalo se da sila F_r ima negativnu vrednost suprotno od pretpostavke. Postoji nekoliko razloga za ovakvu orijentaciju sile. Prvi razlog za to je negativna orijentacija u najvećem delu, sile F_x . Drugo, prečnik glodala je veći od širine glodanja što je suprotno preporukama. Usled ovakvog prečnika glodala, dobijene su se velike udarne sile na ulasku rezne pločice u obradak, što veoma utiče na dobijene rezultate.

Takođe se na dijagramima se vidi da je najveća aksijalna sila rezanja F_a . Razlog za povećanu vrednost aksijalne sile je teško obradiv materijal pri čemu se javljaju veliki aksijalni udari na vreteno glodalice.

Takođe pri obradi pomenutog materijala javlja se velika pohabanost reznih pločica, gde čak dolazi i do loma rezne ivice.

Rezna pločica koršćena pri ovom ispitivanju bila je »SPKR 12 03 ED R-WH 3040«. Na slici 5 prikazan je izgled nove rezne ivice, dok je na slici 6 prikazana pohabana pločica. Treba napomenuti da su maksimalne gornje granice promenljivih parametara bile u dijapazonu preporučenih vrednosti režima obrade. Shodno tome da se habanje pločica javilo u većem broju ispitivanih režima, može se zaključiti da bi preporučeni interval za režime obrade trebalo korigovati odnosno suziti.

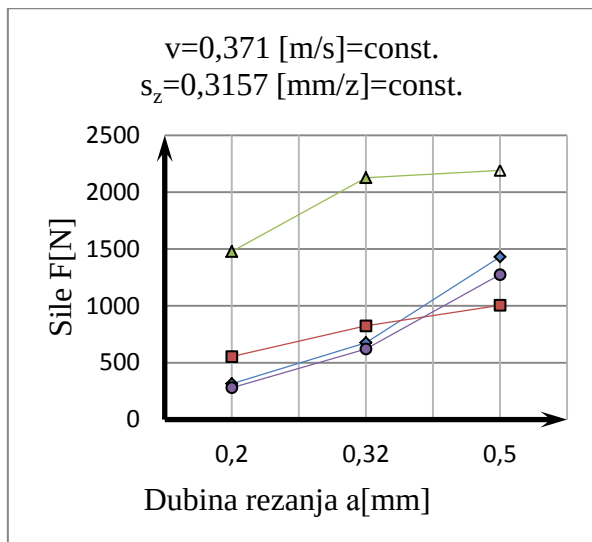


Slika 5. Nova rezna ivica

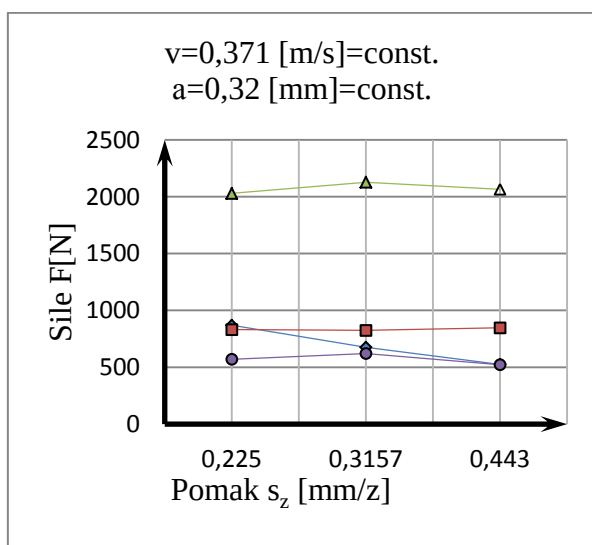


Slika 6. Pohabana rezna ivica

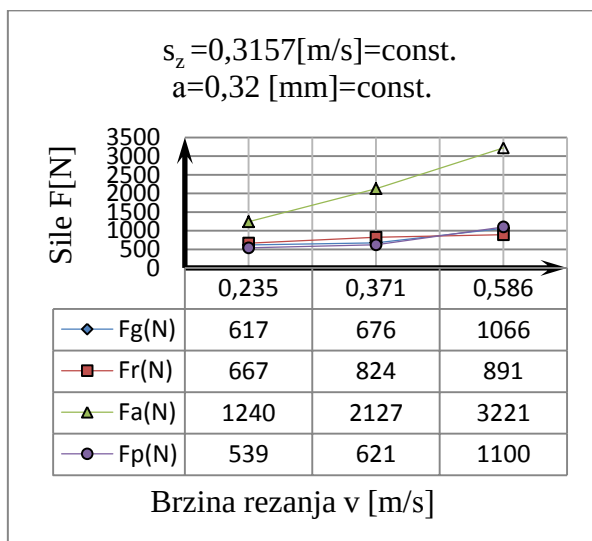
Daljom analizom predstavljeni su dijagrami na slikama 7, 8 i 9 gde se vidi da sa porastom vrednosti režima obrade uglavnom rastu i sile rezanja. Treba spomenuti da se u nekim slučajevima javlja i negativan trend promene sila rezanja što se može pripisati pojavi habanja rezne ivice i grešci prilikom merenja. Na osnovu ova tri dijagrama vidi se da je najveći porast sila rezanja pri variranju dubine rezanja, slika 7.



Slika 7. Zavisnost dubine rezanja i sile rezanja



Slika 8. Zavisnost pomaka i sile rezanja



Slika 9. Zavisnost brzine rezanja i sile rezanja

5. ZAKLJUČAK

Modeliranje procesa obrade i optimizacija ulaznih parametara imaju značajno mesto u savremenoj proizvodnji. Međutim, procesi obrade su veoma složeni, gde je potrebno istovremeno pratiti više izlaznih parametara. Jedan od izlaznih parametara su sile rezanja. One su jedan od osnovnih kriterijuma za ocenu obradivosti materijala i kao takve privlače veliku pažnju istraživača iz ove oblasti. Zbog uticaja velikih broja parametara na sile rezanja eksperimentalno dobijeni rezultati vrlo često odstupaju od literaturnih vrednosti. Osnovu za razvoj mašina alatki i samu optimizaciju predstavljaju upravo ta odstupanja. Takođe, postoje i neki drugi faktori koji utiču na sile rezanja, a ne mogu se predvideti.

Prilikom izvođenja ovog eksperimenta korišćena je starija mašina alatka i nije korišćeno sredstvo za podmazivanje i hlađenje. U toku ispitivanja uočeno je habanje alata u velikoj meri. To objašnjava zašto dolazi do pomene odstupanja od teorijskih očekivanja.

Zaključci koji su rezultat ovog istraživanja su sledeći:

- Materijal obratka, visoko-hromirano liveno gvožđe ima nepovoljnu obradivost, odnosno treba precizno odrediti režime obrade kao i izbor alata.
- Izlazni podaci u samom eksperimentu su sile rezanja koji pre svega zavisi od mernog instrumenta i od uslova merenja.
- Pri ispitivanju korišćene su rezne pločice do momenta loma i odnosno gubitka funkcije. Shodno tome ovaj vid habanja alata svakako utiče na dobijene rezultate.

6. LITERATURA

- [1] P. Kovač, D. Milikić, „Rezanje Metala”, Novi Sad, Februar, 1998.
- [2] Milikić, D., Gostimirović, M., Sekulić, M.: Osnove tehnologije obrade rezanjem, Fakultet tehničkih nauka, Novi Sad, 2008.
- [3] M. Sekulić, „Razvoj podloga za proračun sile rezanja”, Magistarski Rad, FTN, Novi Sad, Oktobar, 1998.
- [4] Savković, B.: Modeliranje funkcija obradivosti pri procesu obrade glodanjem, Doktorska disertacija, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, Novi Sad 2015.
- [5] Mijić, A.: Modeliranje sile rezanja pri čeonom glodanju, Fakultet tehničkih nauka, Novi Sad, 2009.

Kratka biografija:



Čedomir Ždrnja rođen je u Linz-u, Austrija 1992. god. Diplomski rad na Fakultetu tehničkih nauka odbranio je 2015. godine.



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

**ДЕТЕКЦИЈА КАВИТАЦИЈСКИХ РЕЖИМА У ЦЕНТРИФУГАЛНИМ ПУМПАМА
ПРИМЈЕНОМ ВИБРОДИЈАГНОСТИЧКИХ МЕТОДА****CAVITATION DETECTION IN CENTRIFUGAL PUMPS USING VIBRO-DAGNOSTIC
METHODS**

Радивојка Вучинић, Факултет техничких наука, Нови Сад

Област – МАШИНСТВО

Кратак садржај – У оквиру рада приказан је поступак експерименталног испитивања појаве кавитације на примјеру центрифугалне пумпе у пумпној станици „Лиман“ у Новом Саду. Током експеримента паралелно су мјерени хидраулички радни параметри пумпе и вибрације на одабраним тачкама на кућишту пумпе. Објашњени су појмови кавитације и вибродијагностике и приказани су поступак и резултати испитивања, као и анализа резултата.

Кључне речи: вибрације, вибродијагностика, кавитација, центрифугалне пумпе,

Abstract – Within the paper the procedure of experimental testing of the occurrence of cavitation on the example of one centrifugal pump in the pumping station „Liman“ in Novi Sad is presented. The experiment comprised parallel measurement of the pump hydraulic parameters and vibrations at selected points on the pump casing. The concepts of cavitation and vibration diagnostics are explained and the procedure and test results are presented as well as the analysis of the results.

Key words: vibrations, vibration diagnostic, cavitation, centrifugal pumps

1. УВОД

Кавитација је један од главних погонских проблема са којим се сусрећу центрифугалне пумпе. Појава кавитације праћена је ерозијом материјала, нестабилним радом, буком и вибрацијама. Будући да су вибрације релативно лако мјерљив пратилац кавитације, могуће је на основу њих установити појаву и развој кавитације у струјном току унутар пумпе.

2. ВИБРОДИЈАГНОСТИКА

Вибрације су неизбежан пратилац рада ротационих машина, често непожељан, будући да изазивају замор и хабање и врло често су одговорне за отказ у раду. Под вибрацијама које се јављају на машинама подразумевамо кретање механичких компоненти машине, изазвано унутрашњим и спољашњим силама.

Вибродијагностика представља скуп активности који се проводе са циљем утврђивања стања техничког система на основу појаве вибрација, ради доношења одлука о даљим активностима одржавања или давања прогнозе понашања техничког система у будућности.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био доцент др Слободан Ташин.

Један од узрока појаве вибрација код хидрауличких машина јесте кавитација.

3. КАВИТАЦИЈА

Кавитација настаје када локални притисак течности у струјном току опадне испод притиска засићене паре при датој температури, па долази до испаравања течности и стварања парних мјехурића. Ношени струјом течности парни мјехурићи улазе у зону повишеног притиска гдје имплодирају изазивајући при томе интензивне локалне пулсације притиска.

Најозбиљнија последица кавитације је кавитацијска ерозија материјала. Различити материјали могу бити мање или више отпорни на дејство кавитације али не постоји материјал који може трајно да издржи њено дејство. У случају центрифугалних пумпи најугроженије мјесто од настанка кавитације јесте улазна ивица лопатице радног кола.

3.1. Кавитацијска резерва

За проучавање појаве кавитације у центрифугалним пумпама потребно је дефинисати величину која се може лако измјерити и која на неки начин описује кавитацијске процесе у пумпама. Таква величина, данас опште прихваћена у инжењерској пракси, је кавитацијска резерва $NPSH$ (Net Positive Suction Head).

Кавитацијска резерва дефинише се као нето позитивна специфична енергија флуида у уписном пресеку пумпе, односно као разлика тоталног притиска на упису (p_t) и притиска засићене паре (p_v) за дату температуру [1]:

$$NPSH = \frac{p_t - p_v}{\rho g} - (z_0 - z_l) \quad (1)$$
$$= \frac{p_t - p_v}{\rho g} + \frac{v_l^2}{2g} - (z_0 - z_l)$$

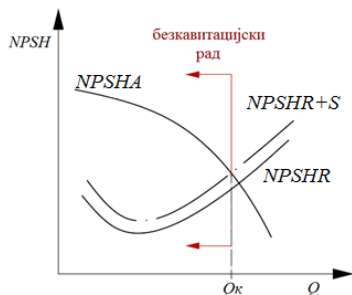
Инжењерски прорачун опасности од рада пумпе у кавитацијском режиму своди се на познавање потребне кавитацијске резерве пумпе $NPSHR$ и расположиве кавитацијске резерве инсталације $NPSHA$ [1].

Сматра се да пумпа ради у режиму без кавитације све док је задовољен услов:

$$NPSHA > NPSHR + S \quad (2)$$

гдје је S сигурносни додатак (мин. 0,5 m) који узима у обзир тип пумпе и материјал од којег је израђено радно коло пумпе.

Графички приказ услова (2) дефинише област безкавитацијског рада пумпе (слика 1):



Слика 1. Услов безкавитацијског рада

Произвођачи пумпи договорно дефинишу критеријум за одређивање $NPSHR$. Овај критеријум одговара таквом интензитету кавитације при којем долази до 3%-ог смањења напора пумпе при константној брзини обртања и протоку [2].

Често се уместо кавитацијске резерве $NPSH$ за анализу кавитације користи бездимензијски параметар назван кавитацијски коефицијент:

$$\sigma = \frac{NPSH}{\frac{1}{2g} U_0^2} \quad (3)$$

гдје је: U_0 – брзина струјања у погодном одабраној, референтној тачки струјног тока [m/s].

У табели 1. приказани су типични односи карактеристичних кавитацијских коефицијената центрифугалних пумпи при $Q = Q_{opt}$ [3].

Табела 1. Типични односи карактеристичних кавитацијских коефицијената центрифугалних пумпи при $Q = Q_{opt}$

Тип радног кола	$\sigma_i/\sigma_{3\%}$	$\sigma_{0\%}/\sigma_{3\%}$	$\sigma_b/\sigma_{3\%}$
Радна кола нормалне конструкције	4 - 6	1,1 - 1,3	0,8 - 0,9 $\approx 1,0$ за веће n_{ss}
Са специјално обликованим улазним ивицама лопатица	2		
Радна кола са малим бројем лопатица ($z = 3$)	8 - 10		

σ_i - регистрован почетак кавитације у пумпи

$\sigma_{0\%}$ - регистрован почетак опадања напора пумпе услед кавитације

$\sigma_{3\%}$ - регистрован пад напора пумпе од 3% услед кавитације

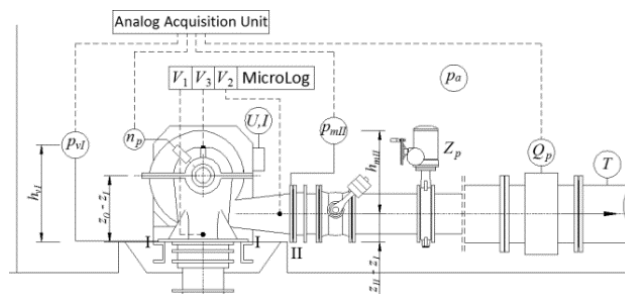
σ_b - потпун губитак напора пумпе услед кавитације

4. ЕКСПЕРИМЕНТАЛНО ИСПИТИВАЊЕ ПОЈАВЕ КАВИТАЦИЈЕ У ПС „ЛИМАН“

Основна улога ПС „Лиман“ јесте да пуни резервоар „Институт“, који служи за снабдевање водом виших зона новосадског водовода. У ПС „Лиман“ уграђена су три пумпна агрегата типа „Југотурбина ДХ 58-35“. Пумпе су двострујне, центрифугалне, у хоризонталној изведби. Погонски електромотори су номиналне снаге 480 kW, брзине обртања 1485 min^{-1} .

4.1. Мјерне величине

За потребе испитивања кавитације мјерене су следеће хидрауличке величине (слика 2): проток пумпе Q_p , потпритисак на усису пумпе p_{vl} , натпритисак на потису пумпе p_{ml} , атмосферски притисак p_a , температура воде T (потребна за израчунавање притиска zasiћене паре) и брзина обртања пумпе n_p . Мјерење вибрација вршено је у три тачке на кућишту пумпе: на уисном конфузору V1, на потисном дифузору V2 и на лејају на страни пумпе V3.



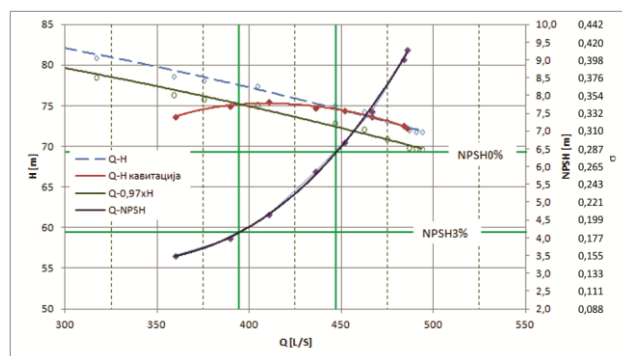
Слика 2. Мјерне величине у ПС „Лиман“ [1]

Мјерења хидрауличких параметара вршена су у складу са стандардом ISO 9906:2012, коришћењем уобичајене мјерне опреме (ултразвучно мјерило протока, вакуум-метар, манометар, барометар, инфрацрвени термометар, тахометар). Мјерења вибрација вршена су у складу са стандардом ISO 10816-5:2000 коришћењем пиезоелектричних акцелерометара „Wilcoxon Research 780 B“.

Изазни мјерни сигнали уређаја за мјерење натпритиска, потпритиска, протока и брзине обртања повезани су са уређајем за аквизацију. Мјерни сигнали вибрација прикупљани су и анализирани помоћу вишенаменског анализатора типа „SKF Microlog CMVA 60“.

4.2. Резултати мјерења хидрауличких радних параметара

На основу мјерења хидрауличких параметара добијене су најприје радне карактеристике испитиване пумпе а затим и кавитацијски дијаграми од којих је један, за радни режим „P1“, приказан на слици 3.



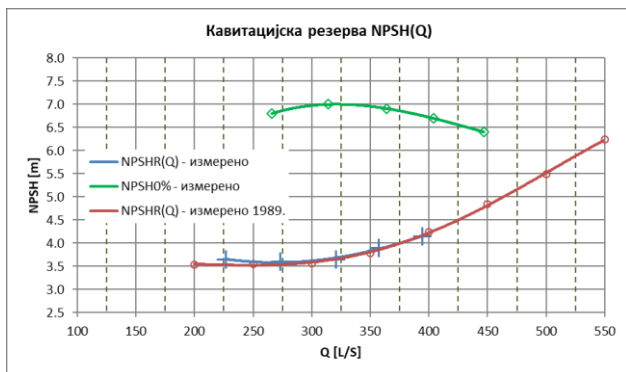
Слика 3. Кавитацијски дијаграм – режим „P1“

Кавитацијски дијаграми добијени су методом снижавања притиска на усису пумпе, при чему је проток промјенљива величина, што спада у стандардне методе испитивања, дефинисане у ISO 9906:2012.

У пресеку напора пумпе при пригушивању (црвена крива) и карактеристике пумпе 0,97H (зелена крива)

очитава се $NPSHR$ пумпе ($NPSH_{3\%}$) која се према ISO стандарду проглашава за $NPSHR$ пумпе.

Будући да је мјерење урађено за пет различитих радних режима пумпе добијено је пет оваквих дијаграма на основу којих је добијена $NPSHR$ карактеристика пумпе (слика 4).



Слика 4. Зависност кавитацијске резерве од протока (поређење $NPSHR$ мјерено 1989. године и у току експеримента) [1]

Са слике 4 је уочљиво да се измјерена $NPSHR$ крива практично у потпуности поклапа са кривом која је за исту пумпу измјерена 1989. године, такође у радним условима.

Интересантно је уочити да за оптимални проток пумпе од $Q = 485$ L/s потребна кавитацијска резерва износи $NPSHR = 5,4$ m а да је у том радном режиму измјерена расположива кавитацијска резерва од $NPSHA = 9,3$ m. Будући да је према томе испуњен услов бескавитацијског рада $NPSHA > NPSHR + S$ ($9,3$ m $>$ $5,4$ m + $0,5$ m), могло би се закључити да пумпа ради без кавитације.

4.3. Резултати мјерења вибрација

Као резултат мјерења вибрација добијени су спектрограми за радне режиме од $I = 0$ (без пригушења на усису пумпе, условно речено – бескавитацијски режим) до $I = N$ (режим потпуно развијене кавитације).

Спектрограми снимљени за радни режим „P1“ приказани су на сликама 5, 7 и 9.

Измјерени фреквентни спектри накнадно су анализирани на персоналном рачунару како би се уочио фреквентни домен у којем је приметан утицај кавитације. У тако издвојеном фреквентном домену ($f_i \in [f_a, f_b]$) израчунати су следећи карактеристични параметри:

- укупна ефективна вриједност вибрацијске брзине у радном режиму „I“:

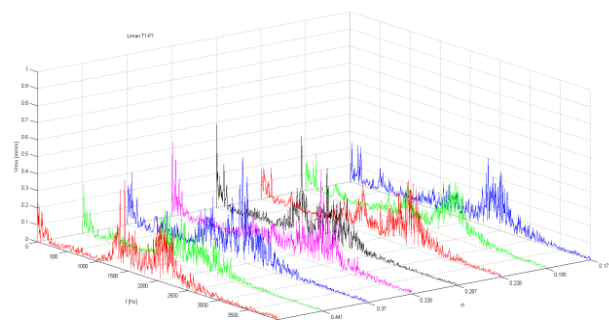
$$V_{rms}^I = \sqrt{\sum_{i=a}^b (v_{rms,i}^I)^2} \quad I = 0, 1 \dots N \quad (4)$$

- нормирана укупна ефективна вриједност вибрацијске брзине у радном режиму „I“:

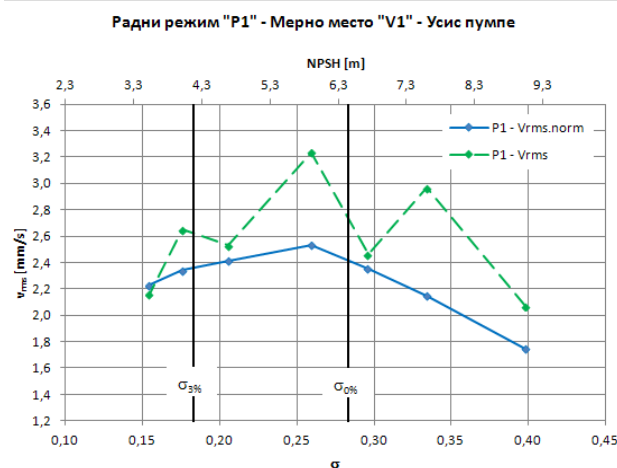
$$V_{rms, norm}^I = \sqrt{\sum_{i=a}^b (v_{rms,i}^I - v_{rms,i}^0)^2} \quad I = 1, 2 \dots N \quad (5)$$

Као крајњи резултат мјерења и обраде резултата мјерења добијени су дијаграми зависности ефективних вриједности вибрацијских брзина V_{rms} и $V_{rms, norm}$ од кавитацијског коефицијента σ (слике 6, 8. и 10).

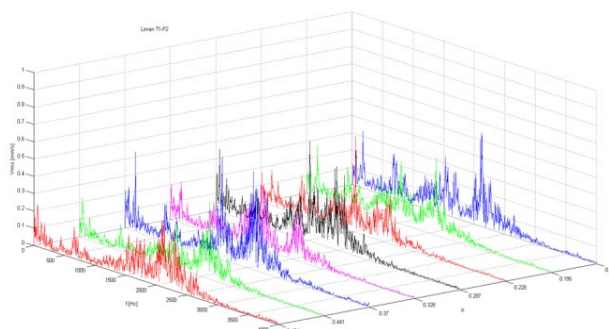
Дијаграми $V_{rms}-\sigma$ и $V_{rms, norm}-\sigma$ омогућавају лако праћење и анализу промјене наведених карактеристичних параметара вибрација са развојем кавитације у пумпи. Другим ријечима ови дијаграми омогућавају сагледавање везе између интензитета вибрација и интензитета кавитације у центрифугалним пумпама. Ово се посебно односи на дијаграм зависности $V_{rms, norm}-\sigma$.



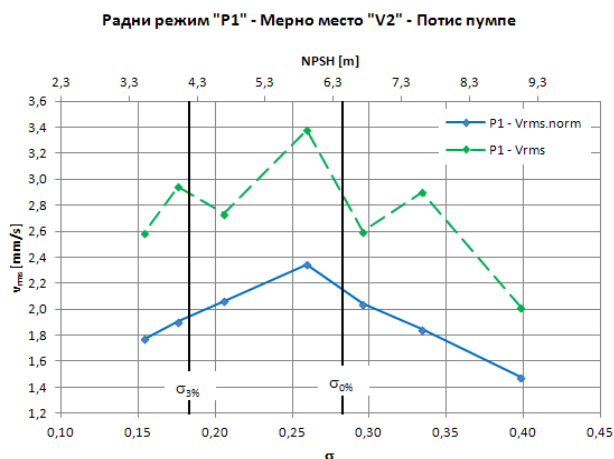
Слика 5. Спектри вибрација измјерени у радном режиму „P1“ на мјерном мјесту V1 – усис пумпе [1]



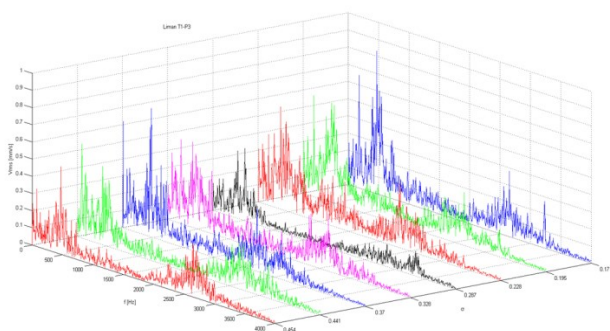
Слика 6. Зависности ефективних вриједности вибрацијских брзина $V_{rms}-\sigma$ и $V_{rms, norm}-\sigma$ у фреквентном опсегу [200 Hz, 4000 Hz]: радни режим „P1“, мјерно мјесто V1 – усис пумпе [1]



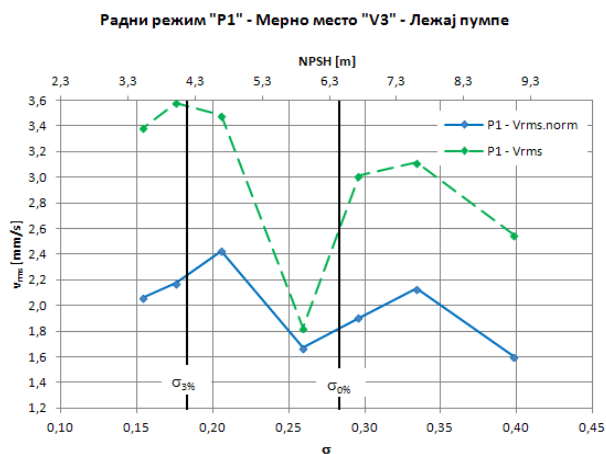
Слика 7. Спектри вибрација измјерени у радном режиму „P1“ на мјерном мјесту V2 – потис пумпе [1]



Слика 8. Зависности ефективних вриједности вибрацијских брзина V_{rms} - σ и $V_{rms.norm}$ - σ у фреквентном опсегу [200 Hz, 4000 Hz]: радни режим „P1“, мјерно мјесто V2 –потис пумпе [1]



Слика 9. Спектри вибрација измјерени у радном режиму „P1“ на мјерном мјесту V3 –лежај пумпе [1]



Слика 10. Зависности ефективних вриједности вибрацијских брзина V_{rms} - σ и $V_{rms.norm}$ - σ у фреквентном опсегу [200 Hz, 4000 Hz]: радни режим „P1“, мјерно мјесто V3 –лежај пумпе [1]

5. ЗАКЉУЧАК

На основу вибродијагностичких мјерења која су вршена на испитиваној пумпи установљено је да пумпа практично стално ради у режиму кавитације. Анализом спектограма приказаних на сликама 5, 7. и 9. може се установити да су у читавом посматраном фреквент-

ном опсегу измјерене интензивне вибрације, на сва три мјерна мјеста V1, V2 и V3. Вибрације су мјерене у фреквентном домену до 5000 Hz.

У свим фреквентним спектрима регистрована су два пика. Први пик на око 25 Hz одговара брзини обртања пумпе, а други пик, који се јавља на око 75 Hz одговара фреквенцији пролаза лопатица радног кола.

У области фреквенција изнад 200 Hz регистроване су интензивне вибрације у широком фреквентном домену које могу бити последица механичких, хидрауличких или кавитацијских феномена.

Посматрајући дијаграме на сликама 6, 8. и 10. примјетно је да са опадањем кавитацијског коефицијента σ , ефективне вриједности вибрацијских брзина расту, а посебно у области између вриједности кавитацијског коефицијента $\sigma_{0\%}$ и $\sigma_{3\%}$.

Интересантно је примјетити да на овим дијаграмима зависности не постоји хоризонтални дио, што указује на то да испитивана пумпа у ПС „Лиман“ стално ради у режиму кавитације. Највјероватније је ријеч о кавитацији ниског интензитета, која у кратком временском року не може да оштети радно коло, али у дужем временском року сигурно може, што се уосталом и догађало током досадашње експлоатације ПС „Лиман“.

На основу дијаграма са слика 3 и 4 може се закључити да пумпа не ради у режиму кавитације са обзиром да је услов бескавитацијског рада ($NPSHA > NPSHR + S$) испуњен. Међутим, на основу вибродијагностичких мјерења недвосмислено је показано да пумпа стално ради у режиму кавитације, истина релативно ниског интензитета. До индиректне потврде овог закључка може се доћи и на основу литературних података датих у табели 1. Наиме, на основу ових података, коефицијент почетка кавитације (за случај да су улазне ивице лопатице специјално обликоване) требало би да износи $\sigma_i = 2 \times \sigma_{3\%} = 2 \times 0,239 = 0,478$, односно $NPSH_i = 10,8$ m док је измјерена расположива кавитацијска резерва у оптималном радном режиму $NPSHA = 9,3$ m (слика 3).

6. ЛИТЕРАТУРА

- [1] Ташин, С. „Оптимални дијагностички пакет параметара за детекцију кавитацијских режима у центрифугалним пумпама“, докторски рад, Нови Сад, Факултет техничких наука, 2016. год.
- [2] Вуковић, Ж., Халкијевић, И. „Кавитација у центрифугалним црпкама“, Грађевинар 63-1, 2011.г.
- [3] Hergt P., Nicklas, A., Mollenkopf, G., Brodersen, S. „The suction performance of centrifugal pumps possibilities and limits of improvements“. Proceedings of the 13th International Pump Users Symposium, Houston, p. 13-25, 1996.

Кратка биографија:



Радивојка Вучинић рођена је у Невесињу 1993. год. Основну и средњу школу завршила је у Невесињу. Мастер рад на Факултету техничких наука из области Машинства одбранила је 2019. год.

VIBRODIJAGNOSTIKA STANJA ELEKTROMOTORA SA OŠTEĆENIM ROTOROM**VIBRATION ANALYSIS OF ELECTRIC MOTOR WITH DAMAGED ROTOR**Siniša Kraljević, Ninoslav Zuber, Dejan Reljić, *Fakultet tehničkih nauka, Novi Sad***Mašinstvo – MEHANIZACIJA I KONSTRUKCIONO MAŠINSTVO**

Kratak sadržaj – Održavanje mašina po stanju, merenje vibracija rotirajućih mašina u cilju određivanja stanja mašine, dijagnostika stanja elektromotornih pogona na bazi merenja vibracija, osnovni pokazatelji oštećenja elektromotornih pogona, eksperiment.

Ključne reči: Vibracije, održavanje po stanju, stanje rotora elektromotora, stanje statora elektromotora

Abstract – Maintenance of machines by condition, vibration measurement on rotating machines in order to determine the machine operating condition, diagnostics of the condition of electric motor drives based on vibration measurement, basic indicators of damage of electric motor drives, experiment.

Keywords: Vibrations, Condition based maintenance, condition of rotor in electric motor, condition of stator of electric motor

1. UVOD

Mera uspešnosti konstrukcije mašine se osim ocene ispunjenja predviđene funkcije ogleda i u dužini njenog životnog veka, pouzdanosti, preciznosti, potrošnji energije i nivou buke i vibracija koje se emituju u toku rada. Stoga je veoma bitno da se u fazi razvoja krajnje verzije tehničke dokumentacije mašine, odnosno kroz ispitivanje prototipova, izvrši lokalizacija svih prisutnih izvora buke i vibracija u toku rada. Nakon identifikacije izvora buke i vibracija, vrši se modifikacija tehničke dokumentacije, modifikacija ili izrada novog prototipa i njegovo ispitivanje [1].

Osim u fazi projektovanja, ispitivanje vibracija se koristi i za održavanje mašina po stanju. Tokom rada mašine, njeni delovi se habaju, čime se menjaju dimenzije delova. Promena dimenzija delova dovodi do neuravnoteženog rada i stvaranja vibracija. Kako bi se izbegle havarije, potrebno je obavljati redovnu vibrodiagnostiku mašina, kako bi se uočile promene koje su nastale kao posledica rada mašine. Detektovanjem uzroka povišenih vibracija pre nastanka kvara, omogućeno je sprečavanje neplanskih zastoja i pratećih posledica. Vibrodiagnostika je metoda ispitivanja bez razaranja. Njenom primenom nije potrebno rastaviti mašinu da bi se uočio kvar i nije potrebno zaustaviti radni proces da bi se obavilo merenje.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Ninoslav Zuber, vanr. prof.

2. POSTUPAK EKSPERIMENTA**2.1. Opis radnog stola**

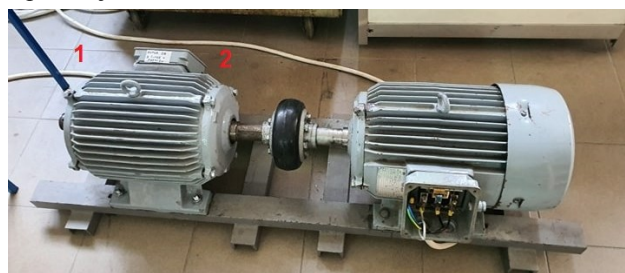
Radni sto je izveden od dva elektromotora, od kojih jedan radi u režimu elektromotora, dok drugi radi u generatorskom režimu i obavlja funkciju kočenja elektromotora. Tokom izvođenja eksperimenta, ispitivani motor je napajan direktno iz mreže. Druga mašina, koja je predstavljala opterećenje, napajana je iz četvorokvadratnog pretvarača (ACS 800), proizvođača ABB. Ova mašina je radila u generatorskom režimu, a regulisana je po momentu. Na taj način je bilo moguće podesiti vrednosti momenta opterećenja ispitivanom motoru.

Prilikom snimanja zaletne karakteristike ispitivanog motora, u cilju snimanja kritičnog broja obrtaja, ispitivani motor je napajan iz energetskog pretvarača (FC 302), proizvođača Danfoss. Zalet neopterećenog ispitivanog motora obavljen je po rampi čije je vreme podešeno na 30 sekundi.

Podaci o motorima:

- Tip: 1.RZK 160 M-4, trofazni asinhroni kavezni motor
- Proizvođač: ATB Sever, Subotica
- Nominalna snaga: 11 kW
- Nominalni napon: 420 V
- Nominalna struja: 22 A
- Nominalna frekvencija: 50 Hz
- Nominalna brzina obrtanja: 1438 min^{-1}
- Sihrona brzina obrtanja: 1500 RPM (min^{-1})
- Broj polova: 4 (četvoropolni motor)
- Broj rotorskih štapnih provodnika: 28
- Broj statorskih utora (žlebova): 36
- Nepogonski i pogonski ležajevi: 6210-2Z/C3 SKF

Elektromotor je sa generatorom spojen elastičnom spojnicom, kako bi se kompezovala nesaosnost vratila, radi česte izmene rotora, radi obavljanja različitih ispitivanja.



Slika 1. Izgled radnog stola

2.2. Primenjena oprema tokom ispitivanja

Veza mernog pretvarača vibracija i objekta ispitivanja je ostvarena korišćenjem jakog dvopolnog magneta prilikom merenja nad svim obavljenim testovima.

Za snimanje signala vibracija i struje motora pri različitim testovima koji su sprovedeni, korišćena je sledeća oprema:

1. Prenosivi četvorokanalni analizator vibracija (OneProd Falcon) u Premium verziji (Slika 2).

Na grafičkom prikazu uređaja, za odabrano merno mesto, vibrodijagnostičar ima spisak svih definisanih vibracionih iskaza. Nakon uspešno završenog snimanja, vibrodijagnostičar povezuje prenosivi analizator vibracija sa računarom i prebacuje podatke u softver.



Slika 2. Analizator vibracija

2. Bežični troaksijalni akcelerometar (OneProd WLS) (Slika 3)

Primena akcelerometra sa šiljatim vrhom je bila neophodna, radi nemogućnosti montaže drugih akcelerometara. Primenom ovog tipa akcelerometra dobijamo najlošiju prenosnu karakteristiku mer og pretvarača. Sopstvena frekvencija je značajno narušena zbog smanjene krutosti i značajno dodate mase na sam senzor - gornja granica linearnog opsega je 500 Hz – 800 Hz. Ovo znači da ispravnim smatramo samo snimljene komponente vibracionog signala do 800 Hz a sve preko toga nije merodavno.



Slika 3. Bežični troaksijalni akcelerometar u radu

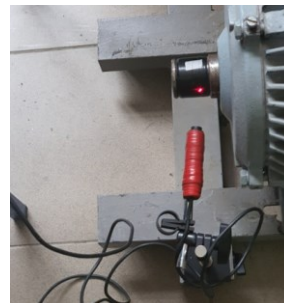
3. Kablovski jednoaksijalni akcelerometri (model: ASH201)



Slika 4. Kablovski jednoaksijalni akcelerometar

4. Softverski paketi: OneProd XPR, OneProd Nest, dbFA Suite i MeScope
5. Prenosivi merač faze (model: SPSR IM)

Prilikom montaže merača faze, potrebno je za vratilo zalepiti reflektujuću traku, koja služi da vrati laserski zrak u merač. Time se dobija zavisnost broja obrtaja [Hz] i vremena zaletanja [s].



Slika 4. Laserski davač faze

6. Strujna klemena (model: A662, Tektronix)



Slika 5. Strujna klemena

3. REZULTATI EKSPERIMENTA

3.1. Vrsta primenjenih ispitivanja

Merno mesto „1“ je kod nepogonskog ležaja, dok je mesto „2“ kod pogonskog ležaja (Slika1). Merno mesto za snimanje struje mora se nalaziti na fazi broj 1 (Slika 4).

Izvršena su sledeća ispitivanja:

1. Snimanje vibracija i struje motora u stacionarnom radu pri sledećim nivoima opterećenja:
 - a) Prazan hod
 - b) 50% opterećenja
 - c) 75% opterećenja
 - d) 100% opterećenja

Za različite tipove merenja definisani su različiti skalarni i vektorski iskazi vibracionih signala.

Snimljeni su sledeći skalarni iskazi vibracionih signala:

1. Ovr1: Vibration velocity – Sumarna brzina vibracija sa RMS normalizacijom amplitude u frekvencijskom opsegu od 2 Hz do 1 kHz, mm/s
2. Ovr1: Acceleration – Sumarno ubrzanje sa RMS normalizacijom amplitude u frekvencijskom opsegu od 2 Hz do 20 kHz, g ($1g=9,81 \text{ m/s}^2$).
3. Ovr1: Absolute displacement – Sumarni iskaz apsolutnih pomaka sa True Peak-Peak normalizacijom amplitude u frekvencijskom opsegu od 2 Hz do 300 Hz, μm
4. DEF: Faktor oštećenosti kotrljajnog ležaja

5. KURT: Kurtosis parametar za ocenu prisustva udarnih komponenti u signalu
6. LF – 2/200 Hz: Sumarno ubrzanje sa RMS normalizacijom amplitude u frekvencijskom opsegu od 2 Hz do 200 Hz, g ($1g=9,81 \text{ m/s}^2$)
7. LF – 200/2000 Hz: Sumarno ubrzanje sa RMS normalizacijom amplitude u frekvencijskom domenu od 200 Hz do 2000 Hz, g ($1g=9,81 \text{ m/s}^2$)
8. LF – 2000/20000 Hz: Sumarno ubrzanje sa RMS normalizacijom amplitude u frekvencijskom domenu od 2000 Hz do 20000 Hz, g ($1g=9,81 \text{ m/s}^2$)
9. Spektralne ekstrakcije: komponenta rotorskih štapnih provodnika – RBF

Snimljeni su sledeći vektorski iskazi vibracionih signala:

1. Frekventni spektar u jedinicama ubrzanja u opsegu od 2 Hz do 200 Hz sa 3200 linija
2. Frekventni spektar u jedinicama ubrzanja u opsegu od 2 Hz do 2000 Hz sa 6400 linija
3. Frekventni spektar u jedinicama ubrzanja u opsegu od 2 Hz do 20000 Hz sa 3200 linija
4. Frekventni spektar u jedinicama brzine u opsegu od 2 Hz do 200 Hz sa 3200 linija
5. Frekventni spektar u jedinicama brzine vibracija u opsegu od 2 Hz do 2000 Hz sa 6400 linija
6. Frekventni spektar u jedinicama brzine vibracija u opsegu od 2 Hz do 20000 Hz sa 3200 linija
7. Frekventni spektar u jedinicama pomaka u opsegu od 2 Hz do 200 Hz sa 3200 linija
8. Frekventni spektar u jedinicama pomaka u opsegu od 2 Hz do 2000 Hz sa 6400 linija
9. Frekventni spektar u jedinicama pomaka u opsegu od 2 Hz do 20000 Hz sa 3200 linija
10. Vremenski zapis ubrzanja snimljen frekvencijom uzrokovanja od 25.6 kHz dužine 512000 tačaka
11. Vremenski zapis brzine vibracija snimljen frekvencijom uzrokovanja od 25.6 kHz dužine 512000 tačaka
12. Spektri envelope u opsegu od 2 Hz do 200 Hz sa 3200 linija

Za snimanje struje snimljeni su sledeći vektorski iskazi vibracionih signala:

1. Vremenski zapis ubrzanja snimljen frekvencijom uzrokovanja od 5.12 kHz dužine 512000 tačaka

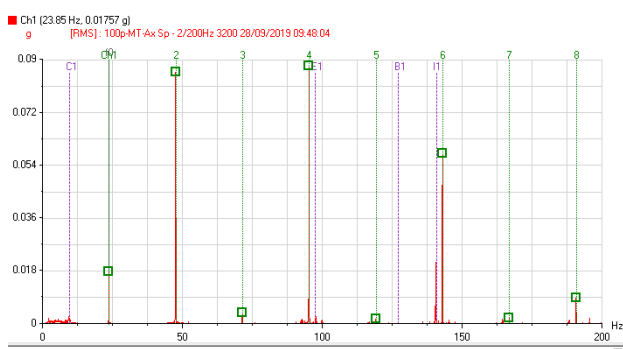
3.2. Dijagramski prikaz rezultata eksperimenta

Primenjene oznake na dijagramima:

- f_0 – marker broja obrtaja
- E1 – marker oštećenja ležaja na spoljašnjoj stazi
- I1 – marker oštećenja ležaja na unutrašnjoj stazi
- B1 – marker oštećenja ležaja na kotrljajnom telu
- C1 – marker oštećenja kaveza ležaja
- RBF – komponenta rotorskih štapnih provodnika

Ležajevi motora su blago oštećeni. U spektrima vibracija se vide komponente oštećenja unutrašnje staze. U

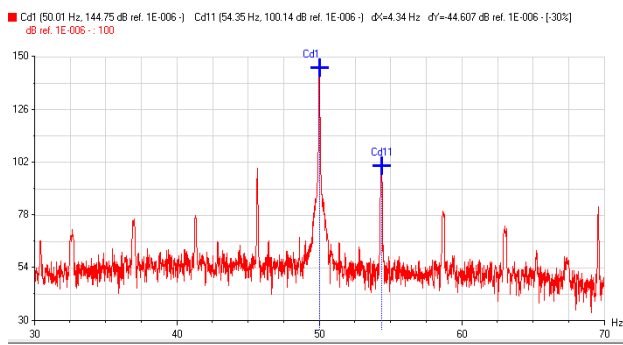
frekventnom spektru vibracija se vidi oštećenje ležaja na unutrašnjoj stazi I1.



Dijagram 1. Frekventni spektar, Sp 2/200 Hz na mestu pogonskog ležaja u aksijalnom pravcu, [g], pri 100% opterećenja

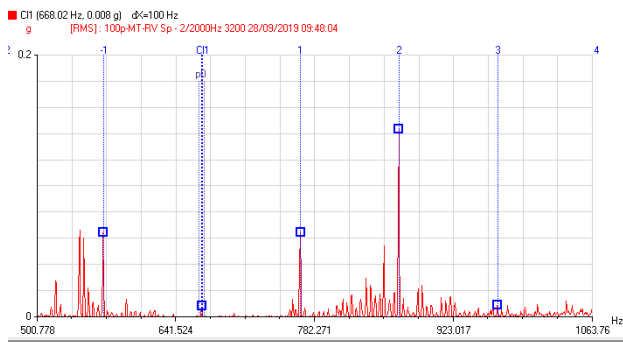
Rotor poseduje oštećenje štapnih provodnika, što se jasno vidi na spektrima struje, gde je izmereno 44.67 dB relativne razlike amplitude između mrežne frekvencije i bočnog pojasa komponente prolaska između polova (Dijagram 2).

Kako i očekivano, najmanja relativna razlika je izmerena pri najvećem opterećenju motora.



Dijagram 2. Spektr struje na 100 % opterećenja, [dB]

Ujedno i u spektrima vibracija vidi oštećenje rotora i kroz postojanje amplitude modulacije dvostruke mrežne frekvencije (100 Hz), oko komponente rotorskih štapnih provodnika ($RBF=28xf_0$) – Dijagram 3.



Dijagram 3. Frekventni spektar, Sp 2/2000 Hz na mestu pogonskog ležaja u vertikalnom pravcu, zumiran oko RBF, [g], pri 100% opterećenja

4. ZAKLJUČAK

Ležajevi motora su blago oštećeni, tako da se preporučuje češća kontrola ležajeva. Trenutno, zamena ležajeva nije potrebna, samo češće praćenje stanja.

Iz analize spektra struje je izmereno 44.67 dB relativne razlike amplitude između mrežne frekvencije i bočnog pojasa komponente prolaska između polova. Kada se ovaj podatak uporedi sa preporukama, zaključuje se da postoji veliki otpor u spojnim prstenovima ili da su napukli rotorski štapni provodnici – rotor motora je potrebno remontovati.

5. LITERATURA

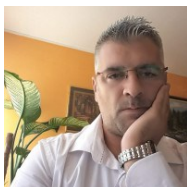
- [1] N. Zuber, Vibrodijagnostika rotirajućih mašina, Fakultet tehničkih nauka, Novi Sad, 2014.
- [2] D. Reljić, Otkrivanje kvarova rotora kaveznog asinhronog motora primenom tehnika analize terminalnih veličina, Doktorska disertacija, Fakultet tehničkih nauka, Novi Sad, 2017.

Kratka biografija:



Siniša Kraljević je student Master studija na Fakultetu tehničkih nauka u Novom Sadu na Katedri za mašinske konstrukcije, transportne sisteme i logistiku. Diplomski rad je odbranio 2018. godine na temu: „Vibracije mostne dizalice“. Oblasti interesovanja su teorijska i primenjena mehanika, mašinske vibracije i nelinearne oscilacije.

kontakt: sinisakraljevic92@gmail.com



dr Ninoslav Zuber je vanredni profesor na Fakultetu tehničkih nauka u Novom Sadu i šef Katedre za mašinske konstrukcije, transportne sisteme i logistiku. Sertifikovani je instruktor i član tehničkog komiteta Mobius instituta za obuku vibrodijagnostičara. Oblasti interesovanja su metode vezane za metode eksperimentalne mehanike i vibrodijagnostička i akustička ispitivanja rotacione opreme.

kontakt: zuber@uns.ac.rs



dr Dejan Reljić je docent na Fakultetu tehničkih nauka u Novom Sadu. Oblasti interesovanja su elektromotorni pogoni i detekcija i identifikacija kvarova električnih mašina.

kontakt: reljic@uns.ac.rs

**IDENTIFIKACIJA PARAMETARA EKVIVALENTNE ŠEME ASINHRONE MAŠINE
METODOM DVA TEST SIGNALA****INDUCTION MOTOR PARAMETER ESTIMATION TECHNIQUE BASED ON TWO
TEST SIGNALS**

Mladen Vučković, Vladimir Popović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je prikazana metoda bazirana na injektiranju dva naponska test signala različitih učestanosti tokom self-commissioning procesa asinhronog motora. Za potrebe verifikacije razvijen je Matlab/Simulink model. Rezultati simulacije poređeni su sa stvarnim parametrima motora koji je namenjen za potrebe u okviru električne vuče.

Ključne reči: parametarska identifikacija, asinhrona mašina, ekvivalentne šeme AM.

Abstract – In this paper the method based on injection of two voltage signals of different frequencies during the self-commissioning process is presented. For the verification purposes Matlab/Simulink model is derived. The results of simulation were compared with ideal motor parameters which is used for traction drive applications.

Keywords: Parameter estimation, induction machine, IM equivalent circuit.

1. UVOD

Električna energija predstavlja vid energije koji se relativno lako proizvodi, konvertuje u druge oblike energije i troši na koristan rad. Većina nastale električne energije konvertuje se u mehaničku energiju posredstvom električnih motora. Asinhroni motori zbog svoje visoke robusnosti i pouzdanosti predstavljaju glavnu pokretačku snagu današnje industrije, ali se mogu naći i na drugim mestima kao što su transportni sistemi i obična domaćinstva. Prema nekim procenama asinhroni motori predstavljaju 50 % potrošnje ukupne proizvedene električne energije i čak 90 % električne energije svih elektromotornih pogona na svetu. Kako bi se postigle visoke performanse kontrole i energetska efikasnost pogona ključno je poznavanje parametara ekvivalentne šeme asinhronog motora. Iako se parametri mogu dobiti od proizvođača motora, oni su podložni varijacijama usled samog postupka izrade mašine, temperature ambijenta ili drugih spoljašnjih uticaja. U radu je za potrebe identifikacije parametara istražena jedna *self-commissioning* metoda bazirana na injekciji dva naponska sinusoidna test signala različitih učestanosti.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Veran Vasić, red. prof.

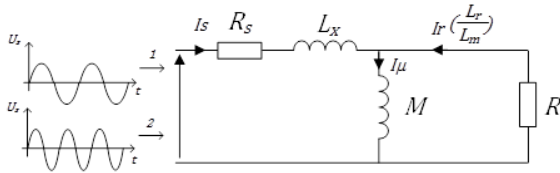
Metoda identifikacije parametara ekvivalentne šeme asinhronne mašine sa dva test signala se zasniva na inverznoj „G“ ekvivalentnoj šemi. Ova metoda određuje gotovo sve parametre ekvivalentne šeme asinhronne mašine osim omske otpornosti namotaja statora koja mora biti unapred poznata. Cilj ovog rada je da, postepeno, planski i praktično prikaže proces razvoja, implementacije i verifikacije metode upotrebom Matlab/Simulink programskog paketa.

2. OSNOVNI OPIS METODE

Realizacija same metode zasniva se na injektiranju dva naponska test signala sa statorske strane asinhronne mašine u cilju određivanja njenih parametara. Napajanje tokom testa mora biti jednofazno kako se u mašini ne bi razvilo obrtno polje koje bi prouzrokovalo razvijanje elektromagnetnog momenta i pokretanje mašine. Ako je mašina napajana monofaznim naizmeničnim test signalom u njoj se razvija samo pulzaciono magnetno polje koje se po Leblanovoj teoremi može rastaviti na dva obrtna magnetna polja suprotnih smerova. Pošto se dejstva ova dva polja poništavaju vratilo mašine ostaje stacionarno ili se vrlo malo pomera u odnosu na neku referentnu osu usled nesavršenosti mašine. Kako bi se uspešno koristila ova metoda za proračun parametara upotrebljava inverznu „G“ ekvivalentnu šemu koja je prikazana na slici 1. Razlog za korišćenje baš ove ekvivalentne šeme je zbog njene jednostavnosti i zato što se relativno lako njeni parametri prebacuju na „T“ ekvivalentnu šemu koja se koristi u većini primena i proračuna. Parametri koji se određuju sa ovom test metodom su sledeći:

- Otpornost rotora svedenu na statorsku stranu: $R = \left(\frac{L_m}{L_r}\right)^2 R_r$
- Induktivnost magnećenja: $M = (1 - \sigma)L_s$
- Rasipne induktivnosti statora i rotora svedene na stranu statora: $L_x = \sigma L_s$

Upotrebljavajući inverznu „G“ ekvivalentnu šemu električnu mašinu možemo predstaviti njenim zamenskim kolom. Ulazna impedansa zamenskog kola ima svoj realan i imaginaran deo. Ova impedansa zavisi od učestanosti test signala kojim se napaja mašina. Opšta predstava metode identifikacije parametara ekvivalentne šeme asinhronne mašine sa dva test signala na određenoj učestanosti prikazana je na slici 1.



Slika 1. Opšta predstava metode na osnovu inverzne „G“ ekvivalentne šeme.

Kompleksna impedansa Z može se dobiti kao količnik fazora napona na namotaju motora $\bar{U}_s$ i fazora struje koja protiče kroz dati namotaj $\bar{I}_s$, gde φ_1 predstavlja fazni pomeraj struje u odnosu na napon pri određenoj učestanosti test signala ω_1 kao što je i prikazano jednačinom (1).

$$Z(\omega_1) = R_\Gamma + jX_\Gamma = \frac{\bar{U}_s}{\bar{I}_s} = \frac{|\bar{U}_s| \angle 0}{|\bar{I}_s| \angle \varphi_1} \quad (1)$$

Ne treba izostaviti iz razmatranja vrednost učestanosti naponskog test signala. Ona treba biti reda veličine nazivnog apsolutnog klizanja asinhronne mašine. Kako bi tokom testa površinski efekat u namotaju rotora asinhronne mašine mogao biti zanemaren, ovu preporuku treba poštovati. Ulazna impedansa $Z(\omega_1)$ za zamensko kola sa slike 1 može se predstaviti na sledeći način:

$$\begin{aligned} Z(\omega_1) &= R_s + j\omega_1 L_x + \frac{R(j\omega_1 M)}{R + j\omega_1 M} \\ &= R_s + j\omega_1 L_x + \frac{R(\omega_1 M)^2 + jR^2 \omega_1 M}{R^2 + (\omega_1 M)^2} \end{aligned} \quad (2)$$

Pri čemu su realni i imaginarni deo izraženi kao:

$$R_\Gamma = R_s + \frac{R(\omega_1 M)^2}{R^2 + (\omega_1 M)^2}; \quad (3)$$

$$X_\Gamma = \omega_1 L_x + \frac{R^2 \omega_1 M}{R^2 + (\omega_1 M)^2} \quad (4)$$

Od realnog dela treba oduzeti već izmerenu vrednost omske otpornosti namotaja statora R_s dok imaginarni deo ne treba posebno modifikovati. Na osnovu prethodnog dobijamo sledeće jednačine:

$$R_{Tot1} = \frac{R(\omega_1 M)^2}{R^2 + (\omega_1 M)^2}; \quad (5)$$

$$X_{Tot1} = \omega_1 L_x + \frac{R^2 \omega_1 M}{R^2 + (\omega_1 M)^2} \quad (6)$$

Jednačine (5) i (6) predstavljaju realni i imaginarni deo impedanse od interesa za metodu na određenoj test učestanosti (ω_1) koja je poznata veličina. Jednačine (5) i (6) predstavljaju sistem jednačina sa tri nepoznate veličine – R , M , L_x .

Pošto takav sistem jednačina nema jednoznačno rešenje potrebne su dodatne jednačine. Do njih se dolazi injektovanjem drugog naponskog test signala, iste amplitude kao u prvom slučaju i drugačije učestanosti (ω_2). Novom test signalu odgovara druga ulazna impedansa čiji su realni i imaginarni deo:

$$R_{Tot2} = \frac{R(\omega_2 M)^2}{R^2 + (\omega_2 M)^2}; \quad (7)$$

$$X_{Tot2} = \omega_2 L_x + \frac{R^2 \omega_2 M}{R^2 + (\omega_2 M)^2} \quad (8)$$

Uvođenjem drugog test signala sistem jednačina postaje predimenzionisan. Za tri nepoznata parametara na raspolaganju su nam ukupno četiri jednačine (5) do (8).

Nelinearni sistem jednačina (5)–(8) ima analitičko rešenje kombinovanjem (5), (7) i (8):

$$R = \frac{R_{Tot1} R_{Tot2} (\omega_2^2 - \omega_1^2)}{R_{Tot1} \omega_2^2 - R_{Tot2} \omega_1^2}; \quad (9)$$

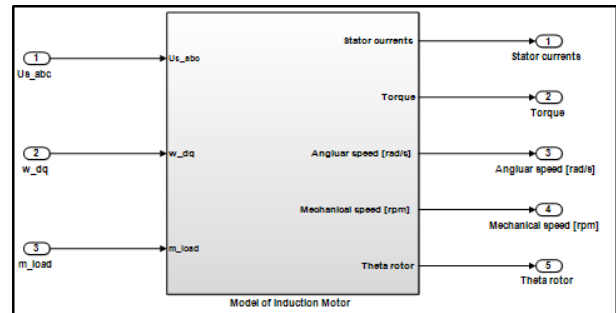
$$M = \frac{R_{Tot1} R_{Tot2} (\omega_2^2 - \omega_1^2)}{\omega_1^2 \omega_2^2 (R_{Tot1} \omega_2^2 - R_{Tot2} \omega_1^2)} \sqrt{\frac{R_{Tot1} \omega_2^2 - R_{Tot2} \omega_1^2}{R_{Tot1} - R_{Tot2}}} \quad (10)$$

$$L_x = \frac{X_{Tot2}}{\omega_2^2} + \frac{R^2 M}{R^2 + (\omega_2 M)^2} \quad (11)$$

Metoda identifikacije parametara ekvivalentne šeme asinhronne mašine sa dva test signala predstavlja jedinstven i direktan pristup za proračunavanje svih parametara ekvivalentne šeme asinhronne mašine osim omske otpornosti statora. U daljem delu rada biće obrađeni rezultati simulacija sa osvrtima na detalje i probleme tokom realizacije istih.

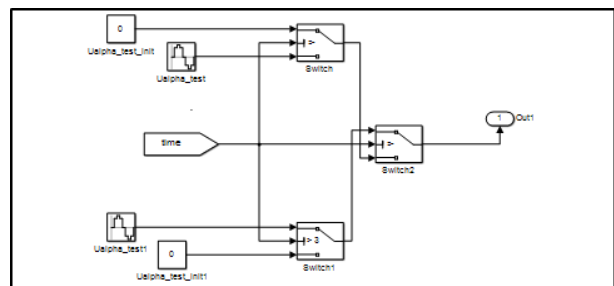
3. SIMULINK MODEL RAZVIJEN ZA POTREBE VERIFIKACIJE METODE

Za potrebe prvobitne provere metode kreiran je model asinhronne mašine tipa YDQ-6730 proizvođača Kinetek u Matlab/Simulink razvojnom okruženju na osnovu jednačina matematičkog modela mašine. Centralni podsistem modela prikazan na slici 2 predstavlja realizovani model asinhronne mašine. Osnovni ulazi u model su sistem trofaznih napona U_{s_abc} i momenat opterećenja m_load , dok je treći ulaz w_dq uveden radi mogućnosti implementacije vektorske kontrole.



Slika 2. Model asinhronne mašine u programskom alatu Matlab/Simulink.

Na samom početku ove metode motor se pobuđuje odgovarajućim test signalima. Unutrašnjost podsistema koji služi za dovođenje odgovarajućeg test signala na ulaz modulatora prostornog vektora data je na slici 3.



Slika 3. Podsistem koji služi za generisanje sinusnih test signala.

Kako bi se realizovala jednačina (1), odnosno dobila impedansa na test učestanosti mora se preći u dq koordinatni sistem. Takvim prelaskom jednačine (3) i (4) postaju:

$$R_r = \frac{U_{sd}I_{sd} + U_{sq}I_{sq}}{I_{sd}^2 + I_{sq}^2}; \quad (12)$$

$$X_r = \frac{U_{sq}I_{sd} - U_{sd}I_{sq}}{I_{sd}^2 + I_{sq}^2} \quad (13)$$

Ovakvu jednostavnu realizaciju proračuna na osnovu jednačina (12) i (13) narušava jedna bitna činjenica, a to je da u slučaju jednofaznog napajanja imamo nesimetričan režim tj. veličine u dq domenu nisu jednosmerne. Pošto metoda zahteva monofazno napajanje kao što je gore navedeno dobijamo pulzaciono magnetno polje koje se može interpretirati kao dva obrtna polja suprotnih smerova obrtanja.

Prilikom *Park* transformacije koordinatni sistem vezujemo za jedno obrtno polje, dok drugo obrtno polje stvara komponente na dvostrukoj učestanosti koje se javljaju u dq domenu. Da bi realizovali proračun veličine u dq domenu moraju biti jednosmerne, odnosno samo tako ih možemo predstavljati odgovarajućim reprezentom u vidu kompleksnog broja.

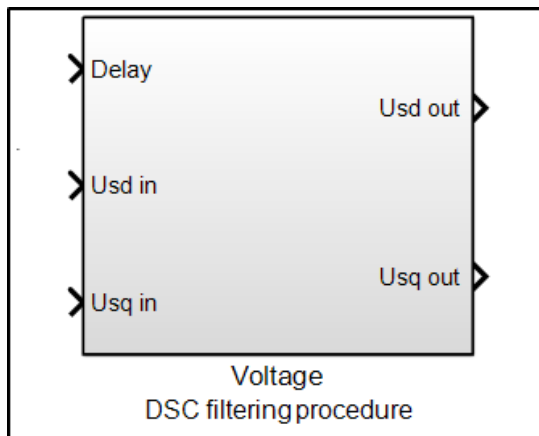
Rešenje koje je prihvaćeno za potrebe filtriranja napona i struja u dq domenu je DSC (engl. *Delayed Signal Cancellation*) filter. Osnovni način funkcionisanja DSC filtra je sabiranje originalnog signala koji sadrži nepoželjnu harmonijsku komponentu i istog originalnog signala koji je fazno smaknut (zakašnjen).

Ovim odgovarajućim pomeranjem dobija se poništavanje nepoželjne harmonijske komponente. Nakon ove operacije dobijeni signal treba podeliti sa dva zbog prethodnog sabiranja.

U simulacionom modelu dva DSC filtera vezana su kaskadno na red, i smeštena su u podsistem prikazanom na slici 4.

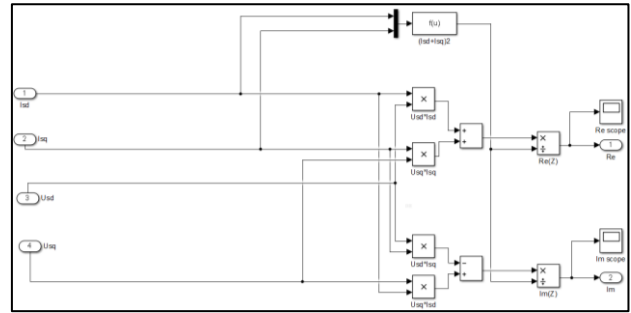
Prvi DSC filter služi za suzbijanje komponente koja potiče od inverznog obrtnog polja, a drugi za suzbijanje komponente koja potiče od jednosmernog *offset*-a.

Jednosmerni *offset* se koristi za potrebe eksperimenta kako bi se poništio uticaj nelinearnosti invertora, a koji je zbog verodostojnosti simulacije i ovde realizovan.



Slika 4. Podsistem za realizaciju DSC filtera za napone u dq domenu.

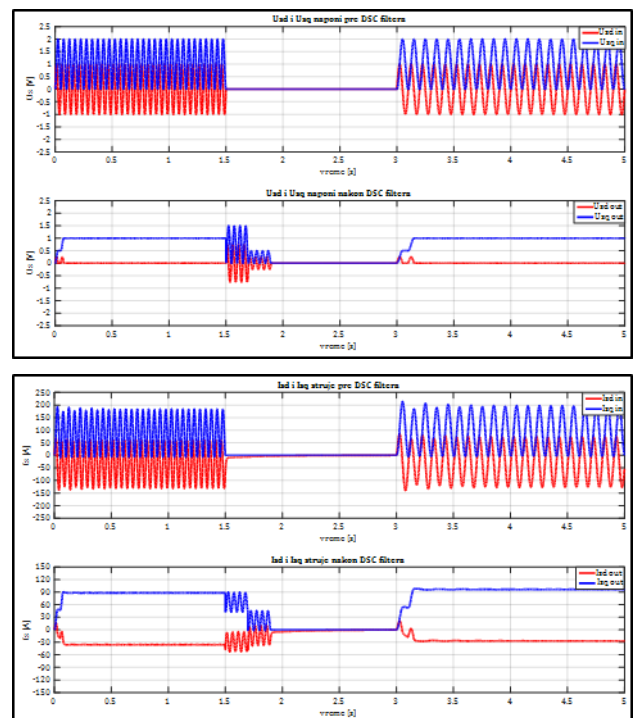
Nakon filtriranja napona i struja u dq domenu, može se realizovati proračun prema jednačinama (12) i (13). Unutrašnjost podsistema za računanje realnog i imaginarnog dela impedanse data je na slici 5.



Slika 5. Podsistem za proračun realnog i imaginarnog dela impedanse.

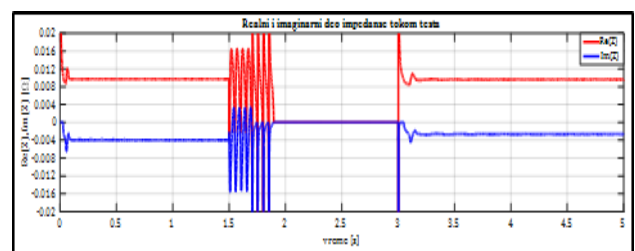
4. REZULTATI SIMULACIJA

Prikazani rezultati nastali su na osnovu simulacija sprovedenih na test učestanostima 10 Hz i 5 Hz. Na slici 6 prikazani su naponi i struje pre i posle filtriranja DSC filterom.



Slika 6. Naponi i struje u dq domenu.

Realizovanjem jednačina (12) i (13) koje izračunavaju realni i imaginarni deo impedanse dobijaju se vremenske zavisnosti ovih delova impedanse koje su prikazane na slici 7.



Slika 7. Realni i imaginarni deo impedanse tokom simulacije.

Za potrebe simulacije metode napravljena je .m skripta koja na osnovu vrednosti realnog i imaginarnog dela impedanse u određenim trenucima računa sve parametre

ekvivalentne šeme osim omskog otpora namotaja statora koji mora biti unapred poznat.

```
% Proračun parametara Rr, M i Lx

Re1= Re(22188); % Realni deo impedanse u tenutku 1,42 s
Re2= Re(75756); % Realni deo impedanse u tenutku 4,84 s
Rtot1=Re1-Rs;
Rtot2=Re2-Rs;
Xtot1=-Im(22188); % Imaginarni deo impedanse u tenutku 1,42 s
Xtot2=-Im(75756); % Imaginarni deo impedanse u tenutku 4,84 s

Rr=(Rtot1*Rtot2*((wtest2*wtest2)-(wtest*wtest)))/(Rtot1*(wtest2*wtest2)-Rtot2*(wtest*wtest))

M=((Rtot1*Rtot2*((wtest2*wtest2)-(wtest*wtest)))/(wtest*wtest2*(Rtot1*(wtest2*wtest2)-Rtot2*(wtest*wtest)))*sqrt((Rtot1*(wtest2*wtest2)-Rtot2*(wtest*wtest))/(Rtot2-Rtot1))

Lx=(Xtot2/wtest2)-(R*R)/(R*(R)+(wtest2*M)*(wtest2*M))
```

Slika 8. Matlabova .m skripta za proračun parametara.

Nakon više izvršenih simulacija sa različitim učestanostima test signala proračunati su parametri ekvivalentne šeme i sačinjena je tabela sa rezultatima (tabela 1). Svi testovi rađeni su sa test signalima iste amplitude koja iznosi 2 V. Nakon izračunavanja parametara u poslednje tri kolone date su relativne greške proračunatih parametara izražene u procentima. Relativne greške računane su u odnosu na „stvarne“ parametre asinhronog motora YDQ-6730 koji su dobijeni od samog proizvođača (Kinetek).

Tabela 1. Rezultati simulacija sa dva test signala iste amplitude

RAĐENE SIMULACIJE SA DVA TEST SIGNALA ISTE AMPLITUDE										
ω_1 [rad/s]	ω_2 [rad/s]	U_{1m} [V]	U_{2m} [V]	R_r [Ω]	M [H]	L_x [H]	G_r [%]	G_m [%]	G_{Lx} [%]	
10	15	2	2	0.005194	0.000915	0.000055	-2.922582	0.195449	-0.771576	
10	20	2	2	0.005184	0.000917	0.000055	-3.115299	0.406032	0.244523	
15	20	2	2	0.005187	0.000913	0.000055	-3.055571	0.067025	-0.241192	
20	25	2	2	0.005179	0.000924	0.000056	-3.205453	1.251504	0.874623	
20	30	2	2	0.005185	0.000919	0.000055	-3.103289	0.631393	0.399490	
20	40	2	2	0.005189	0.000914	0.000055	-3.020665	0.140081	0.129503	
20	50	2	2	0.005186	0.000918	0.000055	-3.083667	0.513899	-0.090498	
30	50	2	2	0.005186	0.000917	0.000055	-3.081298	0.420053	-0.108764	
40	50	2	2	0.005194	0.000863	0.000054	-2.919492	-5.462342	-1.453220	
50	60	2	2	0.005176	0.001037	0.000056	-3.268107	13.642395	1.629765	
10	50	2	2	0.005186	0.000916	0.000055	-3.079970	0.367172	-0.119815	

5. ZAKLJUČAK

Savremene metode vektorskog upravljanja temelje se na poznavanju parametara ekvivalentne šeme asinhronne mašine. Za potrebe preciznog, robusnog i energetski efikasnog upravljanja od presudnog značaja je poznavanje parametara ekvivalentne šeme.

U radu je za potrebe identifikacije parametara istražena jedna *self-commissioning* metoda bazirana na injekciji dva test signala.

Metoda se zasniva na inverznoj „G“ ekvivalentnoj šemi i izuzetno je primenljiva obzirom da određuje gotovo sve parametre ekvivalentne šeme asinhronne mašine. U radu je metoda proverena i verifikovana na osnovu simulacija u programskom paketu Matlab/Simulink. Na osnovu rezultata simulacija može se zaključiti da Simulink model verodostojno oslikava pojave u mašini tokom testa i da na osnovu proračuna daje parametre ekvivalentne šeme asinhronne mašine sa zadovoljavajućom tačnošću. Ako se na dobijene rezultate testova primene dodatni napredniji matematički algoritmi za obradu podataka može se značajno uvećati preciznost prikazane merne metode.

6. LITERATURA

- [1] Marčetić P. Darko, „Mikroprocesorsko upravljanje energetskim pretvaračima“, Novi Sad: Fakultet tehničkih nauka, 2014.
- [2] Shafiq Ahmed Odhano, „Self-Commissioning of AC Motor Drives“, PhD thesis, Torino: Politecnico di Torino, 2014.
- [3] Francisco A. S. Neves, „Digital Filters for Fast Harmonic Sequence Component Separation of Unbalanced and Distorted Three-Phase Signals“, Časopis IEEE Transactions on industrial electronics, 2012.

Kratka biografija:



Mladen Vučković rođen je u Vršcu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2019.god. Od 2018. god. zaposlen je na Fakultetu tehničkih nauka u zvanju saradnika u nastavi.

kontakt: mladen.vuckovic@uns.ac.rs

**PRIMENA SPRING CLOUD PROGRAMSKOG OKVIRA ZA RAZVOJ
MIKROSERVISNIH APLIKACIJA****USE OF SPRING CLOUD FRAMEWORK FOR DEVELOPING MICROSERVICE-BASED
APPLICATIONS**Stefan Bratić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Prikaz osnovnih prednosti i mana mikroservisne arhitekture, kao i konkretna primena Spring Cloud radne biblioteke uz pomoć Jhipster generatora na konkretnom projektu.

Ključne reči: *Microservisi, Jhipster, Spring Cloud, Kubernetes, Docker*

Abstract – *Representation of basic advantages and flaws in the microservice architecture, as well as a concrete implementation through use of Spring Cloud Framework and Jhipster generator.*

Keywords: *Microservices, Jhipster, Spring Cloud, Kubernetes, Docker*

1. UVOD

Proces pravljenja softverskih rešenja se sastoji iz niza procesa, koji su neophodni kako bi konačno rešenje ispunilo sve zahteve klijenta. U pomenute procese spadaju planiranje, razvoj, testiranje i isporuka softverskog rešenja.

Sastavni deo procesa planiranja čini odabir softverske arhitekture. Razlog tome je, što svaka softverska arhitektura predstavlja stub izgradnje softvera i utiče na to kako će se odvijati proces razvoja softvera.

Da bi se moglo znati koja je softverska arhitektura najpogodnija, potrebno je znati karakteristike date arhitekture.

U skladu sa tim, glavni fokus ovog rada će biti mikroservisna arhitektura, koja danas ima sve veću primenu.

Teme koje će biti obrađene u radu su sledeće:

- osnovne karakteristike mikroservisne arhitekture
- razlike između mikroservisne i monolitne arhitekture
- osnovne komponente mikroservisne arhitekture
- primena *Spring Cloud* radnog okvira za razvoj mikroservisnih aplikacija

Cilj rada predstavlja detaljan prikaz mikroservisne arhitekture upotrebom postojećeg radnog okvira. Takođe, uradiće se analiza u kojim slučajevima je mikroservisna arhitektura dobro rešenje za razvoj softvera.

Kompletna analiza će biti prikazana kroz konkretno softversko rešenje.

1.1 Mikroservisna arhitektura

Mikroservisnu arhitekturu čini skup aplikacija zvanih mikroservisi. Mikroservisi su autonomni i izolovani entiteti, čiji je cilj izvršavanje dela funkcionalnosti kompletnog sistema i skladno komuniciranje sa ostalim učenicima u datom sistemu [1].

Glavne odlike mikroservisne arhitekture su:

- autonomnost
- tehnološka heterogenost
- robustnost
- skalabilnost
- pojednostavljen deployment postupak
- organizacijsko poravnanje
- kompozitna struktura

1.2 Autonomnost

Autonomnost se ogleda u tome što je svaki mikroservis zaseban entitet, koji se posebno *deploy-uje* na platformu ili izvršava kao poseban računarski proces izolovan od ostalih učesnika u sistemu. Svaki vid komunikacije se vrši isključivo preko računarske mreže i ne postoji nikakva sprega na nivou samog koda [1].

1.3 Tehnološka heterogenost

Pošto su mikroservisi definisani kao autonomne i izolovane jedinice, samim tim, programski kod, koji se koristi za razvijanje određenih delova sistema, je odvojen. To daje mogućnost da se delovi sistema razvijaju u različitim programskim jezicima, tehnologijama i da se koriste različiti izvori perzistencije [1].

1.4 Robustnost

Pošto je prethodno ustanovljeno da mikroservisi funkcionišu kao autonomni i izolovani entiteti, oni u skladu sa tim po istim principima reaguju na sistemske greške.

Robustnost mikroservisne arhitekture se ogleda u tome što ne postoji propagacija sistemske greške jednog mikroservisa na ceo sistem. Samim tim, sistem je otporniji na greške u pojedinim delovima sistema, jer je u mogućnosti da i pored greške funkcioniše dalje i usput vrši oporavak delova sistema [1].

1.5 Skalabilnost

Svaki mikroservis čini jedan proces, preodređen da izvršava jednu funkcionalnost. Ograničavanje jedne funkcionalnosti po mikroservisu omogućava bolje

NAPOMENA:

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

raspoređivanje hardverskih resursa, zato što je omogućeno skaliranje mikroservisa, čija funkcionalnost u toku upotrebe zahteva veće resurse da bi opslužila korisnike [1].

1.6 Pojednostavljen *deployment* postupak

Svaka mikroservisna aplikacija se *deploy-uje* zasebno i promena koda jedne mikroservisne aplikacije ne zahteva lansiranje celokupnog sistema, već samo dela, kojem pripada promenjeni kod.

Pojednostavljen deployment postupak se ogleda u tome što se promene mogu lakše ispratiti i teže mogu izazvati grešku kod ostalih delova sistema [1].

1.7 Organizacijsko poravnanje

Izolovanost mikroservisnih aplikacija daje prostora da se određeni timovi ili članovi tima fokusiraju na samo određen mikroservis ili skup mikroservisa. Ovakav pristup je u skladu sa *Conway's law*, koji glasi: **"Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure"**.

Prevedeno znači da svaka organizacija, koja definiše neki sistem, će definisati svoju strukturu na osnovu organizaciono-komunikacione strukture koju sama koristi.

To se pokazuje kao dobra praksa, zato što su timovi odgovorni samo za određeni deo koda celog sistema i spram toga moguće je lakše održavati, razvijati sistem i komunicirati sa ostalim timovima u vezi sa integracijom delova sistema [1].

1.8 Kompozitna struktura

Svaka mikroservisna aplikacija čini jednu komponentu sistema sa određenim funkcionalnostima. Mikroervisna arhitektura omogućava da se skup komponenti orkestrirano koristi u cilju izvršavanja složenih procesa [1].

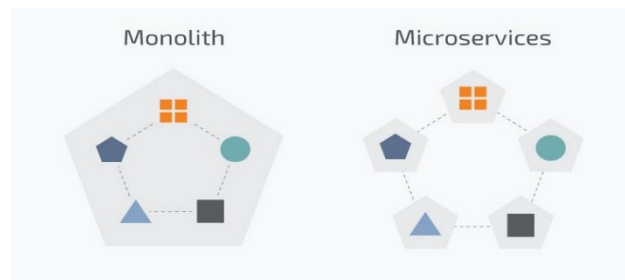
2. KOMPARATIVNI PRIKAZ MIKROSERVISNE I MONOLITNE ARHITEKTURE

Mikroservisna arhitektura sa sobom nosi mnoge prednosti, a takođe i izazove. Ona ne predstavlja rešenje za svaki mogući slučaj. U skladu sa tim, da bi se sagledale činjenice u kojim situacijama je mikroservisna arhitektura pogodna kao rešenje, uradiće se komparativni prikaz mikroservisne i monolitne arhitekture.

Monolitna arhitektura, u odnosu na mikroservisnu arhitekturu, predstavlja jednostavniji način razvijanja softvera, jer sistem nije distribuiran i celokupan kod je u jednoj tehnologiji i programskom jeziku. Međutim, kod aplikacija, koje izlaze van okvira jednog razvojnog tima, je teže održavati stabilnim ako nad tim okvirom koda treba da radi veliki broj ljudi.

Takođe, navedeni način struktuiranja aplikacije može se primenjivati na nivou mikroservisa, jedina razlika je što postoji granulacija funkcionalnosti po entitetima.

Slika 1 daje vizuelni prikaz razlike između monolitne arhitekture, koja sadrži sve funkcionalnosti u okviru jednog entiteta, i mikroservisne arhitekture, gde se svaka funkcionalnost nalazi izolovano u zasebnom entitetu.



Slika 1: Prikaz razlike mikroservisne u odnosu na monolitnu arhitekturu

3. SPRING RADNI OKVIR

Spring predstavlja radni okvir napisan u *Java* programskom jeziku i razvijen je od strane firme Pivotal. U početku je *Spring* predstavljao biblioteku za uvođenje *Dependency Injection Pattern-a* u softver, a kasnije se namena biblioteke proširila na širok dijapazon problema, koji se tiču razvoja *Enterprise* softvera.

3.1. Spring Cloud

Spring Cloud, kao deo *Spring* radnog okvira, pretežno je namenjen za razvijanje distribuiranih sistema, gde se pojavljuju problemi poput: upravljanja konfiguracijom i distribuiranim sesijama, otkrivanja usluga, dinamičkog rutiranja itd.

Pošto u okviru *Spring Cloud* radnog okvira postoji veliki broj rešenja, prikazaće se oni delovi radnog okvira, koji rešavaju probleme navedenih *Design Pattern-a* u mikroservisnom okruženju.

Ti delovi radnog okvira su:

- Spring Cloud Config
- Spring Cloud Netflix
- Spring Cloud Gateway
- Spring Cloud Security
- Spring Cloud Sleuth

3.2. Spring Cloud Config

Spring Cloud Config, koji implementira *Central Configuration Pattern*, nudi klijentsku i serversku podršku za eksternu konfiguraciju u distribuiranim sistemima, gde se sa konfiguracionim serverom uvodi centralno mesto za upravljanje eksternom konfiguracijom za aplikaciju u okviru svih okruženja [3].

Odlike *Spring Cloud Config Server-a* su:

- HTTP-bazirani API za eksternu konfiguraciju
- enkripcija i dekripcija konfiguracionih vrednosti
- mogućnost korišćenja u okviru *Spring Boot* aplikacije

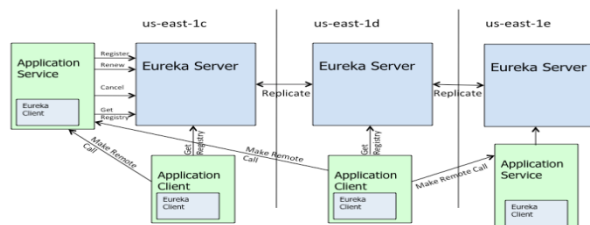
3.3. Spring Cloud Netflix

Spring Cloud Netflix je integracija *Netflix OSS* biblioteke, koja rešava probleme distribuiranih sistema, u *Spring* ekosistemu. *Pattern-i*, koje *Spring Cloud Netflix* obuhvata su:

- otkrivanje servisa pomoću *Eureka*
- pametno rutiranje pomoću *Zuul-a*
- prekidanje puta (*Circuit Breaker*) pomoću *Hystrix-a*

- klijent-bazirano balansiranje opterećenja sa *Ribbon-om*

Eureka je *REST*-baziran servis, koji se primarno koristi za *AWS (Amazon Web Services) Cloud* okruženje u svrhu lociranja servisa i balansiranja opterećenja poslatih zahteva. Uloga *Eureka* servisa je da opslužuje zahteve *Eureka* klijenata, kojima je potrebna informacija o adresi željenih servisa [2].



Slika 2: Prikaz arhitekture *Eureka Server-a*

Slika 2 prikazuje da je *Eureka server* napravljen da se sastoji iz više replikacija po različitim geografskim lokacijama i da pruža indirektni pristup klijentima ka željenoj aplikaciji.

Svaki servis, koji želi da bude na raspolaganju drugim servisima, pri pokretanju se registruje kod *Eureka server-a* i na svakih 30 sekundi treba da šalje *Heartbeat* odnosno poruku da je i dalje u funkciji. Takođe, proces registracije se vrši samo kod jednog servera, a potom se ta informacija o registrovanju servisa šalje svim ostalim instancama *Eureka servera* [2].

Zuul predstavlja pristupnu tačku za sve zahteve, koja se upućuje u sistem od strane klijenata, te omogućava dinamička rutiranja, osmatranja, otpornost i sigurnost zahteva. *Zuul* predstavlja konkretnu implementaciju *API Prolaz Design Pattern-a*.

Zuul je razvijem od strane kompanije Netflix i koristi se za sledeće stvari:

- autentifikaciju
- razne vrste testiranja
- praćenje i nadgledanje zahteva
- dinamičko rutiranje
- integraciju servisa
- bezbednost
- balansiranje opterećenja
- upravljanje saobraćajem

U *Spring Cloud* ekosistemu, *Zuul* je ugrađen u okviru *Spring* aplikacije, kako bi se pojednostavio njen razvoj.

Česta pojava u distribuiranim sistemima jeste da jedan deo sistema otkáže. U takvim okolnostima, zahtev uglavnom izvrši više skokova na više različitih servisa pre nego što vrati odgovor klijentu. Spram toga, ukoliko na putanji zahtev dobije grešku, ta greška se može propagirati kroz više servisa davajući mogućnost da sistem postane nestabilan, a u najgorem slučaju nedostupan.

Kako bi se taj problem prevazišao, uvodi se *Circuit breaker* biblioteka pod imenom *Hystrix*, koja uvodi sledeće funkcionalnosti:

- kontrola kašnjenja i grešaka zahteva
- zaustavljanja proširenja jedne greške na ceo distribuiran sistem
- oporavak sistema u slučaju greške
- nadgledanje sistema

Važnost otpornosti sistema, posebno u kompleksnim distribuiranim sistemima, pokazuje računica da sistem od 30 servisa ima 99.99% dostupnost. Kumulativno se dobija totalna dostupnost sistema $99,99\%^{30}=99,7\%$. Kada se prevede nedostupnost sistema od 3% na bilion zahteva, na mesečnom nivou to predstavlja oko 3 miliona grešaka što rezultira da sistem ne bude u funkciji 2 ili čak više sati na mesečnom nivou [4].

3.4. Spring Cloud Security

Spring Cloud Security predstavlja skup primitiva da se na deklarativan način dobiju funkcionalnosti, koje su potrebne da bi sistem bio siguran, kao što su identifikacija, autentifikacija i autorizacija [5].

Jedne od glavnih funkcionalnosti su:

- transfer *Single Sign On* tokena sa *Frontend* strane na *Backend* stranu preko *Zuul* proksija
- konfigurisanje autentifikacije u *Zuul* proksiji
- transfer tokena između resursnih servera.

3.5. Spring Cloud Sleuth

Spring Cloud Sleuth je implementacija distribuiranog praćenja u *Spring* okruženju. Većina implementacije se svodi na adaptiranje postojećih rešenja za distribuirano praćenje, kao što su: *Zipkin*, *Dapper* i *HTrace*, i njihovo korišćenje u *Spring Cloud*-baziranim sistemima [6].

Osnovna jedinica rada u *Spring Cloud Sleuth-a* predstavlja entitet po imenu *Span*. *Span* predstavlja jedinstveno identifikovani dnevnik vremenski označenih zapisa, koji čuva početak i kraj operacije. Pomenuti zapisi se identifikuju pomoću 64-bitnog identifikatora tzv. *span id* i još jednog 64-bitnog identifikatora zvanog *trace id*, koji predstavlja identifikator *trace-a*, čiji je *span* deo. Pored identifikatora, *span-ovi* mogu sadržati u sebi i druge podatke kao što su: opisi, identifikator procesa tj. IP-adresa i slično [6].

Svaki *span* se programski započinje i završava, te se na taj način čuvaju informacije o vremenu [6].

Korišćenjem *span-ova* i *trace-ova* daje se mogućnost da se operacije koje se dešavaju distribuirano povezuju, te na taj način dovode do lakšeg uočavanja problema u sistemu.

4. ZAKLJUČAK

Na osnovu iznetih podataka o mikroservisnoj arhitekturi, može se zaključiti da je mikroservisna arhitektura po svojoj prirodi pogodna za razvoj sistema, koji su veliki i kompleksni. Veličina sistema se ogleda u količini koda, koja je potrebna da bi se dati sistem razvio. Ukoliko bi se celokupan sistem održavao na jednom mestu, došlo bi do velikog opterećenja pri održavanju i daljem razvoju datog sistema. Stoga, deljenje koda na izolovane celine daje veći stepen indirektnosti delova koda i smanjuje šansu pojave greške u sistemu.

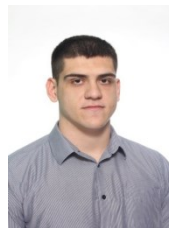
Pored velikih prednosti, koje mikroservisna arhitektura može da pruži, javljaju se drugi problemi sa kojima razvojni tim mora da se nosi. Ti problemi polaze od činjenice da mikroservisna arhitektura razvija sistem kao distribuiran. Samim tim, problem nadgledanja, testiranja, *deploy-ovanja* celokupnog sistema funkcioniše drugačije te je potrebno više uložiti vremena za te stvari u toku razvoja. Za skladan i uspešan rad mikroservisne arhitekture neophodno je podržati proces kontinuirane integracije ili kontinuirane isporuke, kako bi se problemi kod određenih delova sistema u najbržem roku uočili.

Mikroservisna arhitektura kao način razvoj projekta ne predstavlja najbolje rešenje za svaku situaciju. Ova arhitektura sa sobom nosi velike prednosti, ali takođe i traži veliku odgovornost i disciplinu razvojnog tima kako bi to imalo smisla. Stoga, možemo zaključiti da mikroservisna arhitektura predstavlja samo jedno od ponuđenih rešenja. Pre nego što projekat pređe u fazu razvoja, potrebno je dobro razmisliti koja će softverska arhitektura biti odabrana, jer arhitektura utiče na brzinu razvoja, rad i sigurnost sistema.

5. LITERATURA

- [1] S. Newman, Building Microservices, O'Reilly Media, Inc., 2015.
- [2] Pivotal Software, Inc., "Spring Cloud Config," 2019. URL: <https://spring.io/projects/spring-cloud-config>. (pristupljeno 07. 05. 2019).
- [3] "Eureka Netflix," 2019. URL: <https://github.com/Netflix/eureka/wiki/Eureka-at-a-glance>. (pristupljeno 16. 05. 2019).
- [4] "Hystrix wiki," 2019. URL: <https://github.com/Netflix/Hystrix/wiki>. (pristupljeno 30. 5. 2019).
- [5] "Spring Cloud Security," 2019. URL: <https://cloud.spring.io/spring-cloud-security/spring-cloud-security.html>. (pristupljeno 30. 05. 2019).
- [6] "Spring Cloud Sleuth," 2019. URL: <https://spring.io/projects/spring-cloud-sleuth>. (pristupljeno 13. 6. 2019).
- [7] B. H. S. a. L. A. B. a. M. B. a. P. S. a. M. P. a. D. B. a. S. J. a. C. Shanbhag, „Dapper, a Large-Scale Distributed Systems Tracing Infrastructure,“ 2010.

Kratka biografija:



Stefan Bratić rođen je 08.03.1994. godine u Novom Sadu. Godine 2008. završio je Osnovnu školu "Vuk Karadžić" u Novom Sadu. Srednju ekonomsku školu "Svetozar Miletić" u Novom Sadu završio je 2013. godine. Iste godine upisao je Fakultet tehničkih nauka na Univerzitetu u Novom Sadu, smer Softversko inženjerstvo i informacione tehnologije. Godine 2017. diplomirao je sa temom diplomskog rada *Upotrebna X-pack-a u Elastic stack tehnologiji*. Master Akademske Studije, smer Softversko inženjerstvo i informacione tehnologije, upisao je 2018. godine, gde je tokom te godine, dobivši *Best of South-East* stipendiju, pohađao kurseve na Tehničkom univerzitetu u Gracu i položio sve ispite koji su bili predviđeni planom i programom.

kontakt: stefanbraticns@gmail.com

SISTEM ZA PERSONALIZACIJU VIDEO TUTORIJALA ZASNOVAN NA MAŠINSKOM UČENJU**VIDEO TUTORIAL PERSONALIZATION SYSTEM BASED ON MACHINE LEARNING**Nevena Đerić, *Fakultet tehničkih nauka, Novi Sad***Oblast – SOFTVERSKO INŽENJERSTVO I INFORMACIONE TEHNOLOGIJE**

Kratak sadržaj – Problem koji ovaj rad rešava jeste preporuka najpogodnijeg tipa tutorijala. Rad je rezultat istraživanja koje je rađeno nad studentima Fakulteta tehničkih nauka. Studenti su odgovarali na pitanja ankete, gledali video tutorijale o algoritmima sortiranja i odgovarali na pitanja testa znanja o algoritmima sortiranja. Prilikom gledanja jednog od video tutorijala praćen im je pogled na ekranu uz pomoć uređaja za praćenje pokreta očiju. Na taj način formiran je skup podataka, koji je korišćen za izgradnju modela. U modelu sa najvećom tačnošću, za pronalaženje sličnih studenata koristi se KNN algoritam. Na osnovu sličnih studenata pronalazi se najpogodniji tip tutorijala za datog studenta. Ovaj model postigao je tačnost od 75% na validacionom skupu i tačnost od 67% na testnom skupu.

Gljučne reči: *personalizovana nastava, eye-tracker uređaj, klasterovanje, KNN, KMeans*

Abstract – The problem that this paper solves is the recommendation of the most appropriate type of tutorial to the student. This paper is the result of research conducted on students of the Faculty of Technical Sciences. Twenty nine students were questioned. Students answered survey questions, watched video tutorials on sorting algorithms, and answered questions about sorting algorithms. When they were watching one of the video tutorials, their eye movement were followed using an eye-tracking device. In that way was formed dataset that was used to build the model. The model that gave the highest accuracy is consisted of KNN algorithm, which is applied in finding students with the most similar type of movement. The most appropriate type of tutorial is found based on similar students. This model achieved an accuracy 75% on the validation set and accuracy 67% on the test set.

Keywords: *personalized teaching, eye-tracker device, clustering, KNN, KMeans*

1. UVOD

Personalizovana nastava je nastava budućnosti. Razlike u psiho-fizičkim sposobnostima, načinima percepcije okruženja i stečenim navikama kod studenata rezultuju nejednakom efikasnošću usvajanja znanja. Prilagođavanje nastave pojedincu dovodi do boljih rezultata.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, red. prof.

Ovo potvrđuje i činjenica da studenti sve više uplaćuju privatne časove ili pretražuju internet, da bi pronašli reprezentaciju gradiva u željenom obliku [1]. Pitanje koje se postavlja je da li način učenja, koji student smatra da je optimalan za njega, stvarno dovodi do najboljih rezultata u najkraćem vremenskom periodu.

U ovom radu opisan je sistem za preporuku najpogodnijeg video tutorijala. Sistem vrši preporuku na osnovu osobina studenta i njegovog tipa gledanja.

Rad sadrži šest sekcija. U drugoj sekciji prikazana su prethodna rešenja problema, koja u svojim istraživanjima koriste uređaj za praćenje pokreta očiju. U trećoj sekciji opisan je skup podataka, obeležja koja sadrži, način prikupljanja podataka i proces njegovog skladištenja. U četvrtoj sekciji opisani su korišćeni alati prilikom implementacije sistema i opisana je metodologija. U petoj sekciji opisana je evaluacija sistema. U šestoj sekciji izvršen je osvrt na implementirani sistem i prodiskutovani nedostaci sistema, kao i potencijalna buduća poboljšanja.

2. PRETHODNA REŠENJA

Tehnologija za praćenje pokreta očiju (engl. *eye-tracking*) je skup metoda i tehnika koje se koriste za detektovanje i snimanje aktivnosti pokreta očiju. Uređaj za praćenje pokreta očiju (engl. *eye-tracker*) meri pokrete očiju dok se oči kreću gledajući u neki video objekat [2]. Postoje razni radovi koji u svom istraživanju koriste uređaj za praćenje pokreta očiju, kao i neke od metrika za obradu podataka prikupljenih pomoću ovih uređaja. Njihov pregled dat je u nastavku.

U objavljenim radovima koriste se različite metrike za praćenje pokreta očiju. Čak i kada se koristi ista metrika, u različitim radovima, drugačije se naziva. Posledica ovoga su teška poređenja studija. Postoji potreba za standardizacijom metrika koje se koriste za obradu podataka sa uređaja za praćenje pokreta očiju. Rad [3] nastoji da dovede do svesti za konzistentnom upotrebom ovih metrika. U radu dat je pregled metrika korišćenih u ostalim radovima. Metrike bazirane na fiksaciji očiju mogu se podeliti na metrike broja fiksacija i metrike trajanja fiksacija. Fiksacije su ona vremena kada oči prestaju da se pomeraju, drže pogled na jednom mestu, tako da vizuelni sistem može dobiti informaciju o tome šta ispitanik gleda. *Fixation Rate* spada u metrike broja fiksacija. Izračunava se kao odnos broja fiksacija područja od interesa i ukupnog broja fiksacija ili broja fiksacija nekog drugog područja od interesa. *Fixation Time* spada u metrike trajanja fiksacije. Predstavlja zbir trajanja svih fiksacija područja od interesa. Područje od interesa (engl.

Areas of Interest – AOI) je alat za odabir područja za praćenje i izračunavanje metrika. Nije strogo merni alat sam po sebi, ali predstavlja područje na kojem se izračunavaju drugi merni podaci [4].

U radu [5] su korišćene metrike fiksacije: broj fiksacija za svako područje od interesa, ukupno vreme trajanja fiksacija, vreme prve fiksacije određenog područja od interesa. Predstavljena su dva eksperimenta. Prvi eksperiment predstavlja analizu opažanja u medicinskom procesu dijagnostikovanja. Drugi eksperiment predstavljala analizu vizuelne pažnje pacijenata prilikom psihomotornog testa. Za praćenje pokreta očiju u ovom istraživanju korišćen je ekranski uređaj. Da bi prikazali razliku između radiologa eksperta i studenta korišćene su toplotne mape. Toplotna mapa (engl. *heatmap*) predstavlja grafički prikaz frekvencije gledanja pojedinačnih delova ekrana. Delovi se označavaju određenom bojom, zavisno od broja identifikovanih fiksacija oka. Tako se crvenom bojom označavaju delovi koji su privukli najviše pažnje korisnika, nakon koje slede narandžasta, žuta, pa plava boja kojom su označeni delovi sa najmanjim brojem fiksacija [4].

Rad [6] opisuje klasterovanje podataka dobijenih pomoću uređaja za praćenje pokreta očiju. Korišćeni algoritmi klasterovanja su: *KMeans*, *spectral clustering* i *DBscan*. Predloženi pristup za evaluaciju konfiguracije klastera sastoji se od dva koraka.

Prvi je primena interne mere validacije klastera, koja se zove koeficijent siluete (engl. *silhouette score*) [7]. Nedostatak ove mere jeste favorizovanje konveksnih klastera, što možda ne odgovara strukturi podataka.

Drugi korak jeste vizuelizacija rezultata klasterovanja. Metode redukcije dimenzionalnosti projektuju podatke visokih dimenzija u ljudski čitljiv dvodimenzionalni prostor. U ovom radu koristi se t-SNE (*t-Distributed Stochastic Neighbor Embedding*) [8]. Algoritam može otkriti strukturu podataka, ali se ne može koristiti za klasterovanje, jer ne čuva udaljenost.

Još jedan rad [9] bavi se pronalaženjem korelacije između načina gledanja kod ljudi i njihovih osobina. Konkretno, dokazuje različitost muškaraca i žena u načinu posmatranja slika. Snimani su pokreti očiju 52 osobe starosti od 19 do 47 godina prilikom njihove evaluacije u tri različite dimenzije značenja 80 različitih slika. Rezultati pokazuju da su dužine fiksacija kod muškaraca kraće nego kod žena, dok su amplitude brzog pomeranja oka kod žena znatno veće nego kod muškaraca. Pokazano je da su žene sklonije istraživanju, praveći više fiksacija van regiona lica a slikama. Tačnost kojom je u ovom radu klasifikovan pol osobe na osnovu gledanja jedne slike iznosi 63%.

3. OPIS SKUPA PODATAKA

Skup podataka za ovo istraživanje prikupljen je ispitivanjem studenata. Ispitano je 29 studenata, prosečne starosti 20 godina. Svaki student je popunjavao anketu, koja se sastoji od pitanja kao što su koliko često gleda tutorijale, koliko dugo mu gledanje tutorijala ili čitanje drži pažnju, itd, kao i pitanja koja se tiču njegovog prethodnog obrazovanja, pola, mesta rođenja, godina, itd. Nakon popunjavanja ankete student gleda tri video

tutorijala. Tutorijali su pravljani tako da poseduju različite elemente: tutora, animaciju i prevod. Uvodni video tutorijal sadrži sva tri elementa. Prilikom gledanja ovog tutorijala studentu se prati pogled na ekranu. Nakon odgledanog tutorijala, student gleda još dva tutorijala u vezi dva algoritma sortiranja: *Bubble sort* [10] i *Selection sort* [11]. Za svaki algoritam postoje četiri kombinacije (sa tutorom sa prevodom, bez tutora (animacija) sa prevodom, sa tutorom bez prevoda, bez tutora bez prevoda). Student za svaki algoritam gleda jednu kombinaciju. Posebno je obraćena pažnja da se studentu ne pusti ista kombinacija za oba algoritma. Nakon odgledanih tutorijala student odgovara na pitanja testa znanja o prezentovanom algoritmu sortiranja.

Od studentovih odgovora na pitanja ankete i testa, kao i od podataka prikupljenih pomoću uređaja za praćenje pokreta očiju formiran je skup podataka. Prikupljeni podaci se mogu podeliti u tri celine: podaci sa ankete, podaci prikupljeni pomoću uređaja za praćenje pokreta očiju i podaci sa testa znanja. Podaci sa ankete i podaci sa testa se čuvaju u CSV datoteci. Za svakog studenta čuva se vektor obeležja. Podaci praćenja pokreta očiju prikupljeni su uz pomoć softvera za upravljanje uređajem za praćenje pokreta očiju. Ovaj softver za svakog studenta generiše CSV datoteku u koju se za svaki sekund trajanja videa izgeneriše po 60 vektora.

4. METODOLOGIJA I ALATI

U ovoj sekciji opisani su alati korišćeni za implementaciju sistema i prikazana je metodologija.

4.1. Korišćeni alati

U ovom istraživanju koristi se *Gazepoint GP3 eye-tracker* [12] ekranski uređaj. Softver koji se koristi za upravljanje uređajem za praćenje pokreta očiju, kao i za prikupljanje i obradu prikupljenih podataka je *Gazepoint Control and Analysis* [13]. Za implementaciju sistema koristi se programski jezik *Python* [14].

4.2. Metodologija

Metodologija ovog rešenja može se podeliti u tri celine: obrada i analiza podataka, pronalaženje studenata sa najslbližijim tipom gledanja i uspostavljanje korelacije između osobina gledanja i rezultata testa znanja.

4.2.1. Obrada i analiza podataka

Ova podsekcija opisuje proces obrade i vizuelizacije podataka.

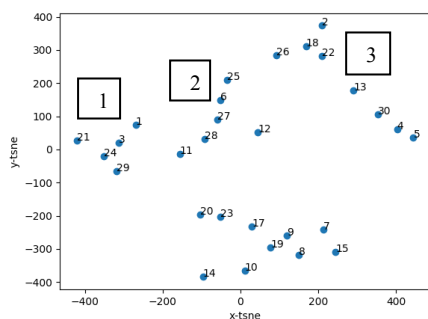
Podaci sa ankete sadrže kategorijske vrednosti. Kategorijske vrednosti zamenjene su numeričkim pomoću metode *Label encoding* [15]. Vrednost atributa mesto rođenja zamenjena je sa brojem stanovnika u tom mestu. Ovo je urađeno kako bi se napravila razlika između malih i velikih mesta.

Podaci sa uređaja za praćenje pokreta očiju obrađeni su pomoću metrika za obradu podataka prikupljenih pomoću uređaja. Za svako područje od interesa (tutora, animaciju i prevod) računata je *Fixation rate* metrika, kao odnos broja fiksacija područja od interesa i ukupnog broja fiksacija. *Fixation Time* metrika predstavlja ukupno vreme fiksacija područja od interesa. Vreme prve fiksacije

područja od interesa predstavlja vreme koje je studentu potrebno da prvi put ugleda područje od interesa.

Odgovori studenata na pitanja sa testa znanja zamenjeni su sa brojem poena. Svaki tačan odgovor vrednovan je sa jednim poenom. Prilikom računanja broja poena u obzir je uzeto i vreme koje je bilo potrebno da se odgovori na pitanje. Na svaki poen tačnog odgovora dodat je i koeficijent, brzine davanja tačnog odgovora, koji je računat prema formuli $\frac{10}{t}$, gde je t broj sekundi koji je bio potreban studentu da odgovori na pitanje. Analizom skupa podataka ova formula je odabrana kao najpogodnija da napravi razliku između brzine potrebne za odgovor na pitanje.

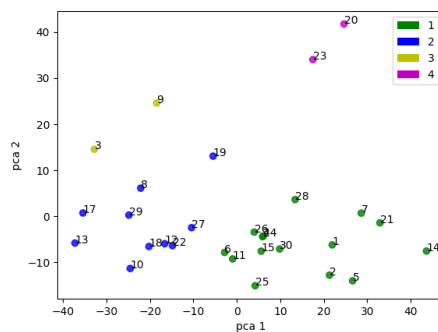
Prikupljeni podaci su visoko dimenzionalni. Da bi se ovakav skup podataka mogao vizuelizovati najpre bi trebalo primeniti neku od metoda za redukciju dimenzionalnosti. Na slici 1. prikazana je vizuelizacija redukovanih podataka sa ankete pomoću t-SNE algoritma. Na slici se može primetiti izdvajanje četiri klastera. Brojevi pored tačaka su jedinstveni identifikatori studenata. Brojevi pored grupa tačaka predstavljaju brojeve klastera. Studenti prvog klastera su muškarci, koji često gledaju video tutorijale i pohađali su srednju tehničku školu. Studenti drugog klastera su muškarci, koji provode više od 4 sata dnevno na internetu i više vole da čitaju kraće forme, kao što su časopisi ili novine. Studenti trećeg klastera nemaju iskustva u programiranju i više vole da čitaju knjige i gledaju filmove. Četvrtom klasteru pripadaju studentkinje, koje imaju po 19 godina i ne bave se sportom.



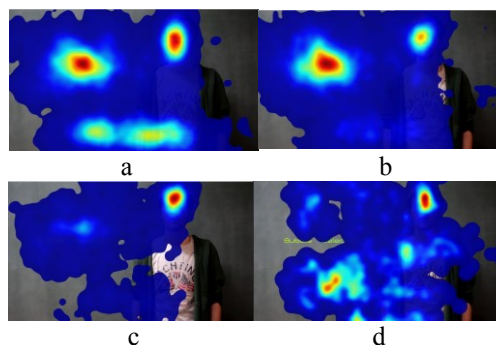
Slika 1. Vizuelizacija skupa podataka sa ankete

Na preprocesirane podatke dobijene pomoću uređaja za praćenje pokreta očiju primenjen je PCA (*Principal Component Analysis*) [16] algoritam za redukciju dimenzionalnosti. Na tako redukovane podatke primenjen je *KMeans* algoritam za klasterovanje. Pomoću lakatnog metoda i koeficijenta siluete za k je dobijen broj 4. Za inicijalizaciju centara klastera korišćen je *KMeans++* algoritam. Na slici 2. prikazano je klasterovanje studenata prema tipu gledanja. Za svaki klaster izgenerisane su toplotne mape, koje su prikazane na slici 3. Toplotna mapa 3. a prikazuje studente koji pripadaju prvom klasteru. Može se primetiti da su studenti ovog klastera pratili pogledom podjednako animaciju i tutora, takođe su gledali i prevod, ali znatno manje nego animaciju i tutora. Toplotna mapa 3. b prikazuje studente koji pripadaju drugom klasteru. Ovi studenti su najviše pažnje obraćali na animaciju. Toplotna mapa 3. c prikazuje studente

trećeg klastera. Ovi studenti su najviše gledali u tutora. Toplotna mapa 3. d prikazuje studente četvrtog klastera. Ovaj klaster predstavlja šum, odnosno u njega su svrstani studenti kojima uređaj za praćenje pokreta očiju nije uspeo da isprati pogled, npr. pomerili su se prilikom gledanja video tutorijala i uređaj za praćenje pokreta očiju nije detektovao oči.



Slika 2. KMeans klasterovanje studenata na osnovu metrika za praćenje pogleda



Slika 3 Prikaz toplotne mape studenata koji pripadaju redom prvom, drugom, trećem i četvrtom klasteru

4.2.2. Pronalaženje studenata sa najbližim tipom gledanja

Za pronalaženje studenata sa najbližim tipom gledanja upotrebljeni su podaci sa ankete i podaci sa uređaja za praćenje pokreta očiju obrađeni pomoću metrika.

Na preprocesirane podatke primenjen je KNN (*K Nearest Neighbors*) [17] algoritam. KNN algoritam je efikasan u rešavanju sličnih problemima, pa je izabran da se koristi u ovom sistemu. Algoritam za prosleđenog studenta pronalazi studente sa najbližim tipom gledanja.

4.2.3. Uspostavljanje korelacije između osobina gledanja i rezultata testa znanja

Ova metoda određuje optimalan tip gledanja za prosleđen skup studenata. *Bubble* i *Selection sort* algoritmi posmatraju se kao ekvivalentni. Za svakog studenta iz skupa posmatra se razlika u tipu, odnosno kombinaciji elemenata, odgledanih video tutorijala. Na primer, ako je student ostvario više poena na testu video tutorijala, koji je sadržao tutora, ali nije imao prevod, nego na testu video tutorijala, koji nije sadržao tutora, ali jeste imao prevod, može se zaključiti da studentu, tutor pomaže u učenju, dok sa prevodom to nije slučaj. Na ovaj način, dodeljivanjem pozitivnih i negativnih bodova vezanih za posto-

janje elementa u video tutorijalu dolazi se do ukupnih bodova. Bodovi upućuju na tip tutorijala, koji je optimalan za prosledeni skup studenata. Pošto prosledeni studenti imaju najbližnje osobine sa studentom za kog se traži odgovarajući tip tutorijala, onda je taj tip tutorijala optimalan za tog studenta. Osim pronalaska optimalnog tipa tutorijala, iz ovih podataka se mogu rangirati i ostali tipovi tutorijala. Krajnji rezultat ove metode je lista tipova tutorijala sortirana od najpogodnijeg do najmanje pogodnog za datog studenta.

5. EVALUACIJA I REZULTATI

U ovom istraživanju performanse su evaluirane na dva načina. Prva evaluacija urađena je podelom skupa podataka na obučavajući i testni. Obučavajući skup podataka sadrži 80% podataka, a testni 20%. Podaci iz testnog skupa nisu se ni na koji način koristili prilikom obučavanja. Druga evaluacija urađena je samo na obučavajućem skupu, pomoću unakrsne validacije (engl. *cross-validation*) sa 4 podskupa. Za metriku je korišćena tačnost (engl. *accuracy*). Za izračunavanje tačnosti upoređeni su tipovi tutorijala koje je sistem predložio sa tipovima koje je student gledao. Student je odgledao samo dva od moguća četiri tipa. Zbog toga se ne gleda samo najpogodniji tip, nego se uzima u obzir lista sortiranih tipova od najpogodnijeg do najmanje pogodnog. Porede se poeni koje je student osvojio na testu znanja i gleda se mesto koje tip odgledanog video tutorijala zauzimaju na listi predloženih. Ukoliko je tip tutorijala, za koji je student dobio više poena, na listi pre tipa za koji je student dobio manje poena znači da je model dao dobru preporuku. Najveća tačnost od 75% dobijena je na validacionom skupu sa izborom 5 suseda. Dok tačnost na testnom skupu iznosi 67%.

6. ZAKLJUČAK

U ovom radu predstavljen je sistem za personalizaciju video tutorijala. Sistem za unete podatke sa ankete za studenta i podatke dobijene pomoću uređaja za praćenje pokreta očiju prilikom gledanja uvodnog tutorijala pronalazi najpogodniji tip tutorijala.

Model koji je dao najbolje rezultate za pronalaženje najbližnjih studenata koristi KNN algoritma. Na osnovu sličnih studenata pronalazi se odgovarajući tip tutorijala za datog studenta. Ovaj model je postigao tačnost od 75% na validacionom skupu i tačnost od 67% na testnom skupu, što znači da je sistem uspešan za četiri od šest slučajeva. Može se smatrati dobrim rezultatom, s obzirom na količinu prikupljenih podataka.

Prikupljanje podataka za ovo istraživanje je predstavljalo najveći izazov. Osnovni problem je bio motivisati studente da učestvuju u ovom istraživanju. Sa druge strane, postoji jedan uređaj za praćenje pokreta očiju, pa se prikupljanje podataka ne može optimizovati, niti paralelizovati. Ispitivanje je vršeno u laboratorijski kontrolisanim uslovima, koje zahteva dosta vremena.

Sledeći korak u uvođenju personalizovane nastave bi bilo proširivanje istraživanja van okvira laboratorije. Za to bi se mogli iskoristiti pametni telefoni i tableti, kao uređaji za praćenje pokreta očiju [18]. Sledeći korak unapređenja sistema je implementacija mobilne aplikacije za

prikupljanje podataka u realnom vremenu. Prilikom prikupljanja velike količine podataka javljaju se drugi problemi, koji se mogu otkloniti pomoću alata za obradu velike količine podataka.

7. LITERATURA

- [1] Milena Kostadinović. Personalizacija obrazovanja (Stilovi učenja), 2015
- [2] Mele, Maria Laura, and Stefano Federici. Gaze and eye-tracking solutions for psychological research. *Cognitive processing*, 2012, 13.1: 261-265.
- [3] Sharafi, Zohreh, et al. Eye-tracking metrics in software engineering. In: 2015 Asia-Pacific Software Engineering Conference (APSEC). IEEE, 2015. p. 96-103.
- [4] <https://imotions.com/blog/7-terms-metrics-eye-tracking/> (pristupljeno u avgustu 2019.)
- [5] Borys, Magdalena, and Małgorzata Plechawska-Wójcik. Eye-tracking metrics in perception and visual attention research. *EJMT*, 2017, 3: 11-23.
- [6] Göbel, Fabian, and Henry Martin. Unsupervised Clustering of Eye Tracking Data. In: *Spatial Big Data and Machine Learning in GIScience*, Workshop at GIScience 2018. p. 25-28.
- [7] Rousseeuw, Peter J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 1987, 20: 53-65.
- [8] Maaten, Laurens van der; HINTON, Geoffrey. Visualizing data using t-SNE. *Journal of machine learning research*, 2008, 9.Nov: 2579-2605.
- [9] Moss, Felix Joseph Mercer, Roland Baddeley, and Nishan Canagarajah. Eye movements to natural images as a function of sex and personality. *PLoS One*, 2012, 7.11: e47870.
- [10] Astrachan, Owen. Bubble sort: an archaeological algorithmic analysis. In: *ACM SIGCSE Bulletin*. ACM, 2003. p. 1-5.
- [11] Hayfron-Acquah, J. B., Obed Appiah, and K. Riverson. Improved Selection Sort Algorithm. *International Journal of Computer Applications*, 2015, 110.5.
- [12] <https://www.gazept.com/product/gazepoint-gp3-eye-tracker/>, (pristupljeno u avgustu 2019.)
- [13] <https://www.gazept.com/tutorials/>, (pristupljeno u avgustu 2019.)
- [14] <https://www.python.org/>, (pristupljeno u avgustu 2019.)
- [15] <https://towardsdatascience.com/choosing-the-right-encoding-method-label-vs-onehot-encoder-a4434493149b>, (pristupljeno u avgustu 2019.)
- [16] Wold, Svante, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 1987, 2.1-3: 37-52
- [17] Peterson, Leif E. K-nearest neighbor. *Scholarpedia*, 2009, 4.2: 1883.
- [18] Kraska, Kyle, et al. Eye tracking for everyone. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. p. 2176-2184.

Kratka biografija:



Nevena Đerić rođena je 05.06.1995. godine u Užicu. Osnovnu školu „Prva osnovna škola kralj Petar Drugi“ završila je 2010. godine. Gimnaziju „Užička gimnazija“ u Užicu, prirodno-matematički smer, završila je 2014. godine. Iste godine upisala se na Fakultet tehničkih nauka, odsek Računarstvo i informatika. Zvanje diplomirani inženjer elektrotehnike i računarstva stekla je 2018. godine. Iste godine upisala je master akademske studije na smeru Softversko inženjerstvo i informacione tehnologije. Uža specijalnost na master studijama joj je inteligentni sistemi. Položila je sve ispite propisane planom i programom.

**PRIMENA UČENJA USLOVLJAVANJEM NA OBUČAVANJE AGENTA ZA
AUTONOMNO KRETANJE U OKRUŽENJU *CarRacing-v0*****USING REINFORCEMENT LEARNING TO TRAIN AN AGENT FOR THE *CarRacing-v0*
ENVIRONMENT**

Novica Šarenac, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – U radu je opisan trening i evaluacija agenta za autonomno kretanje u OpenAI Gym okruženju *CarRacing-v0*. Okruženje predstavlja stazu za vožnju automobila iz ptičije perspektive. Za treniranje agenta korištene su tehnike učenja uslovljavanjem. Poređene su osobine algoritama u pogledu rezultata postignutih u okruženju, dužine treniranja i implementacionih detalja. Implementirani i poređeni algoritmi su: *Deep Q-Network (DQN)*, *Advantage Actor Critic (A2C)* i *Asynchronous Advantage Actor Critic (A3C)*.

Ključne reči: učenje uslovljavanjem, *CarRacing-v0*, *Deep Q-Network*, *Advantage Actor Critic*, *Asynchronous Advantage Actor Critic*

Abstract – This paper presents training and evaluation of the agent for autonomous driving in OpenAI Gym environment *CarRacing-v0*. Environment is a top-down view of racing track. Agent is trained using reinforcement learning techniques. Algorithms are compared in terms of results achieved in the environment, training time and implementation details. Algorithms that are implemented and evaluated are: *Deep Q-Network (DQN)*, *Advantage Actor Critic (A2C)* i *Asynchronous Advantage Actor Critic (A3C)*.

Keywords: reinforcement learning, *CarRacing-v0*, *Deep Q-Network*, *Advantage Actor Critic*, *Asynchronous Advantage Actor Critic*

1. UVOD

U današnje vreme prisutna je velika ekspanzija veštačke inteligencije u svim sferama računarskih nauka. Sve više se teži razvoju softvera koji se na neki način može okarakterisati kao inteligentan. Veliku ulogu u toj ekspanziji ima razvoj hardvera, koji omogućava treniranje složenih modela.

Najčešća upotreba veštačke inteligencije je u detektovanju i prepoznavanju objekata na slikama, razumevanju i klasifikaciji teksta, sistemima za preporučivanje, odgovaranju na pitanja, i slično.

Sve češće veštačka inteligencija se upotrebljava za obučavanje inteligentnih agenata koji fleksibilno i samostalno izvršavaju neke akcije u okruženju.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, vanr. prof.

Tako u auto-industriji mnoge kompanije rade na razvoju autonomnih vozila, u privredi mnogi procesi teže ka automatizaciji, u softveru se razvijaju agenti čiji je cilj ostvarenje što boljih rezultata u simuliranim okruženjima (poput video igara).

U ovom radu opisan je trening i evaluacija agenta za autonomno kretanje u simuliranom okruženju. Okruženje predstavlja vožnju automobila iz ptičije perspektive. Za treniranje agenta korišćena su i upoređena tri algoritma učenja uslovljavanjem: *Deep Q-Network (DQN)* [1], *Advantage Actor Critic (A2C)* i *Asynchronous Advantage Actor Critic (A3C)* [2].

2. METODOLOGIJA

Svaki od pristupa za treniranje agenta koristi različite korake preprocesiranja stanja (observacija) koja daje okruženje, kao i različite metode dubokog učenja uslovljavanjem.

2.1. Okruženje *CarRacing-v0*

CarRacing-v0 [3] je okruženje sa OpenAI Gym platforme. Okruženje podrazumeva 2D simulaciju upravljanja automobilom iz ptičije perspektive.

Stanje koje okruženje vraća prilikom interakcije je slika dimenzija 96x96 piksela, sa tri kanala (RGB).

Nagrade se računaju za svaki frejm (korak u okruženju). Nagrada je -0.1 za svaki frejm i 1000/N za svaku posećenu pločicu na stazi, gde je N ukupan broj pločica na stazi. Kako se staza generiše na slučajnan način za svaku igru, ukupan broj pločica na stazi varira između 230 i 320. Epizoda se završava kada se posete sve pločice na stazi ili kada prođe 1000 frejmova.

Jedna od najvažnijih karakteristika ovog okruženja je što ima kontinualne akcije, za razliku od većine okruženja, kao što su *Atari* igre koje imaju diskretne akcije. Svaka akcija sastoji se od tri kontinualne vrednosti, koje odgovaraju: smeru (levo ili desno), ubrzanju i kočenju. Upravljanje smerom je u rangi između -1.0 i 1.0, gde je -1.0 potpuno skretanje levo i 1.0 potpuno skretanje desno. Ubrzavanje i kočenje je u rangi od 0 do 1.

U donjem delu prozora nalaze se grafički predstavljeni indikatori: brzina, ABS senzori, pozicija volana i žiroskop. Ove informacije nisu dostupne u brojevnom obliku. Smatra se da je okruženje rešeno ako je srednja vrednost nagrade u 100 uzastopnih pokušaja 900.

2.2. Primena DQN metode

Jedna od tri primenjene metode je DQN metoda. Da bi se agent mogao uspešno obučavati potrebno je stanje koje se dobija od okruženja prilagoditi DQN metodi. Takođe je potrebno odabrati optimalnu arhitekturu neuronske mreže, koja je pogodna za rad sa takvim stanjem.

2.2.1 Stanje u DQN metodi

Okruženje *CarRacing-v0* reprezentuje svako stanje sa 96x96x3 *RGB* matricom. Iz razloga kao što su brz konvergencija i trening vrši se niz koraka preprocesiranja originalnog stanja.

Stanje se prvo prebacuje u *grayscale*, čime se dobija matrica dimenzija 96x96. S obzirom na to da se indikatori sa donjeg dela prozora ne mogu iskoristiti u brojevnom obliku, oni se uklanjaju iz stanja. Iseca se prostor dimenzija 12x96 sa donjeg dela svake slike. Takođe se iseca prostor dimenzija 96x6 piksela sa leve i desne strane svake slike. Nakon izvršenih koraka preprocesiranja dobija se slika dimenzija 84x84 piksela.

Takođe se primenjuje tehnika *frame-skipping*. Agent vidi i odabira akciju za svako k -to stanje, umesto za svako stanje. Poslednja odabrana akcija se ponavlja za preskočena stanja. Pravljenje koraka u okruženju je računski veoma jeftino i brzo, za razliku od odabiranja akcija, pa ova tehnika omogućava agentu da odigra k puta više igara bez velikog povećanja vremena treniranja. Vrednost korištena za parametar k je 4.

2.2.2 Arhitektura DQN mreže

Stanje koje se koristi za obučavanje agenta je slika ekrana okruženja, pa je odabrana arhitektura koja predstavlja konvolutivnu mrežu. Arhitektura se sastoji od konvolutivnih, *batch normalization* i potpuno povezanih slojeva. Ulaz u neuronsku mrežu je preprocesirana slika dimenzija 84x84.

Prvi sloj je 2D konvolutivni sloj. Broj kanala na izlazu tog sloja je 16. Koristi se filter dimenzija 5x5 i korak (*stride*) pomeranja filtera 2. Sledeći sloj je *batch normalization* sloj.

Drugi konvolutivni sloj na ulazu očekuje 16 kanala, a broj kanala na izlazu je 32. Dimenzije filtera u ovom sloju su 5x5, a korak pomeranja 2. I nakon ovog sloja vrši se *batch* normalizacija.

Broj kanala na ulazu i izlazu trećeg konvolutivnog sloja je 32. Koristi se filter dimenzija 5x5 sa korakom pomeranja 2. Sledeći sloj je takođe *batch* normalizacija. Izlaz se transformiše u jednodimenzioni vektor, koji je pogodan za potpuno povezane slojeve.

Poslednja dva sloja su potpuno povezani slojevi. Prvi potpuno povezan sloj na ulazu ima dimenziju 1568, a na izlazu 256. Poslednji sloj na ulazu ima dimenziju 256, a na izlazu 4, što se poklapa sa brojem akcija. Izlaz neuronske mreže predstavlja Q -vrednosti za svaku akciju.

2.2.3 Implementacija DQN metode

Obučavanje agenta *DQN* metodom koristi okruženje i korake preprocesiranja opisane u sekciji 2.2.1. U

implementaciji se koriste dve mreže: *policy* mreža i *target* mreža. *Policy* mreža koristi se za odabiranje akcija, a *target* mreža služi za računanje očekivanih Q vrednosti.

U *replay* memoriji se čuvaju parovi: stanje, akcija, nagrada, sledeće stanje i da li je epizoda završena. Kapacitet *replay* memorije koji je korišten je 10 000 primera. Tokom treniranja izvršava se glavna trening petlja. Broj izvršenih koraka u trening petlji je 150 000.

Prvi korak u trening petlji je računanje Q -vrednosti za akcije u trenutnom stanju. Za ovo računanje koristi se *policy* mreža. Akcije su diskretizovane, tako da postoje četiri moguće akcije: skretanje levo, skretanje desno, gas i kočnica. Tokom treninga, za odabir akcije se koristi ϵ -greedy, da bi agent mogao istraživati okruženje. Ukoliko je slučajno odabran broj manji od ϵ agent na slučajan način odabira akciju. U suprotnom uzima se akcija za koju je Q -vrednost maksimalna. Vrednost ϵ se menja tokom treninga (1).

$$\epsilon = \epsilon_{final} + (\epsilon_{start} - \epsilon_{final}) * e^{(-1 * step / \epsilon_{step})} \quad (1)$$

Vrednost ϵ_{start} je 0.9, ϵ_{final} je 0.05 i ϵ_{step} je 30 000. Parametar *step* je korak do koga je trening petlja stigla.

Sledeći korak je izvršavanje akcije u okruženju. Nakon toga se par koji se sastoji od stanja, akcije, novog stanja (dobijenog iz okruženja) i statusa epizode čuva u *replay* memoriji.

Update policy mreže je sledeći korak obučavanja. Prvo se na slučajan način uzima *batch* sakupljenih primera iz *replay* memorije. Veličina *batch*-a je 32. Računaju se Q -vrednosti za urađene akcije za sva stanja u *batch*-u, koristeći *policy* mrežu. Zatim se računaju očekivane Q -vrednosti za sledeća stanja iz *batch*-a koristeći *target* mrežu. Nakon toga se računa *loss* i vrši *update policy* mreže.

Update target mreže vrši se povremeno, na svakih 10 000 koraka. *Update* se vrši tako što *target* mreža preuzme parametre *policy* mreže. Algoritam za optimizaciju i *update* neuronske mreže korišten u implementaciji je *RMSprop* [4].

2.3. Primena actor-critic metoda

Actor-critic metode koje su implementirane su A3C i A2C. Kao i kod DQN metode, i za *actor-critic* metode je potrebno prilagoditi stanje koje se dobija od okruženja i pronaći optimalnu arhitekturu neuronske mreže. Za obe implementirane metode se koriste isti koraci preprocesiranja stanja, kao i ista arhitektura neuronske mreže.

2.3.1. Stanje u actor-critic metodama

Preprocesiranje stanja (96x96x3 *RGB* matrice) koje se dobija od *CarRacing-v0* okruženja sastoji se od niza koraka preprocesiranja.

Prvi korak preprocesiranja je prebacivanje slike u *grayscale*, čime se dobija stanje dimenzija 96x96 piksela.

Sledeći korak preprocesiranja je centriranje piksela oko 0. To se vrši oduzimanjem srednje vrednosti od svakog piksela na slici. Kao i kod DQN metode, indikatori sa

donjeg dela prozora se uklanjaju iz stanja i iseca se prostor dimenzija 12x96 piksela sa donjeg dela svakog stanja. Takođe se iseca prostor dimenzija 96x6 piksela sa leve i desne strane stanja. Nakon preprocesiranja, dimenzija stanja je 84x84 piksela.

Krajnje stanje, koje se koristi u *actor-critic* metodama formira se od nekoliko poslednjih observacija (preprocesiranih 84x84 slika) koje daje okruženje. Observacije se čuvaju i stanje koje je ulaz u neuronsku mrežu za A3C i A2C algoritme je dimenzija 84x84xN. N je broj observacija koji se čuva. Vrednost N koja je korištena je 5.

2.3.2 Arhitektura mreže kod A2C i A3C metoda

Arhitektura neuronske mreže korištene kod A2C i A3C metoda je konvolutivna neuronska mreža. Arhitektura se sastoji od konvolutivnih i potpuno povezanih slojeva. Ulaz u neuronsku mrežu opisan je u sekciji 2.3.1 i ima dimenzije 84x84x5.

Prvi sloj je 2D konvolutivni sloj. Broj kanala na izlazu tog sloja je 16. Koristi se filter dimenzija 8x8 i korak pomeranja filtera 4.

Drugi konvolutivni sloj na ulazu ima 16 kanala, a broj izlaznih kanala je 32. Dimenzije filtera u ovom sloju su 3x3, a korak pomeranja filtera 2.

Izlaz drugog konvolutivnog sloja transformiše se u jednodimenzioni vektor, koji je pogodan za potpuno povezane slojeve. Sledeći sloj je potpuno povezan i na ulazu ima dimenziju 2592, a na izlazu 256.

Neuronska mreža ima dva izlaza. Jedan predstavlja funkciju vrednosti, a drugi politiku. Funkcija vrednosti i politika se razlikuju samo u poslednjem sloju.

Ulazna dimenzija za poslednji sloj funkcije vrednosti je 256, a izlazna je 1. Izlaz iz tog sloja je vrednost trenutnog stanja. Isto kao kod funkcije vrednosti, i kod poslednjeg sloja politike ulazna dimenzija je 256. Izlazna dimenzija politike je 4 i poklapa se sa brojem akcija.

2.3.4 Implementacija A3C metode

A3C metoda koristi okruženje i korake preprocesiranja opisane u sekciji 2.3.1. Kako je A3C metoda zasnovana na multiprocesingu prvi korak u implementaciji je da se inicijalizuje više procesa, od kojih svaki ima svoju trening petlju. Broj procesa koji je korišten je 8. Postoji jedan globalni model, čiji *update* vrše svi procesi.

Svaki proces ima lokalnu instancu modela, instancu okruženja, kao i referencu ka globalnom modelu. Inicijalizuje se i memorija koja čuva parove: korak, vrednost stanja, nagrada, logaritmi verovatnoće akcije, entropija i da li je epizoda završena. Podaci sačuvani u toj memoriji koriste se za jedan *update* mreže, pa je kapacitet memorije jednak broju koraka koji su potrebni za jedan *update*. Broj koraka korišten u implementaciji je 40. Takođe se izvršava glavna trening petlja. Broj izvršenih koraka u trening petlji u jednom procesu je 150 000.

Prvi korak u trening petlji je *update* lokalne mreže i preuzimanje parametara globalne mreže. Sledeći korak je petlja koja izvršava broj akcija potreban za jedan *update*

globalne mreže. Prvo se računaju verovatnoće akcija, logaritmi tih verovatnoća i vrednost trenutnog stanja. Zatim se, koristeći verovatnoće, bira akcija koja će se izvršiti u okruženju. Akcije su kontinualne i dobijaju se tako što se prvo odabere akcija koja ima maksimalnu verovatnoću. Moguće akcije su skretanje levo, skretanje desno, gas i kočnica. Zatim se taj vektor pomnoži sa verovatnoćom i ta akcija se iskoristi kao akcija u okruženju. Nakon odabira akcije računa se entropija, koja podstiče istraživanje agenta.

Sledeći korak u trening petlji je izvršavanje akcije u okruženju, čime se dobija novo stanje i nagrada. Korak, vrednost stanja, nagrada, logaritmi verovatnoće akcije, entropija i da li je epizoda završena čuvaju se u memoriji.

Kada se izvrši potreban broj koraka, računa se *loss* i vrši *update* globalne mreže. Prvo se računaju kumulativne nagrade. Zatim, koristeći kumulativne nagrade i vrednosti stanja računa se *advantage* i *loss* za funkciju vrednosti.

Sledeći korak je računanje *loss*-a za politiku, koji koristi izračunati *advantage* i logaritme verovatnoća akcija iz memorije. Koristeći *loss* funkciju vrednosti, *loss* funkciju politike i entropiju računa se krajnja vrednost *loss*-a. Vrednost koeficijenta entropije korištena u implementaciji je 0,01.

Kada se izračuna *loss* vrši se *update* globalnog modela. Algoritam za optimizaciju i *update* neuronske mreže korišten u implementaciji je Adam [5].

2.3.5 Implementacija A2C metode

A2C metoda koristi okruženje i korake preprocesiranja opisane u sekciji 2.3.1. I A2C metoda zasnovana je na paralelnim agentima, koji istražuju više okruženja. Agenti su sinhroni i koriste isti globalni model, čiji *update* se vrši tek kada svi agenti izvrše potreban broj koraka. Agenti imaju zajedničku trening petlju.

Inicijalizuju se procesi od kojih svaki ima po jednu instancu okruženja. Broj korištenih procesa je 8. Kada se uradi *reset* ili izvrši korak u okruženjima, čeka se da svi procesi urade *reset* ili korak. Takođe, kao i kod A3C metode, inicijalizuje se memorija, u kojoj se čuvaju parovi: korak, vrednosti stanja, nagrade, logaritmi verovatnoće akcije, entropije i da li su epizode završene. Podaci sačuvani u memoriji služe za *update* neuronske mreže, pa je kapacitet jednak broju koraka koji se izvrše za jedan *update* po procesu. Broj koraka korišten u implementaciji je 5. Broj koraka koji se izvršava u glavnoj trening petlji je 125 000.

Prvi korak u trening petlji je izvršavanje akcija potrebnih za jedan *update* mreže. Izvršava se 5 akcija po procesu. Prvo se računaju verovatnoće akcija, logaritmi tih verovatnoća i vrednost trenutnih stanja. Od dobijenih verovatnoća bira se akcija koja će se izvršiti u okruženju. Akcije su, kao i kod A3C metode kontinualne i biraju se na isti način.

Nakon odabira akcija računa se entropija. Zatim se odabrane akcije izvršavaju u svim okruženjima u isto vreme. Nakon izvršavanja, od svakog procesa se dobija novo stanje i nagrada.

Koraci, vrednosti stanja, nagrade, logaritmi verovatnoća akcija, entropije i da li su epizode završene čuvaju se u memoriji. Kada se izvrši potreban broj koraka, vrši se *update* mreže. Prvo se računaju kumulativne nagrade. Pomoću njih i vrednosti stanja se računa *advantage*, a zatim i *loss* za funkciju vrednosti. Za računanje *loss*-a za politiku koriste se izračunati *advantage* i logaritmi verovatnoća akcija, koji su sačuvani u memoriji.

Krajnji *loss* se računa pomoću *loss*-a funkcije vrednosti, *loss*-a politike i entropije. Koeficijent entropije koji je korišten je 0,1. Nakon toga vrši se *update* modela. Algoritam za optimizaciju i *update* mreže korišten u implementaciji je Adam.

3. REZULTATI

Okruženje *CarRacing-v0* smatra se rešenim ukoliko agent ostvari prosečnu vrednost nagrade veću od 900 u 100 uzastopnih pokušaja. Ti rezultati su daleko iznad rezultata koje može ostvariti čovek. Mera tačnosti korištena za evaluaciju sva tri modela je prosečna ostvarena nagrada u 100 uzastopnih epizoda.

Kod DQN metode broj koraka prilikom treninga koji se pokazao kao optimalan je 150 000. Agent i sa manjim brojem koraka (oko 100 000) obučavanja nauči da se kreće unapred, ali ne može da savlada krivine. Tek sa većim brojem koraka agent uči da skreće u krivinama. DQN agent može savladati čak i velike krivine, sa kojima ostali modeli imaju poteškoća. Prosečna osvojena nagrada u 100 uzastopnih epizoda ovog modela je 178.

Optimalan broj koraka prilikom treninga A3C agenta je 150 000. Agent jako brzo (za manje od 40 000 koraka) nauči da prati stazu i kreće se, međutim, kao i DQN agentu, potreban mu je veći broj koraka obučavanja da bi naučio da skreće u krivinama. U 100 uzastopnih epizoda A3C agent ostvaruje prosečnu nagradu 241.

Kod A2C metode optimalan broj koraka prilikom treninga je 125 000. Slično A3C agentu, vrlo brzo nauči da se kreće unapred i prati stazu, ali tek sa velikim brojem koraka obučavanja počinje pravilno skretati u krivinama. Prosečna osvojena nagrada u 100 uzastopnih epizoda ovog modela je 263.

Iz tabele 1. vidi se da A2C metoda ostvaruje najveću prosečnu nagradu u 100 uzastopnih epizoda.

Broj koraka treniranja	Metoda	Prosečna osvojena nagrada u 100 epizoda
Slučajne akcije	-	-54
150 000	DQN	178
150 000	A3C	241
125 000	A2C	263

Tabela 1: Poređenje rezultata svih implementiranih metoda

4. ZAKLJUČAK

U ovom radu opisano je obučavanje agenta za autonomno kretanje u simuliranom okruženju *CarRacing-v0*. Agent je obučavan koristeći metode učenja uslovljavanjem. Implementirane metode su: *Deep Q-Network* (DQN), *Advantage Actor Critic* (A2C) i *Asynchronous Advantage Actor Critic* (A3C).

Na osnovu rezultata može se zaključiti da agenti trenirani metodama učenja uslovljavanjem mogu naučiti da se snalaze u okruženjima poput *CarRacing-v0*. Metode korištene u radu su se pokazale kao veoma uspešne u učenju direktno iz vizuelnih podataka. Takođe, sve metode su veoma osetljive na promene hiperparametara. Performanse svih modela u velikoj meri zavise od načina preprocesiranja slike i formiranja stanja.

Actor-critic metodama potrebno je mnogo manje vremena da počnu da konvergiraju i uče. Kod njih se učenje može primetiti već nakon 40 000 koraka, gde agent počinje da prati stazu, ali ima probleme prilikom skretanja. Kod DQN metode učenje se može primetiti nešto kasnije, nakon 100 000 koraka obučavanja. Takođe vreme potrebno za trening *actor-critic* metoda je značajno manje nego kod DQN metode, što je rezultat korištenja multiprocesinga.

Metode A2C i A3C imaju slične karakteristike upravljanja automobilom. Agenti obučavani korištenjem A2C i A3C metoda imaju karakteristike brze vožnje i na taj način pokušavaju osvojiti veće nagrade. Ti agenti imaju pravilno skretanje u većini krivina. Agent obučavan korištenjem DQN metode ima karakteristike sporije vožnje, ali pravilnog skretanja u krivinama i na taj način pokušava osvojiti što veću nagradu. Oba načina vožnje imaju prednosti i mane. Problem koji se pojavljuje kod agenata obučavanih A2C i A3C metodama je skretanje u velikim krivinama, dok problem agenta obučanog DQN metodom je povratak na stazu nakon silaska sa staze.

5. LITERATURA

- [1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, D. Hassabis „*Human-level control through deep reinforcement learning*“
- [2] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Harley, T.P. Lillicrap, D. Silver, K. Kavukcuoglu „*Asynchronous Methods for Deep Reinforcement Learning*“
- [3] <https://gym.openai.com/envs/CarRacing-v0/> [pristupljeno 7.9.2019.]
- [4] S. Ruder „*An overview of gradient descent optimization algorithms*“
- [5] D. P. Kingma, J. L. Ba „*Adam: A Method for Stochastic Optimization*“

Kratka biografija:



Novica Šarenac rođen je u Nevesinju 1994. godine. Osnovne akademske studije iz oblasti Računarstvo i automatika, završio je na Fakultetu tehničkih nauka, 2017. godine., kada upisuje i master akademske studije iz iste oblasti. Položio je sve ispite predložene planom i programom.

NAVIGACIJA U TRODIMENZIONALNOM OKRUŽENJU *Obstacle Tower* UPOTREBOM UČENJA USLOVLJAVANJEM**USING REINFORCEMENT LEARNING TO TRAIN AN AGENT FOR THE *Obstacle Tower* ENVIRONMENT**

Predrag Njegovanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Učenje uslovljavanjem u današnjem poretku stvari kada je veštačka inteligencija u velikom usponu predstavlja povoljno polje za nova istraživanja.. Jedan od problema koji je rešavan poslednjih godinu ili dve jeste problem kontrole okruženja ili navigacije. U ovom radu predstavljen je jedan vid rešenja problema navigacije i generalizacije u trodimenzionalnom okruženju pri postojanju ograničenja nagrada, formiranjem autonomnog agenta tehnikama dubokog učenja. Evaluacija performansi agenta izvršena je poredbom sa ljudskim performansama i rezultatima već opisanih u pratećim naučnim radovima.

Ključne reči: Učenje uslovljavanjem, trodimenzionalno okruženje, kontrola okruženja, navigacija, ograničene nagrade, neuronske mreže

Abstract – Reinforcement learning in today's order of things when artificial intelligence is on the rise is a favorable field for new research. One of the problems that was trying to be solved in the last year or two is the problem of environmental control or navigation. This paper presents one form of solution to the problem of navigation and generalization in a three-dimensional environment is presented while there are limits to rewards, by forming an autonomous agent with deep learning techniques. An evaluation of the agent's performance was performed by comparing it with the human performance and results already described in the accompanying scientific papers.

Keywords: Reinforcement learning, three-dimensional environment, environment control, navigation, sparse rewards, neural network

1. UVOD

Učenje uslovljavanjem je računarski način učenja iz okruženja sa fokusom na postizanje zadatog cilja. S tim na umu učenje uslovljavanjem svoje primene pronalazi u zadacima optimalnog upravljanja, optimizovanja i monitoringa. Problemi učenja uslovljavanjem počinju sa potrebom za još većom količinom podataka, koja uglavnom može biti dobijena kroz simulacije, i iz tog razloga povezana je sa oblastima koja mogu pružiti ovakva okruženja, npr. video igre i robotika.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, vanr. prof.

Pored ogromne količine podataka reprodukcija rezultata iz naučnih radova, nestabilnost modela kao i interpretacija i detekcija grešaka jesu problemi sa kojima se ova oblast mašinskog učenja susreće. Sve ovo jesu razlozi zbog kojih učenje uslovljavanjem nije još pronašlo svoje mesto u industriji i nije isplativo u velikoj meri [1]. Kako su spomenuta okruženja za simulaciju realnog okruženja, poslednja istraživanja svoj fokus stavljaju na posebno ponašanje u uslovima koji oponašaju realan prostor i snalaženje u istom. Naučiti računarske sisteme kako da nauče važne koncepte u svojoj interakciji sa okruženjem predstavlja dodatan problem koji je i dalje u fokusu istraživanja. Pored, ali i zbog svih ovih problema učenje uslovljavanjem predstavlja najzujljiviju oblast mašinskog učenja.

Tema ovog rada dotiče se svih nabrojanih problema i stavlja akcenat na učenje kretanja softverskog agenta u trodimenzionalnom prostoru pri realnom osvetljenju i znatnom većem nivou generalizacije prostora.

2. POSTOJEĆA REŠENJA

Korišćenje tehnika dubokog učenja u učenju uslovljavanjem dovelo je do razvoja autonomnih agenata čije su performanse u igrama, pre svega Atari igrama, prestigle ljudske. Sve Atari igre su dvodimenzionalne prirode i njihova stanja su potpuno vidljiva agentu u svakom trenutku (engl. *fully observable*), što podrazumeva da agentu nije potrebna memorija o prethodnim stanjima kako bi doneo optimalnu odluku.

Za razliku od ovakvih okruženja, pravi svet je trodimenzionalan i donošenje optimalne odluke zahteva pozivanje na niz od prethodnih n stanja. Ovakva okruženja su razmatrana u narednom radu, koji su u potpunosti ili jednim delom dotakao problema navigacije autonomnog agenta kroz prostor.

Rad koji se bavio tematikom primene pseudo nagrada na kretanje agenata jeste rad *Large-Scale Study of Curiosity-Driven Learning* od strane Burda et al. [2] sa kraja 2018. godine. Autori rada su predstavili opširnu studiju u kojoj su obuhvatili veliki broj okruženja, kako 2D tako i 3D, testirajući hipotezu da česte spoljašnje nagrade koje agent dobija od okruženja nisu skalabilne već da je neophodno agentu pružiti unutrašnje nagrade. Ovaj tip nagrada su oni nazvali znatiželja agenta (engl. *curiosity*). Znatiželja agenta se može interpretirati u kontekstu problema o kom pričamo u ovom radu i kao želja za posetom novih stanja i istraživanja okruženja. Autori definišu znatiželju kao unutrašnju nagradu koju definiše sam agent, koja

predstavlja predviđanje posledica akcija u zavisnosti od trenutnog stanja. Kreiranje unutrašnjih nagrada zadatak je modula koji je nazvan *Intrinsic-Curiosity module* (ICM). Model nad kojim se vrše eksperimenti je u ovom radu u drugom planu. Fokus rada je deo modela koji vrši ekstrakciju *feature*-a iz vizuelnih signala. Rezultati opsežnog eksperimenta pokazali su da agenti uspevaju da ostvare optimalnu politiku čak i ukoliko koriste samo generisane unutrašnje nagrade. Korišćenje samo unutrašnjih nagrada pokazalo se korisnim u Atari okruženjima, Super Mario okruženju i dr. Naravno, moguće je koristiti spoljašnju i unutrašnju nagradu u kombinaciji, što je poželjno u nekim okruženjima koja su dizajnirana sa namerom eksploatacije spoljašnje nagrade.

3. METODOLOGIJA

U ovoj sekciji pristupa se detaljnom opisivanju okruženja sa kojim agent vrši interakciju i ideja koje se kriju iza implementacije ovakvog zadatka učenja uslovljavanjem. Agent je implementiran u programskom jeziku *Python* [3]. Programski jezik *Python* postao je defakto *go-to* jezik za razvoj modula mašinskog učenja i aplikacija veštačke inteligencije zbog brzine razvijanja prototipa rešenja do postojanja mnogobrojnih biblioteka koje podržavaju razvoj.

3.1. Obstacle Tower video igra

Video okruženje *Obstacle Tower* [4] je trodimenzionalna video igra gde korisnik kontroliše agenta sa pogledom kamere iz trećeg lica (slika 1). Okruženje se sastoji od 100 spratova (prve verzije igre su sadržale samo 24 sprata), pri čemu agent počinje sa nultog sprata.



Slika 1. Igra Obstacle Tower

Cilj igre je rešiti toranj, odnosno proći kroz svih 100 spratova. Svaki sprat sadrži makar sobu u kojoj počinje sprat i u kojoj se završava. Spratovi mogu sadržati više soba od kojih neke mogu sadržati slagalice, neprijatelje, prepreke ili mogu zahtevati ključ kako bi se nastavilo dalje. Sa porastom spratova kompleksnost igre raste. Kretanje agenta u okruženju ograničeno je *in-game* vremenom koje se nadoknađuje rešavanjem sprata ili skupljanjem loptica (engl. *time orbs*) koje su nasumično razbacane po sobama.

Okruženje se renderuje koristeći osvetljenje i senčenje u realnom vremenu sa mnogo detaljnijim teksturama i modelom. Okruženje *Obstacle Tower* uključuje u sebe nekoliko varijacija koje se ogledaju kroz texture, svetlosne uslove i geometriju objekata.

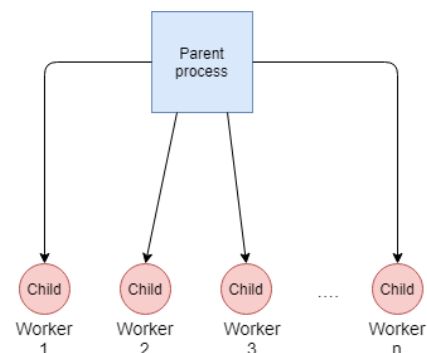
Prostor stanja agenta se sastoji od dva tipa informacija. Prvi tip observacije jeste renderovana piksel reprezentacije slike okruženja iz perspektive trećeg lica. Slika je renderovana u veličini 168x168 kao RGB slika (sadrži sva tri kanala boja, crveni, zeleni i plavi). Pored slike okruženja, drugi tip podataka jeste vektor pomoćnih vrednosti. U pomoćne informacije spadaju: broj ključeva koje agent poseduje u datom trenutku, vreme koje mu je preostalo u okviru trenutne igre i trenutni sprat na kom se nalazi.

Prostor akcija agenta je multi-diskretan što znači da se sastoji od skupa manjih diskretnih prostora akcija čija unija odgovara jednoj akciji u okruženju. Podprostori akcija obuhvataju sledeće skupove akcija: kretanje unapred, kretanje unazad, bez kretanja, zatim kretanje ulevo, kretanje udesno, mirovanje, skok ili bez skoka i rotacija kamere u smeru kazaljke na satu, obrnuto od kazaljke sata i bez rotacije kamere.

Okruženje *Obstacle Tower* jeste okruženje sa "retkim nagradama". *Obstacle Tower* pruža agentu nagradu +1 ukoliko uspešno završi ceo sprat, a +0.1 ukoliko uspešno otvori vrata, reši slagalicu ili pokupi ključ. Razređenost nagrada stavlja dodatan akcenat na korišćenje unutrašnjih nagrada i posebno oblikovanje nagrade dobijene od strane okruženja.

3.2. Paralelizacija okruženja

Pri rešavanju zadataka učenja uslovljavanjem potrebno je kontrolisati okruženje. Pod kontrolom okruženja podrazumeva se izdavanje komandi za pokretanje sledećeg koraka izvršavanja, postavljanje okruženja u početno stanje i uzorkovanje akcija iz prostora akcija. Na slici 2. predstavljen je na konceptualnom nivou dijagram koji opisuje rad modula za paralelizaciju.



Slika 2. Dijagram komunikacije između procesa

Svaki *worker* otvara po jednu instancu *Obstacle Tower* okruženja. *Parent* proces održava kontrolu nad *child* procesima i šalje im komande koje je potrebno izvršiti. Komande koje *parent* proces šalje su komande kontrole nad okruženjem. U ove komande spadaju komanda uzorkovanja koja vraća nasumičnu akciju iz prostora akcija, zatim komanda narednog koraka koja govori okruženju da pređe u naredni vremenski korak vraćajući pri tom observacije i komanda resetovanja koja postavlja okruženje u početno stanje. Prilikom slanja komandi, *parent* proces prolazi kroz listu svih *child* procesa i šalje im signal. Dobijeni podaci se agregiraju u *parent* procesu u obliku tenzora i prosleđuju dalje.

3.3. Model i interfejs ka modelu

Modeli su definisani kroz *PyTorch*, biblioteku za mašinsko učenje. Konvolutivna neuronska mreža ima strukturu zasnovanu na radu [5] objavljenom u *Nature* časopisu. Prvi sloj neuronske mreže ima 32 filtera sa veličinom kernela 8x8 i korakom pomeranja kernela 4. Drugi sloj mreže ima 64 filtera, kernel 4x4, korak 2 i treći sloj takođe ponavlja 64 filtera sa veličinom kernela 3x3 i korakom 1. Na izlaz iz svakog kernela se primenjuje *LeakyReLU* aktivaciona funkcija.

Između slojeva je upotrebljena i *batch normalization* tehnika [6] za uklanjanje pomeraja u distribuciji podataka. Rekurentna neuronska mreža inicijalizovana je sa GRU ćelijama i veličinom matrice za stanje ćelije od 512 kolona. Zadržan je samo jedan sloja rekurentne mreže. Linearni slojevi za aproksimaciju funkcije stanja i politike su definisani shodno traženim dimenzijama tj. bile su uslovljene veličinom izlaza rekurentne neuronske mreže.

Arhitektura ICM modela ispraćena je kroz naučni rad koji ju je i predstavio. Blok za ekstrakciju *feature*-a je identična konvolutivna mreža koju koristi i osnovni model uz razliku u izlaznoj transformaciji, gde izlazni vektor sadrži 288 elemenata. Forward model je kombinacija tri potpuno povezana sloja sa 256, 512 i 288 neurona. Aktivacija svakog sloja je *LeakyReLU*. Inverse model jeste potpuno povezana mreža sa četiri sloja izlaznih dimenzija 256, 512, 512 i veličinom prostora akcija, respektivno. Izlazna aktivaciona funkcija jeste *softmax* funkcija.

Interfejs ka modelima sakriva modele i otkriva metode za manipulaciju istima. Metode koje Tower Agent pruža su: metoda za propagaciju podataka unapred kroz osnovni model, metoda za propagaciju podataka unapred kroz ICM model, metoda za računanje funkcije gubitka sinhronog *actor-critic* algoritma i metoda za računanje funkcije gubitka *proximal policy optimization* algoritma.

3.4. Skladištenje observacija

Skladištenje observacija poslužilo je kao prikladan način za uzorkovanje trajektorija jednog od više paralelnih okruženja kao i za računanje očekivanih nagrada za svaki vremenski korak do kraja epizode.

Skladište se sledeće informacije: stanje okruženja, nagrade, terminalna stanja, prediktovane vrednosti funkcije stanja, odabrane akcije za svaki vremenski korak i vrednosti politike za svaki vremenski korak. Dodatno se čuvaju i vektori stanja koji se koriste prilikom računanja funkcije greške ICM modela. Osnovne operacije su dodavanje, brisanje ali i uzorkovanje prilikom treninga agenta. Uzorkovanje iskustava se vrši za sinhronu *actor-critic* metodu kao i za *proximal policy optimization*.

3.5. Obučavanje agenta

Parametri koji kontrolišu tok obučavanja agenta su: ukupan broj vremenskih koraka agenta, količina prikupljenog iskustva, broj epoha ažuriranja, stopa učenja i algoritam obučavanja. Preostali parametri inicijalizuju ostale neophodne module aplikacije. Ukupan broj koraka agenta se uobičajeno zadaje pre početka treninga i direktno određuje dužinu treninga. Usled hardverskih ograničenja, izabrani broj koraka je 5 miliona. Uobičajeno se agent trenira sa 20-50 miliona koraka. Trening agenta započinje izračuna-

vanjem broj ažuriranja, a dva najvažnija koraka treniranja agenta jesu prikupljanje iskustva agenta i primena iskustva na ažuriranje modela. Petlja za prikupljanje iskustva agenta iznosi 128 koraka. Zbog jednostavnosti problem je definisan kao epizodni, što podrazumeva da se prikupljanje iskustva završava kada se dođe do terminalnog stanja ili nakon ispunjenja svih 128 koraka. Biranje sledeće akcije se vrši *on-policy* pristupom, iz trenutne politike se uzorkuje nasumična akciju i primeni se na okruženje. Observacije koje okruženje pruža puštaju se kroz osnovni model i ICM model gde se kao izlaz dobijaju vrednosti funkcije stanja, politike i unutrašnje nagrade. Treba napomenuti da se pored stanja, osnovnom modelu pruža i poslednje stanje ćelije rekurentne mreže.

Na taj način održavamo zavisnost između vremenskih koraka i modelujemo sekvencu događaja u rekurentnoj neuronskoj mreži. Ovi podaci smeštaju se u bafer iskustva. Nakon prikupljanja iskustva pristupa se ažuriranju modela. Ažuriranje modela zavisi od primenjenog algoritma. U slučaju *Synchronous actor-critic* ne postoje ciklusi ažuriranja niti višesutruka ažuriranja jer priroda algoritma to ne dozvoljava. Uzorkovanjem bafera iskustva se dobijaju relevantni podaci neophodni za ažuriranje modela. Definisani algoritam za optimizaciju funkcije gubitka, u slučaju oba algoritma, jeste Adam [7] i inicijalno postavljena vrednost stope učenja iznosi 0.0004 što se pokazalo kao dovoljno velika vrednost da agent počne brzo da uči ali ne i prevelika da učenje divergira. Vrednost funkcije gubitka se normalizuje brojem epoha ažuriranja u slučaju PPO algoritma i vrši se propagacija unazad. Svi delovi funkcije gubitka se čuvaju u definisanim tenzorima radi prikazivanja na grafikonima.

4. EVALUACIJA REŠENJA I DISKUSIJA REZULTATA

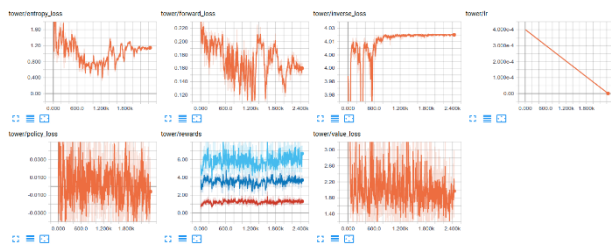
Standardna procedura po završetku obučavanja modela jeste provera performansi modela koja će pokazati da li model zadovoljava postavljene kriterijume.

Prosečne ljudske performanse u okruženju *Obstacle Tower* prezentovane su u već spomenutom naučnom radu koji je pratio objavu okruženja [4] i iznose 15 spratova po prosečnoj igračkoj sesiji bez dinamičkog generisanja nivoa. Sa istim postavkama okruženja najbolji modeli zasnovani na čistom *PPO* i *Rainbow* algoritmu dostižu 5 odnosno 7 spratova. Model *Tower agent* je obučavan u perspektivama sa pogledom iz trećeg lica i sa pogledom iz prvog lica. Pretpostavka vezana za perspektivu iz prvog lica jeste da će otkloniti suvišan prikaz prostora a s tim i *feature*-e koje pruža perspektiva iz trećeg lica i da će time ostvariti bolje performanse. Modeli su puštani 10 puta i uzimala se maksimalna vrednost broja spratova koje je agent uspeo da reši. Rezultati su prikazani tabelom 1.

	<i>Proximal Policy Optimization</i>	<i>Synchronous actor-critic</i>
<i>First person perspective</i>	1	1
<i>Third person perspective</i>	3	2

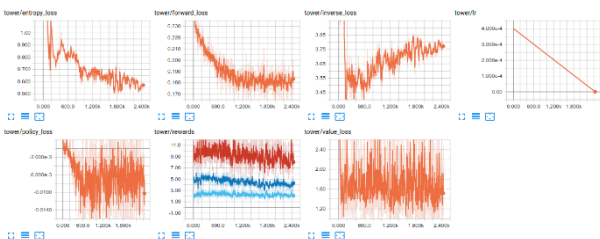
Tabela 1. Prosečan broj rešenih spratova u zavisnosti od perspektive kamere

Na osnovu rezultata iz table 1. izvlače se zaključci da je postavljena hipoteza zadovoljena u slučaju kada je pogled na okruženje u perspektivi trećeg lica uz bolje rezultate *PPO* algoritma. Pokazuje se da agent ostvaruje lošije performanse u perspektivi prvog lica. Posmatranjem agenta u fazi zaključivanja primećuju se problemi koje ima sa samim okruženjem. Često tokom kretanja, realistično osvetljenje i probijanje ivica okruženja “hvata” nepotrebne *feature*-e koji u krajnjem slučaju ne vode do dobrih rezultata. Grafici (slika 3) koji opisuju obučavanje *A2C* algoritmom pokazuju nestabilnost prilikom treninga i veliko oscilovanje greške funkcije politike.



Slika 3. Tok obučavanja *A2C* algoritma za perspektivu iz prvog lica

Performanse agenta obučanim *PPO* algoritmom sa perspektivom kamere iz trećeg lica, što i jeste podrazumevano podešavanje, zadovoljava postavljenu hipotezu o prelasku jednog sprata ali ne uspeva da ostvari rezultate prezentovane u naučnom radu koji prati okruženje. Maksimalan sprat koji agent dostiže jeste sprat broj tri. Međutim uzevši u obzir da je trening agenta koji se opisuje u ovom radu trajao deset puta kraće, rezultati su zadovoljavajući. Proces obučavanja kroz grafike funkcija gubitaka i nagrade prikazan je slikom 4.



Slika 4. Tok obučavanja *PPO* algoritam za perspektivu iz trećeg lica

5. ZAKLJUČAK

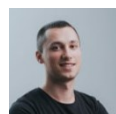
Kroz ovaj rad predstavljena je aplikativna primena učenja uslovljavanjem na jednom od poslednjih istraživačkih problema. Za potpunu kontrolu nad okruženjem potrebna su neka od kompleksnijih rešenja kao npr. *behavioral cloning* algoritam gde agent uči da oponaša ljudski performans u okruženju, korišćenje KL-divergence faktora umesto funkcije entropije, GAIL (engl. *Generative Adversarial Imitation Learning*) metode ili kombinacija istih. Sam proces obučavanja agenta je složen i zahtevan sa strane vremenskih i hardverskih resursa. Novi trend obučavanja agenta kroz skupljanje iskustva kroz više paralelnih okruženja zadovoljava potrebu za velikim količinama podataka ali povećava hardverske zahteve, pre svega u vidu procesorske moći, radne memorije ali najviše video memorije. Naravno da bi svi resursi bili optimalno iskorišćeni, celokupnu aplikaciju je potrebno napisati na pametan način. *Obstacle Tower* okruženje se

pokazalo kao vrlo izazovno za savladavanje i spremno je da gurne granice istraživanja u učenju uslovljavanjem. Izazovi sa vizuelnog aspekta, sa aspekta kontrole i planiranja ali i nagrada koje pruža okruženje zahtevaju ozbiljno poznavanje materije i ostavljaju prostor za dalja istraživanja. Ovo okruženje svakako potkrepljuje tvrdnju da vremenom nastaju sve kompleksnija okruženja koja bivaju gurnuta na pozornicu u ovoj oblasti. Poslednji pravci istraživanja u trenutnoj postavci stvari obuhvataju probleme vizije, planiranja, kontrolisanja trodimenzionalnih okruženja kao i okruženja sa retkim nagradama. Pored svih nabrojanih problema, implementacionih i aplikativnih, učenje uslovljavanjem jeste jedna od najuzbudljivijih oblasti veštačke inteligencije i veliki uspesi na ovom polju se tek očekuju.

6. LITERATURA.

- [1] AY Ng. (2017, December 14). Practical applications of reinforcement learning in industry. Retrieved June 24, 2019 from <https://www.oreilly.com/ideas/practical-applications-of-reinforcement-learning-in-industry>
- [2] Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., & Efros, A. A. (2018). Large-scale study of curiosity-driven learning. arXiv preprint arXiv:1808.04355.
- [3] Python. (n.d). Retrieved from <https://www.python.org/>
- [4] Juliani, A., Khalifa, A., Berges, V. P., Harper, J., Henry, H., Crespi, A., ... & Lange, D. (2019). Obstacle Tower: A Generalization Challenge in Vision, Control, and Planning. arXiv preprint arXiv:1902.01378.
- [5] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.
- [6] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
- [7] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kratka biografija:



Predrag Njegovanović je rođen 4.10.1994. u Vukovaru, Republika Hrvatska. Osnovnu školu „Slobodan Bajić Paja“ završio je 2009. godine. Iste godine upisuje srednju tehničku školu „Nikola Tesla“, smer elektrotehničar računara. Srednju školu završava 2013. godine i upisuje osnovne akademske studije na smeru Računarstvo i automatika, Fakulteta tehničkih nauka u Novom Sadu. Zvanje diplomirani inženjer elektrotehnike i računarstva stiče 2017. godine, sa prosečnom ocenom 9,59, uz specijalizaciju računarske nauke i informatika, modul inteligentni sistemi. Nakon toga, iste godine, upisuje master akademske studije na Fakultetu tehničkih nauka u Novom Sadu, odsek Elektrotehnika i računarstvo, smer Računarstvo i automatika, modul Inteligentni sistemi. Položio je sve ispite predviđene planom i programom master studija uz prosečnu ocenu 10,0.

ULAZNI STEPEN EDUKATIVNOG ELEKTROFIZIOLOŠKOG AKVIZICIONOG UREĐAJA**FRONTEND OF AN EDUCATIONAL ELECTROPHYSIOLOGICAL ACQUISITION DEVICE**

Boris Knežević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Rad predstavlja realizaciju edukativnog elektrofiziološkog akvizicionog uređaja pomoću *Mikromedia for Xmega* razvojnog okruženja i akvizicionog modula baziranog na ADS1299 integrisanom kolu. U radu su detaljno opisane komponente sistema, kao i programski kod. Na kraju rada dati su rezultati i njihova analiza.

Ključne reči: Mikrokontroler, elektrofiziološki signali

Abstract – The paper represents the realization of an educational electrophysiological acquisition device using the *Mikromedia for Xmega* developing platform and the ADS1299 microchip. It describes the components of the system as well as the program code. The end of the paper displays the results and the analysis of those results.

Keywords: Microcontroller, electrophysiological signal

1. UVOD

Biomedicinsko inženjerstvo predstavlja multidisciplinarnu oblast sastavljenu od niza užih naučnih delatnosti. Sa tehničke strane, to su metode obrade signala, elektronika, sistemi automatskog upravljanja, merno-akvizicioni sistemi itd. Sa medicinske strane, to su anatomija, fiziologija, neurologija, histologija itd [1].

Na Fakultetu tehničkih nauka Univerziteta u Novom Sadu realizovan je studijski program Biomedicinsko inženjerstvo koji se upravo bavi izučavanjem prethodno navedenih oblasti. Određeni broj kurseva na ovom studijskom programu se bavi akvizicijom i analizom elektrofizioloških signala.

Zbog tehničke složenosti akvizicionih elektrofizioloških uređaja i standarda koji oni moraju da zadovoljavaju, pre svega u smislu zaštite subjekata nad kojima se ti signali snimaju, studenti koji izučavaju ovaj studijski program veoma teško mogu samostalno da snime elektrofiziološke signale koje bi nakon toga mogli da obrađuju i analiziraju.

Otud je proistekla ideja za realizacijom edukativnog elektrofiziološkog akvizicionog uređaja koji ima zadatak da bolje upozna studente sa ovom granom biomedicinskog inženjerstva i da im pruži neophodno praktično iskustvo koje će im pomoći u daljem obrazovanju i usavršavanju.

U poslednjih nekoliko godina, ova oblast je značajno počela da se razvija i proširuje, pa tako već postoji značajan broj naučnih radova koji se upravo bave analizom i razvojem elektrofizioloških akvizicionih uređaja koji se primenjuju u edukaciji. Edukativni elektrofiziološki merno-akvizicioni sistem opisan je u radu [3]. Ovaj naučni rad opisuje prototip merno-akvizicionog sistema čiji je osnovni zadatak da reši probleme prenosivosti, lošeg odnosa signal-šum, pristupačnosti korisnicima itd. Sistem je, pre svega, služio za akviziciju EKG signala i fotopletizmografiju, a realizovan je primenom mikrofona i zvučne kartice. Za obradu i prikazivanje prikupljenih podataka korišćeno je razvojno okruženje MATLAB.

Merno-akvizicioni uređaj opisan u radu [4] takođe je korišćen u edukativne svrhe. Uređaj je razvijen na *Raspberry PI* platformi i služio je za merenje EKG signala, a pored toga, implementirano je i merenje impedanse interfejsa elektroda-tkivo. Prikupljeni podaci su se, zatim, slali putem *Bluetooth* konekcije na PC računar gde se vršila njihova obrada i vizuelizacija.

U naučnom radu [5] opisan je merno-akvizicioni sistem koji je primarno realizovan kako bi ga koristili student u toku edukacije. Signali su se prikupljali putem senzora, a do PC računara su se slali putem RS232 interfejsa. Za prikazivanje prikupljenih podataka korišćeno je *LabVIEW* razvojno okruženje.

Pored prethodno navedenih merno-akvizicionih sistema koji najčešće koriste PC računare i aplikacije, postoje i sistemi koji uključuju pametne telefone. Uređaj opisan u radu [6] razvijen je na *Raspberry PI* platformi. Prikupljeni podaci su putem *wireless* komunikacije dalje prosledeni do mobilnog telefona, gde se vršila vizuelizacija tih podataka.

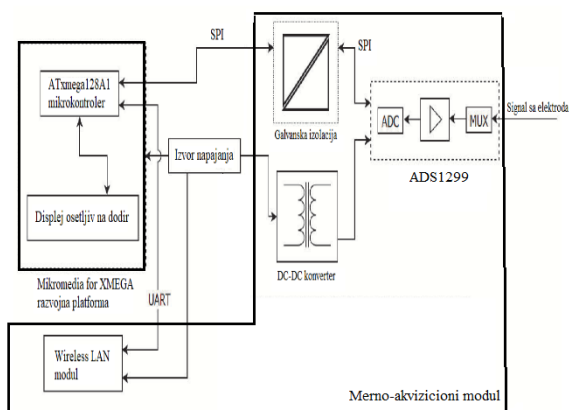
Cilj ovog rada je da prikaže realizaciju edukativnog elektrofiziološkog akvizicionog uređaja primenom *Mikromedia for Xmega* razvojnog sistema razvijenim oko mikrokontrolera ATxmega128A1 i akvizicionog modula baziranog na ADS1299 integrisanom kolu. Uređaj je prenosiv, intuitivan i jednostavan za upotrebu, a namenjen je za snimanje električne aktivnosti srca i mišića. Navedene tri osobine ukazuju na to da je uređaj pogodan upravo za studentsku upotrebu u toku izvođenja laboratorijskih vežbi.

2. HARDVER

Hardverski deo uređaja se sastoji iz dva dela: *Mikromedia for XMEGA* razvojne platforme [7,8] i akvizicionog modula, razvijenog na Fakultetu tehničkih nauka u Novom Sadu, baziranog na ADS1299 integrisanom kolu [9]. Strukturni blok dijagram sistema prikazan je na slici 1.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Nikola Jorgovanović, red. prof.



Slika 1. Strukturni blok dijagram sistema

2.1. Merno-akvizicioni modul

Centralni deo merno-akvizicionog modula je ADS1299 integrisano kolo. Proizvod je kompanije *Texas Instruments* i najčešće se koristi za snimanje električne aktivnosti kore velikog mozga (elektroencefalogram, tj. EEG signal) i električne aktivnosti srca (elektrokardiogram, tj. EKG signal). Osnovna uloga ovog modula jeste snimanje i prikupljanje elektrofizioloških signala, obrada i pojačanje tih signala i njihova konverzija iz analognog u digitalni oblik. U pitanju je osmokalni modul koji sadrži osam diferencijalnih ulaza sa implementiranim *antialiasing* filtrom, osam 24-bitnih delta-sigma ($\Delta\Sigma$) analogno-digitalnih konvertora i osam pojačavača sa promenljivom vrednošću pojačanja (*Programmable Gain Amplifier, PGA*). Pored toga, modul sadrži i multiplekser, interni generator test signala, interni oscilator, DRL (*Driven Right Leg*) kolo i SPI (*Serial Peripheral Interface*) modul za komunikaciju sa drugim uređajima. Pored ADS1299 merno-akvizicioni modul sadrži još i modul za napajanje i galvansko razdvajanje, interfejs za pulsni oksimetar i WiFi komunikacioni modul.

2.2. Mikromedia for XMEGA

Mikromedia for XMEGA razvojna platforma proizvod je kompanije *MikroElektronika* i predstavlja najvažniju hardversku komponentu ovog sistema. Ova razvojna pločica sadrži sve neophodne resurse neophodne za pravilnu akviziciju elektrofizioloških signala. Centralni deo razvojne pločice čini 8-bitni mikrokontroler ATmega128A1. Osnovni zadatak ovog mikrokontrolera je da reguliše procese akvizicije, obrade i vizuelizacije signala.

Kao što je već napomenuto, osnovna komponenta razvojne pločice je mikrokontroler ATmega128A1. U pitanju je mikrokontroler iz AVR familije mikrokontrolera koji sadrži 8-bitni mikroprocesor RISC (*Reduced Instruction Set Computer*) arhitekture i 100 ulazno/izlaznih pinova. Predviđen napon napajanja se kreće u granicama između 1,6 V i 3,6 V. Što se tiče memorije, mikrokontroler poseduje 128 kilobajta programske ili fleš (*flash*) memorije, 8 kilobajta SRAM (*Static Random Access Memory*) memorije i 2048 bajtova EEPROM (*Electrically Erasable Programmable Read-Only Memory*) memorije. Mikrokontroler sadrži osam 16-bitnih tajmera i brojača, dva 12-bitna A/D konvertora, dva 12-bitna D/A konvertora, kao i modul za serijsku komunikaciju. Frekvencija rada mikroprocesora iznosi 32 MHz.

3. FIRMVER

Za pisanje firmvera korišćeno je radno okruženje *mikroC PRO for AVR*, razvijeno od strane kompanije *Mikroelektronika*. *MikroC PRO for AVR* je programsko okruženje za programiranje mikrokontrolera u programskom jeziku C.

3.1 Inicijalizacija

Dovođenjem odgovarajućeg napajanja, mikrokontroler počinje sa radom, nakon čega kreće njegova inicijalizacija. Prva funkcija koja se izvršava je funkcija *Init()*, koja služi za definisanje vrednosti flegova, koji će imati važnu ulogu u kontrolisanju daljeg toka programa. Flagovi spadaju u promenljive tipa bit i mogu imati dve vrednosti: 1, tj. *TRUE* i 0, tj. *FALSE*. Prilikom inicijalizacije, vrednosti svih flegova se postavljaju na *FALSE*, osim promenljive *I_bit* čija se vrednost postavlja na *TRUE*. Promenljiva *I_bit* kontroliše globalnu dozvolu za izvršavanje prekidnih rutina. Pored *I_bit*-a, da bi se omogućilo izvršavanje prekidnih rutina svih prioriteta, potrebno je podesiti poslednja tri bita 8-bitnog *PMIC_CTRL* registra na 1. Nakon izvršavanja ove funkcije, kreće inicijalizacija prekidnih rutina koje će se koristiti u toku izvršavanja glavnog programa. Prekidne rutine koje treba inicijalizovati su prekid SPI komunikacije (funkcija *SPIInit()*) i eksterni prekid na pinu EEG_DRDY (funkcija *DRDYInit()*).

SPI komunikacioni protokol se koristi za razmenu podataka između ATmega128A1 mikrokontrolera i ADS1299 integrisanog kola. Mikrokontroler ATmega128A1 predstavlja master uređaj, a integrisano kolo ADS1299 predstavlja slave uređaj, što je u kodu definisano podešavanjem *SPID_CTRL* registra.

Poslednji prekid koji je potrebno inicijalizovati je DRDY eksterni prekid na pinu E6. Eksterni prekid na određenom pinu se definiše postavljanjem 1 na na odgovarajućem bitu *PORTE_INT0MASK* registra. Mikroprocesor izvršava ovu prekidnu rutinu u trenutku kada je završeno prikupljanje novih podataka sa svih A/D konvertora integrisanog kola ADS1299. Ova prekidna rutina predstavlja osnovu akvizicije podataka, i zbog toga je prioritet ovog prekida postavljen na najviši nivo. To je urađeno postavljanjem 11 na poslednja dva bita registra *PORTE_INTCTRL*.

3.2 Akvizicija signala

Postavljanjem vrednosti *EEG_ENABLE* i *EEG_START* pinova na *TRUE*, započinje akvizicija signala. U trenutku kada su prikupljene vrednosti signala na svim kanalima (3 bajta za svaki kanal, ukupno 24 bajta podataka), aktivira se eksterni DRDY prekid na pinu E6. U toj prekidnoj rutini, vrednost *flag-a read_data*, koji omogućava čitanje prikupljenih podataka, postavlja se na *TRUE*. Podaci se nakon toga iščitavaju u prekidnoj rutini SPI komunikacije. Prekidna rutina SPI komunikacije započinje iščitavanjem *SPID_STATUS* i *SPID_DATA* registara. Prikupljeni podaci nalaze se u *SPID_DATA* registru. U toku jednog izvršavanja prekidne rutine moguće je iščitati samo jedan bajt podataka. Zbog toga se uvodi brojač *read_byte_counter* koji sadrži informaciju koliko je bajtova podataka iščitano. Iščitani podaci se smeštaju u niz *signal_all_channels* i šalju se dalje na obradu. Kada se preuzmu svih 24 bajtova podataka, vrednost *flag-a read_data* se postavlja na *FALSE*. Postupak se ciklično ponavlja svaki put kada se prikupi nova 24 bajta podataka.

3.3 Obrada podataka

Nakon akvizicije podataka kreće njihova obrada. Kao što je već rečeno, vrednost signala snimljenog na jednom od 8 kanala opisana je sa 3 bajta koji su smešteni u 3 različite promenljive. Prvi bajt predstavlja najviši bajt u paketu, dok je treći najniži bajt u paketu. Prvi zadatak jeste ta 3 bajta upakovati u jednu 32-bitnu promenljivu tipa *long*. Prvi korak ovog zadatka predstavlja eksplicitnu konverziju ta tri 8-bitna podatka u tri 32-bitna podatka. Nakon toga, primenjuje se operacija nad bitima poznata kao šiftovanje. Prvi bajt sadrži najznačajnije bite, pa se zbog toga oni pomeraju (šiftuju) za 16 mesta ulevo. Drugi bajt sadrži bite srednje značajnosti, pa se zato oni šiftuju za 8 mesta ulevo, dok treći bajt sadrži najmanje značajne bite, pa zbog toga oni ostaju na pozicijama na kojima se trenutno nalaze. Nakon šiftovanja, tri 32-bitna podatka se spajaju u jednu promenljivu primenom logičke operacije ili (*OR*).

Dobijena 32-bitna promenljiva predstavlja vrednost signala u komplementu dvojke. Dobijena vrednost je neoznačena (sve vrednosti su pozitivne), i nalazi se u opsegu od 0 do $2^{24} - 1$. Ove vrednosti je potrebno konvertovati iz neoznačenog (*unsigned*) tipa u označeni (*signed*) tip. Kod reprezentacije brojeva u komplementu dvojke, prvu polovinu vrednosti (od 0 do $2^{23} - 1$) predstavljaju pozitivni brojevi, dok drugu polovinu vrednosti (od 2^{23} do $2^{24} - 1$) predstavljaju negativni brojevi. Zbog toga, da bi se dobila označena vrednost, neophodno je proveriti da li je ova dobijena vrednost veća od polovine vrednosti celokupnog opsega. Ukoliko jeste, od vrednosti se oduzima maksimalna vrednosti opsega, tj. 2^{24} . Ukoliko nije, vrednost ostaje nepromenjena.

Naredni zadatak koji je potrebno obaviti je izvršiti konverziju vrednosti signala iz komplementa dvojke u decimalnu vrednost, tj. u vrednost u voltima. To se radi uz pomoć sledeće formule:

$$1 \text{ LSB} = (2 \times V_{REF} / \text{Gain}) / 2^{24} \quad (1),$$

pri čemu V_{ref} iznosi 4,5 V, a $Gain$ predstavlja pojačanje PGA pojačavača i iznosi 24. Dobijena konvertovana vrednost je izražena u voltima. Iz razloga što elektrofiziološki signali imaju amplitude reda milivolti, čitav izraz je pomnožen sa 1000.

Opisanom konverzijom dobijeni su odbirci snimanog elektrofiziološkog signala. Pre njegovog iscertavanja, potrebno ga je filtrirati. Snimljeni signal poseduje pomešanu baznu liniju i određeni visokovremenski šum. Za eliminaciju pomerene bazne linije realizovan je visokopropusni IIR (*Infinite Impulse Response*) filter trećeg reda i granične učestanosti 0,5 Hz. Visokofrekventni šum eliminisan je iz signala primenom tehnike usrednjavanja. Usrednjena vrednost se dobija računanjem srednje vrednosti 15 prethodno izračunatih odbiraka signala. Elektrofiziološki signal se nakon svih obrada šalje na iscertavanje.

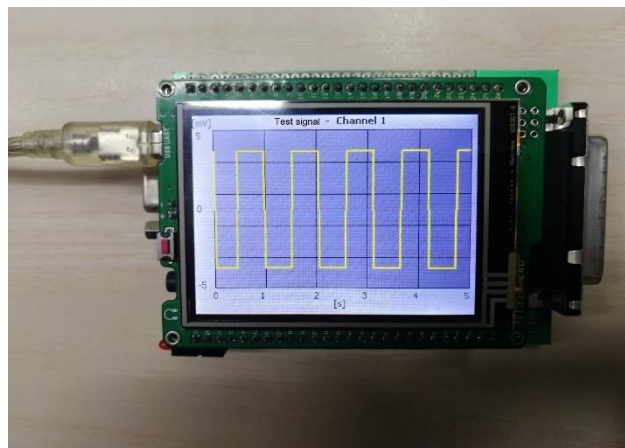
4. REZULTATI

Proces testiranja rada uređaja podeljen je na dve faze:

1. Snimanje signala sa test generatora,
2. Snimanje elektrofiziološkog signala.

4.1 Test signal

Signal dobijen preko internog test generatora predstavlja povorku pravougaonih impulsa amplitude 3,75 mV i periode od 0,97 s. Na slici 2 prikazan je snimljeni signal na vremenskom intervalu od 0 do 5 sekundi i amplitudskom rezolucijom od ± 5 mV.

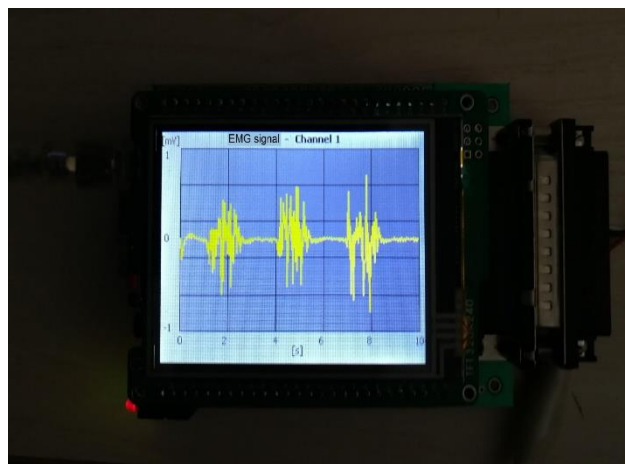


Slika 2. Test signal

4.2 Elektrofiziološki signali

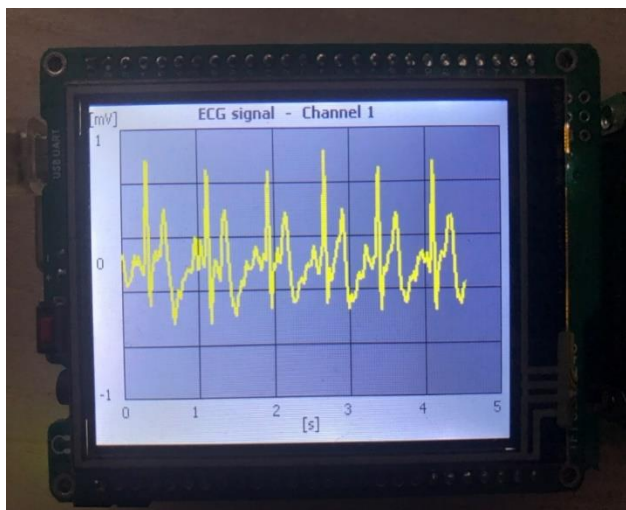
Prilikom testiranja rada uređaja snimljena su dva tipa elektrofizioloških signala: EMG i EKG signal.

Za snimanje EMG signala korišćena je bipolarna konfiguracija elektroda (pozitivna, negativna i DRL elektroda) postavljena na *flexor carpi ulnaris* mišiću. Frekvencija odabiranja uređaja je podešena na 500 Hz. Razlog tome je veći frekvencijski opseg koji obuhvata sam EMG signal (najznačajnije frekvencije se nalaze u opsegu od oko 50 do oko 150 Hz). Filtrirani EMG signal sniman na vremenskom intervalu od 0 do 10 sekundi i sa amplitudskom rezolucijom od ± 1 mV prikazan je na slici 3.



Slika 3. EMG signal

Za snimanje EKG signala korišćen je prvi bipolarni odvod. Pozitivna elektroda je postavljena na desnu ruku, dok su negativna i DRL elektroda postavljene na levu ruku. Frekvencija odabiranja je podešena na 250 Hz. Razlog tome je niži frekvencijski opseg koji obuhvata sam EKG signal. Filtrirani EKG signal sniman na vremenskom intervalu od 0 do 5 sekundi i sa amplitudskom rezolucijom od ± 1 mV prikazan je na slici 4.



Slika 4. EKG signal

5. ZAKLJUČAK

Realizovani edukativni elektrofiziološki akvizicioni uređaj zasnovan na Mikromedia for XMEGA razvojnoj pločici i integrisanom kolu ADS1299 omogućava korisniku, pre svega studentima, da vrše merenja elektrofizioloških signala tokom izvođenja laboratorijskih vežbi i pritom steknu praktično znanje iz oblasti elektrofiziologije i merno-akvizicionih sistema. Uređaj je jednostavan za upotrebu, prenosiv, a odlikuje ga stabilnost u radu.

U radu su detaljno opisane hardverske komponente koji čine ovaj uređaj, kao i firmver koji se izvršava na mikrokontroleru Atxmega128A1. Posebna pažnja posvećena je procesu akvizicije samih signala.

Uzimajući u obzir elektrofiziološke akvizicione sisteme opisane u uvodu ovog rada, performanse realizovanog uređaja se mogu poboljšati implementacijom merenja impedanse interfejsa elektroda-tkivo na samom mikrokontroleru. Konačno, uređaj bi bio potpun kada bi se isprogramirao i modul za pulsnu oksimetriju, što bi obezbedilo i mogućnost merenja nivoa oksigenisanosti krvi u organizmu.

6. LITERATURA

- [1] D. Popović, M. Popović, „Biomedicinska merenja i instrumentacija“, Akademska misao, Beograd 2007.
- [2] O. Tomašević, L. Mejić, D. Stanišić, V. Ilić, F. Gašparić, „A Portable Device for Physiological Measurements in Biomedical Engineering Education“, Serbian Journal of Electrical Engineering, Vol. 16, No.1, 55-77, 2019.
- [3] R. Stojanović, D. Karadaglić, „An Economical and Feasible Teaching Tool for Biomedical Education“, Proceedings of the 24th International Symposium on Computer-Based Medical Systems (CBMS), Bristol, Ujedinjeno Kraljevstvo, 2011.
- [4] A. Hafid, S. Benouar, M. Kadir-Talha, F. Abtahi, M. Attari, F. Seoane, „Full Impedance Cardiography Measurement Device Using Raspberry Pi3 and System-on-Chip Biomedical Instrumentation Solutions“, IEEE Journal of Biomedical and Health Informatics, Vol. 22, No. 6, 1883-1894, 2018.

- [5] C. A. Valliappan, A. Balaji, S. R. Thandayam, P. Dhingra, V. Baths, „A Portable Real Time ECG Device for Arrhythmia Detection Using Raspberry Pi“, Proceedings of the 6th International Conference on Wireless Mobile Communication and Healthcare (MobiHealth 2016), Italija, 2016.
- [6] T. Cruz, S. Brás, S. C. Soares, J. M. Fernandes, „Monitoring Physiology and Behavior Using Android in Phobias“, Proceedings of the 37th International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Italija, 2015.
- [7] „Mikromedia for XMEGA Manual“, Ver. 1.11, MikroElektronika, 2012.
- [8] „8-bit Atmel XMEGA A Microcontroller XMEGA“, A MANUAL 8077I-AVR 11/2012, Atmel.
- [9] „ADS1299-x Low-Noise, 4-, 6-, 8-Channel, 24-Bit, Analog-to-Digital Converter for EEG and Biopotential Measurements, ADS1299, ADS1299-4, ADS1299-6“, Texas Instruments, SBAS499C, 2017.
- [10] N. Jorgovanović, M. Stanković, Ž. Tepić, V. Ilić, D. Bojanić, „Laboratory System for Physiological Measurements“, Proceedings of the 1st International Conference on Electrical, Electronic and Computing Engineering, IcETRAN, Srbija, 2014.

Kratka biografija:



Boris Knežević rođen je u Subotici 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Biomedicinsko inženjerstvo odbranio je 2019. god.
kontakt: borisknezevic@uns.ac.rs

AUTOMATSKA DETEKCIJA STAVKI MENIJA UNUTAR TEKSTOVA RECENZIJA RESTORANA**AUTOMATIC DETECTION OF MENU ITEMS IN RESTAURANT REVIEWS**Igor Trpovski, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Cilj ovog istraživanja jeste prezentovanje jednog pristupa za detekciju stavki menija unutar tekstova recenzija restorana. Nekoliko modela mašinskog i dubokog učenja istrenirano je da detektuje pominjanja hrane unutar recenzija restorana. Nakon toga, nekoliko algoritama poklapanja stringova primenjeno je kako bi se pominjanja hrane uparila sa odgovarajućim stavkama menija. Podaci su prikupljeni sa sajta Donesi.com i ručno anotirani. Svi upotrebljeni modeli i algoritmi su evaluirani.

Ključne reči: analiza teksta, obrada prirodnog jezika, prepoznavanje imenovanih entiteta

Abstract – The purpose of this research is to present one approach for automatically detecting menu items in restaurant reviews. Several machine and deep learning models were trained in order to detect food mentions. Afterwards, several string matching algorithms were applied in order to match food mentions with corresponding menu items. Data was acquired from the website Donesi.com and manually annotated. All used models and algorithms were evaluated.

Keywords: text mining, natural language processing, named entity recognition

1. UVOD

Zahvaljujući ekspanziji i povećanoj dostupnosti interneta u poslednjoj deceniji, mnogi ga koriste kako bi pronašli informacije koje bi im pomogle pri donošenju svakodnevnih odluka kao što su izbor restorana i hrane za jelo. Postoji veliki broj sajtova koji prikazuju informacije o određenim restoranima, ali pronaći informacije o kvalitetu i popularnosti pojedinačnih stavki menija iz restorana može biti veoma vremenski zahtevan i umarajući posao, jer uglavnom podrazumeva čitanje potencijalno velikog broja recenzija.

Zbog različitih načina pisanja naziva jela unutar recenzija, može biti teško zaključiti na koju se stavku menija pomenuto jelo odnosi. Takođe, ukoliko nekoga interesuje kvalitet neke konkretne stavke menija, umesto da mu automatski budu prikazane samo recenzije u kojima se ta stavka pominje, on opet mora čitati sve recenzije.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, vanredni profesor

Tema ovog rada je automatska detekcija pominjanja hrane unutar recenzija restorana i za svako od njih određivanje na koju stavku menija se ono odnosi. Najveći izazov predstavlja činjenica da su recenzije podaci sa mnogo šuma. Drugim rečima one sadrže veliki broj pravopisnih grešaka, specijalnih znakova i često su pisane upotrebom kolokvijalnih i regionalnih izraza. Tretiranjem pominjanja hrane kao imenovanih entiteta, njihova detekcija unutar tekstova recenzija svodi se na klasičan problem prepoznavanja imenovanih entiteta (eng. Named Entity Recognition - NER). NER je proces koji ima za cilj da locira i klasifikuje pominjanja imenovanih entiteta iz nestrukturiranih tekstova u prethodno definisane kategorije kao što su imena, lokacije, organizacije, itd. Uzima neanotiran i proizvodi anotiran deo teksta sa označenim imenovanim entitetima. Ovaj problem proučavan je i ranije, ali koliko je poznato autoru ovog rada, nikada za recenzije pisane na srpskom jeziku.

Kreiran je skup podataka sačinjen od recenzija restorana dobavljenih sa Donesi.com [1], popularnog sajta za naručivanje hrane u regionu Srbije, Crne Gore i Bosne i Hercegovine. Kako bi bilo moguće trenirati modele da prepoznaju koji delovi teksta predstavljaju hranu, a koji ne, bilo je neophodno manuelno anotirati pominjanja hrane unutar recenzija. Modeli koji su korišćeni za ovaj problem su: Conditional Random Fields (CRF) [2], bidirekcioni Long Short-term Memory (LSTM) [3] i bidirekcioni Gated Recurrent Units (GRU) [4]. Za bidirekcione LSTM i bidirekcione GRU modele iskorišćeno je nekoliko različitih načina za kodiranje ulaza: one-hot reprezentacija reči, one-hot reprezentacija reči i karaktera i FastText [5] vektorska reprezentacija reči. Takođe isproban je hibridni pristup korišćenja CRF-a zajedno sa bidirekcionim LSTM i bidirekcionim GRU modelima isto koristeći prethodno spomenute metode za kodiranje ulaza. Kako bi odredili koja stavka menija odgovara određenom entitetu hrane isprobano je nekoliko različitih metoda poklapanja stringova: potpuno poklapanje, podstring poklapanje, fuzzy poklapanje i parcijalno poklapanje. Cilj rada je uporediti sve iskorišćene metode, kao i odrediti onu koja daje najbolje rezultate za zadatak za koji je namenjena.

2. METODOLOGIJA

U ovom poglavlju prezentovane su primenjene metodologije za detekciju stavki menija unutar recenzija restorana. Takođe opisani su i alati upotrebljeni u njenoj implementaciji.

2.1. Prikupljanje podataka

Svi potrebni podaci pribavljeni su sa sajta Donesi.com. Koliko je poznato autoru ovog rada, on nema API za prikupljanje podataka, pa je bilo neophodno napisati web crawler. Napisan je u jeziku Python koristeći biblioteku Selenium [6], a parsiranje HTML stranica izvršeno je pomoću biblioteke BeautifulSoup4 [7]. Odlučeno je da se dobave svi podaci sa područja Srbije. Konkretno, dobavljeno je 169.486 recenzija za 1.658 različitih restorana. Takođe prikupljeni su jelovnici svih restorana.

2.2. Filtriranje recenzija

Filtriranje podataka odrađeno je radi stvaranja skupa podataka koji sadrži samo one recenzije restorana koje u svom naslovu ili tekstu sadrže jedno ili više pominjanja hrane. Da bi se olakšao ovaj zadatak implementiran je mali program čiji je interfejs prikazan na slici 1. Program iterira kroz skup podataka i za trenutnu recenziju prikazuju se naslov i tekst. Osoba koja izvršava ovaj zadatak odlučuje da li se unutar recenzije nalazi pominjanje hrane. Ako odluči potvrdno, recenzija biva sačuvana unutar nove kolekcije podataka. Novokreirani skup podataka sastoji se iz 20.079 filtriranih recenzija. Neophodno je napomenuti da pre čuvanja recenzije unutar nove kolekcije, ukoliko je njen tekst napisan u Ćirilici, on biva konvertovan u Latinicu. Takođe svi latinični specijalni karakteri su zamenjeni. Ova dva prethodna koraka odrađena su kako bi se sve recenzije svele na isti način pisanja i time olakšao posao algoritmima koji će biti korišćeni u narednim koracima.

```
Naslov: Zvrk
Sadržaj: Kapricioza je vrhunska, jos kad zbudzis extra kackavalj...
I svaki put mi donesu mnogo pre isteka 45 minuta...
Da li se u recenziji spominje neka hrana? (Y/N):
```

Slika 1. Interfejs programa za filtriranje recenzija

2.3. Anotacija pominjanja hrane

Kako bi bilo moguće obučiti modele koji detektuju pominjanja hrane unutar recenzija restorana, bilo je neophodno anotirati delove naslova i teksta recenzije koji ih sadrže. BILOU šema kodiranja korišćena je za predstavljanje pominjanja hrane. Pomoću nje model se može obučiti da prepozna početne, unutrašnje i krajnje tokene entiteta čiji naziv sadrži više tokena, kao i entitete čiji naziv sadrži samo jedan token. O klasa se koristi za predstavljanje tokena koji ne pripadaju nekom entitetu. Na primer, entitet hrane "Piletina sa kikirikijem" označava se sa labelama "B-FOOD", "I-FOOD" i "L-FOOD", odnosno entitet "Kapricioza" sa labelom "U-FOOD". Anotiranje je vršeno pomoću alata MAE [8]. Najpre je neophodno kreirati datoteku koja sadrži klase koje će se koristiti prilikom anotiranja. U ovom slučaju, svaki token mogao je biti anotiran sa jednom od ovih klasa: B-FOOD, I-FOOD, L-FOOD ili U-FOOD. Nakon toga bilo je neophodno kreirati datoteku u XML formatu koja sadrži tekst za anotaciju. Budući da recenzije sadrže naslov i tekst, izvršena je njihova konkatencija sa karakterom "\n" između. Dakle, za svaku recenziju koristili smo taj konkatencirani tekst za kreiranje odgovarajućih XML datoteka. Posle završene anotacije jedne recenzije, u

njenoj odgovarajućoj XML datoteci kreira se lista svih anotiranih tokena sa dodeljenim klasama. Zbog vremenskih ograničenja anotirano je 10 000 recenzija.

2.4. Obučavajući i test skup podataka

Za podelu teksta u rečenice, tokene, POS tagovanje i lematizaciju korišćena je Python biblioteka ReLDD [9-13], jer je razvijena specifično za obradu južnoslovenskih jezika. Takođe je korišćena i Python implementacija stemera za Srpski jezik [14]. Skup podataka od 10 000 recenzija sa anotiranim entitetima hrane i procesiranih od strane ReLDD biblioteke i stemera za srpski jezik nasumično je podeljen na obučavajući skup od 9.000 i test skup od 1.000 recenzija.

2.5. Detekcija pominjanja hrane

Prvobitno je isproban algoritam mašinskog učenja CRF, jer se kroz razna istraživanja pokazao kao najbolji algoritam tog tipa za problem prepoznavanja imenovanih entiteta. Razlog za to je što je po prirodi sekvencioni model, tj. za sekvencu primera predviđa sekvencu labela. Drugim pri predikciji labela za trenutni primer on uzima u obzir predikcije za prethodne. U slučaju linear-chain CRF modela koji je ovde korišćen uzima se u obzir samo labela prethodnog primer. Kada govorimo o NER-u, sekvencu predstavlja rečenica odnosno skup tokena.

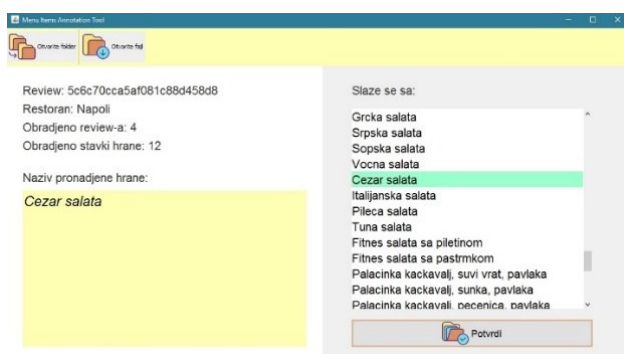
CRF omogućava specificiranje određenog broja dodatnih atributa koji će dodatno opisati svaki token. U ovom radu su za ulaz u CRF model pored tokena korišćeni su sledeći atributi: POS tag, lema, stem, token pisan svim velikim slovima, token pisan svim malim slovima, token pisan velikim početnim i svim ostalim malim slovima, da li je token broj, da li je token znak interpunkcije, da li je token napisan samo velikim slovima, da li je token napisan sa pa prvim početnim, a svim ostalim malim slovima i da li token predstavlja krajnji token rečenice. Takođe je korišćena okolina od 3 tokena oko trenutnog, odnosno 3 prethodna i 3 naredna tokena, kao i POS tagovi svih tih tokena. Svi prethodno navedeni atributi izabrani su proučavanjem sličnih rešenja i generalnih upotreba CRF algoritma za NER, kao i otkrivanjem šablona u korišćenim podacima. Iskorišćena je implementacija CRF algoritma u programskom jeziku C++ zvana CRF++ [15]. Važno je napomenuti da su pri treniranju korišćeni baš svi navedeni atributi, tj. ovaj rad se nije bavio određivanjem optimalnog skupa atributa. Takođe za proces treniranja korišćeni su predefinisani parametri CRF++ modela.

Nakon toga isprobani su modeli dubokog učenja LSTM i GRU. Oni predstavljaju varijacije standardnih modela rekurentnih neuronskih mreža i imaju za cilj da otklone probleme često vezane za njih kao što su nestajući gradijent i rukovanje dugoročnim zavisnostima. Korišćeni su birirekcionni LSTM i GRU modeli, jer se tako pri predikciji trenutnog tokena uzimaju u obzir informacije iz prošlih i budućih stanja, tj. informacije o prošlim i budućim tokenima iz sekvence. Kako ovi modeli za ulaz ne mogu uzeti tekstualne podatke oni su morali na neki način biti kodirani. Isprobane su varijante one-hot reprezentacija reči, kao i reči i karakterata.

Takođe isprobane su i FastText vektorske reprezentacije reči. FastText vektori dobijeni su treniranjem neuronske mreže nad skupom podataka od 267 362 rečenica recenzija sa Donesi.com koje sadrže 6 ili više tokena. Iskorišćeni su predefinisani parametri Python FastText implementacije. Takođe iskorišćena je varijanta LSTM i GRU modela sa dodatnim CRF slojem. Razlog za to je što modeli bez tog sloja donose odluke o klasifikaciji svakog tokena zasebno, odnosno ne obezbeđuju predikciju najverovatnije čitave sekvence. To dovodi do narušavanja ograničenja koja BILOU šema postavlja. Npr. dešava se da se “U-FOOD” tag javlja odmah nakon “B-FOOD” taga. CRF sloj, sa druge strane, vrši predikciju klase trenutnog tokena tako što uzima u obzir klasu prethodnog tokena i tako na kraju odredi najverovatniju čitavu sekvencu. Za kreiranje arhitekture svih modela dubokog učenja korišćena je biblioteka Keras [16].

2.6. Anotacija stavki menija

Bilo je neophodno anotirati pominjanja hrane iz recenzija sa odgovarajućom stavkom menija iz restorana na koju se ta recenzija odnosi. Iz tog razloga kreiran je poseban alat za anotaciju. Na slici 2. prikazan je njegov interfejs. Ovaj korak bio je neophodan kako bi se testirali algoritmi koji će automatski odrediti koja stavka menija odgovara kom pominjanju hrane. Kao ulaz u ovaj alat iskorišćen je skup podataka od 1.000 recenzija kod kojih je izvršena anotacija pominjanja hrane. Taj skup se takođe koristi i za testiranje algoritama za detekciju pominjanja hrane. Jedno pominjanje hrane može se odnositi na nijednu, jednu ili više stavki iz menija restorana. Ukoliko se pominjanje hrane očigledno ne odnosi ni na jednu stavku menija ili je nemoguće napraviti izbor između nekoliko stavki menija, doneta je odluka da se u tom slučaju pominjanje hrane anotira sa “None”. U suprotnom anotator je morao izabrati jednu stavku menija na koju se pominjanje hrane odnosi. Pri završetku anotiranja poslednjeg pominjanja hrane iz jedne recenzije, generiše se izlazna datoteka u JSON formatu, koja sadrži spisak pominjanja hrane sa anotiranim stavkama menija.



Slika 2. Interfejs programa za anotaciju stavki menija

2.7. Uparivanje pominjanja hrane sa odgovarajućim stavkama menija

Kako bi se automatski odredilo na koju stavku menija se pominjanje hrane odnosi isprobano je nekoliko različitih metoda poklapanja stringova: potpuno poklapanje, podstring poklapanje, fuzzy poklapanje i parcijalno poklapanje. Kod svih tekstova su pre primene algoritama za poklapanje velika slova konvertovana u mala.

Potpuno poklapanje je najstrožiji od svih algoritama. Smatra se uspešnim ukoliko se tekst stavke menija potpuno poklapa sa tekstom pominjanja hrane. Ova metoda nije uspeła da pronađe previše poklapanja, osim u slučajevima kada je autor recenzije naveo puno ime stavke menija pri njenom pominjanju u recenziji.

Pominjanje hrane i stavka menija poklapanju se ukoliko je tekst pominjanja hrane podstring teksta stavke menija ili obrnuto, u zavisnosti od toga koji tekst sadrži više karaktera. Bilo je vrlo uobičajeno da se pominjanje hrane poklopi sa više stavki menija. Isprobana su dva različita načina za rešavanje tog problema. Prvi je da se izabere stavka menija čiji je tekst najkraći, a drugi je da se ne izabere nijedna stavka menija odnosno da se pominjanje hrane označi sa “None”.

Mnoga pominjanja hrane sadržala su greške u pisanju. U tim slučajevima prethodne metode nisu mogle biti uspešne. Samim tim isprobane su tri metode fuzzy poklapanja: Damerau-Levenštajново rastojanje [17], Jaroova sličnost [18] i Jaro-Winklerova sličnost [19]. Stavka menija koja odgovara pominjanju hrane jeste ona za koju je Damerau-Levenštajново rastojanje najmanje, gde je ono takođe manje od zadate vrednosti praga (eng. threshold). Korišćena je Python implementacija ovog rastojanja koja je deo biblioteke strsim [20]. Stavka menija koja odgovara pominjanju hrane pri korišćenju Jaroove sličnosti je ona za koju je ta sličnost najveća, a da je ona pritom veća od zadate vrednosti praga. Isti princip korišćen je i za Jaro-Winklerovu sličnost. Korišćena je Python biblioteka pyjarowinkler [21] koja sadrži implementacije ova dva algoritma.

Pre vršenja parcijalnog poklapanja tekstovi pominjanja hrane i stavki menija su tokenizovani, izvršeno je stemovanje i uklonjene su stop reči, odnosno reči koje ne nose prevelik značaj. Za tokenizaciju i stemovanje korišćene su Python implementacije tokenizera i stemera za srpski jezik. Poklapanje se smatra uspešnim ukoliko se u skupu stemovanih tokena pominjanja hrane nalazi bar polovina stemovanih tokena stavke menija i beleži se tačan broj stemovanih tokena u preseku. Stavka menija sa kojom je najveći broj stemovanih tokena u preseku se uzima kao ona stavka na koju se pominjanje hrane odnosi.

3. EKSPERIMENTALNA EVALUACIJA I REZULTATI

Eksperimentalna evaluacija se vrši radi određivanje performansi korišćenih algoritama za detekciju pominjanja hrane i određivanja stavke menija na koje se svako pominjanje odnosi.

Metrike performansi algoritama za detekciju pominjanja hrane i određivanja stavke menija na koje se svako pominjanje odnosi koje su korišćene u ovom radu su preciznost, odziv i F_1 -mera. Važno je napomenuti da je pominjanje hrane koje je detektovao model jedino smatrano tačnim ukoliko se svi njegovi tokeni poklapaju sa tokenima pominjanja označenog kao ispravnog. Drugim rečima nije se razmatralo nikakvo parcijalno poklapanje, odnosno detektovana pominjanja koja se parcijalno poklapaju sa očekivanim smatrana su kao netačna.

Što se tiče detekcije pominjanja hrane svi modeli koji su postigli najbolje metrike su neuralni i koriste FastText vektorske reprezentacije reči. Najbolju preciznost od 0.9376 dostigao je bidirekcion GRU-CRF treniran u 5 epoha. Najbolji odziv od 0.9335 bidirekcion LSTM i najbolju F_1 -meru od 0.9272 bidirekcion LSTM-CRF, oba trenirana u 10 epoha. Sa druge strane za uparivanje pominjanja hrane sa odgovarajućim stavkama menija najbolju preciznost od 0.8865 dostiglo je Jaro-Winklerovo poklapanje sa pragom od 0.8, a najbolji odziv od 0.8683 i F_1 -meru od 0.8468 dostiglo je parcijalno poklapanje.

4. ZAKLJUČAK

U ovom radu predstavljeno je jedno rešenje za detekciju stavki menija u recenzijama restorana. Kreiran je skup podataka recenzija na srpskom jeziku, tako što su one dobavljene sa sajta Donesi.com. Izvršena je ručna filtracija i anotacija recenzija kako bi se obeležili oni delovi teksta koji sadrže pominjanja hrane. Nekoliko modela mašinskog i dubokog učenja istrenirano je da detektuje pominjanja hrane. Nakon toga, nekoliko algoritama poklapanja stringova primenjeno je kako bi se pominjanja hrane uparila sa odgovarajućim stavkama menija. Svi korišćeni modeli i algoritmi postigli su zadovoljavajuće rezultate.

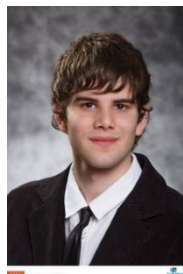
Rad bi mogao dalje da se proširi izborom optimalnog skupa atributa za treniranje CRF modela. Moglo bi se dodatno poraditi na pronalaženju optimalnih parametara za treniranje FastText neuronske mreže i neuralnih modela korišćenih za NER. Takođe mogao bi se kreirati veći skup podataka sa anotiranim pominjanjima hrane eksperimentisanjem sa polu-automatizovanom anotacijom. Pošto metode koje određuju na koju stavku menija se pominjanje hrane odnosi ne mogu da detektuju sinonime, mogao bi se napraviti rečnik čestih sinonima i on koristiti da se taj problem razreši. Takođe se pored metoda za poklapanje stringova može eksperimentisati i sa korišćenjem metoda mašinskog učenja ili neuronskih mreža.

5. LITERATURA

- [1] www.donesi.com
- [2] Lafferty, J., McCallum A., and Pereira F., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- [3] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, 9 no.8, pp.1735-1780.
- [4] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*
- [5] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T., 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, pp.135-146.
- [6] Jason Huggins, et al, 2004. Selenium, <https://www.seleniumhq.org>
- [7] Leonard Richardson 2014, BeautifulSoup4 <https://www.crummy.com/software/BeautifulSoup>
- [8] Kyeonmin Rim, "MAE2: Portable Annotation Tool for General Natural Language Use". In Proceedings of the 12th Joint ACL-ISO Workshop on Interoperable Semantic Annotation, Portorož, Slovenia, May 28, 2016.

- [9] Ljubecic, Nikola, Tomaz Erjavec and Darja Fiser. "Corpus-Based Diacritic Restoration for South Slavic Languages." *LREC* (2016).
- [10] Ljubecic, Nikola and Tomaz Erjavec. "Corpus vs. Lexicon Supervision in Morphosyntactic Tagging: the Case of Slovene." *LREC* (2016).
- [11] Ljubecic, Nikola, Filip Klubicka, Zeljko Agic and Ivo-Pavao Jazbec. "New Inflectional Lexicons and Training Corpora for Improved Morphosyntactic Annotation of Croatian and Serbian." *LREC* (2016).
- [12] Agic, Zeljko and Nikola Ljubecic. "Universal Dependencies for Croatian (that work for Serbian, too)." *BSNLP@RANLP* (2015).
- [13] Fišer, D., Ljubešić, N. & Erjavec, T. Lang Resources & Evaluation (2018). <https://doi.org/10.1007/s10579-018-9425-z>
- [14] Milosevic, Nikola "Stemmer for Serbian Language." *CoRR* abs/ 1209.4471 (2012): n. pag.
- [15] Taku Kudo, "CRF++: Yet another CRF toolkit" 2005, <https://taku910.github.io/crfpp>
- [16] Chollet, Francois et al. "Keras" 2015, <https://keras.io>
- [17] Damerau, F.J., 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3), pp.171-176.
- [18] Jaro, M.A., 1989. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406), pp.414-420.
- [19] Winkler, W.E., 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage.
- [20] Zhuo Yang Luo. python-string-similarity 2018 <https://github.com/luozhouyang/python-string-similarity>
- [21] Jean-Bernard Ratte. Jaro Winkler Distance 2015, <https://github.com/nap/jaro-winkler-distance>

Kratka biografija:



Igor Trpovski rođen je u Novom Sadu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Računarstva i automatike – Sistemi za istraživanje i analizu podataka odbranio je 2019.god. kontakt: trpovski@gmail.com

PRIMENA MAŠINSKOG UČENJA U ANALIZI RELEJNE ZAŠTITE PRI STARTU ASINHRONIH MOTORA**APPLICATION OF MACHINE LEARNING IN RELAY PROTECTION DURING THE START OF ASYNCHRONOUS MACHINES**

Milorad Maksimović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu se razmatra primena mašinskog učenja u analizi sigurnosti relejne zaštite pri startu asinhronih motora. Cilj je da se prikažu prednosti i različite pogodnosti primene ovakve vrste analize u odnosu na standardne proračune pri startu asinhronih motora. Razmatranja su izvršena na velikom broju različitih topoloških struktura u cilju dobijanja što preciznijeg modela predviđanja.

Ključne reči: mašinsko učenje, sigurnost relejne zaštite, asinhroni motori, mašinsko učenje

Abstract – This paper deals with application of machine learning in relay protection during the start of asynchronous machines. The goal is to show advantages and different benefits of this type of analysis in regards to standard calculations during the motor start. Considerations have been made on a large number of different topologies in order to achieve the most accurate prediction model.

Keywords: machine learning, relay protection safety, asynchronous machines, machine learning

1. UVOD

U ovom radu razmatra se realizacija algoritma metoda potpornih vektora u analizi sigurnosti relejne zaštite pri startu asinhronih motora. Sigurnost predstavlja zahtev da se zaštitom ne deluje kada ne treba (nekorektno delovanje pri normalnim pogonskim uslovima) [1]. Pokretanje velikih asinhronih motora može da izazove ozbiljne probleme u radu samog motora, ali i ostalih već priključenih motora kao i bilo kog potrošača u blizini priključenja motora, usled prvenstveno propada napona u čvorovima lokalne mreže.

Uobičajena situacija jeste da motor krene u start bez problema, ali onda odreaguje relejna zaštita i isključi motor od sistema pre nego što postigne punu brzinu. Relejna zaštita ne bi trebala da odreaguje u ovakvim situacijama, jer se start asinhronog motora ubraja u normalno pogonsko stanje. Sa razvojem brzih i efikasnih algoritama, kao i modela zasnovanih na analizi podataka koji omogućavaju obradu u realnom vremenu, mašinsko učenje je u stanju da analizira velike količine podataka i na osnovu njih izvrši tačna i precizna predviđanja. Ovo je obrađeno u drugom delu rada.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Duško Bekut, redovni profesor.

U trećem delu obrađena je problematika pokretanja motora, a u četvrtom je ispitano i primenjeno formiranje modela predviđanja sigurnosti relejne zaštite, zasnovanog na prethodno opisanim mehanizmima veštačke inteligencije. U poslednjim delovima rada izneseni su zaključak i prednosti ovakvog vida analize kvaliteta relejne zaštite, kao i literatura korišćena za izradu ovog rada.

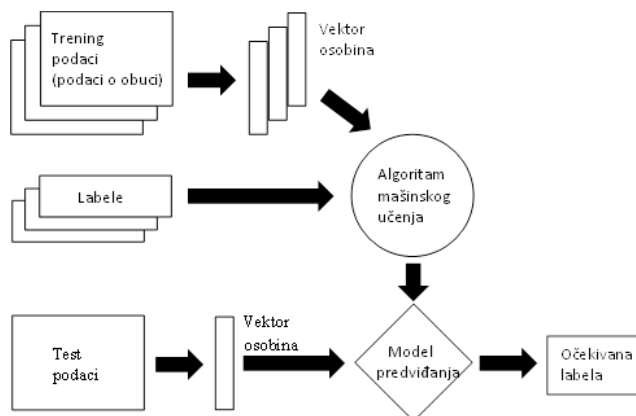
2. MAŠINSKO UČENJE – ALGORITAM METODA POTPORNIH VEKTORA

Mašinsko učenje, ili kako se još naziva automatizovano učenje, je posebna podoblast nauke informacionih tehnologija. Naziv „mašinsko učenje“ odnosi se na automatizovano uočavanje smislenih obrazaca u velikom skupu podataka. Razvijena iz raznih studija o prepoznavanju obrazaca i računarske teorije učenja u veštačkoj inteligenciji, mašinsko učenje predstavlja posebnu granu koja se bavi otkrivanjem i razvijanjem algoritama koji mogu da uče i izvode pretpostavke na osnovu ulaznih podataka. Jednu od definicija mašinskog učenja je dao *Arthur Samuel*, naučnik koji je i skovao termin „mašinsko učenje“. Ona glasi: „Mašinsko učenje daje računarima sposobnost učenja bez njihovog eksplicitnog programiranja“ [2].

Danas postoji veliki broj razvijenih algoritama mašinskog učenja. Razlikuju se 3 vrste algoritama prema načinu učenja:

- 1) nadgledano učenje,
- 2) nenadgledano učenje,
- 3) polu-nadgledano učenje.

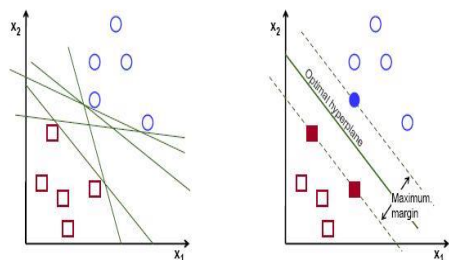
Na slici 2.1 je prikazan princip nadgledanog učenja.



Slika 2.1 – Princip nadgledanog učenja

Algoritam metoda potpornih vektora (engl. *Support vector machines*) [3] predstavlja jedan od najpoznatijih i najčešće korišćenih algoritama mašinskog učenja, zbog brojnih prednosti koje ima u odnosu na druge algoritme. Osnovni cilj metoda potpornih vektora jeste pronalaženje funkcije koja aproksimira odnos između ulaznih i izlaznih podataka, tj. formiranje hiper-ravni razdvajanja između klasa objekata. Metode potpornih vektora predstavljaju binarni klasifikator, koji dodeljuje objekte nekoj od klasa.

Na slici 2.2 prikazane su razne mogućnosti izbora hiper-ravni razdvajanja, kao i optimalna hiper-ravan.



Slika 2.2 – Mogućnosti izbora hiper-ravni razdvajanja i optimalna hiper-ravan [4]

3. ASINHRONI MOTORI – POKRETANJE

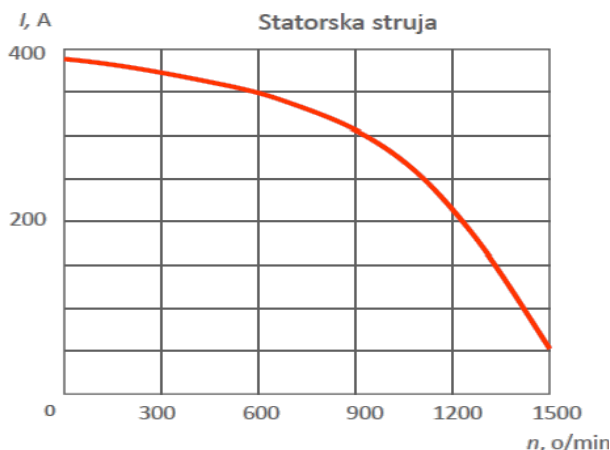
U ovom poglavlju predstavljene su tehnike pokretanja asinhronog motora razmazrane u ovom radu. To su neke od najčešće primenjivanih tehnika pokretanja asinhronih motora i u njih spadaju:

- 1) pokretanje direktnim priključenjem na mrežu,
- 2) pokretanje zvezda-trougao preklopkom,
- 3) pokretanje preko autotransformatora.

Sve tehnike pokretanja se svode na to da se izvrši uspešan zalet motora i pogona uz što manje struje zaleta i što kraće trajanje zaleta. Uticaj na mrežu mora biti u prihvatljivim granicama dozvoljenog pada napona izazvanog strujom polaska motora.

3.1 POKRETANJE DIREKTNIM PRIKLJUČENJEM NA MREŽU

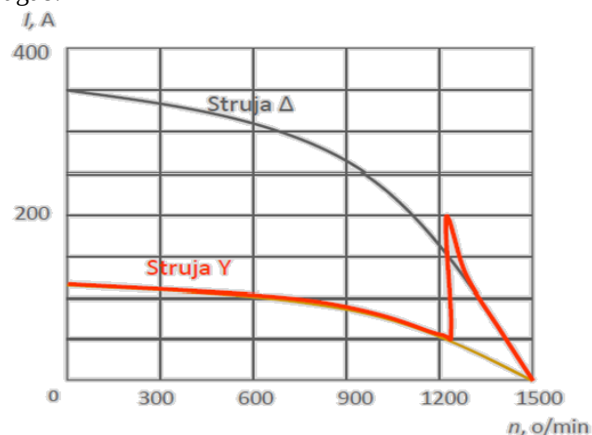
Pokretanje asinhronog motora direktnim priključenjem na mrežu je najjednostavniji i najjeftiniji način pokretanja, ali je praćen najvećim polaznim strujama motora. Na slici 3.1 je prikazana statorska struja pri startu asinhronog motora.



Slika 3.1 – Statorska struja asinhronog motora pri direktnom priključenju na mrežu

3.2 Pokretanje zvezda – trougao preklopkom

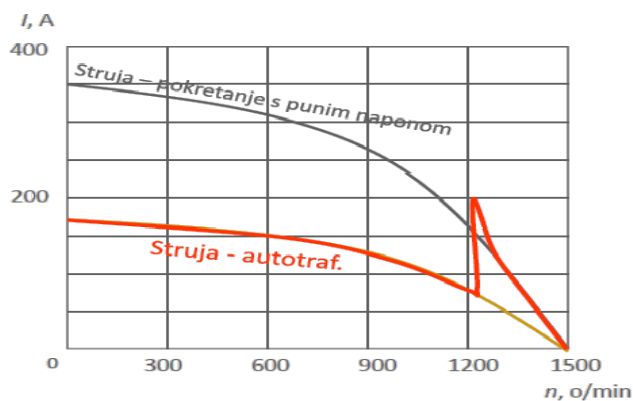
Kavezni asinhroni motor za trajan rad u vezi trougao (Δ), pri pokretanju se veže u zvezdu (Y). Napon po faznom namotaju je $\sqrt{3}$ puta manji pa kroz namotaj teče $\sqrt{3}$ puta manja struja, a struja iz mreže je još $\sqrt{3}$ puta manja jer se iz jednog faznog voda napaja samo jedan namotaj. U spoju Y iz mreže se povlači 1/3 polazne struje u spoju Δ . Kada se obavi zalet u zvezdi, motor se spoji u vezu trougao.



Slika 3.2 – Statorska struja asinhronog motora pri startu zvezda-trougao preklopkom

3.3 POKRETANJE ASINHRONOG MOTORA PREKO AUTOTRANSFORMATORA

Pri pokretanju asinhronog motora preko autotransformatora zatvoreni su prekidači koji statorske namotaje povezuju sa mrežom preko autotransformatora, dok su u normalnom pogonu zatvoreni prekidači koji direktno povezuju statorske namotaje sa mrežom.



Slika 3.3 – Statorska struja asinhronog motora pri startu preko autotransformatora

4. PRIMENA ALGORITMA U ANALIZI RELEJNE ZAŠTITE

U ovom poglavlju biće predstavljena mogućnost primene algoritma metoda potpornih vektora na određivanje sigurnosti relejne zaštite pri startu asinhronih motora. Opisana je primena algoritma na primerima pokretanja asinhronog motora direktnim priključenjem, zvezda-trougao preklopkom i pomoću autotransformatora. Kao što je već rečeno, pokretanje asinhronog motora je normalno pogonsko stanje sistema i relejna zaštita bi trebala radi pouzdano i sigurno i u takvim uslovima rada.

U standardnim tehnikama učenja, „učeniku“ (algoritmu) je potrebno obezbediti/definirati proizvoljan skup X , koji predstavlja skup ulaznih objekata u algoritam kojima je potrebno dodeliti odgovarajuću labelu. Ovaj skup se naziva domenski skup, a elementi skupa se nazivaju instance (objekti). U ovom slučaju, skup X predstavlja skup raznih topologija mreže sa priključenim asinhronim motorom. Cilj je na osnovu ulazne topologije odrediti kvalitet relejne zaštite na počecima izvoda na koje je priključen asinhroni motor, bez proračuna tokova snaga. Ove topologije su predstavljene vektorom obeležja O . Ovaj vektor sadrži podatke o snazi asinhronog motora, nominalnom naponu motora, udaljenosti mesta priključenja asinhronog motora od relejne zaštite na početku izvoda i tipu starta asinhronog motora.

Stoga, vektor obeležja se zapisuje na sledeći način:

$$O = [S_{AM}, V_n, Z_{thev}, \alpha_s], \quad (4.1)$$

gde su:

S_{AM} – snaga asinhronog motora,
 V_n – nominalni napon asinhronog motora,
 Z_{thev} – vrednost Tevenenove impedanse sa mesta priključenja asinhronog motora,
 α_s – tip starta (direktno, zvezda-trougao, autotransformator).

Važno je primetiti da ovaj vektor ne sadrži podatke o vrednostima struja kroz prekidače na počecima izvoda, što je i bio cilj.

S obzirom da se radi o nadgledanom učenju, poznati su željeni rezultati algoritma. Druga bitna stvar koju je potrebno definisati je skup labela Y . Taj skup, u ovom slučaju će sadržati samo dva elementa i to:

$$Y \in \begin{cases} -1, & \text{loša sigurnost zaštite} \\ +1, & \text{dobra sigurnost zaštite} \end{cases} \quad (4.2)$$

Važno je kao mogući rezultat imati skup od dva elementa, jer je zadatak algoritma mašinskog učenja da kreira binarni klasifikator sposoban da odgovori na pitanje pripadnosti određene topologije nekoj od klasa (-1 ili +1).

Da bi se efikasno primenio algoritam metoda potpornih vektora, potrebno je formirati ulazni skup obučavajućih podataka. U ovom slučaju, formira se sledeći skup:

$$S = (x_1, y_1), \dots, (x_m, y_m), \quad (4.3)$$

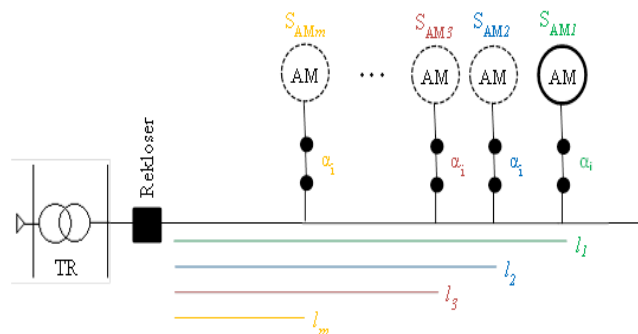
gde su:

- S – skup podataka o obuci,
- x_i – i -ta topologija,
- y_i – i -ta labela za datu topologiju,
- m – broj obrađenih topologija sa poznatim labelama.

Dakle, trening skup sadrži uređene parove topologija-labela. Formira se premeštanjem asinhronog motora kroz čvorove na određenom izvodu, dobijajući tako uvek novu topološku strukturu mreže. Takođe, pored ove promene obeležja l , ujedno je moguće menjati snagu i napon priključenog asinhronog motora, kao i tip starta motora, i tako dobiti raznovrstan skup podataka za obuku.

Na slici 4.1 šematski je prikazana mogućnost izbora različitih topologija, priključivanjem asinhronog motora u bilo koji čvor na datom izvodu (bilo koji asinhroni motor je nacrtan isprekidanom linijom). Sa S_{AMi} ($i = 1, 2, \dots, m$)

su označene različite snage asinhronog motora, a sa l_i ($i = 1, 2, \dots, m$) su označene različite udaljenosti asinhronog motora od relejne zaštite na početku izvoda. Sa α_i (i =direktno, zvezda-trougao, autotransformator) je označen tip priključenja i pokretanja asinhronog motora.



Slika 4.1 – Primeri različitih topoloških struktura korišćenih prilikom obučavanja

Drugi deo uređenih parova, labela, dobijaju se klasičnim postupcima za određivanje kvaliteta relejne zaštite pri startu asinhronih motora. Ti postupci podrazumevaju simulaciju tokova snaga i poređenje struje kroz zaštitnu opremu na početku izvoda i podešenja relejne zaštite, za svaku topologiju iz obučavajućeg skupa.

Ovaj postupak se ponovi za svaku topologiju iz obučavajućeg skupa i na osnovu pomenutih poređenja se dolazi do korektnih labela za poznate topologije.

Na taj način je formiran skup podataka o obuci, i ovi skupovi predstavljaju ključne podatke za algoritam. Od algoritma se zahteva da na osnovu njih kreira model predviđanja, ili funkciju klasifikacije $f: X \rightarrow Y$, koja će biti korišćena svaki sledeći put kada kao ulaz u algoritam dođe nepoznata topologija.

Dobro istreniran algoritam će kreirati kvalitetan model predviđanja koji ima mogućnost da gotovo trenutno prilikom priključenja asinhronog motora na mrežu pruži informaciju o kvalitetu relejne zaštite na početku izvoda, klasifikujući do tad neviđenu topologiju u jednu od klasa (dobra ili loša sigurnost).

Time se željeni rezultat dobija bez dugotrajnog i komplikovanog proračuna tokova snaga, uz veliku uštedu kako vremena, tako i hardverskih i softverskih resursa.

U nastavku rada će biti predstavljeni rezultati simulacije formiranog modela predviđanja kvaliteta relejne zaštite pri startu asinhronih motora.

Obučavanje algoritma, formiranje modela predviđanja i klasifikacija novih ulaznih topologija je izvršena koristeći programski paket MATLAB.

Skup obučavajućih podataka (izraz 4.3) je formiran na već opisan način, i ima sledeći oblik:

$$X_i(a,b,c,d)=$$

gde su:

a – snaga asinhronog motora u [MW],

b – nominalni napon aishronog motora u [kV],

c - ekvivalentna impedansa sa mesta priključenja motora u $[\Omega]$,

d – tip starta asinhronog motora (1 – direktno, 2 – zvezda-trougao, 3 – autotransformator).

Labele za ove poznate topologije su date u sledećem obliku:

$$Y_i =$$

gde su:

- 1 – dobra sigurnost zaštite,
- -1 – loša sigurnost zaštite.

Kreirani model predviđanja je testiran sa novim, za njega do sada nepoznatim skupom topologija i dobijeni su rezultati prikazani u tabeli 4.1

Dodatnom verifikacijom je potvrđeno da su dobijeni rezultati korektni.

Tabela 4.1 – Rezultati kalsifikovanja nepoznatih topologija

Snaga AM [MW]	Nominalni napon AM [kV]	Ekvivalentna impedansa [Ω]	Tip starta	Labela
0.5	10	2.580	1	1
0.5	10	0.780	1	1
0.6	10	0.950	2	1
0.6	10	1.050	3	1
0.7	10	1.110	3	1
0.7	10	0.880	1	1
0.8	10	0.990	2	1
0.8	10	1.550	3	1
0.9	10	2.660	1	1
0.9	10	3.550	3	1
1.1	10	0.650	2	1
1.1	10	1.880	1	1
1.3	10	2.410	3	1
1.3	10	1.770	2	1
1.5	10	1.220	1	1
1.5	20	1.630	3	-1
1.5	20	5.230	2	-1
1.5	20	1.070	1	-1
1.8	20	1.550	2	-1
2.2	10	2.060	3	1
2.2	10	2.330	1	1
2.6	10	0.770	2	-1
0.6	20	3.990	3	-1
0.7	10	0.820	2	1
0.9	20	3.780	2	-1
1.2	20	0.790	1	-1
1.1	10	4.450	3	1
1.4	10	3.570	2	1
3.5	10	2.580	1	-1
5	10	0.920	1	-1

5. ZAKLJUČAK

U cilju smanjenja računarskog vremena potrebnog za precizne proračune režima pri startu asinhronog motora, ispitana je i primenjena mogućnost korišćenja algoritma metoda potpornih vektora, kao mehanizma za predviđanje kvaliteta sigurnosti relejne zaštite. Prednosti ovog algoritma su gotovo trenutno dobijanje informacije o kvalitetu sigurnosti relejne zaštite pri priključenju asinhronog motora, dok su nedostaci velika količina vremena potrebna za kreiranje što boljeg skupa za obuku, kao i uočavanje prave kombinacije vektora obeležja kojim bi se dobili tačni rezultati.

6. LITERATURA

- [1] Duško Bekut, Relejna zaštita, Fakultet Tehničkih Nauka Novi Sad, 2009.
- [2] Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers"
- [3] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.
- [4] http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

Kratka biografija:



Milorad Maksimović je rođen u Loznici 1994. godine. Osnovne studije je završio na Fakultetu tehničkih nauka 2017. godine na departmanu Elektrotehnika i računarstvo – Elektroenergetski sistemi. Master rad je odbranio 2019. godine, na istom fakultetu na smeru Energetika, elektronika i telekomunikacije – Elektroenergetski sistemi.

**KOORDINACIJA PREKOSTRUJE ZAŠTITE SA INVERZNOM KARAKTERISTIKOM
REAGOVANJA U DISTRIBUTIVNOJ MREŽI****COORDINATION OF OVERCURRENT PROTECTION WITH INVERSE
CHARACTERISTIC IN DISTRIBUTION NETWORK**Viktor Trifunjagić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Ovaj rad bavi se problemom podešenja relejne zaštite na distributivnom vodu. Korišćene su prekostrujna zaštita sa inverznom karakteristikom reagovanja sa različitim stepenom inverznosti i topljivi osigurači. Zatim je vršena provera koordinacije između zaštita u dve situacije. Prvi slučaj je kada su u mreži samo releji, dok je druga situacija kada je korišćena kombinacija releja i topljivih osigurača. Na kraju je izvršena provera koordinacije u realnim mrežama.

Ključne reči: zaštitni releji, topljivi osigurači, prekostrujna zaštita, distributivna mreža

Abstract – This paper deals with adjustment of relay protection on distribution feeder. Protection relays with different types of inverse characteristics are being used, as well as fuses. It is checked if coordination between the protections is being satisfied in two cases. The first situation covers the case of protection by relays only, while the second situation covers the protection combination of relays and fuses. In the end, it is checked coordination of protection in real networks.

Keywords: protection relays, fuses, overcurrent protection, distribution network

1. UVOD

Kvarovi u elektroenergetskom sistemu predstavljaju poremećeno stanje mreže i veoma su česta pojava. Velika razorna moć kvara predstavlja opasnost po opremu koja se nalazi u sistemu, kao i po potrošače koji ostaju bez napajanja. Zbog karaktera kvarova i njihove učestale stohastične prirode, zaštita elektroenergetskih sistema je postala sastavni deo svakog prenosnog i distributivnog sistema. Zadatak zaštite je sprečavanje nastanka kvarova ili njihovo što kraće uklanjanje. Kako se tehnologija razvijala, tako se razvijao sve veći broj različitih vrsta zaštite. Najčešće korišćene zaštite su prekostrujni releji i topljivi osigurači. Princip delovanja navedene zaštite je da će reagovati za određeno vreme kada struja koja prolazi kroz poziciju na koju su postavljeni releji premaši podešenu vrednost. Jedna od važnih osobina zaštite jeste I odgovarajući izbor njene pozicije u sistemu i potrošačkom području pošto će zaštite svojim reagovanjem, ostaviti taj deo bez napajanja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Duško Bekut, redovni profesor.

Kako bi se uvek isključio najmanji mogući broj potrošača, postavlja se više nivoa zaštite u mrežama. Međutim, prednost koja se dobija postavljanjem više nivoa zaštite se gubi ako ta zaštita nije dobro koordinisana.

U drugom delu rada je su kratko predstavljene distributivne mpeže, dok je u trećem dat pregled modela elemenata distributivne mreže. U četvrtom delu su analizirani primeri koordinacije mreže. U poslednja dva dela su dati zaključak i literatura korišćena za pisanje ovog rada.

2. DISTRIBUTIVNE MREŽE

Distributivna mreža služi za distribuciju električne energije do krajnjih potrošača, te na nju mogu biti priključene manje elektrane iz kojih distributivna mreža preuzima deo električne energije. Takođe preuzima električnu energiju iz prenosne mreže u transformatorskim stanicama sa 110/35, 30, 20 ili 10 kV [1].

Kao glavne karakteristike distributivne mreže mogu se istaći sledeće karakteristike:

- 1) Kod distributivnih mreža prenos snage se odvija na manje udaljenosti.
- 2) Izvedba distributivnih mreža je jednostavnija od prenosnih bez obzira što se sastoje od istih elemenata zbog toga što su projektovane na manje nazivne napone.
- 3) Niskonaponske i srednjenaponske mreže nemaju mogućnost dvostrukog napajanja.
- 4) U distributivnoj mreži radi uštede nemaju svi elementi svoju zaštitu ni prekidače. Nakon kratkog spoja (KS) potrošač imaju prekid napajanja u trajanju od 0.5 pa čak do 8 i više časova.

Što se tiče osnovne strukture srednjenaponske distributivne mreže, ona može biti izvedena sa dva naponska nivoa, najčešće 35-10 kV ili sa naponom od 20 kV. Distribucija električne energije prema niskom naponu se odvija pomoću preko transformacije 10-0.4 kV ili 20-0.4 kV.

**2.1. PRORAČUN STRUJA KRATKIH SPOJEVA U
DISTRIBUTIVNOJ MREŽI**

Prema [2] pojava kratkih spojeva u distributivnoj mreži je neizbežna iz tog razloga što izolacija elemenata ne može da bude 100% pouzdana. Dakle, mora se unapred predvideti mogućnost pojave kvara i preduzeti odgovarajuće mere da bi se negativne posledice svele na po ljude bezopasan nivo, ali i na ekonomski opravdanu meru.

Kratak spoj (KS) kao tip kvara može da bude simetričan trolpolni (3pKS) i nesimetričan dvopolni (2pKS), dvopolni sa zemljom (2pKS) i jednopolni (1pKS). Jednopolni KS je u praksi najčešći. Trolpolni KS su praćeni najintezivnijim strujama KS. Prekidne moći savremenih prekidača koji se primenjuju su date u tabeli 2.1.

Tabela 2.1. – Prekidne moći prekidača

Un[kV]	Prekidna moć[kA]	In[kA]
10	12 ... 25	400 ... 2500
20	12 ... 16	800 ... 1250
110	18 ... 39	630 ... 2000
SF6 prekidači		
110	40	2000 ... 3150

Distributivne mreže prema [3] u većini slučajeva, u normalnom režimu rade kao radijalne, a vrlo retko kao slaboupetljane.

2.2. RELEJNA ZAŠTITA U DISTRIBUTIVNOJ MREŽI

Ono što jeste jedan od osnovnih ciljeva i uloge relejne zaštite u DM (distributivnoj mreži) jeste da se elementi najbrže moguće isključe ili/i deo sistema sa kvarom, ali uz očuvanje funkcionalnosti ostalog dela sistema. Zahtevi koji se prema [4] trebaju ispoštovati radi ispunjenja ovog cilja su sledeći:

- 1) Selektivnost.
- 2) Brzina reagovanja.
- 3) Osetljivost.
- 4) Pouzdanost i sigurnost.
- 5) Jednostavnost.
- 6) Ekonomičnost.

3. MODELI ELEMENATA U DISTRIBUTIVNOJ MREŽI

Pod modelom elementa EES ili celog sistema podrazumeva se manje ili više uprošćena predstava tog elementa ili sistema koja je pogodna za simulaciju različitih pojava koje se dešavaju na elementu ili u sistemu. Razlikuju se matematički i fizički modeli. U matematičkim modelima pojave se opisuju matematičkim relacijama ili jednačinama, kojima se nastoji što vernije očuvati sličnost sa pojavama na objektima koji se proučavaju.

Fizički model u direktnom smislu predstavlja redukciju stvarnog objekta ili sistema, pri čemu model treba da je takav da očuva identičnu vezu između pojedinih elemenata objekta, odnosno sistema kakva postoji u realnosti, ili, drugim rečima, mora se očuvati visoki stepen sličnosti između modela i realnog objekta, odnosno sistema.

3.1. MODEL VODA

Vodovi su elementi elektroenergetskog sistema namenjeni prenosu i distribuciji ("transportu" i "raspodeli") napona i električne energije [5]. Postoje dve vrste vodova: nadzemni (vazdušni) i kablovski (kablovi). Osnovni nominalni podaci za vodove su pogonski parametri voda: R , L , G , C , dužina voda l , kojima su pridruženi nominalni linijski napon V_n i nominalna struja voda I_n .

Nominalni napon i struja određuju nominalnu snagu voda:

$$S_n = 3U_n I_n = \sqrt{3}V_n I_n. \quad (3.1)$$

3.2. MODEL TRANSFORMATORA

U cilju efikasnog prenosa velikih snaga sa održavanjem gubitaka aktivne snage i energije na tolerantnom nivou, neophodno je u EES-u prenos električne energije realizovati na visokim naponima [5]. Ovako visoke napone nemoguće je ostvariti na naponskom nivou proizvodnje električne energije, pošto raspoloživi izolacioni materijali ograničavaju naponske nivoe u generatoru na granici od 25 kV, već se traženi visoki naponi proizvode u transformatorima.

3.3. MODEL PREKOSTRUJNOG RELEJA

Relej je naprava koja se koristi za prekidanje ili uspostavljanje strujnog kola putem elektromagneta koji otvara i zatvara strujne kontakte [6]. Elektromagnet se obično sastoji od mnogobrojnih namotaja bakrene žice na železnom jezgru. Kada struja teče kroz žicu (primarno strujno kolo), oko elektromagneta se stvara magnetno polje koje privlači železnu kotvu. Kotva nosi na sebi električne kontakte, koji onda otvaraju ili zatvaraju sekundarno strujno kolo (strujni krug). Kada se prekine struja kroz elektromagnet, elektromagnet više ne privlači železnu kotvu, i ona se vraća u početni položaj, obično uz pomoć opruge. Time električni kontakti prekidaju ili uspostavljaju strujno kolo, u zavisnosti od tipa kontakata (Normalno otvoreni/Normalno zatvoreni).

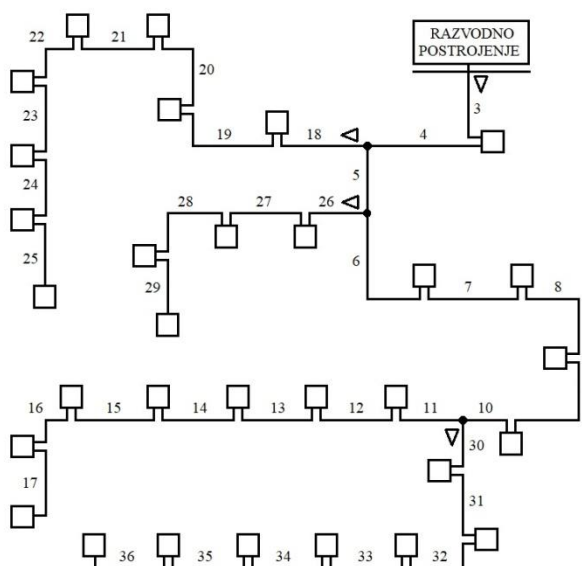
3.4. TOPLJIVI OSIGURAČI

Postoje dve vrste osigurača i to su niskoučinski, koji sadrže brze ili spore topljive umetke i visokoučinski, čiji umeci imaju završetke u vidu noževa. Maksimalna podnosiva struja, odnosno prekidna moć svih navedenih umetaka je nezavisna od nazivne struje umetka, koja iznosi čak 50 kA [7]. Glavni deo jednog topljivog osigurača je topljivi umetak ili drugim rečima patrona, koja sadrži posebno dimenzionisan elektroprovodni deo čiji se topljenjem ostvaruje prekidanje prevelike struje. Na taj način se vrši zaštitna funkcija ovih elemenata.

4. PRIMERI ZAŠTITE MREŽE

Test mreža je prikazana na slici 4.1. Potrošači su na slici predstavljeni kvadratima, dok su ekvivalent napojne mreže i transformator 35/10 kV/kV predstavljeni blokom "razvodno postrojenje".

Za potrebe analiza u ovom radu razvijena je jednostavna programska podrška. Ona se koristi za sve proračune. Prvo će biti testirana mreža koja sadrži samo releje sa strujno zavisnom karakteristikom, a zatim će biti testirana mreža koja sadrži relej sa strujno zavisnom karakteristikom i osigurače na lateralnim izvodima (mesta ugradnje označene malim trouglovima na slici 4.1).



Slika 4.1. – Test mreža

Test mrežu čini ekvivalent napojne mreže (snaga tropolnog kratkog spoja 750 MVA), transformatorska stanica 35/10 kV/kV (jedan transformator 20 MVA; $u_k=6\%$) i jedan izvod 10 kV Cu 95 mm² sa tri ogranka. Fider je podeljen na sekcije, na čijem svakom kraju se nalazi potrošački centar, pri čemu je svaka sekcija numerisana.

4.1. ZAŠTITA MREŽE RELEJIMA SA INVERZONOM KARAKTERISTIKOM

U ovom potpoglavlju se analizira situacija kada su na sekcijama 3, 18, 26 i 30 postavljeni releji sa strujno zavisnom karakteristikom reagovanja. Na osnovu vrednosti struja u normalnom radnom režimu, struje releja proračunavaju se na sledeći način:

$$I_{pod} = \frac{k_{sigurnosti} \cdot k_{samopuštanja} \cdot I_{radno}}{a}, \quad (4.1)$$

gde je:

$k_{sigurnosti}$ – koeficijent sigurnosti (tipično 1.2),

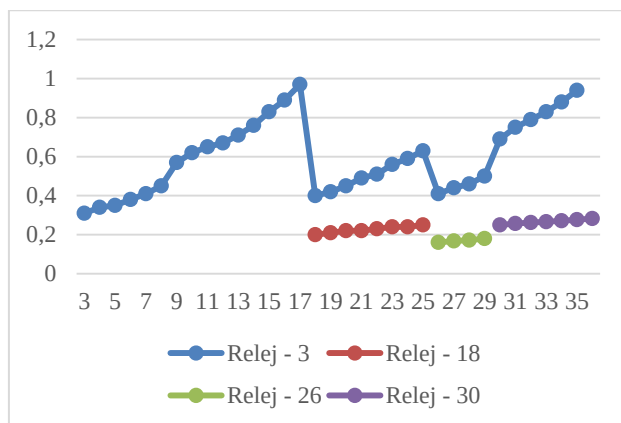
$k_{samopuštanja}$ – koeficijent samo puštanja (1-6),

a – koeficijent otpuštanja,

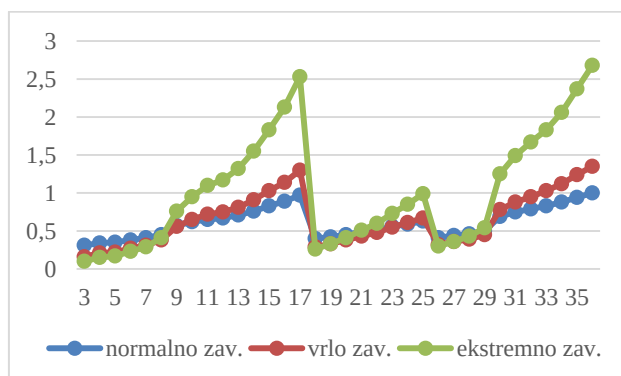
I_{radno} – radna struja.

Na osnovu formule i usvojenih vrednosti koeficijenata se mogu izračunati struje podešenja releja na sekcijama 3, 18, 26 i 30, pri čemu one iznose 1230 A, 250 A, 110 A i 220 A, respektivno. Analiziraju se situacije kada je karakteristika releja: normalno zavisna, vrlo zavisna, ekstremno zavisna i vrlo dugo zavisna. Na slici 4.2. su prikazani releji sa normalno zavisnom karakteristikom.

Sa slike se može primetiti da je vreme reagovanja releja duže udaljavanjem mesta kvara od ekvivalenta napojne 35 kV mreže. Takođe se može zaključiti da su vremena reagovanja releja na lateralima kraća od vremena reagovanja releja na sekciji 3 za minimum 0,2 s, što znači da je uslov selektivnosti ispunjen – uvek će pre reagovati relej koji je bliže kvaru. Slični zaključci se mogu zaključiti i za ostale tipove karakteristika releja. Na slici 4.3. prikazane su sve karakteristike reagovanja releja sa strujno zavisnom karakteristikom koji je dat u delu 3.



Slika 4.2. – Releji sa normalno zavisnom karakteristikom



Slika 4.3. – Poređenje karakteristika reagovanja releja iz sekcije 3

Sa grafika 4.3. se može videti da releji sa većim stepenom zavisnosti reaguju brže za veće struje kvara (kvarovi bliže razvodnom postrojenju), a sporije za niže vrednosti struje kvara (kvarovi koji su u dubini mreže).

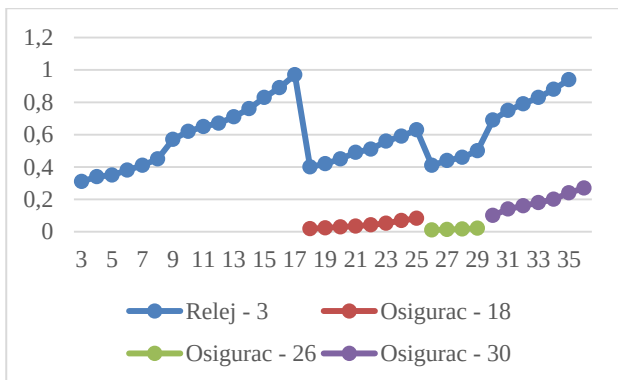
4.2. ZAŠTITA MREŽE RELEJIMA SA INVERZONOM KARAKTERISTIKOM I OSIGURAČIMA

U ovom potpoglavlju će biti analizirana situacija kada je na sekciji 3 postavljen relej sa inverznom karakteristikom reagovanja, dok će na sekcijama 18, 26 i 30 biti postavljeni osigurači. Nominalne struje osigurača će se izabrati na sledeći način:

$$I_{min,pod} = k_{sigurnosti} \cdot I_{max,radno}. \quad (4.2)$$

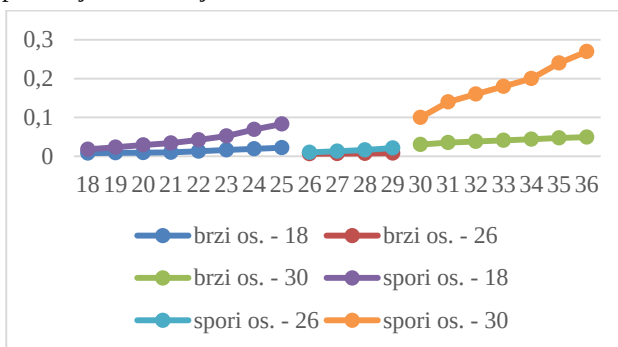
Minimalne podešene vrednosti struja osigurača iznose 205 A, 92 A i 180 A, zbog čega su izabrati osigurači sa nominalnim strujama od 100 A, u sekciji 26, i 200 A, u sekcijama 18 i 30. Analiziraju se situacije kada relej ima različite vrste inverznih karakteristika dok su osigurači sa brzim i tromim umetkom. Na slici 4.4. je prikazana koordinacija releja i osigurača sa tromim umetkom.

Sa slike 4.4 se može primetiti da su vremena reagovanja osigurača na lateralima kraća od vremena reagovanja releja na sekciji 3 za minimum 0,3 - 0,4 s, što znači da je uslov selektivnosti ispunjen – uvek će pre reagovati relej koji je bliže kvaru iz čega se može zaključiti da je zaštita dobro koordinisana.



Slika 4.4. – Koordinacija releja sa normalno zavisnom karakteristikom i osigurača sa tromim umetkom

Na slici 4.5. prikazane su karakteristike reagovanja osigurača sa tromim, odnosno brzim umetkom, koji su postavljeni u sekcijama 18, 26 i 30.



Slika 4.5. – Poređenje karakteristika reagovanja osigurača sa brzim i tromim umetkom

4.3. ZAŠTITA REALNIH MREŽA

Analizira se realna elektroenergetska mreža. Svi parametri i podaci o mreži su isti kao i podaci o test mreži koja je opisana u poglavlju 4.1. Osim svega navedenog, u realnim mrežama postoji i zaštita potrošačkih područja. Potrošačka područja se štite osiguračima od 100 A koji se nalaze ispred elektroenergetskog transformatora 10/0,4 kV/kV. U situacijama kada osim releja na glavnom izvodu i releja ili osigurača na lateralnim izvodima postoji i osigurač ispred potrošačkog centra, potrebno je izvršiti proveru usaglašenosti vremena reagovanja svih zaštitnih uređaja. Zaštitni uređaj koji je, gledajući prema napojnoj transformatorskoj stanici, najbliži mestu kvara bi trebao da ima najkraće vreme reagovanja.

Na lateralnom izvodu 18-25 najmanja struja koja prolazi kroz relej u sekciji 18 se dobija kada se kvar desi u čvoru 25. Osigurač u potrošačkom području 25 će videti kvar na niskonaponskim sabirnicama tog transformatora, što je ekvivalentno struji kvara u čvoru 25 i ta struja iznosi 3,69 kA. Tromi osigurač od 100 A u potrošačkom području 25 će odreagovati za 0,01 s pri navedenoj struji kvara. Struja reagovanja releja na sekciji 18 pri kvaru u čvoru 25 za:

- 1) normalnu karakteristiku iznosi 0,253 s,
- 2) vrlo zavisnu karakteristiku iznosi 0,098 s,
- 3) ekstremno zavisnu karakteristiku iznosi 0,037 s.

Može se zaključiti da je za sve tipove karakteristika zaštita dobro koordinisana jer u svim situacijama osigurač ispred potrošačkog područja reaguje pre releja u sekciji 18.

5. ZAKLJUČAK

U radu je izvršeno podešenje i provera koordinacije različitih tipova zaštite u test mreži. Zaštita je postavljena na samom početku distributivnog voda i na svakom njegovom lateralnom izvodu. Podešenje zaštite je izvršeno na osnovu proračunatih struja u normalnom radnom režimu, a zatim su određivana vremena reagovanja zaštite na osnovu struja kratkog spoja u svakom od čvorova mreže. Ispitivanje koordinacije je vršeno po principu zadovoljenja uslova da zaštita koja je bliže kvaru ima kraće vreme reagovanja od zaštite koja je dalje od kvara. Na samom vrhu distributivnog voda se nalazi prekostrujni relej dok su na izvodima analizirane situacije kada se nalaze prekostrujni releji i topljivi osigurači. Analizom situacija su izvedeni sledeći zaključci:

- 1) Vreme reagovanja releja je duže udaljavanjem mesta kvara od pozicije releja u mreži.
- 2) Vremena reagovanja zaštite na lateralima su kraća od vremena reagovanja releja na početku distributivnog voda za minimum 0,2 s, što znači da je uslov selektivnosti ispunjen.
- 3) Releji sa većim stepenom inverznosti reaguju brže za veće struje kvara (kvarovi bliže razvodnom postrojenju), a sporije za niže vrednosti struje kvara (kvarovi koji su u dubini mreže).
- 4) Povećanjem dužina vodova oblast zaštite se smanjuje.
- 5) Osiguraču sa brzim umetkom potrebno je manje vremena da odreaguje od osigurača sa tromim umetkom. Oba tipa se mogu upotrebiti za zaštitu u ovom primeru.

5. LITERATURA

- [1] N. Rajaković, D. Tasić, G. Savanović, *Distributivne i industrijske mreže*, Beograd, 2004.
- [2] M. Nimrihter, *Elektrodistributivni sistemi*, FTN Izdavaštvo, Novi Sad, 2009.
- [3] N. Rajaković, D. Tasić, *Distributivne i industrijske mreže*, Akademska misao, Beograd, 2008.
- [4] D. Bekut, *Relejna zaštita*, FTN Izdavaštvo, Novi Sad, 2009.
- [5] V.C.Strezoski, *Osnovi elektroenergetike*, Fakultet tehničkih nauka Novi Sad, 1996.
- [6] Releji – teorija upravljanja, <https://www.automatika.rs/bazaznanja/teorijaupravljanja/releji.html>
- [7] M. Milanković, D. Perić, I. Vlajić-Naumovska, *Osnovi elektroenergetike*, Beograd, 2016.

Kratka biografija:



Viktor Trifunjagić je rođen u Zrenjaninu 1995. godine. Osnovne studije je završila na Fakultetu tehničkih nauka 2018. godine na departmanu Elektrotehnika i računarstvo – Elektroenergetski sistemi. Master rad je odbranio 2019. godine, na istom fakultetu na smeru Elektroenergetski sistemi.

ZAŠTITA KABLOVA

CABLE PROTECTION

Marija Sarić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu se razmatra zaštita kablova, osnovne karakteristike i izvođenje krivih oštećenja kablova, kao i koordinacija sistema. Cilj je da se prikažu prednosti primene korišćenja vremensko-strujnih krivih kao grafičke tehnike za ilustraciju selektivne koordinacije zaštite. Za softversku verifikaciju rada se koristi program za koordinaciju releja.

Ključne reči: kablovi, prekostrujna zaštita, prekostrujna koordinacija

Abstract – The paper deals with cable protection, basic characteristics and performance of cable damage curves as well as system coordination. The aim is to show the benefits of applying the use of time-current curves as a graphical technique to illustrate selective system coordination. A program for relay coordination is used for software verification of operation.

Keywords: cables, overcurrent protection, overcurrent coordination

1. UVOD

Električna energija predstavlja univerzalni vid energije bez kojeg se ne može zamisliti funkcionisanje savremenog društva.

Da bi elektroenergetski sistem mogao da funkcioniše potrebno je obezbediti zaštitu svakog njegovog elementa, a takođe i pojedinih celina u okviru elektroenergetskog sistema od kvarova.

Da bi se razumeli osnovni principi zaštite opreme potrebno je poznavanje osnovnih karakteristika mogućeg opterećenja opreme. Ovaj rad ce se baviti zaštitom kablova, osnovnim karakteristikama i razmatranjem krivih mogućih opterećenja kablova.

U drugom delu rada su predstavljeni osnovni principi zaštite elektroenergetskih sistema, a u trećem prekostrujna zaštita. Četvrti deo je posvećen zaštiti kablova. U petom delu je napisano uopšteno o prekostrujnoj koordinaciji zaštite. Šesti deo je softverska verifikacija koordinacije zaštitnih uređaja sa krivama oštećenja kabla, a poslednja dva dela su zaključak i literatura.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Duško Bekut, redovni profesor.

2. OSNOVNI PRINCIPI ZAŠTITE ELEKTROENERGETSKOG SISTEMA

Zaštita elektroenergetskih sistema se vrši od nenormalnih stanja. Pod nenormalnim stanjima se podrazumevaju:

- Opasna pogonska stanja.
- Stanja sa kvarom (kvarovima).

Stanje u kom osnovni parametri sistema, napon, struja i frekvencija, ne izlaze iz dozvoljene tolerancije naziva se normalno pogonsko stanje sistema.

Opasno pogonsko stanje predstavlja "lakši vid" odstupanja od normalnog pogonskog stanja, dok se stanja sa kvarom kategorizuju kao "teži vid" odstupanja.

Osnovni cilj primene relejne zaštite je najbrže moguće isključenje elementa ili dela EES-a sa kvarom uz očuvanje funkcionalnosti ostalog dela sistema [1].

2.1 Opšti zahtevi koji se postavljaju pred relejnu zaštitu

Osetljivost:

Osetljivost je sposobnost releja da odreaguje na najmanju očekivanu struju kvara. Relejna zaštita na fiderima mora biti podešena tako da odreaguje i u uslovima minimalne proizvodnje, kada je struja kvara takođe minimalna.

Selektivnost:

Cilj selektivno podešene zaštite je da minimalan broj opreme reaguje i da se izolovanjem sekcije sa kvarom bez napajanja ostavi najmanji mogući broj potrošača.

Brzina:

Brzina je sposobnost releja da reaguje u predviđenom vremenskom intervalu otkloniti što je pre moguće.

Pouzdanost:

Pouzdanost predstavlja sigurnost da će relejna zaštita pravilno raditi, bez obzira na spoljne faktore.

Jednostavnost:

Jednostavnost predstavlja zahtev da se uz minimalno angažovanje zaštite obezbede prethodno navedeni zahtevi.

Ekonomičnost:

Ova karakteristika predstavlja obezbeđenje maksimuma zaštite uz minimalne troškove.

2.2 Zaštitni uređaji u EES-u

Uređaji koji se koriste za izolaciju kvara su osigurači, prekidači i releji.

Relej je uređaj koji služi za detekciju nenormalnih pogonskih stanja uređaja i delova elektroenergetskih sistema i inicijalizaciju odgovarajućih upravljačkih akcija

za obezbeđenje normalnog pogona. Pod nenormalnim pogonskim stanjima se podrazumeva stanje sa kvarom ili opasno pogonsko stanje [1].

Za zaštitu manjih delova sistema se koriste osigurači, koji ujedno detektuju i izoluju kvar. Osnovna funkcija osigurača je da prekine protok struje nakon kvara ili preopterećenja.

2.3 Podela releja prema principu rada

Prema principu rada releji se dele na [1]:

- Elektromehaničke.
- Statičke.
- Mikroprocesorske.

2.3.1 Elektromehanički releji

Osobina elektromagnetne indukcije u kombinaciji sa mehaničkim delovima je korišćena za detekciju prekostrujnih i prenaponskih kvarova u sistemu.

2.3.2 Statički releji

Statički releji su se pojavili šezdesetih godina dvadesetog veka. Najveća prednost ovih releja je što nemaju pokretne delove i mogu se daljinski kontrolisati. Takođe, brzina delovanja je bila znatno veća, kao i vreme potrebno za reset.

2.3.3 Mikroprocesorski releji

Mikroprocesorski releji predstavljaju vrlo savremen i kvalitetan vid zaštite. Osnovna prednost je što je omogućen daljinski pristup i daljinska kontrola sa brojnim naprednim funkcijama.

2.4 Podela releja prema vrsti kontrolisane veličine

Prema vrsti kontrolisane veličine releji se dele na:

- Strujne – ovi releji deluju pri povećanju struje.
- Naponske – deluju kada napon na koji su priključeni odstupa od definisanih granica.
- Frekventne – deluju kada frekvencija odstupa od definisanih granica.
- Releje snage – kod kojih delovanje zavisi samo od smera snage.
- Impedantne – kod kojih delovanje zavisi od vrednosti impedance.
- Termičke – kojima se deluje pri porastu temperature ($T > T_0$).
- Mehaničke – kojima se deluje pri povećanju broja obrtaja $n > n_0$.

3. PREKOSTRUJNA ZAŠTITA

Za zaštitu vodova se koriste sledeće vrste zaštita:

- prekostrujna,
- distantna,
- diferencijalna.

Na vodovima srednjih napona najčešće se koristi prekostrujna zaštita.

Prekostrujna zaštita je najjednostavnija zaštita i ona se ostvaruje delovanjem prekostrujnih releja. Ovi releji su dizajnirani tako da reaguju kada kroz njih teče struja koja je veća od predefinisane vrednosti.

3.1 Trenutni prekostrujni releji

Ovim relejima se deluje čim struja premaši određenu vrednost. Trenutni nadstrujni releji u praksi se koriste za trenutno isključivanje jako velikih struja kvara. Njihova svrha je sprečavanje velike štete na skupim uređajima, kao što su transformatori, usled delovanja struje kratkog spoja.

3.2 Prekostrujni releji sa strujno nezavisnom vremenskom karakteristikom

Prekostrujni releji sa strujno nezavisnom vremenskom karakteristikom se pobuđuju kada struja premasi određenu vrednost ali se njima deluje tek nakon nekog vremena (vremena kašnjenja). Kod trenutnih nema razlike između pojma delovanje i pobuđivanje.

3.3 Prekostrujni releji sa strujno zavisnom vremenskom karakteristikom

Ovim relejima se deluje u zavisnosti od intenziteta struje. Što je struja veća, vreme delovanja je kraće i obrnuto. Vreme delovanja releja se može izračunati prema formuli:

$$t = \frac{k}{I} \left(\frac{I}{I_{\text{podešenja}}} \right)^{\alpha} \quad (3.1)$$

gde su:

t – vreme delovanja releja pri struji I ,
 I – struja kvara koju relej meri,
 $I_{\text{podešenja}}$ – podešena vrednost struje,
 α, k – koeficijenti osnovnog podešenja,
 k_f – koeficijent finog podešenja.

4. ZAŠTITA KABLOVA I ODREĐIVANJE KRIVIH OŠTEĆENJA KABLOVA

U ovoj glavi je dat kratak opis vrsta povodnika, temperaturni rast zbog struje opterećenja, a zatim i izvođenje vremensko-strujnih karakteristika (engleski *time current characteristic TCC*) kablova.

Zadatak provodnika kablova je da provodi električnu energiju od izvora do potrošača. Pri tom prenosu se stvaraju gubici koji zavise od dužine, preseka i tipa provodnika. Dva su materijala za provodnike koji se najčešće koriste: bakar i aluminijum.

Mora se biti oprezan kod postavljanja provodnika u prostorijama sa visokom temperaturom okoline, tako da radna temperatura ne prelazi onu koja je određena kao maksimalna za dati tip izolovanog provodnika [2].

Zaštita kablova podrazumeva primenu uređaja za zaštitu od kratkog spoja i prevelike struje, kao i pravilno dimenzionisanje kabla. Cela šema zaštite se zasniva na rejtingu kablova koja je prilagođena okolini i uslovima rada.

Kabl treba zaštititi od pregrevanja usled prekomerne struje kratkog spoja u provodniku. Kvar može biti na delu zaštićenog kabla ili na bilo kom drugom delu električnog sistema. Deo kabla sa kvarom se treba zameniti nakon čišćenja kvara.

Tokom jednofaznog kvara I^2R gubici u faznim provodnicima prvo podižu temperaturu provodnika, zatim izolacije kabla, zaštitnoj oblozi (omotača) i konačno okoline [3].

Za gole provodnike, provodnike bez izolacije, dozvoljena vrednost struje kratkog spoja je znatno veća od one koja je navedena za izolovane provodnike [4].

Zavisnost vremena t i struje I za ove provodnike je data sledećim jednačinama:

Aluminijumski provodnici bez izolacije:

$$t = \left(0.0671 \frac{A}{I} \right)^2, \quad (4.1)$$

Al/Če provodnici bez izolacije:

$$t = \left(0.0862 \frac{A}{I} \right)^2, \quad (4.2)$$

gde je:

A – površina poprečnog preseka kabla,

I – struja kratkog spoja [A].

Za izolovane kablove, "Damage curve" je kriva koja opisuje granicu struje kratkog spoja koja, ako se prekorači, oštećuje izolaciju kabla. Kriva se računa pod pretpostavkom da provodnik adijabatski zagreva pri kratkom spoju [5].

Ove vrednosti su povezane sledećom jednačinom:

$$t = k \log_{10} \frac{T_2 + m}{T_1 + m} \left(\frac{A}{I} \right)^2, \quad (4.3)$$

gde je:

t – vreme trajanja kratkog spoja 0,01s-10s,

T_1 – radna temperatura kabla [°C],

T_2 – maksimalna temperatura kabla prilikom kratkog spoja [°C],

k i m su konstante koje zavise od tipa provodnika. Vrednosti za ove konstante su dati u tabeli 4.1.

Tabela **Error! No text of specified style in document..1**

Vrednosti konstanti k i m za različite tipove provodnika

Tip provodnika	k	m
Cu	0,0297	234
Al	0,0125	228

5. PREKOSTRUJNA KOORDINACIJA

Dobrom koordinacijom među zaštitnim elementima se omogućuje selektivnost u radu sistema. Selektivnost obezbeđuje da se pri trajnim kvarovima, taj kvar izoluje na vreme, tako da ostavlja bez napajanja najmanji broj potrošača.

S druge strane postoji zabrinutost za zaštitu komponenti sistema. Step en štete koja nastaje prilikom kvara je proporcionalan količini vremena za koje kvar postoji i kvadrata struje kratkog spoja. Energija koja se tako

oslobađa je ogromna, zato se oprema mora zaštititi. Najbolja zaštita se dobija upotrebom uređaja koji su osetljivi na minimalnu količinu struje kvara koja bi mogla teći, a uređaj bi trebao izolovati grešku što je brže moguće.

5.1 Selektivnost

Deo sistema koji mora da se izoluje treba da bude najmanji mogući. Više zaštitnih uređaja može odgovoriti na grešku, cilj je da prvi zaštitni uređaj iznad kvara reaguje i tako isključi minimalnu oblast pod kvarom. Ako reaguje neki drugi, neki iznad njega, može doći do nepotrebnog izolovanja dela sistema, dela koji nije pod kvarom.

5.2 Vremensko-strujne karakteristike

Razumevanje vremensko-strujnih karakteristika (engleski *time current characteristic TCC*) je osnova za izvođenje koordinacione studije.

TCC kriva određuje vreme rada zaštitnog uređaja za različite vrednosti radnih struja.

Vremensko-strujna kriva je kriva zaštitnog uređaja iscrtana na grafiku sa logaritamskom skalom gde vertikalna osa predstavlja vreme a horizontalna osa predstavlja struju.

5.3 Koordinacija zaštitnih uređaja sa krivama oštećenja kablova

TCC grafik takođe prikazuje i krive oštećenja elemenata sistema. U ovom radu pažnja će se posvetiti kablovima. Izvođenje krivih je u glavi 4. Zaštitu treba postaviti tako da svaki kabl može izdržati struju kvara koja teče kroz njega. Da bi se sprečilo oštećenje kabla uređaj za prekid struje mora ograničiti struju kvara na levo od krive oštećenja.

Krive kablova obično se crtaju na vremensko-strujnim graficima i koriste se za osiguranje da će zaštitni sistem sprečiti oštećenja zbog struje koje prolaze kroz te provodnike. Krive oštećenja komponenti takođe se mogu nazvati 'Damage' krivama, jer pokazuju nivo struje i količinu vremena za koje komponenta može da održi potencijalno oštećujuću struju bez pregrevanja i oštećenja [6].

6. SOFTVERSKA VERIFIKACIJA KOORDINACIJE ZAŠTITNIH UREĐAJA SA KRIVAMA OŠTEĆENJA KABLA

Funkcionalnost koja se koristi za koordinaciju zaštite u programu za koordinaciju releja je Relay Protection – Coordination Analysis.

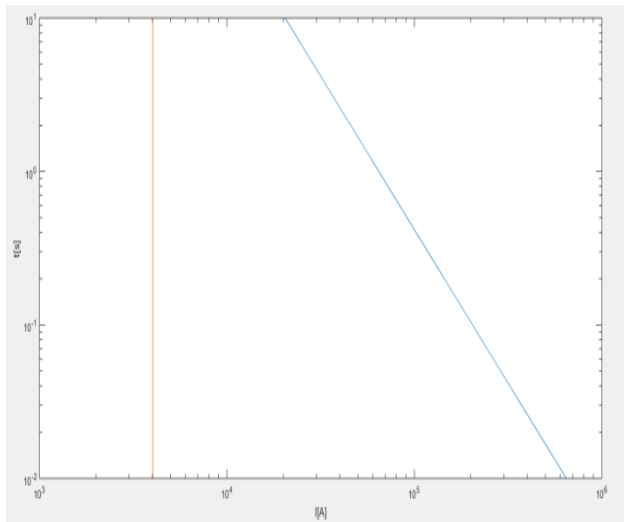
Za softversku verifikaciju je puštana funkcija za određeno mesto kvara i dobijen je tabelarni prikaz koordinacije zaštitnih uređaja na putanji kvara kao i vrednost struje kratkog spoja, a zatim je primenom MATLAB-a nacrtan grafik krive oštećenja kablova kroz koje prolazi struja kratkog spoja.

Rezultat funkcije je prikazan u tabeli 6.1.

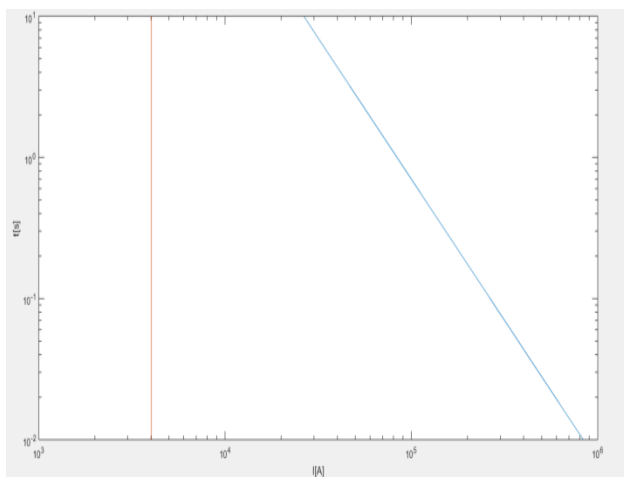
Struja kratkog spoja na mestu kvara i kroz kablove iznosi 4022.31A. Na osnovu formula (4.2) i (4.3) dobijaju se krive oštećenja Al/Če provodnika i Cu kabla. Grafici su prikazani na slikama 6.1 i 6.2 [7].

Tabela **Error! No text of specified style in document..1**
– Tabelarni prikaz rezultata analize koordinacije releja

Tip zaštite	Redni broj delovanja	Strujno podešenje uređaja [A]	Struja kvara [A]	Vreme delovanja [s]
51 - Fast	1	220.0	4022.3	0.402
Fuse	2	900.0	4022.3	0.419
Fuse	2	900.0	4022.3	1.079
51 - Slow	3	270.0	4022.3	1.544
51	4	420.0	4022.3	3.460
50	N/A	7670.0	4022.3	-



Slika **Error! No text of specified style in document..1**.
Dijagram struje kvara i kriva oštećenja Al/Če provodnika



Slika **Error! No text of specified style in document..2**.
Dijagram struje kvara i kriva oštećenja kabla

Rezultat softverske verifikacije jeste da su kablovi dobro zaštićeni, jer su napravljeni da mogu da izdrže izračunatu struju kvara.

Na osnovu tabele 6.1 vidi se da su uređaji su dobro koordinisani i da reaguju na struju kvara.

Prilikom kvara reagovala je brza karakteristika reklozera. Prekidač će se otvoriti, izolovaće mesto kvara, sačekati neko vreme i zatvoriti. Ako kvar i dalje postoji, onda će biti izvršena akcija reklozera. Posle dva otvaranja smatra se da je kvar trajni, karakteristika reklozera se prebacuje na sporu i otvara se fuses koji trajno izoluje kvar.

7. ZAKLJUČAK

Sa povećanjem snage i napona povećavaju se i potencijalne opasnosti i štete. Visoka temperatura koja nastaje zbog kontinuiranog preopterećenja, nelinearnih opterećenja ili neusklađene zaštite od kvarova je čest uzrok smanjenog veka kabla i njegovog kvara. Pravilan izbor osobina kabla osigurava da je kabl dovoljno siguran za očekivanu struju. Pogodna zaštita osigurava da porast temperature kabla ne postane preteran i sprečava kvar kablova.

Korišćenje vremensko-strujnih krivih kao grafičke tehnike za ilustraciju selektivne koordinacije zaštite olakšava prikazivanje da li je dobijena selektivnost zaštite i da li zaštitni uređaji adekvatno štite distribucijsku opremu. Na više primera u radu je pokazano kao se koriste ove krive kao i da su dobijeni rezultati zadovoljavajući, odnosno, da su izabrani provodnici dobro dimenzionisani.

8. LITERATURA

- [1] Duško Bekut, „Relejna zaštita”, Fakultet tehničkih nauka, Novi Sad, 2009.
- [2] IEEE Recommended Practice for Electric Power Distribution for Industrial Plants, IEEE Std 141-1993
- [3] IEEE Recommended Practice for Protection and Coordination of Industrial and Commercial Power Systems, Published by the Institute of Electrical and Electronics Engineers, Inc., IEEE Std 242-2001™
- [4] <http://www.skm.com/applicationguides8.html>
- [5] Thomas P. Smith, The ABC's of Overcurrent Coordination, P.E. January 2006
- [6] David Paul P.E., Understanding Time Current Curves, MAVERICK Technologies
- [7] ETAP ADMS demo software.

Kratka biografija:



Marija Sarić rođena je u Valjevu 1993. godine. Osnovne studije je završila na Fakultetu tehničkih nauka 2016. godine na departmanu Elektrotehnika i računarstvo – Elektroenergetski sistemi. Iste godine je upisala master studije na istom smeru.

KROVNE VETROELEKTRANE - MODELOVANJE I PREGLED SOFTVERSKIH ALATA ROOF WIND POWER PLANTS – MODELING AND REVIEW OF SOFTWARE TOOLS

Marija Jovanović, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Glavni fokus ovog rada je na primeni i iskorišćenju energije vetra u arhitekturi. Date su glavne prednosti i nedostaci vetrogeneratora u arhitekturi. Takođe predstavljeni su matematički modeli malih vetroelektrana, kao i njihove osnovne vrste i mesta primene. Analizirani su i softverski alati koji se mogu koristiti.

Ključne reči: Male vetroelektrane, matematički model, softverski alati

Abstract – The main focus of this paper is on the application and utilization of wind energy in architecture. The main advantages and disadvantages of wind turbines in architecture are given. Also presented are mathematical models of small wind farms, as well as their basic types and places of application. Software tools that can be used are also analyzed.

Keywords: Small wind farms, mathematical model, software tools

1. UVOD

Energija vetra je formirala globalno tržište koje uključuje veliki broj kompanija koje proizvode, snabdevaju, upravljaju vetroagregatima i integiraju svoju proizvedenu energiju u postojeće elektroenergetske mreže. Energija vetra je obnovljiv izvor energije koji potiče od snage vetra. Vetar je zapravo oblik solarne energije, jer je uzrokovan neravnomernim zagrevanjem atmosfere od sunca, nepravilnostima površine zemlje i rotacijom zemlje. Strujanja vetra su oblikovana od strane terena zemlje, vode i vegetacije. Industrija vetra je jedna od najbrže rastućih industrija širom sveta sa godišnjim rastom globalnog tržišta od skoro 10%.

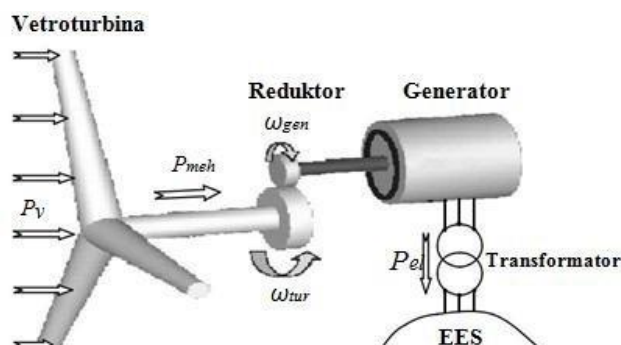
Energija vetra osim što je čisto, efikasno i održivo rešenje za proizvodnju energije, ono se takođe visoko ceni od strane javnog mnjenja. Ankete su pokazale da je velika većina građana prihvatila energiju vetra kao održivo rešenje za smanjenje CO₂ i da su u korist energije vetra posebno u Evropi, 89 % javnosti ima pozitivan stav prema energiji vetra.

Takođe, energija vetra je obnovljiva, ne zagađuje vazduh, ne prouzrokuje kisele kiše, ne zrači i ne razara ozonski omotač. Sada i u budućnosti energija vetra se pokazala kao najozbiljniji obnovljiv izvor pri dostignutom razvoju tehnologije. Osnovni razlozi za to su:

- neizmerna količina energije,
- pad cena vetrogeneratora i prateće opreme,
- ekološki potpuno čist način pretvaranja energije,
- mala zauzetost zemljišta.

2. TEHNOLOGIJA VETROGENERATORA

Kinetička energija vetra se transformiše u mehaničku energiju pomoću vetroturbine. Brzinu obrtanja vetroturbine obično je potrebno prilagoditi zahtevanoj brzini generatora. Zato se koristi mehanički reduktor. Vetrogenerator može biti sinhroni ili asinhroni, pri čemu može raditi sa fiksnom ili promenljivom brzinom obrtanja.



Slika 1. Principalska šema konverzije energije vetra u električnu [1]

2.1. Vrste vetroturbina

Generalno postoje dva velika tipa turbine:

Vetroturbina sa vertikalnom osovinom (tzv. eng VAWT – „vertical axis wind turbine“) kod koje vetar struji normalno na osu rotacije pa ove vrste vetroturbina ne moraju imati sisteme za detekciju i prilagođavanje smeru duvanja vetra. Generator se kod ove vrste turbine smešta u podnožje te nisu potrebni jaki stubovi, ali opterećenje na donjem kraju osovine je ogromno i zbog velike težine same turbine, a i snage vetra koja se kroz nju prenosi strdaju mehanički elementi pre svega ležajevi u samom podnožju. Primer ovakve turbine je Darrius-ova turbina, koja se obično gradi sa dva ili tri luka i koja ima nizak stepen iskorišćenja, pa se iz tih razloga retko sreće i to uglavnom u području malih snaga.

Vetroturbina sa horizontalnom osovinom (tzv. eng. HAWT – „horizontal axis wind turbine“) može biti postavljena uz i niz vetar. Vetroturbine postavljene niz vetar se same prilagođavaju smeru vetra. Lopatice pri rotaciji prolaze kroz zavetrinu stuba pa se stvaraju neželjene mehaničke vibracije i buka što naravno smanjuje koeficijent energetske efikasnosti vetroturbine, pa se ovaj koncept ne koristi pri većim snagama.

NAPOMENA:

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

3. MALE VETROELEKTRANE

Mali vetroagregati su po načinu rada uglavnom jednaki velikim vetroagregatima i s razvojem industrije primenjuju se novi materijali, čime se bitno poboljšala pouzdanost i raspoloživost uređaja. Paralelno s razvojem velikih vetroagregata počeli su se razvijati i mali vetroagregati kao zasebno tržište. Iako ovakav sastav zahteva prilična početna ulaganja, cenom je postao konkurentan tradicionalnim izvorima energije, ako se uzme u obzir celi životni vek postrojenja i izuzeće troškova priključka na električnu mrežu. Malim vetroagregatima se smatraju jedinice do 10 kW, koje su namenjene zadovoljenu energetske potreba na nivou domaćinstva.

3.1. Prednosti integrisanja vetrogeneratora u zgradarstvu

Karakteristika brzine vetra jeste da se povećava sa visinom u odnosu na tlo, delimično i zbog manjeg uticaja drveća i drugih prepreka. Postavljanje vetrenjača na krov, naročito visokih zgrada, zvuči logično s obzirom da nije potrebno postavljati visoki stub kao nosač turbine [2].

U nekim slučajevima oblik i anelopa objekta mogu uticati na performanse integrisanih vetrenjača. Nekoliko proizvođača vetrogeneratora za integraciju u objekte koristi prednosti pojačanih strujanja u zoni atike (niskog obodnog zida, uglavnom kod ravnih krovova) gde se vetar penje uz fasadu većih zgrada i zatim oko zida atike na vrhu pravi jači kovitlac.

Najveći pojedinačni potrošač ukupne proizvedene električne energije u najvećem broju zemalja su upravo zgrade, pa proizvodnja na licu mesta (tj. na samoj parceli) smanjuje potrebu za skupim sistemima prenosa - infrastrukturom. Ovo takođe umanjuje gubitke kod prenosa, kao i količine potrebnog materijala (kablova, stubova i sl).

Kao još jednu prednost ovakvog pristupa u kreiranju koncepta arhitekture nekog objekta treba pomenuti i jasno vizuelno izražavanje stava da je vlasnik objekta posvećen očuvanju resursa i planete. Ovaj slučaj nemamo kod fotovoltaznih (PV - photovoltaic) solarnih panela, jer oni samo stoje negde na krovu, i često se ne vide sa ulice dok proizvode električnu energiju.

3.2. Mesto postavljanja malih vetrogeneratora

Vetroturbine dizajnirane za urbano okruženje mogu biti instalirane direktno na krovovima ili se mogu instalirati na prostoru pored zgrada. Lokacija instalacije zavisi od veličine i težine turbine koja zavisi od njenog tipa, primene i eventualno izlazne snage. Stoga, u zavisnosti od gore navedenih faktora, uzimaju se u obzir različiti parametri u pogledu opreme za montažu, strukturnih opterećenja i potrebnih delova.

Jedan od najvažnijih parametara je visina kule koja će ili smestiti lopatice i generator na vrhu (HAWT) ili će delovati kao osa rotacije (VAWT). Opšte je mišljenje da što je toranj veći, to je kvalitet vetra (nizak nivo turbulencije) bolji što će rezultirati efikasnijim i glatkim radom turbine.

Bez obzira na to što je toranj veći, biće potrebno više materijala da se izdrže sve gravitacione i aerodinamičke sile kako bi se osigurala sigurna konstrukcija. Više materijala direktno utiče na ukupnu težinu. Ako se turbina nalazi na zemljištu pored zgrada gde je težina manje važna od maksimiziranja proizvodnje energije, toranj može doseći i do 25 m kako bi se izbegla inducirana turbulencija stabala, zgrada i drugih konstrukcija.

S druge strane, ako se turbina postavi direktno na krov, dozvoljena visina će biti praktično ograničena težinom kule, jer su bezbednosna i vibraciona pitanja od većeg značaja nego maksimiziranje proizvodnje energije. Većina instaliranih HAWT-a na krovu koristi čelike od 5 do 8 metara ili druge materijalne stubove.

3.3. Neke od turbina dostupnih na tržištu EU I SAD

3.3.1. Aero Vironment AVX1000

Ova kompanija je lider u tehnologiji integrisanih turbina. Godine 2006., kompanija je predstavila turbinu od 400W dizajniranu da iskorišćava vetar na atikama objekta. Turbine su dizajnirane da se postavljaju u nizu, npr. 20 komada na zgradi.



Slika 2. Aero vironment AVX1000 [2]

3.3.2. Aerotecture International - turbine na vetar sa spiralnim rotorom

Ovaj tip turbine je visine 3m i 1,5 m u prečniku, 510 V, dizajnirana je za vertikalno instaliranje sa predviđenim izlazom od 1 kW na vetru od 14 m/s. Polazna brzina na kojoj turbina počinje da proizvodi energiju je 2,8 m/s.



Slika 3. Spiralna Aerotecture turbina na zgradi u Čikagu [2]

3.3.3. Quiet Revolution QR5 – turbina sa vertikalnom osom

Turbina Quiet Revolution QR5 trenutno je dostupna samo na tržištu Velike Britanije. Zasnovana je na elegantnoj spiralojnoj jajolikoj formi (Darrieus), a kao materijal za izradu korišćena su vlakna od karbona i fiberglasa.

Visoka je 5 m (bez nosećeg stuba), sa 3,1 m u prečniku turbina može stajati na nezavisnom stubu ili na krovu zgrade.



Slika 4. Quiet Revolution turbine [2]

3.3.4. Vetrogenerator Swift Wind

Prečnik ove turbine je 2,1 m. Ima pet krakova i spoljni ram, kao i dva moguća položaja ugla. Dizajnirana je da stoji na krovovima kuća, na aluminijumskom stubu, sa udaljenjem od krova za samo 0,6 m.

Proizvođač tvrdi da je skoro nečujna. Pravi se od polimera armiranim nano vlaknima.



Slika 5. Swift Wind turbina [2]

4. MATEMATIČKI MODEL VETROTURBINE

Kinetička energija raspoloživa u vetru predstavlja kinetičku energiju velikih vazdušnih masa, koje se kreću iznad zemljine površine. Ove vazdušne mase krećući se kroz lopatični prostor turbine predaju deo kinetičke energije koja se transformiše u mehanički rad ili dalje u električnu energiju, zavisno od želje potrošača. Posebno u prošlosti mehanička energija se koristila za pokretanje mlinova za mlevenje žitarica ili za pumpanje vode. Efikasnost konverzije energije vetra u druge korisne energetske oblike zavisi od efikasnosti interakcije rotorskih lopatica sa strujom vetra.

4.1. Matematički model trokrake turbine

Kinetička energija struje vazduha mase m i brzine V data je jednačinom:

$$E_k = \frac{1}{2} m V^2 \quad (4.1)$$

Masa čestice je određena gustinom ρ , površinom kroz koju se kreće S , brzinom V i vremenom kretanja t , $m = \rho SVt$, tako da izraz (4.1) dobija sledeći oblik:

$$E_k = \frac{1}{2} \rho SV^3 t \quad (4.2)$$

Snaga posmatrane čestice, odnosno snaga vetra, određuje se kao izvod kinetičke energije po vremenu, dE_k/dt i iznosi:

$$\frac{dE_k}{dt} = P_{vetra} = \frac{1}{2} V^2 \frac{dm}{dt} \quad (4.3)$$

$$\frac{dm}{dt} = \rho SV \quad (4.4)$$

$$P_{vetra} = \frac{1}{2} \rho SV^3 \quad (4.5)$$

Prema (4.5) jasno je da je snaga vetra proporcionalna gustini vazduha i trećem stepenu njegove brzine. Gustina vazduha zavisi od temperature, pritiska i vlažnosti. Standardizovana vrednost je $\rho = 1,225 \text{ kg/m}^3$ za pritisak od 101325 Pa i temperaturu od 15 °C na površini mora.

4.2. Matematički model Savoniusove turbine

Chauvin je predložio matematički model zasnovan na padu pritiska na obe strane lopatica, da bi se procenila potencijalna snaga rotora sa dve lopatice, sa razmakom između njih [3]. Trenutni vektor rotacije je $\vec{\omega} = \dot{\alpha} \vec{k}$ i zbog simetrije Savoniusovog rotora obrtni momenat turbine je:

$$Q = \sum_i (\vec{OM} \times \vec{F}_i) \cdot \vec{k} \quad (4.6)$$

Pretpostavimo da je razlika pritiska na oštrici koja se povlači i na oštrici koja se pruža Δp_M i Δp_D respektivno, pa se obrtni momenat turbine može zapisati i kao:

$$Q = 2r^2 H \int_0^{\pi/2} (\Delta p_M - \Delta p_D) \sin 2\theta d\theta \quad (4.7)$$

Gde je:

θ – ugao zakreta lopatica,

r – poluprečnik lopatice,

H – visina rotora.

Prosečna snaga P se dobija integracijom obrtnog momenta u granicama od 0 do π .

$$P = \omega Q = \frac{\omega}{\pi} \int_0^\pi Q d\alpha \quad (4.8)$$

gde je: ω – ugaona brzina, α – azimutni ugao.

Prema eksperimentalnom istraživanju doslo se do zaključka da Darrieus-ova turbina nije samostalna, dok Savonius-ov rotor ima veoma dobre startne momentne karakteristike. Kod Savoniusovog rotora sa dve lopatice postoji velika varijacija statičkog momenta u odnosu na ugaoni položaj, dok su te varijacije kod Savoniusovog rotora sa tri lopatice manje. Savoniusov rotor će startovati iz bilo kog početnog ugla, kada je statički aerodinamički obrtni momenat veći od opterećenja i momenta trenja. Koeficijent maksimalne snage kod Savoniusovog rotora sa dve lopatice je približno 1.5 puta veći od konfiguracije sa tri lopatice. Koeficijenti aerodinamičkih performansi Savoniusovog rotora se normalizuju pomoću koeficijenta ($q_\infty V_\infty 4rH$) koji je konstantan za sve konfiguracije [3].

5. ANALIZA I PREGLED SOFTVERSKIH ALATA

Najzastupljeniji specijalizovani programi za analizu strujanja vetra koji služe kao pomoć pri modelovanju i izgradnji vetroelektrane su Meteodyn WT (UrbaWind), Furow, WAsP, WindSim, WindStation. Osnovna uloga ovih softverskih alata je prevashodno proračun strujanja vetra kao i prikaz godišnje proizvodnje energije. Svaki od njih je drugačiji na svoj način ali uglavnom svi imaju iste ili slične metode za rad.

Pored osnovne uloge, ovi programi imaju sposobnost da obavljaju i neke funkcije za projektovanje vetroelektrana kao što su izračunavanje proizvodnje turbina na određenim lokacijama, određivanje idealnih pozicija za njihovo instaliranje (s obzirom na strujanja vetra), određivanje trag efekta i sl.

Ovi softverski alati su detaljnije opisani u master radu. Opisani programi su dosta slični zato što su namenjeni da rade iste stvari. Svaki od njih sadži CFD metodologiju koja je dostupna samo kad se program plati u potpunosti.

Nakon njihove analize i ocene, najpogodniji softverski alat za projektovanje je WAsP [4].

6. ZAKLJUČAK

Predmet naučne analize u ovom radu jeste predstavljanje matematičkih modela vetroturbina i opis softverskih paketa koji se mogu koristiti za projektovanje malih vetroelektrana.

Na samom početku rada opisan je potencijal vetra u svetu i u Srbiji, tehnologija vetrogeneratora, kao i njihove osnovne vrste i karakteristike. Drugi deo rada posvećen je opisu malih vetroelektrana. Male vetroelektrane su namenjene za lokalno korišćenje, za proizvodnju snage od 1 kW do 10 kW. Većina malih vetrogeneratora su horizontalno orijentisani, međutim vertikalno orijentisani vetrogeneratori se sve više upotrebljavaju i njihova prodaja je u porastu.

Pošto projekat malih vetroelektrana nije dovoljno realan u gradovima, rešenje bi bilo korišćenje preciznijih alata za procenu vetra i predviđanja koja uzimaju u obzir efekte senčenja i turbulencije okolnih prepreka u urbanim sredinama.

Mana malih vetroelektrana jeste velika buka i vibracije, pa rešenje leži u tome da se dizajniraju novi noževi za njihovo smanjenje. Takođe, još jedna mana jeste da male turbine nisu bezbedne (za ljude i okolinu) u gradovima. Rešenje je u primeni sistema nadzora i prikupljanja podataka (SCADA sistem) za poboljšanje rada i održavanja – kontrola turbina i pitanja sigurnosti.

Ovakva rešenja bi trebalo da pomognu u smanjenju investicionih troškova i povećanju atraktivnosti i primeni malih vetroelektrana.

U radu su opisane glavne prednosti i nedostaci integrisanja malih vetroelektrana u arhitekturi, kao i njihova mesta postavljanja. Na samom kraju tog dela predstavljene su neke vrste vetroelektrana, različitih proizvođača širom Evrope i SAD.

Nakon ovog dela u radu je predstavljen matematički model trokrake i Savoniusove turbine i opis i pregled softverskih alata. Softverski alati se koriste za projektovanje vetroelektrana. Da bi se one projektovale potrebno je prvo odgovoriti na neka od pitanja: koliko energije turbine proizvode na određenim lokacijama, kako napraviti optimalan raspored turbina obzirom na duvanja vetra i ograničenja zbog zaštite životne sredine, koje su posledice priključenja na već postojeću mrežu, koliki su troškovi za izgradnju, koliki kapacitet vetroparka možemo dobiti i dr...

7. LITERATURA

- [1] <https://sites.google.com/site/tehskolamojaucionica/home/obnovljivi-izvori-energije/princip-konverzije-energije-vetra-u-elektricnu/>
- [2] https://gradjevinarstvo.rs/tekstovi/652/820/ku%C4%87a-vetrenja%C4%8Da-%E2%80%93da-li-je-integracija-vetrogeneratora-u-objekte-mogu%C4%87a-i-isplativa?fbclid=IwAR0QnOc6Sq3jeNB6qB5PBfpJB4BlN1RrWgSsMCnPx9_rCAC_F2ScuHeDEIc
- [3] Ion Paraschivoiu „Wind Turbine Design“, Polytechnic International Press
- [4] <https://www.wasp.dk/>

Kratka biografija:



Marija Jovanović rođena je u Užicu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Distribuirani elektroenergetski resursi odbranila je 2019. god. kontakt: jovanovic.marija031@gmail.com



dr Vladimir Katić red.prof. rođen je u Novom Sadu 1954. godine. Doktorirao je na Univerzitetu u Beogradu 1991. godine. Oblasni interesovanja su mu energetska elektronika, obnovljivi izvori električne energije, kvalitet električne energije i električna vozila.

СИСТЕМ ЗА УПРАВЉАЊЕ РАЊИВОСТИМА У СОФТВЕРУ**SOFTWARE VULNERABILITY MANAGEMENT SYSTEM**Марија Ковачевић, *Факултет техничких наука, Нови Сад***Област – СОФТВЕРСКО ИНЖЕЊЕРСТВО И ИНФОРМАЦИОНЕ ТЕХНОЛОГИЈЕ**

Кратак садржај – У раду је описана имплементација система за управљање рањивостима у софтверу и објашњени су механизми за откривање јавно идентификованих рањивости.

Кључне речи: *Моделовање претњи, дијаграм тока података, база знања, NVD, CPE, CVE, XML*

Abstract – *This paper presents implementation of the Software Vulnerability Management System. Description of the mechanisms for detecting publicly known vulnerabilities is given.*

Keywords: *Threat modelling, Data flow diagram, Knowledge base, NVD, CPE, CVE, XML*

1. УВОД

У данашње време, напади на софтверске системе постају све учесталији. Нападач може да угрози читав систем нападом на само један ток података или један елемент система. Произвођачи софтвера се константно боре са овим проблемом и раде на његовом сузбијању. Један од начина за решавање овог проблема јесте разматрање безбедносних аспеката током читавог животног века софтвера укључујући и фазу дизајна производа [1]. Дугогодишњим радом, велики број техника је развијен у сврху решавања различитих безбедносних проблема. Једна од тих техника је моделовање претњи система.

Моделовање претњи система захтева разумевање комплексности целог система и идентификовање свих могућих претњи, без обзира да ли напад може одмах да се реализује [1]. Овај посао треба радити систематично и на време, јер се само на тај начин може гарантовати да ће потенцијалне претње и рањивости бити откривене од стране програмера, а не нападача. Ако се моделовање претњи ради након што је производ већ у употреби, сузбијање напада и отклањање рањивости ће бити много захтевније и скупље, него да се вршило у фази дизајна система.

Тема овог рада је креирање Система за управљање рањивостима у софтверу. Систем треба да на основу дијаграма тока података креира извештај са потенцијалним нападима на софтвер и листом јавно идентификованих рањивости у компонентама софтвера.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био проф. др Горан Сладић.

2. МЕХАНИЗМИ ЗА ОТКРИВАЊЕ РАЊИВОСТИ

Сви софтвери, који поседују механизме за откривање рањивости, у великој мери умањују степен ризика од потенцијалних напада и обезбеђују ефикаснију израду буџета за дати софтвер.

2.1. Дијаграм тока података

Дијаграм тока података (ДТП) је графички дијаграм за конструкцију и визуелизацију модела система [2]. Он служи за дефинисање захтева у графичком приказу. Користи се приликом моделовања претњи, јер је лак за разумевање и усмерен је на податке и њихов ток [3]. ДТП има 4 врсте компоненти [4]:

- Процес - активност или функција, која извршава одређену бизнис логику.
- Складиште података - репозиторијум где се подаци чувају или одакле се добављају.
- Ток - репрезентује податке, који се преносе између елемената.
- Спољни елементи - елементи ван система, који су у интеракцији са њим.

2.2. Шема за именовање производа

Common Platform Enumeration (CPE) је стандардизован начин за описивање и идентификацију класа апликација, оперативних система и хардверски уређаја [5]. CPE идентификује производе тако што за сваки наводи низ парова атрибут-вредност.

2.3. Листа рањивости и изложености

Листа рањивости и изложености (енг. *Common Vulnerabilities and Exposures*) (CVE) је листа јавно познатих рањивости и изложености из поља рачунарске безбедности [6]. Ова листа је бесплатна и доступна свима. Састоји се од ставки, где свака ставка представља рањивост и омогућава лако приступање информацијама о њој уз помоћ више референци које је чине [6]. Ставку чине:

- идентификатор,
- опис,
- минимум једна референца ка јавно познатој рањивости.

2.4. Национална база рањивости

Национална база рањивости (енг. *National Vulnerability Database*) (NVD) је база владе Сједињених Америчких Држава објављена 2005. године од стране Националног института за стандарде и технологију [7]. Задатак базе је да преузима ставке из CVE листе и да врши њихову анализу. Експерти за безбедност анализирају CVE ставке и проширују их са додатним метаподацима попут: CPE, CVSS, итд. [8].

2.5. Систем за оцењивање рањивости

Систем за оцењивање рањивости (енг. The Common Vulnerability Scoring System – CVSS) обрађује главне техничке карактеристике рањивости софтвера и хардвера. Резултат анализе су нумеричке вредности које указују на озбиљност анализиране рањивости у односу на друге рањивости [9].

3. СИСТЕМИ ЗА МОДЕЛОВАЊЕ ПРЕТЊИ

Моделовање претњи је техника анализе дизајна [3]. Може да буде базирана на ресурсима, (енг. *Asset-centric*), нападачу (енг. *Attacker-centric*) или софтверу (енг. *Software-centric*). Резултат моделовања су рангиране претње моделованог система.

Моделовање могу да раде експерти, инжењери или и експерти и инжењери заједно. Тежи се оспособљавању инжењера да самостално моделују, јер ангажовање експерата може да буде веома скупо [3].

Главни циљ моделовања је унапређење безбедности софтвера [3]. Моделује се у раним фазама развоја, јер нападачи могу да направе велику штету произвођачима. Штета која је материјалне природе се надокнадити, али нарушен кредибилитет се тешко поново стиче, зато произвођачи све озбиљније схватају зашто је моделовање битно.

У овом поглављу су описана два алата за моделовање претњи: *Threat Modelling Tool* и *Систем за проналажење напада*. Систем за проналажење напада је настао по узору на *Threat Modelling Tool* и исправио је суштинске недостатке тог алата.

3.1. Threat Modelling Tool

Threat Modeling Tool (TMT) је алат који омогућава софтверским архитектурама да рано открију и уклоне потенцијалне проблеме из домена безбедности софтвера.

Рано уочени проблеми у великом мери смањују укупне трошкове и време утрошено за развој софтвера [10].

Цео алат је изузетно једноставан за употребу, јер је дизајниран тако да особе које нису експерти из области безбедности, могу да направе и анализирају дијаграме тока података [10].

3.2. Систем за проналажење напада

Систем за проналажење напада (енг. *Exploits Detection System*) (СПН) је софтверски алат који се користи за моделовање претњи [11].

СПН као улазне параметре прима дијаграм тока података у XML формату, XML датотеку са дефиницијом ресурса и XML датотеку са дефиницијом напада. Као резултат кориснику се враћа извештај о потенцијалним нападима.

СПН је исправио суштинске недостатке које постоје у TMT-у [11]:

- Немогућност извоза креираних ДТП-ова.
- Непостојање ресурса (енг. *Asset*) као фактора при анализи и креирању ДТП-а
- Ограниченост шаблона при анализи дијаграма

4. МОДЕЛ СИСТЕМА

Декомпоновање дијаграма тока података резултује листом шаблона. Сваки шаблон је представљен класом *DiagramPattern* (слика 1). Он садржи почетни и крајњи елемент тока података, као и ресурсе на тим елементима. Када се заврши анализа шаблона и пронађу потенцијални напади, они се смештају у листу *foundExploits*. Касније се, приликом увезивања дефиниција напада, вредности из ове листе замене описима напада и контрамерама за сузбијање истих.

DiagramPattern	
- element	: BlockElement
- assets	: List<Assets.Asset>
- traceStart	: BlockElement
- assetsOnTraceStart	: List<Assets.Asset>
- trace	: List<Element>
- foundExploits	: List<String>
- exploitValues	: List<ExploitDefinition>
- assetValues	: List<String>
+ DiagramPattern (BlockElement element, BlockElement traceStart, List<Element> trace)	
+ setAssetValues (Assets assets)	
+ addExploitValue (ExploitDefinition exploitDefinition)	
+ addExploit (String exploit)	
+ removeExploit (String exploit)	

Слика 1. Класа шаблона

Систем за управљање рањивостима у софтверу приликом генерисања потенцијалних напада пролази кроз две велике фазе:

- Анализа дијаграма и генерисање рањивости за комплексне елементе.
- Генерисање рањивости за цео дијаграм и креирање извештаја.

5. ИМПЛЕМЕНТАЦИЈА СИСТЕМА

У наредном поглављу бити представљена имплементација веб-апликације Система за управљање рањивостима у софтверу.

5.1. Имплементација базе знања

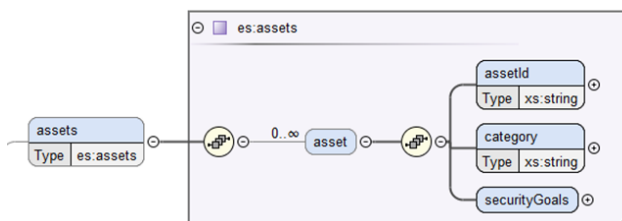
Сваки пронађени шаблон на ДТП бива прослеђен бази знања са циљем да се анализира и да се пронађу потенцијалне претње. Анализа шаблона резултује листом напада, који могу да се изврше над њим.

База знања система је имплементирана уз помоћ *Drools* технологија. База се састоји од једног документа у ком се налазе сва правила.

Правила су подељена на подгрупе у зависности од типа елемента, а то су:

- правила за процесе,
- правила за складишта података,
- правила за спољне елементе.

Елементи ДТП-а садрже листу ресурса. Сваки ресурс има ID, категорију ресурса којој припада и листу сигурносних циљева. На слици 2. налази се приказ дела шеме ДТП-а.



Слика 2. Део шеме ДТП – приказ ресурса

Сваки сигурносни циљ има назив и приоритет. Сигурносни циљеви су:

- поверљивост
- интегритет и
- доступност

Сигурносни циљеви ресурса се преносе на елементе. Када елемент садржи ресурс чија нпр. поверљивост треба да се заштити, онда се тај сигурносни циљ пребацује на елемент.

Ако елемент *складиште података* поседује ресурс *лозинке*, који има сигурносни циљ *поверљивост*, онда се штити поверљивост целог елемента тј. *складишта података*.

На листингу 1. налази се пример правила за детектовање *sniffing* напада. Овај напад се дешава у ситуацијама када се неовлашћено пресрећу пакети података.

Правило пролази кроз све ресурсе на елементу и проверава да ли је један од сигурносних циљева поверљивост. Ако се штити поверљивост ресурса, следи да се штити и поверљивост елемента. Уколико се користи HTTP протокол као веза између елемената, онда се веома лако може нарушити поверљивост података реализовањем *sniffing* напада.

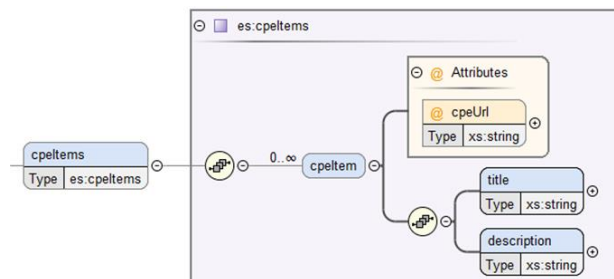
```
rule "AP:_Check_Confidentiality"
agenda-group "ap_check_primary"
when
  pattern: DiagramPattern($list : assets)
  $asset : Assets.Asset() from $list
    $securityGoals : Assets.Asset
      .SecurityGoals() from
        $asset.securityGoals
    $goal : Assets.Asset.SecurityGoals
      .SecurityGoal() from
        $securityGoals.securityGoal
  Boolean(booleanValue == true) from $goal
    .name == "confidentiality"
  elementInTrace : Element() from pattern
    .trace
  DiagramPattern(elementInTrace instanceof
    Http)
then
  pattern.addExploit("ed_sniffing");
end
```

Листинг 1. Једно од правила које посматра цео ток података

Ако је услов из правила задовољен тј. потенцијални напад је откривен, онда се извршава последица правила и идентификатор *sniffing* напада „*ed_sniffing*” се чува у оквиру тренутно анализираног шаблона.

5.2. Имплементација компоненте за откривање рањивости

Елементи ДТП-а, осим листе ресурса, могу да садрже листе дефинисаних CPE-ова (слика 3). Свака CPE ставка из листе садржи два елемента: наслов и опис. Најбитнији део CPE ставке је атрибут *cpeUrl*. У овом атрибуту ће се налазити путања, која идентификује CPE ставку и користи се приликом претраге базе рањивости.



Слика 3. Део шеме ДТП – листа CPE-ова

Систем може у сваком тренутку да контактира NVD и да затражи листу свих рањивости са последњим, најновијим изменама. Листа свих рањивости се добија у виду JSON датотеке. Прво се врши парсирање датотеке, а потом следи претрага листе за задати CPE.

5.3. Имплементација комплексног процеса

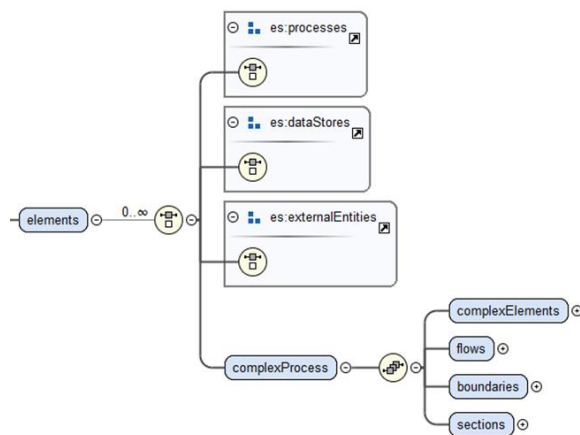
Шема ДТП-а поседује 4 елемента:

- елементи (енг. *elements*)
- токови (енг. *flows*)
- границе (енг. *boundaries*)
- области (енг. *sections*)

Елементи могу бити:

- процеси,
- спољни елементи,
- складишта података или
- комплексни процес.

Комплексни процес је нова врста елемента. То је чвор који поседује сопствени дијаграм тока података на ком је разложен у више целина. На слици 4. се налази део XML шеме ДТП на ком су приказани типови елемената.



Слика 4. Део шеме ДТП – врсте елемената

На почетку процеса анализе дијаграма, један од корака је декомпоновање система. Сваки пут када се пронађе комплексни процес, он се смешта у листу

комплексних процеса у оквиру дијаграма. Пронађени комплексни процеси се даље декомпонују на шаблоне, који се прослеђују бази знања на анализу.

Пошто се активирају одговарајућа правила из базе знања и идентификатори потенцијалних напада сачувају у шаблонима, учитава се XML датотека са дефиницијама напада. Следи замена идентификатора са целом дефиницијом напада.

Сви пронађени напади се смештају у листу и враћају кориснику. Корисник анализира потенцијалне нападе и одлучује које жели да пренесе на дијаграм. Напади који „опстану“ улазе у финални извештај.

5.4. Имплементација компоненте за генерисање извештаја

Финални извештај садржи:

- датум и време креирања извештаја,
- назив датотеке ДТП, који је анализиран,
- листу шаблона са пронађеним претњама,
- листу претњи које су пронађене на комплексним процесима,
- листу јавно идентификованих рањивости за пронађене CPE-ове и
- листу свих претњи са описом и контрамерама.

Добијени извештај у XML формату и креирани XSLT документ се прослеђују XSLT процесору, који врши трансформацију. Резултат трансформације је XHTML од ког се креира PDF.

6. ЗАКЉУЧАК

Тема овог рада је креирање система за управљање рањивостима у софтверу. Представљен је систем који на основу дијаграма тока података креира извештај са листом потенцијалних напада и листом јавно идентификованих рањивости у компонентама софтвера.

Објашњени су неки од механизма за откривање јавно идентификованих рањивости у софтверу. Описана је Листа рањивости (CVE), Национална база рањивости (NVD), као и ефикасан систем за оцењивање рањивости (CVSS).

У поглављу 3 описана су два алата: Threat Modelling Tool и Система за проналажење напада.

У поглављу 4 је објашњен модел система и кораци у раду.

Систем за управљање рањивостима у софтверу се ослања на СПН и врши његово унапређење. У поглављу 5 је приказана имплементација веб-апликације система и кључне новине, које је донела. Једна од новина је база знања, која више није базирана на ресурсима. Такође, имплементирана је и нова компонента, која служи за откривање јавно идентификованих рањивости. У овом поглављу је описан и комплексни процес, нови елемент ДТП. Поглавље се завршава приказом рада компоненте за генерисање извештаја, која генерише извештај у PDF формату.

Препорука за даљи развој веб-апликације је побољшање перформанси претраге јавно идентификованих рањивости. Тренутно претрага JSON датотеке са свим рањивостима, одузима највише времена у процесу

анализе. Размотрити употребу *Elasticsearch*-а или неког другог алата.

Даљи развој веб-апликације може да иде у смеру увођења компоненте за графички интерфејс. Увођењем ове компоненте драстично би се унапредио рад система, јер улазни дијаграм тока података не би могао више да се описује у текстуалној датотеци.

7. ЛИТЕРАТУРА

- [1] Rosziati Ibrahim, Siow Yen Yen, „Formalization of the data flow diagram rules for consistency check“, International Journal of Software Engineering & Applications (IJSEA), 2010
- [2] Suvda Myagmar, Adam J. Lee, William Yurcik, „Threat Modeling as a Basic for Security Requirements“, National Center for Supercomputing Applications (NCSA)
- [3] Adam Shostack, „Experiences Threat Modeling in Microsoft“, Microsoft
- [4] Marwan Abi-Antoun, Daniel Wang, Peter Torr, „Checking Threat Modeling Data Flow Diagrams for Implementation Conformance and Security“
- [5] <https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7695.pdf> (приступљено у септембру 2019)
- [6] CVE Official Specification, <https://cve.mitre.org/>
- [7] <https://nvd.nist.gov/general> (приступљено у септембру 2019)
- [8] Clement Elbaz, Louis Rilling, Christine Moris, „Towards Automated Risk Analysis of "One-day" Vulnerabilities“
- [9] <https://www.first.org/cvss/specification-document> (приступљено у септембру 2019)
- [10] <https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool> (приступљено у септембру 2019)
- [11] Немања Миладиновић, Проналажење рањивости у софтверу на основу дијаграма тока података, Fakultet tehničkih nauka, Novi Sad, 2017.

Кратка биографија:



Марија Ковачевић рођена је у Сомбору 1995. год. Основну школу „Мирослав Антић“ у Оџацима завршила је 2010. године. Исте године уписује гимназију „Јован Јовановић Змај“ у Оџацима, општи смер. Гимназију завршава 2014. године као носилац Вукове дипломе и звања ђака генерације. Године 2014. уписује Факултет техничких наука у Новом Саду, смер Софтверско инжењерство и информационе технологије. Полаже све испите предвиђене планом и програмом са просечном оценом 9.88. 2018. године завршава основне студије и уписује мастер академске студије на истом факултету. Полаже све испите мастер студија предвиђене планом и програмом са просечном оценом 10.00.

SISTEM ZA UPRAVLJANJE BEZBIJEDNOŠĆU RAČUNARSKIH RESURSA

ASSET MANAGEMENT CYBER SECURITY SYSTEM

Nevena Lazarević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je izložena implementacija sistema za upravljanje bezbijednošću računarskih resursa organizacije. Detaljno su objašnjeni model i implementacija sistema. Dodatno, ukazano je na svrhu sistema i navedene su funkcionalnosti koje pruža sistem.

Ključne reči: *Resurs, Upravljanje, Bezbijednost, Generisanje izvještaja*

Abstract – *The paper presents the implementation of Asset management cyber security system. Detailed description about its model and implementation is given. Additionally, purpose and functionalities of system are pointed out.*

Keywords: *ITAM, Asset, Cyber security, Report generation*

1. UVOD

Kako bi se obezbijedilo uspješno poslovanje u organizacijama velikih razmjera potrebno je zadovoljiti ogromne tehnološke zahtjeve. Sa povećanjem resursa unutar organizacija, njihovo praćenje postaje sve složenije.

Pod resursom (engl. *Asset*) se može smatrati sve što je vrijedno za organizaciju. Resurs može biti u obliku fizičkog uređaja, virtuelne mašine, podatka ili pak softverskog sistema koji se koristi u sklopu same organizacije. Dakle, sve što ima vrijednost i podržava poslovanje organizacije može se nazvati resursom.

Ogroman zahtjev je obezbijediti sigurnost organizacije. Zbog velike konkurencije na tržištu, postoje stalne bezbijedonosne prijetnje koje ugrožavaju uspješno funkcionisanje organizacije. Da bi organizacija uspjela opstati uprkos konstantnim prijetnjama, potrebno je osigurati sve resurse da bi sama organizacija bila bezbijedna.

Bezbijednost pruža mnoge prednosti kao što su povjerenje kupaca, ugled i značajna tržišna nadmoć. Da bi se ispunili navedeni zahtjevi, od velike je važnosti koristiti centralizovan sistem za kontrolu, čiji zadatak jeste da koristi obezbjeđuje i upravlja svim računarskim resursima određene organizacije.

NAPOMENA:

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

U radu je izložena implementacija AMCSS (Asset Management Cyber Security System) [1] sistema koji je izgrađen sa ciljem da pruži platformu za upravljanje i osiguravanje računarskih resursa u organizaciji.

Cilj rada jeste da ukaže na značaj obezbjeđivanja i vođenja evidencije o računarskim resursima koje posjeduje organizacija. Korišćenjem ovakvog softverskog rješenja, u ranoj fazi se mogu detektovati neočekivana ponašanja resursa u organizaciji. Dodatno, moguće je detektovati i prijetnje koje bi mogle narušiti bezbijednost organizacije ili na neki drugi način ugroziti njeno poslovanje.

2. UPRAVLJANJE BEZBIJEDNOŠĆU RAČUNARSKIH RESURSA

Računarski resursi obuhvataju različite tehnološke resurse kako hardverske, softverske, virtuelne tako i informacione.

S tim u vezi, sistem za upravljanje računarskim resursima obuhvata četiri ključne cjeline [2]:

- Sistem za upravljanje hardverskim resursima,
- Sistem za upravljanje softverskim resursima,
- Sistem za upravljanje virtuelnim resursima i
- Sistem za upravljanje informacionim resursima.

Sistem za upravljanje hardverskim resursima (engl. *Hardware Asset Management*) ima za cilj da upravlja i obezbjeđuje sve fizičke uređaje koje organizacija posjeduje. Hardverski resursi obuhvataju servere, radne stanice, prenosne računare, mrežne uređaje, mobilne uređaje i slično [3]. Sistem za upravljanje softverskim resursima (engl. *Software Asset Management*) ima zadatak da upravlja i osigurava sve softverske resurse.

Softverskim resursima smatraju se aplikacije, softverska rješenja, biblioteke i slično [4]. Sistem za upravljanje virtuelnim resursima (engl. *Virtual Asset Management*) upravlja i obezbjeđuje sve virtuelne mašine koje organizacija posjeduje [5].

Sistem za upravljanje informacionim resursima (engl. *Information Asset Management*) ima za cilj da upravlja i osigurava sve podatke koje su od značaja organizaciji. Informacioni resursi obuhvataju različite ugovore, standarde, reklame, datoteke i slično [6].

Da bi se bezbijedno upravljalo svim računarskim resursima u organizaciji, potrebno je pratiti sve faze kroz koje resurs prolazi u toku svog životnog ciklusa. Dodatno, bitno je shvatiti koji je cilj svake faze što će biti detaljno objašnjeno u narednom odjeljku [7].

2.1. Životni ciklus resursa

Ključni korak u upravljanju resursima je razumijevanje njihovog životnog ciklusa [8]. Ukoliko se resursom upravlja pravilno, to jeste u skladu sa fazom u kojoj se nalazi, mogućnosti resursa će biti iskorišćene maksimalno. U skladu sa time, ukoliko organizacija iskoristi sve svoje resurse na najbolji mogući način, neosporna je činjenica da će profitirati u svakom segmentu svog poslovanja. Resurs prolazi kroz četiri ključne faze u toku svog životnog ciklusa, i to [9]:

1. Nabavka,
2. Registracija,
3. Aktivni vijek i
4. Odlaganje.

Nabavka predstavlja prvu fazu životnog ciklusa određenog resursa i ona obuhvata planiranje nabavke i samu nabavku resursa [9]. Planiranje nabavke se zasniva na utvrđivanju potreba za novom opremom u organizaciji. Utvrđivanje potrebe za resursima zasniva se na procjeni efikasnosti postojećih sredstava i utvrđivanje njihovih performansi i mogućnosti da ispune dodijeljene operativne zadatke. Drugi korak predstavlja proces nabavke novih resursa. Potrebno je donijeti nabolju odluku za kupovinu novog resursa. Odluka mora biti takva da napravi balans između traženih zahtjeva i dozvoljenih troškova.

Glavni cilj u ovom koraku jeste nabaviti resurs čije ulaganje će se isplatiti organizaciji u određenom vremenskom roku [9].

Registracija predstavlja drugu fazu životnog ciklusa resursa. Nakon što je resurs nabavljen, potrebno je registrovati ga i otpočeti sa njegovim korišćenjem. Da bi se resurs uspješno registrovao, resurs je potrebno prethodno: konfigurirati, instalirati u skladu sa definisanim specifikacijama, povezati na mrežu organizacije na siguran i pouzdan način, definisati fizičku lokaciju resursa, dodijeliti vlasniku koji će biti odgovoran za njega i obučiti radnike (ukoliko se radi o složenom resursu i postoji potreba za obučavanjem) [10].

Nakon što je resurs u eksploataciji, započinje njegov aktivni životni vijek i tada ispunjava sve dodijeljene operativne zadatke. Resurs se nalazi u fazi aktivnog vijeka sve dok se ne nađe u takvom stanju da je neophodno njegovo uklanjanje iz organizacije jer je postao neefikasan [10]. Tokom ove faze, fokus treba biti na odgovarajućem održavanju, praćenju, upravljanju i potencijalnom poboljšanju resursa. Cilj je smanjivanje poteškoća u prilagođavanju na novonastale operativne zahtjeve [10].

Kada završi aktivni vijek resursa, on se smatra viškom, neefikasnim ili neadekvatnim na bilo koji drugi način. Odlaganje treba posmatrati kao prvi korak u povećavanju vrijednosti i profita organizacije.

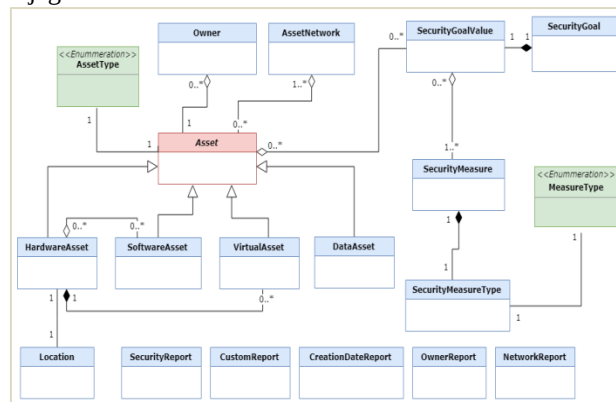
Nakon završenog odlaganja određenog resursa, započinje životni ciklus novog resursa koji će ga zamijeniti. Kao što je ranije navedeno, životni ciklus započinje nabavkom, što znači da nabavka započinje odmah nakon završenog odlaganja [9].

Kada je riječ o hardverskim resursima, prije njihovog odlaganja neophodno je nepovratno ukloniti sve podatke koji se nalaze na njima. Dodatno, recikliranje tehnoloških uređaja, odnosno hardverskih resursa, je takođe stavka o kojoj treba voditi računa jer predstavlja odličnu osnovu za održivo upravljanje računarskim resursima, a takođe utiče i na ugled same organizacije.

U skladu sa time, potrebno je voditi računa o tome da li postoje određene zakonske regulative koje je potrebno ispoštovati prilikom odlaganja.

3. MODEL SISTEMA

U ovom odeljku biće opisan model sistema za upravljanje bezbjednošću računarskih resursa u organizaciji. Za efikasno upravljanje bezbednosnim karakteristikama, potrebno je pratiti sve računarske resurse koje organizacije posjeduju, od fizičkih uređaja, preko softverskih aplikacija, do virtuelnih mašina i ostalih. Takođe, svaki resurs neophodno je osigurati i pratiti tokom njegovog životnog ciklusa. Slika 1 prikazuje dijagram klasa sistema.



Slika 1 - Dijagram klasa sistema

Model sistema podijeljen je u tri cjeline (modula), i to:

1. Modul resursa,
2. Bezbijedonosni modul i
3. Modul za izvještavanje.

3.1. Modul resursa

Osnovna komponenta sistema je resurs na kome se zasniva kompletna struktura sistema. U modelu resursa postoji samo jedan učesnik, a to je administrator organizacije. On ima pravo da registruje novi ili ukloni postojeći resurs. Prilikom registracije novog resursa (klasa *Asset*), administrator ima obavezu da resursu dodijeli vlasnika (klasa *Owner*). Resursi mogu biti grupisani. Grupa resursa, čiji jedini uslov jeste da sadrži minimalno dva resursa, se naziva mrežom i predstavljena je klasom *AssetNetwork*. Mreža može sadržati resurse istog tipa, svrhe, funkcije i slično.

Da bi se zadovoljila činjenica da resurs može biti sve što organizacija posjeduje, od velike je važnosti definisati različite kategorije kojima resurs može pripadati. U skladu sa time, resurs može biti klasifikovan kao:

1. Hardverski resurs,
2. Softverski resurs,
3. Virtuelni resurs ili
4. Informacioni resurs.

3.2. Bezbijedonosni modul

Resurs se smatra ranjivim ukoliko ima zadat barem jedan sigurnosni cilj. U bezbijedonosnom modelu postoji jedan učesnik, a to je vlasnik resursa. Pošto je vlasnik odgovoran za dodijeljene resurse, dužan je da prati bezbijednost resursa. Da bi resurs bio siguran, vlasnik ima obavezu da postavi sigurnosne ciljeve resursu. Ovaj korak takođe

podrazumijeva definisanje sigurnosnih mehanizama. Bez-bijedonosni model se može podijeliti u dvije grupe:

1. Sigurnosni ciljevi i
2. Sigurnosna kontrola.

Sigurnosni ciljevi (klasa *SecurityGoal*) predstavljaju ciljeve koji se moraju ispuniti tokom životnog ciklusa određenog resursa da bi resurs bio siguran i zaštićen. Sigurnosna kontrola (klasa *SecurityMeasure*) obuhvata bezbijedonosne mjere koje se vezuju za sigurnosne ciljeve. Da bi organizacija bila sigurna, obje grupe moraju biti na pravilan način implementirane. Dodatno, grupe su međuzavisne. Prvo se definišu ciljevi koje je potrebno ispuniti nad resursom, a nakon toga i mjere koje će se koristiti da bi se ispunili zadati ciljevi.

3.3. Modul za izvještavanje

Sa povećanim brojem resursa unutar organizacije, izazov je voditi evidenciju o svakom od njih. Zadatak postaje još teži ukoliko je potrebno pratiti resurse koji pripadaju određenim mrežama ili resurse određenih vlasnika. Da bi se ovaj problem riješio, sistem nudi mogućnost generisanja izvještaja. Generisanje izvještaja je funkcionalnost putem koje sistem vraća resurse, koji zadovoljavaju definisane kriterijume, na brz i efikasan način.

Krajnji korisnik sistema ima na raspolaganju pet šablona za generisanje izvještaja, i to:

1. Sigurnosni šablon (klasa *SecurityReport*) – koristi se kada se žele pratiti ranjivi resursi,
2. Šablon opšte namjene (klasa *CustomReport*)
3. Šablon vremena kreiranja (klasa *CreationDateReport*) – koristi se kada se žele pratiti resursi registrovani u određenom vremenskom intervalu,
4. Šablon vlasništva (klasa *OwnerReport*) – koristi se kada se žele pratiti resursi koji su dodijeljeni određenim vlasnicima i
5. Mrežni šablon (klasa *NetworkReport*) – koristi se kada se žele pratiti resursi koji pripadaju određenim mrežama resursa.

4. IMPLEMENTACIJA SISTEMA

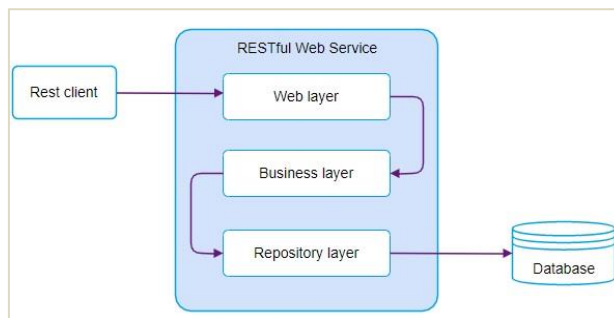
Prilikom implementacije sistema, cilj je napraviti jednostavnu i intuitivnu aplikaciju koju će moći koristiti svi zaposleni u organizaciji, od administratora, preko menadžera, do operatera i ostalih zaposlenih.

4.1. Arhitektura sistema

Sistem je implementiran kao standardna veb aplikacija. Sistem se sastoji iz tri komponente (Slika 2):

1. *Klijentska aplikacija,*
2. *Serverska aplikacija i*
3. *Sistem za upravljanje bazom podataka.*

Prva komponenta sistema jeste klijentska aplikacija. Predstavlja JavaScript [11] baziranu klijentsku aplikaciju i pisana je u Angular 7 [12] radnom okviru. Jedna od najzanimljivijih korišćenih biblioteka koja je zadužena za izgled aplikacije jeste Angular Material [13]. Komunikacija između klijentske aplikacije i serverske aplikacije vrši se korišćenjem HTTP protokola [14] uz slanje JSON [15] objekata. Naredna komponenta jeste serverska aplikacija.



Slika 2 - Arhitektura sistema

Pisana je u Java 8 [16] programskom jeziku, a korišćen je SpringBoot [17] radni okvir. Ova aplikacija je implementirana kao troslojna RESTful (Representational state transfer) [18] aplikacija. Troslojna arhitektura serverske aplikacije obuhvata:

- Mrežni sloj,
- Servisni sloj i
- Repozitorijumski sloj.

Serverska aplikacija osluškuje zahtjeve koji stižu sa klijentske aplikacije. Kada pristigne novi zahtjev mrežni sloj je zadužen za obradu pristiglog zahtjeva. Nakon njegove obrade, mrežni sloj upućuje zahtjev servisnom sloju, koji kasnije upućuje zahtjev sloju repozitorijuma.

Repozitorijumski (engl. Repository) sloj je zadužen za dobavljanje podataka iz baze podataka. Posljednja komponenta sistema jeste sistem za upravljanje bazom podataka. Korišćena je MySQL [19] baza, koja predstavlja relaciju [20] bazu podataka.

List of all hardware assets

1 item(s) selected

☐	Name	Description	Mac Address	IP Address	Serial number
☒	Hp laptop	Small and portable PC	MS-00-25-SS-SS	192.168.100.25	KA010201
☐	Network router	Wireless router	/	/	10SPSWILAN
☐	HP WS 102-S	An old work station	00-40-85-76-SD-E2	100.525.50.15	H01919SAQ0MI
☐	Lenovo Laptop	Personal computer	F0-03-8C-00-8C-0B	192.168.0.15	G5K1CX17K474852

Items per page: 4 1 - 4 of 5

Slika 3 - Komponenta za prikaz resursa

Sistem je izdijeljen na stranice. Za svaku vrstu resursa, navedenu u Modelu resursa, postoji posebna stranica. Svaka stranica se sastoji iz dvije komponente: komponente za prikaz resursa i komponente za modifikovanje resursa. Slika 3 prikazuje komponentu za prikaz resursa. Predstavljena je tabelom koja sadrži prethodno registrovane resurse. Slika 4 prikazuje komponentu za modifikovanje resursa. U obliku je forme koja se popunjava traženim podacima. Pored polja koja opisuju određeni resurs, postoje i akcije koje se mogu izvršiti nad resursom i nalaze se na naslovnoj traci ove komponente. Polja u komponenti će biti nepopunjena ukoliko u komponenti za prikaz resursa nije selektovana nijedna vrsta.

U ovom slučaju, moguće je registrovati novi resurs tako što se prvo popune polja a nakon toga i klikne na dugme + ili Save.

Ukoliko je selektovana jedna vrsta u tabeli registrovanih resursa, polja u formi su automatski popunjena odgovarajućim vrijednostima. Resurs je moguće ažurirati tako što se u odgovarajuća polja unesu izmjene, te se nakon toga klikne na Save ili Update dugme koje se nalazi u

sredini na naslovnoj traci (dugme u obliku olovke). Dodatno, selektovan resurs je moguće izbrisati ukoliko se klikne na Delete dugme, to jeste posljednje dugme u naslovnoj traci (dugme u obliku kante za smeće).

Nakon uspješnog ažuriranja ili brisanja resursa, tabela za prikaz registrovanih resursa je automatski ažurirana.

Slika 4 - Komponenta za modifikovanje resursa

5. ZAKLJUČAK

U radu je opisan sistem za upravljanje bezbjednošću računarskih resursa organizacije sa akcentom na model i implementaciju sistema. Sve tehnologije i modeli koji su korišćeni u realizaciji sistema i razlozi njihovog korišćenja su navedeni u okviru rada.

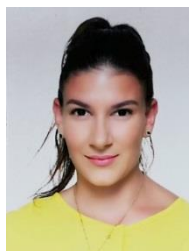
Opisan sistem nalaže da se za svaki resurs koji posjeduje organizacija vezuje zaposleni koji će predstavljati vlasnika. Vlasnik je odgovoran da prati neočekivana ponašanja svojih resursa. U cilju poboljšanja performansi, sistem može preuzeti ovaj zadatak na sebe, te čim uoči ranjivosti kontaktira vlasnika slanjem hitnog upozorenja na službeni broj telefona.

Pored toga, trenutno postoji pet šablona za generisanje izvještaja. U zavisnosti od organizacije u kojoj se koristi sistem, preporučljivo je implementirati još nekoliko šablona koji će zadovoljiti specifične potrebe organizacije.

6. LITERATURA

- [1] 1 NIST Special Publication 1800-5 - IT Asset Management.
- [2] Johannes W. van den Bent, „EXIN ITAMorg IT, Asset Management Foundation WorkBook“, November 2017.
- [3] <https://www.itassetmanagement.net/2015/02/19/importance-effective-ham-hardware-asset-management/>, (pristupljeno u julu 2019.)
- [4] <https://www.techopedia.com/definition/27131/software-asset-management-sam>, (pristupljeno u julu 2019.)
- [5] <http://ogabengineering.com/virtual-asset-management-system/>, (pristupljeno u julu 2019.)
- [6] <https://www.sciencedirect.com/topics/computer-science/information-asset-management>, (pristupljeno u julu 2019.)
- [7] <https://blog.assetsonar.com/it-asset-management-software-functions-benefits-and-implementation-c1d53159a535>, (pristupljeno u julu 2019.)
- [8] <https://road-asset.piarc.org/en/data-and-modeling-lifecycle-planning/lifecycle-planning>, (pristupljeno u julu 2019.)
- [9] <https://pecb.com/article/4-key-stages-of-asset-management-life-cycle->, (pristupljeno u julu 2019.)
- [10] <https://www.lce.com/Life-Cycle-Asset-Management-1112.html>, (pristupljeno u julu 2019.)
- [11] <https://www.javascript.com/>, (pristupljeno u avgustu 2019.)
- [12] <https://angular.io/docs>, (pristupljeno u avgustu 2019.)
- [13] <https://material.angular.io/>, (pristupljeno u avgustu 2019.)
- [14] David Gourley and Brian Tooty “HTTP: The Definitive Guide”, October 2002.
- [15] <https://www.json.org/>, (pristupljeno u avgustu 2019.)
- [16] James Gosling, Bill Joy, Guy Steele, Gilad Bracha, and Alex Buckley “The Java Language Specification”, March 2014.
- [17] https://www.tutorialspoint.com/spring/spring_overview.html, (pristupljeno u avgustu 2019.)
- [18] <https://www.codecademy.com/articles/what-is-rest>, (pristupljeno u avgustu 2019.)
- [19] <https://www.mysql.com/>, (pristupljeno u avgustu 2019.)
- [20] Jan L. Harrington, “Relational Database Design and Implementation”, May 2016.
- [21] http://www.java2s.com/Book/Java/Language-Basics/Boolean_Logical_Operators.htm, (pristupljeno u avgustu 2019.)

Kratka biografija:



Nevena Lazarević rođena je 30. aprila 1996. godine u Zofingenu, u Švajcarskoj. Osnovnu školu „Tamara Begović“ i srednju školu gimnaziju „Vaso Pelagić“ u Brčko Distriktu završava kao nosilac Vukove diplome. Osnovne akademske studije upisuje 2014. godine na Fakultetu tehničkih nauka u okviru Univerziteta u Novom Sadu, smjer Softversko inženjerstvo i informacione tehnologije. 2018. Godine završava osnovne i upisuje master akademske studije na istom fakultetu, na istom smjeru. Položila je sve ispite predviđene planom i programom.kontakt: lazarevic.nevena.43@gmail.com

NAPREDNA KONTROLA PRISTUPA U PAMETNIM MREŽAMA**ADVANCED ACCESS CONTROL IN SMART GRID**Dragan Erić, *Fakultet tehničkih nauka, Novi Sad***Oblast – Elektrotehničko i računarsko inženjerstvo**

Kratak sadržaj – *Proširenje osnovne kontrole pristupa zasnovane na korisničkim ulogama (engl. Role Based Access Control - RBAC) entitetom AOR-a (engl. Area of Responsibility), uz implementaciju servisa za prikupljanje i upravljanje događajima i alarmima u pametnim mrežama.*

Ključne reči: *Kontrola pristupa, RBAC, AOR, Smart Grid*

Abstract – *Extension of Role Based Access Control (RBAC) by AOR (Area of Responsibility) with the implementation of event and alarm collection service in Smart Grid.*

Keywords: *Access Control, RBAC, AOR, Smart Grid*

1. UVOD

Pametne mreže (eng. *Smart Grid*) karakteriše veliki broj korisnika, kritičnih resursa i funkcionalnosti kojima je potrebno upravljati na adekvatan način kako bi bili dostupni svim korisnicima. Tradicionalni elektroenergetski sistemi prolaze kroz duboke promjene koje su rezultovale novim izazovima u procesu upravljanja i nadzora.

Ranije fizički izolovan sistem, sa minimalnim mjerama zaštite ljudskog elementa i uređaja na mreži, danas mora biti prilagođen novim bezbjednosnim prijetnjama. Takve bezbjednosne prijetnje nisu postojale, dok se u elektroenergetski sistem nije počeo uvoditi sve veći broj pristupnih tačaka za razmjenu podataka.

Potrebno je naglasiti važnost izuzetne moći obnovljivih izvora energije kao i njihovog priključivanja na distributivne mreže stvaranjem distribuiranih energetskih resursa (eng. *Distributed Energy Resource* – skr. DER). Na taj način je započeo tranzitivni proces prelaska tradicionalnih pasivnih distributivnih mreža do aktivnih distributivnih sistema, tako da rast entuzijazma za priključivanjem DER-ova poslednjih godina enormno raste.

Sve ovo dovodi do toga da je danas distributivna mreža veoma kompleksna i gotova nemoguća za kontrolu tradicionalnim načinima. Cilj kontrole pristupa jeste ograničenje akcija ili radnji koje legitimni korisnik računarskog sistema može da obavlja.

Korisniku pod kontrolom pristupa zasnovanoj na korisničkim ulogama (eng. *Role Based Access Control*, skr. RBAC) može biti dodijeljena samo jedna uloga u organizaciji.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Čapko, vanr. prof.

AOR (eng. *Area Of Responsibility*) kontrola pristupa, odnosno kontrola pristupa prema oblasti odgovornosti, predstavlja određeni vid nadogradnje RBAC modela. Oblast odgovornosti podrazumijeva skup dopuštenih operacija nad objektima koji imaju zajedničke karakteristike u okviru elektroenergetskog sistema [1].

2. TEORIJSKE OSNOVE**2.1 Tradicionalni elektroenergetski sistemi**

Većina postojećih elektroenergetskih sistema danas ima zajedničke karakteristike. Tri glavne cjeline elektroenergetskih sistema jesu:

- proizvodnja u velikim elektranama,
- prenosna mreža, za prenos energije na visokom naponu i na velika rastojanja do krajnje potrošnje,
- sredjenaponska i niskonaponska distributivna mreža, koja dovodi energiju do krajnjih korisnika.

Konstantan porast zavisnosti od električne energije za obavljanje svakodnevnih aktivnosti i povećanje infrastrukturnih međuzavisnosti prouzrokovali su da tradicionalni sistemi postanu neodrživi. Uvođenjem novih komunikacionih i informacionih tehnologija, pozivajući na decentralizovaniji pristup sistemskim funkcijama nadgledanja i kontrole, elektroenergetski sistem prerasta u inteligentnu mrežu (eng. *Smart Grid*).

2.2 Smart Grid

Proces modernizacije elektroenergetskih sistema podrazumijeva integraciju tradicionalnih elektroenergetskih sistema sa brojnim naprednim sistemima, kao što su nadzorno-upravljački računarski sistemi za automatizaciju procesa upravljanja električnom energijom i optimizaciju potrošnje u zavisnosti od uslova snabdijevanja, zatim napredni mjerni sistemi za upravljanje potrošnjom i brojlilima električne energije, automatizovani sistemi za naplatu i drugi sistemi za upravljanje poslovnim procesima elektroenergetske kompanije.

Najveće dobiti od implementacije pametnih mreža jesu sledeći rezultati koji se mogu kvantifikovati [2]:

- Povećanje pouzdanosti, radnih performansi i cjelokupne produktivnosti.
- Efikasniji način dostavljanja električne energije do potrošača smanjuje potreban broj proizvodnih sistema i broj vodova koje treba izgraditi.
- Pouzdanija isporuka električne energije povećava energetska efikasnost, a dovodi do smanjenja emisije ugljen-dioksida.
- Optimizacija integracije obnovljivih izvora energije.

2.3 Informaciona bezbjednost

Cilj informacione bezbjednosti jeste da se eliminišu ili smanje bezbjednosni rizici, kao i smanjenje posljedica eventualnih incidenata primjenom raznih bezbjednosnih mjera za prevenciju i detekciju napada, kao i reakciju na incidente [1]. Neke od bezbjednosnih mjera uključuju fajl permisije i korisničke kontrole pristupa (eng. *Access Controls*).

2.4 Kontrola pristupa

Cilj kontrole pristupa jeste ograničenje akcija ili radnji koje legitimni korisnik računarskog sistema može da obavlja.

2.4.1 Model kontrole pristupa zasnovan na korisničkim ulogama – RBAC model

Model kontrole pristupa zasnovan na korisničkim ulogama (eng. *Role Based Access Control* skr. RBAC), u kome se pristup zasniva na korisnikovoj funkciji unutar organizacije. Uloga (eng. *role*) predstavlja skup operacija koje korisnik ili više korisnika može izvršavati u kontekstu organizacije. Uloge su orijentisane ka grupama. Instanca korisnikove interakcije sa sistemom se zove sesija (eng. *session*) [3]. Permisije (eng. *permissions* ili *privileges*) su autorizacije za izvođenje odgovarajuće akcije u sistemu [3].

2.5 Postojeće primjene RBAC modela u pametnim mrežama

RBAC je nastao kao alternativa za DAC i MAC modele kako bi se odgovorilo zahtjevima različitih organizacija u državnom i u privatnom sektoru [4]. RBAC je jedan od najzastupljenijih modela za kontrolu pristupa u modernim informacionim sistemima, s obzirom na jednostavnost upravljanja bezbjednosnim politikama, kao i smanjenja troškova i kompleksnosti administracije.

Stoga, autori rada [5] predlažu RBAC model koji se zasniva na dodeli prava pristupa korisnicima na osnovu njihovih uloga i odgovornosti u sistemu, kao i centralizovanoj administraciji bezbjednosnih politika unutar organizacije [1].

Autori u radu [6] navode da postojeće instalacije u digitalnim mrežama često koriste koncept da obavljaju lokalni oblik RBAC-a u zavisnosti od okruženja. Komunikacija između entiteta u kontrolnom centru se, na primjer, vrši na osnovu lokalno ili centralno povezanih korisnika na grupe dozvola. Ovo osigurava da lokalno izvršavanje naredbi može biti izvršeno samo ako su odobrene odgovarajuće dozvole, ali ne mora nužno osigurati udaljenom entitetu da provjeri ko će obaviti namijenjenu operaciju. Pristup opisan u IEC 62351-8 podržava i lokalni trag revizije kroz mogućnost povezivanja informacija o identitetu i pristupu u pristupnom tokenu (eng. *access token*). U trafostanicama, lokalni fizički pristup već može biti dovoljan za pristup entitetima koji komuniciraju. Iako pristup koji koristi tokene za pristup zasnovane na X.509 ima svoje prednosti, on nije odmah primjenljiv u svim slučajevima. Takođe, treba imati na umu da je infrastruktura elektroenergetske mreže tokom godina rasla i da je životni vijek instaliranih uređaja dugačak, 20-25 godina.

Nedavno proučavani različiti sigurnosni modeli u vezi sa kontrolom pristupa koristeći svijest o kontekstu, ali različite usluge koje se nude u pametnim mrežama i

kontroli pristupa u takvom okruženju i dalje imaju ozbiljne ranjivosti [6].

Kao što je navedeno u [6] veb-bazirane usluge zasnovane na XMPP su specificirane za integraciju distribuiranih (decentralizovanih) energetske resursa (DER) u digitalnu energetske mrežu. Ove usluge mogu iskoristiti već postojeće tehnologije koje podržavaju RBAC, kao što je OpenID Connect ili OAuth 2.0, umjesto da grade paralelnu infrastrukturu za rukovanje RBAC baziranim na X.509 [6].

Autori rada [7] navode da podjela odgovornosti između korisnika kojima je dodijeljena ista uloga smanjuje vjerovatnoću (konfiguracionih) grešaka u sistemu. RBAC96 je prilično generički model kontrole pristupa i ne zadovoljava u potpunosti sve sigurnosne zahtjeve kritičnih infrastrukturnih sistema, poput separacije dužnosti i odgovornosti korisnika u skladu sa regionalnim podjelama kritičnih sredstava. U tu svrhu je uveden pojam područja odgovornosti (AOR) još jedan nivo kontrole pristupa u Smart Grid okruženju.

3. ARHITEKTURA SMART GRID SISTEMA

Arhitekturu *Smart Grid* sistema karakteriše kompleksna arhitektura koja je nastala uvođenjem distribuiranih energetske resursa kao što su razni tipovi baterija za skladištenje električne energije, distribuirani generatori. Aplikacija predstavlja softversku platformu koja služi za nadzor, upravljanje i regulaciju distribuiranih izvora energije odnosno *Distributed Energy Resource Management System*, skraćeno DERMS. Termin DERMS se najčešće odnosi na softver koji integriše potrebe dispečera sa mogućnostima energetske resursa sa fleksibilnom potražnjom na kraju mreže.

Korisnik aplikacije je u mogućnosti da unese zathjevani iznos aktivne ili reaktivne snage koji želi da uzme ili da preda u mrežu. Komandovanje tj. postavljanje *setpointa* se oslanja na podatke iz dobavljene vremenske prognoze. **Network Model Service - NMS** Predstavlja komponentu koja drugim servisima pruža statičke podatke o mreži.

Supervisory Control And Data Acquisition - SCADA

Predstavlja sistem za nadzor i upravljanje fizičkim procesima u elektroenergetskim sistemima.

Calculation Engine – CE Centralna uloga mu je da obrađuje zahtjeve za komandovanjem, tj. zahtjeve za povećanje ili smanjenje proizvodnje aktivne i/ili reaktivne snage distribuiranih izvora električne energije.

Weather Forecast Service Daje uvid u vremensku prognozu CE servisu, gdje se analiziraju podaci do 7 dana unaprijed, na nivou jednog časa.

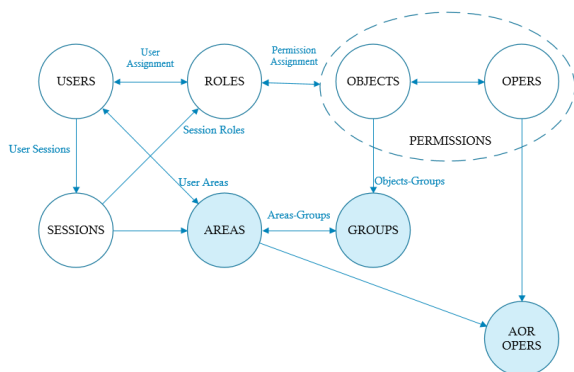
Remote Telemetry Unit – RTU Koriste se prikupljanje izmjerenih analognih i digitalnih ulaza, kao i komandno komunikacioni kontroler za uređaje u polju.

DERMS korisnički interfejs Omogućava nadzor, upravljanje i regulaciju rada distribuiranih izvora električne energije.

4. IMPLEMENTACIJA NAPREDNE KONTROLE PRISTUPA

Bezbjednosni mehanizam pristupa prema oblasti kontrole (eng. *Area Of Responsibility*, skr. AOR) predstavlja koncept podjele odgovornosti prema oblastima, najčešće geografskim, u elektroprenosnoj mreži. Pored AOR-a

implementiran je i prototip sistema za obradu alarma i događaja.



Slika 1. Model proširenog RBAC modela

Model kontrole pristupa demonstriran u ovom radu definiše proširenje RBAC modela, koje predstavlja segmente označene plavom bojom na slici 1. Model karakterišu tri tipična tipa aktivnosti u Smart Gridu (AOR OPERS na slici 1): **nadzor** (akcije praćenja stanja elektroenergetske mreže, statusa analognih ulaza, informacija o alarmima), **kontrola** (korektivne akcije manipulisanjem rada pojedinačnih DER-ova, upravljanje alarmnim događajima), **ažuriranje** (modifikacija statičkih podataka o elementima mreže).

Korisniku (eng. *user*) je moguće pridružiti različite uloge (eng. *role*), od kojih svaka uloga posjeduje određene permisije (eng. *permission*). AOR oblast (eng. *area*) je sačinjena od više AOR grupa (eng. *group*) i poput korisnika posjeduje određene permisije. Operacije koje su sadržane u pripadajućem skupu AOR OPERS se mogu dozvoliti ili ne, u zavisnosti od permisija određene AOR oblasti, koje se poredе sa permisijama koje posjeduje ulogovani korisnik aplikacije. Pripadnost objekta (jednog DER-a) određenoj AOR grupi se opisuje atributom objekta, odnosno relacijom „Objects-Groups“. Podaci o mapiranju AOR grupa na distribuirane energetske resurse su sadržani u Network Model servisu.

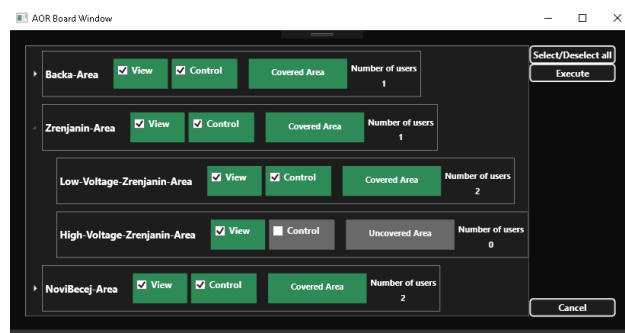
Cache komponenta je locirana unutar AOR servisa i sadrži podatke o AOR grupama, oblastima, permisijama, ulogama, korisnicima i DER-ovima. Grupa predstavlja logički skup resursa koji su blisko povezani. Skup grupa se organizuje u AOR oblasti. Jedna grupa može biti dio više oblasti i svaka oblast može biti član više grupa. Svakoј oblasti se dodjeljuje skup permisija, koje definišu prava pristupa korisnika. Jednom korisniku može biti povjereno više AOR oblasti, dok svaka AOR oblast može biti dodijeljena na više korisnika. Jednoj AOR oblasti može biti dodijeljen i proizvoljan skup drugih AOR oblasti pa na taj način nastaje hijerarhija AOR oblasti. Uloga služi da grupiše određeni broj permisija i dodjeljuje se korisnicima.

Nakon logovanja korisnika se on se *publish-subscribe* mehanizma pretplaćuje na spisak AOR oblasti koje su mu dodijeljene, kako bi dobijao obavještenja kada se dese neke stvari od značaja i to u nekoj od oblasti koje su mu dodijeljene.

AOR Servis Zadužen je za autentifikaciju korisnika sistema. Služi i kao posrednik drugim servisima da dobave podatke iz AOR Cache-a.

Od dodijeljenih oblasti korisnik može da izabere koje od dodijeljenih oblasti želi da nadgleda (eng. *view*) i/ili

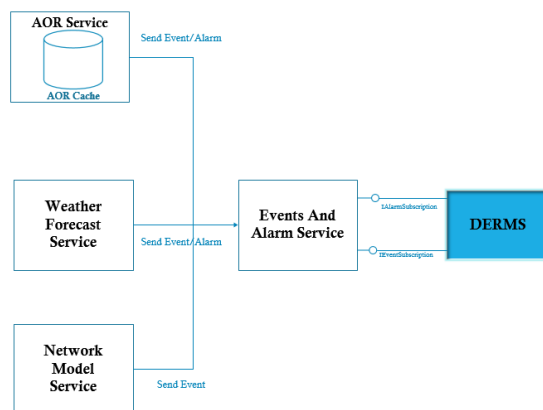
kontrolise (eng. *control*), kao što je prikazano na slici 2. Korisniku će biti prikazani samo alarmi i događaji koji se tiču njemu dodijeljenih AOR oblasti (selektovanih za nadzor ili upravljanje), dok će ostale AOR oblasti biti zanemarene.



Slika 2. Hijerarhijski prikaz AOR oblasti

4.3 Servis za obradu događaja i alarma

Predstavlja centralizovanu komponentu koja u sistemu agregira događaje od značaja. Funkcionisanje zasniva na mehanizmu preplate i objave (eng. *publisher-subscriber*). Arhitektura sistema sa dodatim alarmima i događajima je prikazana na slici 3. Nakon što se korisnik pretplati na njemu dodijeljene AOR oblasti, servis čuva njegov kanal za kasnija obavještenja.



Slika 3. Arhitektura sistema sa dodatim alarmima i događajima

4.3.1 Obrada događaja - Event-a

Događaji mogu biti različitog tipa, obavještenje kada se uloguje novi korisnik koji ima dodijeljenu istu oblast odgovornosti, zatim izdavanja komande za manipulaciju proizvodnje DER-ova ili kada se desi model promocija (primjena novog CIM/XML fajl sa statičkim podacima mreže).

4.3.2 Obrada alarmantnih događaja - Alarma

Alarmantna notifikacija će biti poslata korisniku ukoliko nijedan korisnik (operator) u tom trenutku ne kontrolise navedenu oblast, a trenutno mu je dodijeljena.

U slučaju alarma da je jedna od oblasti ostala nepokrivena, odnosno bez nadzora, korisnik sa ulogom „Administrator“ može ručno izabrati iz liste aktivnih operatera jednog od njih i dodijeliti mu nepokrivenu oblast.

5. TESTIRANJE NAPREDNE KONTROLE PRISTUPA U OKVIRU SMART GRID APLIKACIJE

Model prezentovan u ovom radu je verifikovan simuliranjem na testnom sistemu, tj. simulirano je uprošćeno Smart Grid okruženje koje sadrži obnovljive izvore električne energije. Zbog praktičnih razloga, funkcionisanje dinamičnog i kompleksnog kritičnog infrastrukturnog sistema, kao što Smart Grid svakako jeste, korišćena je simulacija segmenta njegovog rada. Korisnik, u zavisnosti od privilegija koje posjeduje, može komandovati povećanjem ili smanjenjem aktivne ili reaktivne snage određenog segmenta mreže. U simuliranom okruženju je integrisan servis koji vrši kontrolu pristupa, koja se bazira na predstavljenom modelu.

Na osnovu dostupne stručne literature moglo bi se zaključiti da postojeći modeli ne mogu na adekvatan način odgovoriti na aktuelne zahtjeve koji se postavljaju u elektroenergetskoj industriji. Prilikom donošenja odluke o pristupu nekim resursima ABAC model daje mogućnost uvažavanja velikog broja atributa koje treba uzeti u razmatranje. Ipak, ABAC model nije prihvatljiv uslijed definisanja velikog broja autorizacionih pravila, jer Smart Grid obuhvata na hiljade korisnika i opreme. Osim toga, u toku izvršavanja, kada se vrši izračunavanje vrijednosti autorizacionih pravila može doći do ozbiljne degradacije performansi sistema. Takva degradacija performansi u kritičnom sistemu Smart Grida nije prihvatljiva.

Tabela 1 prikazuje skup korisnika sistema sa pripadajućim ulogama i AOR-ima (skraćenice **O**-oblast, **N**-nadzor, **K**-kontrola, **A**-ažuriranje). U nastavku će biti prikazane neke prednosti proširenog modela, u odnosu na RBAC model. Model demonstriran u ovom radu će biti označen kao RBAC-U.

Tabela 1. Skup korisnika sa ulogama i dodijeljenim AOR-ima

Korisnik	Sesija	Korisnička uloga	O1 – N	O1 – K	O1 – A
Operator1	S1	Operator	✓	✓	x
Operator2	S2	Operator	✓	x	x

Kada se uspostavlja korisnička sesija operateri sistema aktiviraju dodijeljene AOR-e. Stanja sesija su demonstrirana tabelom 2 (skraćenice **O**-oblast, **N**-nadzor, **K**-kontrola, **A**-ažuriranje). U tabeli 2 su prikazani rezultati koji uporedno prikazuju RBAC model, kao i model RBAC-U. Interpretacijom rezultata zaključuje se da nepostojanje permisije uzrokuje i nemogućnost izvršenja akcije.

Kada se analiziraju korisnici koji imaju dodijeljen isti tip uloge, može se primjetiti da klasični RBAC ne podržava mogućnost podjele odgovornosti. RBAC-U omogućuje dodatni nivo kontrole pristupa, u odnosu na RBAC.

Tabela 2. Uporedni prikaz modela kontrole pristupa

Korisnik	Sesija	Model kontrole pristupa	N	K	A
Operator1	S1	RBAC	✓	✓	x
		RBAC-U	✓	✓	x
Operator2	S2	RBAC	✓	✓	x
		RBAC-U	✓	x	x

Operateri posjeduju pravo nadgledanja i kontrole samo nad onim dijelom mreže za koji imaju aktivirane AOR-e. (označeno simbolom ✓ u tabeli 2).

6. ZAKLJUČAK

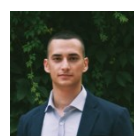
Povećanje broja distribuiranih energetske resursa (skr. DER) stvara decentralizovaniji elektroenergetski sistem i mijenja tradicionalnu dinamiku između lokalnih sistema distribucije i prenosnog sistema na nivou cijele regije. Informaciona bezbjednost postaje sve veća briga kako u fizičkom, tako i u elektronskom domenu. Sajber napadači se trude da pronađu i zloupotrijebe nedostatke sistema poput Smart Grida, pošto su servisi elektroenergetskog sistema od kritičnog značaja za moderno društvo. S obzirom da RBAC model nije najpogodniji model u kritičnim infrastrukturnim sistemima kao što je Smart Grid, jer npr. ne uvažava parametre koji nisu dio identiteta korisnika, iako oni mogu uticati na dozvolu pristupa određenim resursima.

Prilikom praktične primjene razvijenog modela napredne kontrole pristupa utvrđeno je da se proširenjem RBAC modela upravljanje pametnom mrežom može učiniti efikasnijim i pouzdanijim. Segmentiranjem nadzora, kontrole kao i izmjenama statičkog modela distributivnog sistema dobija se efikasniji način upravljanja kritičnom Smart Grid infrastrukturom. Hijerarhijska organizacija oblasti odgovornosti može da smanji broj oblasti koje je potrebno dodijeliti korisnicima čime se olakšava administracija modela kontrole pristupa.

7. LITERATURA

- [1] Rosić D. (2017). Model kontrole pristupa u Smart Grid sistemima. Novi Sad. RS:Faculty of technical sciences, University of Novi Sad
- [2] Borlase, S. (2017). Smart grids: infrastructure, technology, and solutions. CRC press.
- [3] Ferraiolo, D., Kuhn, D. R., & Chandramouli, R. (2003). Role-based access control. Artech House.
- [4] Ferraiolo, D. F., Gilbert, D. M., Lynch, N. An examination of Federal and Commercial Access Control Policy Needs. 16th National Computer Security Conference. Baltimore, Maryland. 1993.
- [5] David F. Ferraiolo and D. Richard Kuhn. (1992) Role-Based Access Controls Reprinted, National Institute of Standards and Technology. 15th National Computer Security Conference Baltimore, pp. 554-563
- [6] Fries, S., Falk, R., & Bisale, C. (2017, May). Handling Rolebased Access Control in the Digital Grid. In ENERGY 2017: The Seventh International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies.
- [7] D. Rosić, I. Lendak, S. Vukmirovic. (2015). A Role-based Access Control Model Supporting Regional Division in Smart Grid System, Acta Polytechnica Hungarica Vol. 12, No. 7

Kratka biografija:



Dragan Erić rođen je 1994. godine u Zvorniku. Završio je srednju ekonomsku školu JU SŠC u Zvorniku 2013. godine. Osnovne akademske studije završio je 2018. godine na Fakultet tehničkih nauka u Novom Sadu. godine.

PROFIBUS INDUSTRIJSKA PERIFERIJA PROFIBUS INDUSTRIAL PERIPHERAL DEVICE

Stevan Srećkov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je prikazana realizacija elektronskog uređaja čija je glavna namena prikupljanje i obrada podataka sa senzora u industrijskim postrojenjima i njihovo slanje glavnom PLC-u putem PROFIBUS mreže u realnom vremenu. Ove podatke je istovremeno moguće snimati i na micro SD karticu, što je pogodno za kasniju analizu. Uređaj poseduje osam digitalnih ulaza, četiri digitalna izlaza, osam analognih ulaza (analogni signali mogu biti strujni ili naponski), jedan spoljni priključak za serijsku komunikaciju, kao i interfejs za povezivanje uređaja na PROFIBUS mrežu. Napajanje uređaja, kao i svi digitalni ulazi i izlazi su prilagođeni standardnom industrijskom naponu od 24 V. Opsezi ulaznih analognih signala takođe odgovaraju industrijskom standardu (0 – 10 V za naponske i 4 – 20 mA za strujne signale).

Gljučne reči: PROFIBUS, PLC, senzori, industrija.

Abstract – This paper describes the development of an electronic device whose main purpose is to collect and process sensor data in industrial plants. Processed data are sent to the main PLC over PROFIBUS network in real time. These data can also be simultaneously saved on a micro SD card for future analysis. The device has eight digital inputs, four digital outputs, eight analog inputs (analog signals can be current or voltage type), one extern connector for serial communication and an interface for connecting to PROFIBUS network. Device's power supply and all digital inputs and outputs are suited for standard industrial voltage level of 24 V. Analog input signals ranges are also industrial standard (0 – 10 V for voltage and 4 – 20 mA for current signals).

Keywords: PROFIBUS, PLC, sensors, industry.

1. UVOD

S obzirom da se industrijska proizvodnja stalno uvećava, a proizvodni procesi usložnjavaju velikom brzinom, postalo je nemoguće kontrolisati ceo proces proizvodnje samo jednim upravljačkim kontrolerom, sa jednog mesta. Efikasniji način upravljanja u složenim industrijskim postrojenjima je decentralizovano upravljanje, odnosno uvođenje manjih upravljačkih jedinica koje su zadužene za upravljanje samo lokalnim procesima. Ove, u hijerarhijskom smislu manje (odnosno niže) jedinice su povezane sa glavnom upravljačkom jedinicom sa kojom

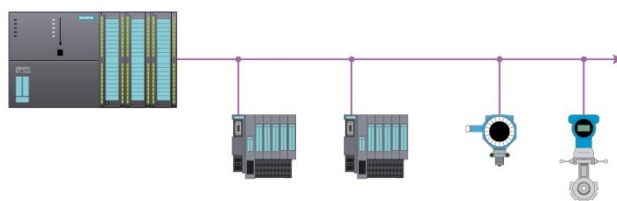
razmenjuju samo osnovne podatke i parametre koji su neophodni za pravilno funkcionisanje celokupnog sistema.

Kako bi se obezbedio kvalitetan i siguran prenos podataka bez grešaka, u često surovim industrijskim uslovima, komunikacija između ovih decentralizovanih jedinica mora biti pouzdana i robusna.

Jedan od standardnih industrijskih (*Fieldbus*) tipova komunikacija koji se u velikoj meri koristi jeste PROFIBUS, koji pored hardverskih performansi, obezbeđuje i internacionalno standardizovan komunikacioni protokol koji obezbeđuje visoku pouzdanost komunikacije koja je neophodna u industrijskim postrojenjima.

Na jednu PROFIBUS magistralu je moguće povezati veći broj uređaja istovremeno, s tim što se najčešće praktikuje da samo jedan od njih bude glavni, odnosno *master*, a ostali *slave*.

Zahvaljujući tome što ima više različitih profila, PROFIBUS može odgovarati različitim specifičnim potrebama korisnika, koji mogu kombinovati različite protokole u skladu sa svojim zahtevima i softverom. Uređaji povezani na ovaj način zadovoljavaju visoke standarde kvaliteta i doprinose visokom kvalitetu mreže. Sledeća slika prikazuje šematski prikaz povezivanja više uređaja na PROFIBUS magistralu.



Slika 1. Šematski prikaz PROFIBUS mreže [1]

U cilju što boljeg i efikasnijeg upravljanja različitim industrijskim procesima, često se javlja potreba za specifičnim perifernim uređajima koji objedinjuju više određenih funkcija i ispunjavaju sve zahteve korisnika, a ne mogu se kao takvi naći na tržištu kao komercijalni uređaji.

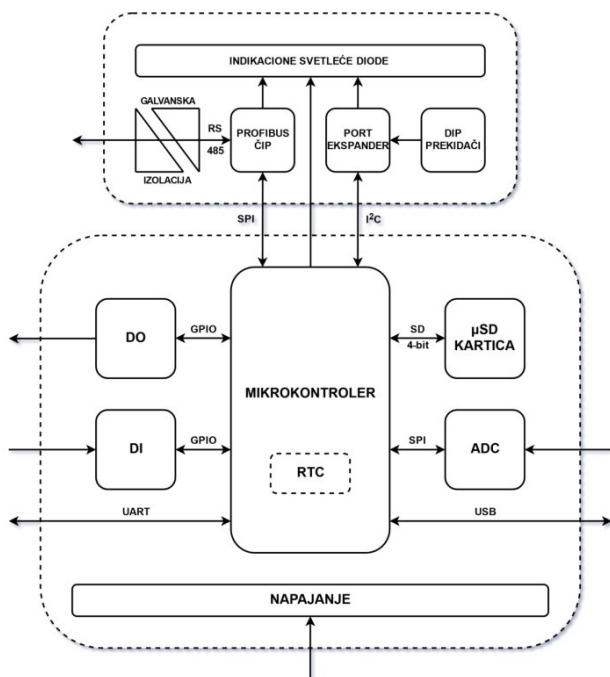
Iz jedne takve potrebe je nastao i ovaj uređaj koji pored mogućnosti kontrole i upravljanja manjim industrijskim procesima, sakupljanja i obrade podataka sa većeg broja različitih senzora, ima i mogućnost jednostavnog povezivanja na već postojeću PROFIBUS magistralu unutar industrijskog postrojenja i razmenu podataka sa glavnom kontrolnom jedinicom, odnosno *master* PLC-om.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Mirjana Damjanović, red.prof.

2. REALIZACIJA HARDVERSKOG DELA UREĐAJA

Kako bi uređaj ispunjavao sve predviđene zahteve i pravilno funkcionisao na željeni način, posebnu pažnju je potrebno obratiti na projektovanje hardvera, kao njegovog osnovnog i najvažnijeg dela. Neophodno je predvideti sve funkcionalne blokove i grupisati ih na pravi način, kako bi se što lakše međusobno povezali, a smetnje maksimalno smanjile. Na slici 2. data je blok šema uređaja.



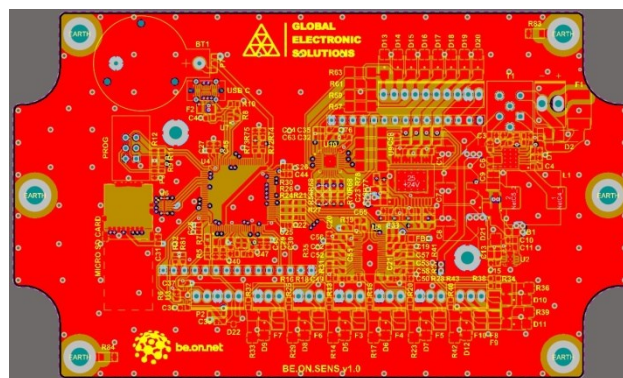
Slika 2. Blok šema uređaja

Iz praktičnih razloga, a i zbog veće modularnosti, elektronske komponente uređaja su razdvojene na dve štampane ploče (PCB-a), koje su međusobno povezane pomoću pin letvica. Na osnovnoj (donjoj) pločici se nalaze glavne komponente sistema: napajanje, mikrokontroler, zatim čipovi za analognu digitalnu konverziju, uključivanje digitalnih izlaza, baferovanje digitalnih ulaznih signala, kao i *micro* SD kartica, baterija za RTC i svi spoljni konektori (osim konektora za PROFIBUS). Druga pločica (gornja), pored svog najvažnijeg dela, čipa koji pruža uređaju mogućnost spajanja i komunikacije preko PROFIBUS magistrale, poseduje još i čip koji služi kao „port ekspander“, tj. daje mogućnost mikrokontroleru da, pomoću I²C veze sa njim, kontroliše ili očitava još dodatnih šesnaest pinova koji mogu biti izlazni ili ulazni. Na gornjoj pločici se takođe nalazi i šesnaest svetlećih dioda, koje služe korisniku za indikaciju različitih stanja u kojima se uređaj može nalaziti, DIP prekidači, pomoću kojih se podešava fizička adresa uređaja na PROFIBUS magistrali, kao i uključivanje, odnosno isključivanje snimanja podataka na SD karticu. Deo svetlećih dioda, kao i DIP prekidači su povezani sa mikrokontrolerom preko port ekspandera. Konektor za PROFIBUS, kao i čip i transformator koji služe za galvansku izolaciju ovog konektora se takođe nalaze na gornjoj pločici.

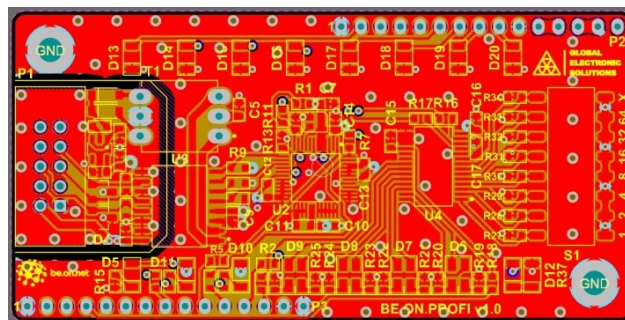
Za projektovanje električne šeme, kao i samih štampanih pločica korišćeno je programsko okruženje „Altium Designer“, verzija 18.1.7.

Prilikom projektovanja, posebno je obraćena pažnja na postavljanje neophodnih kondenzatora za rasprezanje, kao i ESD i prenaponskih zaštita kako bi uređaj bio što robusniji, a njegov rad pouzdaniji. U delu za napajanje se nalazi PTC osigurač kao strujna zaštita, zatim zaštita od obrnutog polariteta dovedenog napona napajanja, prenaponska zaštita u vidu TVS diode, kao i LC filter. Ostavljena je mogućnost spajanja mase uređaja na spolja dovedeno uzemljenje preko džampera u vidu otpornika, ukoliko je to za datu primenu neophodno. Zbog nedostataka prostora nemoguće je prikazati kompletnu električnu šemu uređaja.

Štampane pločice su projektovane prema konkretnim dimenzijama kućišta u koje ih je bilo potrebno postaviti (dimenzije donje pločice su 148x88 mm, a gornje 80x40 mm). Radi što bolje pouzdanosti, izrađene su u četiri sloja, od kojih su gornji i donji slojevi korišćeni za signalne vodove, dok se u srednjim slojevima nalaze masa i napajanja (3,3 V, 5 V i 24 V). Elektronske komponente su SMD tipa i na obe pločice su postavljene samo sa gornje strane. Na slikama 3. i 4. prikazan je *layout* obe pločice.



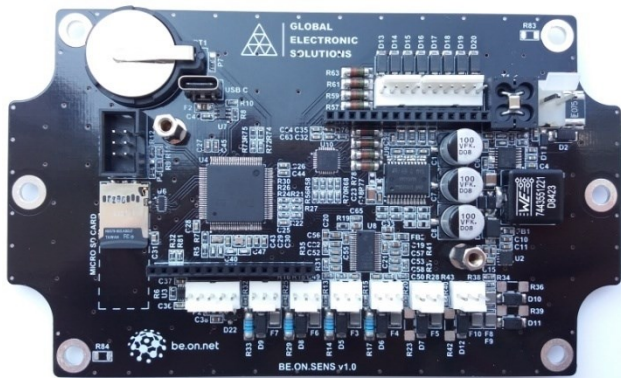
Slika 3. Layout donje pločice



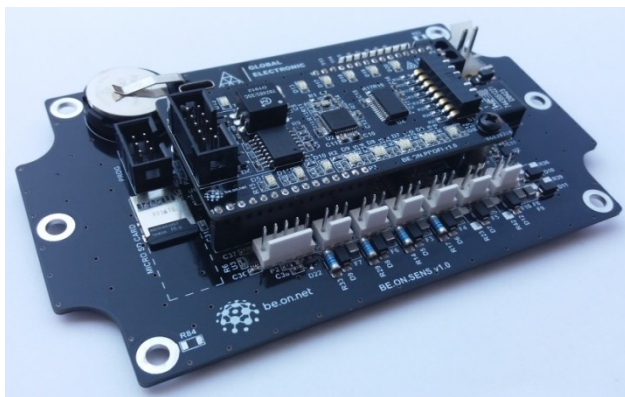
Slika 4. Layout gornje pločice

Nakon izrade pločica, sledi lemljenje svih komponentata i provera svakog funkcionalnog bloka, počevši od napajanja. Uređaj je zatim postavljen u kućište, koje je napravljeno od aluminijuma i kao takvo je vrlo pogodno za obradu, odnosno bušenje otvora za spoljne konektore i svetlovođe koji se nalaze između svetlećih dioda na gornjoj pločici i vrha poklopca. Da bi se izvršila provera uklapanja pločica u kućište, odnosno projekteovala mesta i oblici otvora za konektore i svetlovođe, korišćeni su 3D modeli svih komponentata u programskom okruženju „Autodesk Inventor 2018“. Kućište zadovoljava stepen zaštite IP66 koji je sasvim dovoljan za zaštitu uređaja u industrijskim pogonima.

Senzori se na uređaj priključuju pomoću industrijskih konektora M12 sa pet pinova, dok se za napajanje koristi isto M12 konektor, samo sa tri pina. Za povezivanje na PROFIBUS magistralu, koristi se standardni DB9 konektor. Na slikama 5., 6. i 7. prikazan je uređaj u završnom stadijumu izrade.



Slika 5. Izgled donje pločice



Slika 6. Izgled gornje pločice postavljene na donju



Slika 7. Izgled celog uređaja

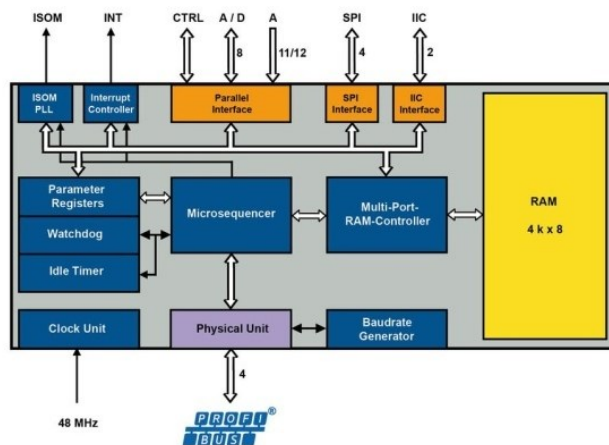
3. REALIZACIJA SOFTVERSKOG DELA

Softver uređaja treba da na optimalan način upravlja radom celog uređaja i omogući korisniku jednostavno korišćenje. S obzirom da je odabran mikrokontroler sa ARM arhitekturom, STM32H743 [2], proizvođača *STMicroelectronics*, najjednostavniji način pisanja programa je u programskom razvojnem okruženju "*STM32CubeIDE*", koje naravno podržava navedeni

mikrokontroler i poseduje sve neophodne biblioteke sa osnovnim funkcijama. U njemu je takođe integrisana i platforma za podešavanje svih važnih parametara koji se odnose na rad mikrokontrolera, kao i za generisanje inicijalnog *C* koda, koji znatno olakšava proces programiranja, s obzirom na kompleksnost ARM mikrokontrolera. Kod je pisan u *C* jeziku, a njegovo spuštanje u mikrokontroler je rađeno programatorom *ST-LINK/V2*.

Početni zadatak pri realizaciji softvera je pravilna inicijalizacija potrebnih integrisanih periferija mikrokontrolera. Najveći izazov je bilo pisanje dela koda koji se odnosi na komunikaciju sa PROFIBUS čipom. Ovaj čip nosi oznaku *VPC3+S*, proizvođača *Profichip*, a služi za direktnu razmenu podataka sa *master* uređajem na PROFIBUS magistrali po PROFIBUS-DP protokolu. Sa druge strane, mikrokontroler sa njim komunicira preko *SPI* interfejsa i na taj način je korisniku omogućeno da na "lak" način razmenjuje podatke preko PROFIBUS magistrale.

Proizvođač ovog čipa je korisniku dao na raspo-laganje osnovni *firmware* koji vodi računa o svim detaljima i parametrima PROFIBUS-DP protokola, a za-datak korisnika je da ga na pravilan način implementira i prilagodi svojim potrebama u skladu sa platformom koju koristi. Slika 8 prikazuje blok šemu ovog čipa.



Slika 8. Blok šema *VPC3+S* čipa [3]

Prilikom instaliranja uređaja u radno okruženje, potrebno je omogućiti korisniku da na jednostavan način podesi osnovne parametre njegovog rada. Ovo je omogućeno implementiranjem odgovarajućih funkcija i povezivanjem uređaja na računar putem dvosmerne *Full-Speed* USB komunikacije, korišćenjem USB tip C konektora na strani uređaja. Neke od osnovnih funkcija koje su implementirane omogućavaju korisniku da putem terminalskog programa na računaru, upravlja radom digitalnih izlaza, očitava digitalne i analogne ulaze, upisuje trenutno vreme i datum u RTC, podešava period upisivanja podataka na SD karticu, itd.

U principu, sam algoritam rada uređaja je jednostavan. Nakon inicijalizacije početnih parametara i promenljivih, glavna petlja se stalno ponavlja dokle god je prisutno napajanje.

Na početku glavne petlje osvežava se *watchdog timer*, zatim se vrši očitavanje ulaznih signala sa senzora i njihova obrada, nakon čega sledi razmena podataka sa PROFIBUS čipom (tj. *master* uređaja na PROFIBUS magistrali), postavljanje digitalnih izlaza i upisivanje podataka na SD karticu.

4. TESTIRANJE UREĐAJA

Da bi se prikazala kompletna funkcionalnost uređaja, potrebno je testirati svaku funkciju pojedinačno, ali i sve zajedno, kako bi se osiguralo da svi delovi pravilno funkcionišu i ne smetaju jedan drugom.

Testiranje digitalnih ulaza i izlaza je vršeno pomoću spolja dovedenih prekidača i svetlećih dioda, dok je testiranje analognih ulaza vršeno uz pomoć senzora temperature Pt100 (strujni signal) i ultrazvučnog senzora razdaljine Sonar-BERO 3RG6176-6GG00 (naponski signal). Na spoljni serijski konektor je doveden senzor EZO-RGB, proizvođača *AtlasScientific*, koji može da meri boju predmeta u kojeg je uperen, razdaljinu od njega i nivo osvetljenja. Na slici 9 je prikazan uređaj u fazi testiranja.



Slika 9. Prikaz uređaja prilikom testiranja

Kako bi se prikazala funkcionalnost i PROFIBUS komunikacije, svi podaci sa ovih senzora su obrađeni, a zatim slati na PLC SIMATIC S7-315-2DP [4], koji je imao ulogu *master* uređaja. Da bi se PLC mogao koristiti za ovo testiranje, prethodno ga je potrebno isprogramirati i podesiti odgovarajuće PROFIBUS parametre, kao što su adresa *slave* uređaja i brzina prenosa podataka. Ovo je urađeno u programskom okruženju "*Simatic Step7*", a pristigli podaci sa uređaja su prikazivani u tabeli za praćenje ulaza i izlaza (VAT tabeli), pomoću koje je moguće slati podatke i ka uređaju i vršiti kontrolu digitalnih izlaza. Sve vreme su podaci sa senzora, kao i trenutno vreme i datum bili snimani na *micro* SD karticu, koja ih je uspešno sačuvala u tabelarnom prikazu.

5. ZAKLJUČAK

Ako se uzme u obzir da je nivo automatizacije u industrijskim postrojenjima u sve većem porastu, sve više mesta se stvara za projektovanje specifičnih industrijskih uređaja, čijom upotrebom bi se proizvodni procesi mogli ubrzati i učiniti efikasnijim, što bi moglo doneti značajnu uštedu vremena i novca proizvođačima.

Rezultati testiranja su zadovoljavajući i u sledećem koraku bi testiranje trebalo nastaviti u pravom industrijskom okruženju. Na osnovu svojih mogućnosti i karakteristika, može se zaključiti da ovaj uređaj ispunjava sve prethodno postavljene zahteve i da je uspešno realizovan, kako hardverski, tako i softverski. Rad na ovom projektu omogućio je autoru bliže upoznavanje sa industrijskim standardima i principima projektovanja modernih industrijskih uređaja.

S obzirom da je gornja pločica zamišljena kao PROFIBUS modul, može se lako zameniti u slučaju potrebe za povezivanjem uređaja na neki drugačiji način. Ovim je obezbeđeno da se osnova uređaja ne mora menjati sa eventualnom promenom zahteva korisnika. Jedno od mogućih budućih poboljšanja bi bila ugradnja displeja sa ekranom osetljivim na dodir umesto indikacionih svetlećih dioda na poklopcu uređaja. Ovo bi korisniku dalo daleko veće mogućnosti u pogledu menjanja parametara rada uređaja u realnom vremenu, a takođe bi mogao da prati i ispis raznih podataka.

6. LITERATURA

- [1] <https://www.prosoft-technology.com/Products/Gateways/AN-X-PB/PROFIBUS-Multi-Slave-or-PROFIBUS-Network-Analyzer-Gateway-non-CE> (pristupljeno u septembru 2019.)
- [2] <https://www.st.com/en/microcontrollers-microprocessors/stm32h743-753.html> (pristupljeno u septembru 2019.)
- [3] <https://www.profichip.com/products/industrial-communication/fieldbus/dp-slave-vpc3-s/> (pristupljeno u septembru 2019.)
- [4] <https://mall.industry.siemens.com/mall/en/WW/-Catalog/Product/6ES7315-2AH14-0AB0> (pristupljeno u septembru 2019.)

Kratka biografija:



Stevan Srečkov rođen je u Novom Sadu 1991. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Primenjena elektronika odbranio je 2017. god.
kontakt: srechkovs@hotmail.com

IZBOR ELEKTRIČNIH MOTORA ZA POGON ELEKTRIČNIH VOZILA SELECTION OF ELECTRIC MOTORS FOR ELECTRIC VEHICLE DRIVE

Miloš Vorkapić, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovm radu je obrađena tema izbora električnih motora koji su najpogodniji za pogon vozila, obzirom da se sve više modela proizvodi i sa električnom varijantom pogona. Deo rada se odnosi na konstrukciju i način rada električnog vozila, razlike i prednosti u odnosu na konvencionalna vozila kao i konstrukciju električnog motora sa parametrima koji ga karakterišu. Na kraju rada predstavljena je simulacija u Matlabu za potrebe prezentacije izgleda signala odziva električnog motora na zahteve gradske vožnje.

Gljučne reči: *Sinhroni motor sa permanentnim magnetom, Električna vozila, Električni motori*

Abstract – This paper deals with the topic of selection of electric motor that is the most suitable for an electric vehicle drive. Structure and methods of operation of the electric vehicles, advantages are presented. Selection of optimal electric motor with parameters that characterize it is given. Matlab/Simulink model is developed and simulations results are presented. Response signals of the electric motor to the requirements of city driving cycle are shown.

Keywords: *PMSM, Electrical vehicles, Electical machines*

1. UVOD

Sa pronalaskom baterije (Allessandro Volta, 1800), stvaranja magnetnog polja iz električne struje (Hans Christian Oersted, 1820) i elektromagneta (William Sturgeon, 1825) postavljena je čvrsta osnova za izgradnju električnog motora. U to vreme još uvek je bilo otvoreno pitanje da li električni motor treba da bude rotacionog tipa ili klipna mašina, tj. da predstavlja jedan klip kao kod parne mašine. Širom sveta, mnogi pronalazači radili su paralelno na ovom zadatku – jer to je tada predstavljalo aktuelan, “moderan” problem. Nove pojave su bile otkrivane svakodnevno, izumi u oblasti elektrotehnike i njihova primena se povećavala iz dana u dan [1].

Prvi električni automobili pojavili su se još pre 180 godina i koristili paralelno sa parnim mašinama, a kasnije i SUS motorima za pokretanje i prevoz u vozila. Zapravo prvi električni automobili su se pojavili neposredno posle konstrukcije prvog električnog motora 1828. godine. Daljim usavršavanjem elektromotora dolazi i do prvih primena u industriji, pa sve do poslednjih godina kada su električna vozila ponovo u zamahu [2].

NAPOMENA:

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

Cilj ovog rada jeste da predstavi izbor optimalnog tipa električnog motora za pogon električnih vozila, modelovanje kompletnog pogona i rezultate simulacije kretanja vozila i dometa baterije primenom standardnog ciklusa vožnje NEDC.

2. POČETAK, RAZVOJ I DOSTIGNUĆA ELEKTRIČNIH VOZILA

Sva pomenuta otkrića i poboljšanja dovela su do toga da su kupci vozila za prevoz i transport početkom XIX veka imali mogućnost izbora između vozila na parni pogon, električni ili benzinski (SUS). Međutim, u tržišnoj utakmici prevagu je odneo SUS motor, odnosno potisnuti su i zaboravljeni parni i elektro motor.

Ipak, pored nekoliko pokušaja tokom XX veka, pravo oživljavanje električnih vozila počelo je u XXI veku. Dva glavna događaja koja su pokrenula povratak električnih automobila u masovniju proizvodnju su:

- Pojava Tojotinog modela Prius (1997. god.), koji je bio prvi masovno proizveden hibridni auto, a u kome su korišćene NiMH baterije.

- Osnivanje kompanije Tesla Motors (2006. god.), koja je počela sa proizvodnjom luksuznih sportskih automobila, ekstremnih preformansi, a koja su mogla da pređu do 320 km, a sada već i do 600 km sa jednim punjenjem [3].

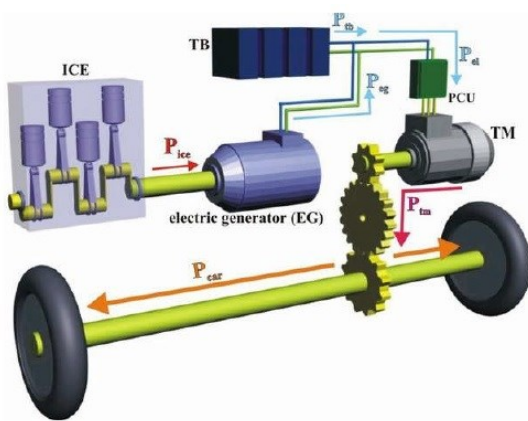
3. TIPOVI I DELOVI ELEKTRIČNIH AUTOMOBILA

Električno vozilo je motorno vozilo, kao što je automobil, kamion ili autobus, koji koristi električnu energiju uskladištenu u punjivim baterijama, umesto benzina, dizela, gasa ili druge vrste fosilnih goriva. Električni motori su mnogo moćniji (u većem delu vremena). Oni postižu izvanredna ubrzanja, to je moguće jer na niskim brzinama, električni motor isporučuju veći obrtni moment nego konvencionalni motor što je i prikazano na slici 2. Sila je ono što je potrebno da bi se poterao auto. Ipak benzinski motor i dalje radi bolje na velikim brzinama nego električni.

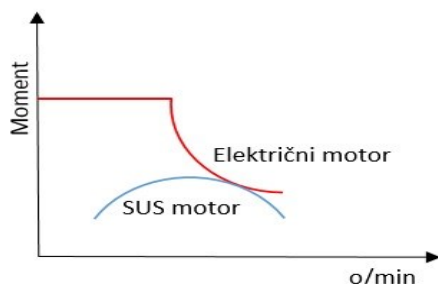
Međutim, kod hibridnih vozila, SUS motor se koristi delimično, dok je kod čisto električnih (baterijskih) SUS motor izbačen, kao i izduvni sistem, a neretko i sistem transmisije.

Kod serijske veze (hibridno vozilo), SUS motor pokreće poseban generator kojim se vučni motor snabdeva električnom energijom i dopunjavaju baterije, slika 1.

Paralelnom kombinacijom motora u hibridnom vozilu je postignuto da se točkovi vozila pokreću i SUS motorom i električnim motor/generatorom.



Sl. 1. Serijska konfiguracija [4]



Sl. 2. Momentne karakteristike električnih i SUS motora

4. ELEKTROMOTOR KAO POGON ELEKTRIČNOG AUTOMOBILA

U cilju dobijanja najprikladnijeg izbora za EV u tabeli 1 navedeni su najvažniji parametri koji figurišu u izboru električnog motora uz uzimanje u obzir sve što je do sada navedeno u analizi. Samo ocenjivanje oslanja se na do sad iznetim prednostima i manama. pa su mane vrednovane sa 1, srednja ocena za datu karakteristiku je 2 a pozitivne strane se vrednuju sa 3 boda.

Kriterijum ocenjivanja je tako izabran iz razloga što svi motori koji se razmatraju koriste se za vuču vozila.

Dobijeni rezultati u tabeli pokazuju da je sinhrona mašina sa premanentnim magnetima (SMPM) najpogodnija najpogodnija za primenu vuče u autoindustriji, sledeći motor je BLDC.

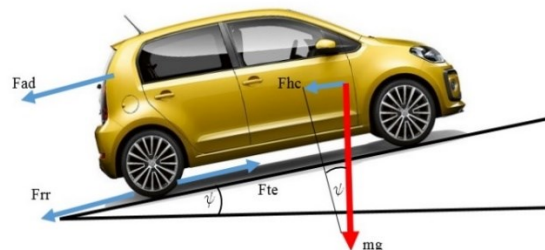
To je očekivano, jer se BLDC motor ne razlikuje u mnogome od SMPM. Razlika je u samoj konstrukciji, inače obe tipa koriste permanentne magnete.

Tabela 1 Ocena električnih motora.

Redni broj	Karakteristike	TIP MOTORA				
		DC sa četkicama	Asihroni	Sinhroni sa PM	Sinhroni reluktantni	DC bez četkica
1	Veliki obrtni moment pri malim brzinama	3	3	3	3	3
2	Nivo kompleksnosti konstrukcije	1	3	3	3	3
3	Efikasnost motora	1	2	3	2	3
4	Održavanje motora	1	3	3	2	3
5	Upravljanje	1	1	2	2	2
6	Gustina energije	2	3	3	3	2
7	Dodatni izvor za stvaranje magnetnog polja	1	1	3	1	3
8	Težina motora	1	2	3	3	3
9	Cena motora	2	2	1	2	1
10	Pouzdanost	1	3	3	2	3
11	UKUPNO	14	23	27	23	26

5. DINAMIČKI MATEMATIČKI MODEL ELEKTRIČNOG MODELA

Prvi i osnovni korak u modelovanju i simulaciji ponašanja električnog vozila je razvijanje jednačine za potrebnu vučnu silu. Ova vrsta sile predstavlja silu koja vuče vozilo napred tj. daje mu mogućnost kretanja i svedena je na tlo preko točkova (slika 3).



Sl. 3. Uticaj sila na kretanje vozila.

Suma svih sila prikazanih na slici 3 predstavlja ukupnu silu vuče električnog vozila i data je izrazom (1):

$$F_{te} = F_{rr} + F_{ad} + F_{hc} + F_{acc} \quad (1)$$

gde su:

- F_{rr} sila otpornosti kotrljanja, data jednačinom

$$F_{rr} = C_r mg \cos \psi \quad (2)$$

- F_{ad} sila otpornosti vazduha, data jednačinom

$$F_{ad} = \frac{1}{2} \rho C_d A v^2 \quad (3)$$

- F_{hc} sila uspinjanja, data jednačinom

$$F_{hc} = mg \sin \psi \quad (4)$$

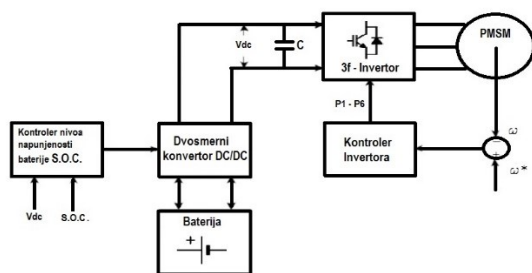
- F_{acc} sila ubrzanja, data jednačinom

$$F_{acc} = ma \quad (5)$$

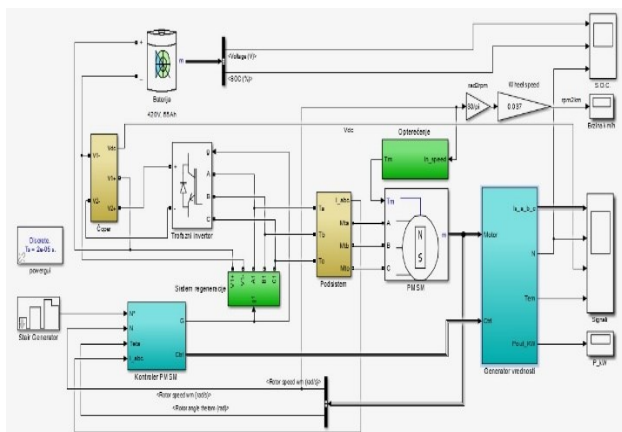
Poslednje dve sile (F_{hc} i F_{acc}) mogu imati negativne vrednosti u slučajevima kada automobil bude usporavao, odnosno kada bude išao nizbrdo (za ovu simulaciju je predviđeno da se vozilo kreće po relativno ravnom gradskom terenu) zbog negativne vrednosti ugla pod kojim je put nagnut) [7].

Dinamički model je kreiran u okviru softvera *Matlab/Simulink*. Dimenzije vozila su kao što je već pomenuto izabrane tako da vozilo najviše liči na vozilo iz „B“ segmenta, ali zbog nepoznavanja parametara ekvivalentne šeme motora i nedostatka informacija o bateriji i ostalim komponentama, izabrani su elementi čije su informacije bili dostupne. Na slici 4 prikazana je uprošćena blok šema najčešće konfiguracije jednog električnog pogona za električna vozila. Neki od osnovnih neizostavnih delova jesu izvor napajanja, DC link veza, Invertor, upravljački sistemi za kontrolu napunjenosti baterije i upravljanje radom invertra i pogonski motor.

Na slici 5 prikazan je razvijeni dinamički model električnog automobila sa odgovarajućim ulazima i izlazima izrađen u *Matlab/Simulinku*. U ulazne parametre sistema ulazi referenca brzine, moment opterećenja koje vozilo mora da savlada i blok napajanja elektro pogona. Izlazi su merenja električnih i mehaničkih veličina koje su od interesa za analizu kao i stanje napunjenosti baterije tokom vožnje. Merenja su prikazana grafički korišćenjem *Scope* blokova.



Sl. 4. Uprošćena blok šema



Sl. 5. Izgled kompletnog modela sa ulazima i izlazima

Parametri koji su uzeti za modelovanje automobila prikazani su tabeli 2, parametri se odnose na uslove na putu i dimenzije vozila, koji su po uzoru na manje vozilo iz „B“ segmenta.

Tabela daje prezentaciju vrednosti koeficijenata i parametara uz pomoć kojih za simulaciju posmatranog sistema.

Za potrebe simulacije, u ovom radu će se koristiti tip motora koji je prema analizi iz poglavlja 4 najpogodniji za vuču vozila. To je PMSM motor sa podacima datim u tabeli 3.

Tabela 2. Karakteristike uslova okoline i vozila

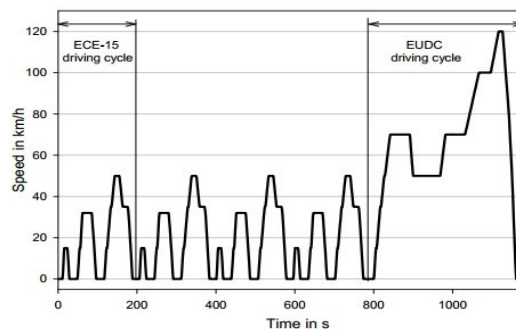
Masa vozila	1050 kg
Gravitaciono ubrzanje	9,81 m/s ²
Koeficijent trenja (suv put)	0,007
Gustina vazduha	1,2041 kg/m ³
Koeficijent aerodinamike	0,3
Površina prednje strane vozila	2,5 m ²
Poluprečnik točka	0,35 m

Tabela 3. Tehnički podaci PMSM [6]

PARAMETAR	VREDNOST
Snaga (kW)	75
Frekvencija (Hz)	50
Max brzina (o/min)	4000
Momenat (Nm)	450
Momenat inercije (kgm ²)	3,011
Masa (kg)	45
Broj pari polova	4

6. NEDC CIKLUS VOŽNJE

Na slici 6 prikazan je izgled NEDC (*New European Driving Cycle*) voznog ciklusa sa naglašenim ciklusima, koji ga sačinjavaju. Trajanje ovakvog ciklusa predloženo je do 1180 s, ali zbog veoma mnogo vremena koliko bi naš softver zahtevao za izvršenje ovakvog tipa simulacije, izvršena je simulacija gradske vožnje sa redukovanim ECE – 15 ciklusom kao referencom brzine. Trajanje od 15 s se pokazalo kao sasvim prihvatljivo vreme za analizu dobijenih rezultata.



Sl. 6. NEDC ciklus vožnje

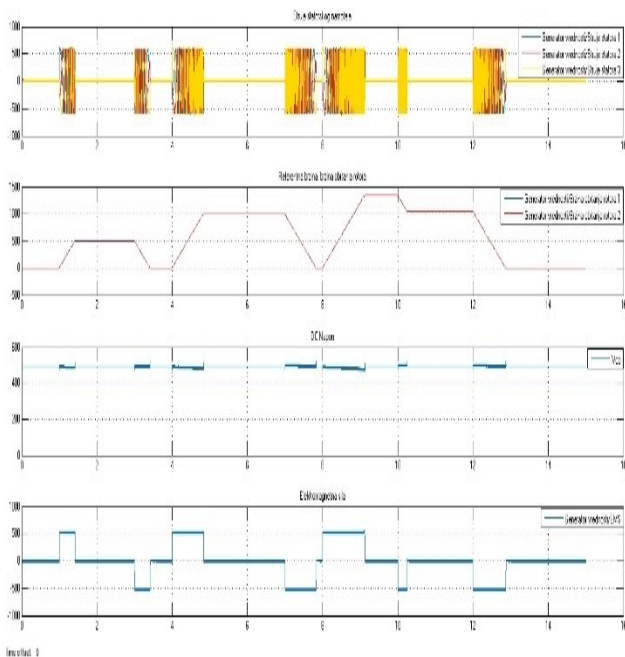
ECE – 15 ciklus ima zadatak da reprezentuje tipične uslove za vožnju u velikim evropskim gradovima u kojima je saobraćaj mnogo gust. Pošto je namenjen vožnji ravnim putem, ne očekuje se mnogo veliko opterećenje. Ograničenje je postavljeno na 50 km/h, koliko i iznosi ograničenje brzine kroz naseljeno mesto. EUDC ciklus predstavlja mnogo agresivniju vožnju, sa brzinama do 120 km/h, ali ovaj ciklus nije uzet u svrhu simulacije usled tehničkih ograničenja u snazi i brzini korišćenog pogonskog motora [7].

7. REZULTATI SIMULACIJA

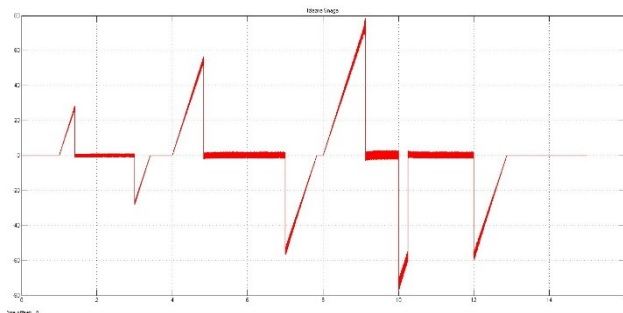
Nakon izvršene simulacije gore predstavljenog dinamičkog modela dobijeni rezultati su prikazani slikama 7-9. Zavisnosti tri statorske fazne struje, brzine, elektromagnetnog momenta i napona DC međukola od vremena dati su na slici 7. Zabeleženi signali daju sliku ponašanja elektropogonskog sklopa električnog automobila prilikom vožnje kroz naseljeno mesto.

Primećuje se brz odziv brzine na referencu što je uzrok relativno malog momenta inercije vozila, koji iznosi 3,011 kgm². Ovaj podatak se unosi na početku inicijalizacije parametara u bloku koji se odnosi na vektorsko upravljanje u okviru mehaničkih veličina motora.

Na slici 8 prikazana je promena snage motora tokom simulacije. Momenat i snaga u direktnoj srazmeri, pa sa porastom razvijenog momenta dolazi do rasta snage koja se troši. U trećoj sekundi, kada dolazi do smanjenja brzine i prelaska u generatorski režim rada, menja se i smer snage. U osmoj sekundi, dolazi do ubrzanja do najveće brzine tokom simulacije što je rezultirano najdužim trajanjem razvijenog momenta, kao i najvećom utrošenom snagom tokom simulacije. Same vrednosti nisu od velikog interesa koliko pregled sistema odnosno zavisnosti promena pojedinih veličina u vremenu i odziva sistema na zadatu referencu.

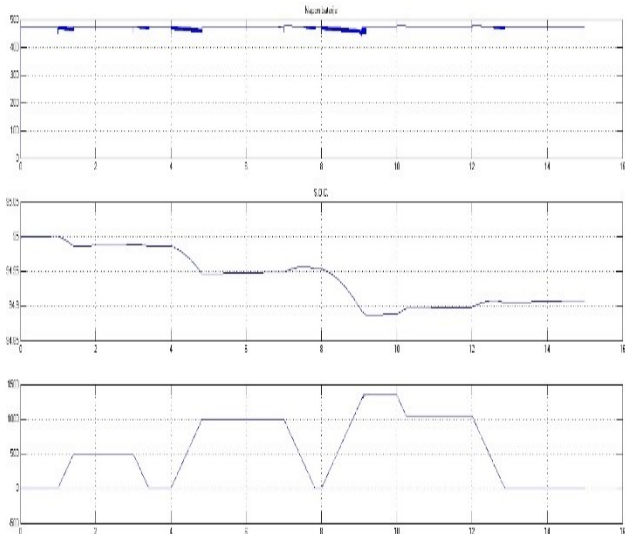


Sl. 7. Odziv sistema



Sl. 8. Aktivna snaga motora na njegovom izlazu

Na slici 9 prikazan je status i način pražnjenja baterije. Sa grafika se može uočiti ciklus pražnjenja i punjenja izvora napajanja prilikom ubrzavanja vozila, odnosno punjenje baterije na račun kočenja vozila (motora) prilikom kočenja. Na samom početku simulacije nivo napunjenosti baterije (SOC) je 95%, a sa kretanjem vozila baterija polako počinje da se troši.



Sl. 9. Ciklus pražnjenja i punjenja izvora napajanja

8. ZAKLJUČAK

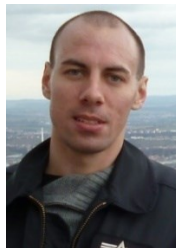
Izneta analiza u ovom radu čini skup osnovne teorije rada, konstrukcije električnim motorima i njihovoj primeni u automobilske industriji, prvenstveno kao pogonske jedinice.

Generalno svaki od datih električnih pogona ima relativno dobre karakteristike tako da je teško napraviti najbolji izbor. Ipak na osnovu svih iznetih činjenica između pomenutih električnih motora i karakteristika koje treba da poseduju kako bi bili upotrebljeni kao pogoni električnih vozila, motori sa premanentnim magnetima (PMSM) imaju blagu prednost nad ostalim pogonima. Trenutno oni su nabolji izbor za električna vozila u pogledu odnosa efikasnost – cena, gustine energije i visoke pouzdanosti.

9. LITERATURA

- [1] <https://www.eti.kit.edu/english/1376.php> (pristupio 25.09.2016.)
- [2] M. Stojkov, D. Gašparović, D. Pelin i dr., "Electric Car – history and components", 5th International Natural Gas, Heat and Water Conference, Hrvatska 2014., (pristupio 19.08.2016.)
- [3] V. Katić i dr., „Mali Električni Automobili - Ispitivanje Osnovnih Pogonskih Karakteristika“, ENEF, Bnja Luka, Nov.2013, dostupno na http://enef.etfbl.net/resources/Zbornik_2013/24_Rad.pdf (pristupio 11.08.2016.) st.:2-3
- [4] Ž. Banković, „Hibridna Vozila“, Seminarski rad, Elektrotehnički fakultet u Nišu, Jun 2010.
- [5] X. D. Xue, K. W. E. Cheng and N. C. Cheung, "Selection of Electric Motor Drives for Electric Vehicles", Hong Kong Polytechnic, University Hong Kong, China, January 2009
- [6] N. Tovilović, "Testiranja pogona električnog automobila – Prva faza", diplomski rad, FTN Novi Sad, 2013 (pristupio 25.10.2016)
- [7] M. Čulibrk, "Modelovanje i simulacija E – pogona modernog električnog automobila", master rad, FTN Novi Sad, 2016.

Kratka biografija:



Miloš Vorkapić rođen je u Vršcu 1988. god. Fakultet tehničkih nauka, studijski program Energetika, elektronika i telekomunikacije upisao je školske 2007/2008. Na studijama se opredelio za smer Elektroenergetika i diplomirao je 2015. god. Master rad iz oblasti Energetske elektronike i električnih mašina. odbranio je 2019. god.

kontakt: milos.vorkapic88@gmail.com



Vladimir Katić rođen je 1954. godine u Novom Sadu. Doktorirao je na Univerzitetu u Beogradu 1991. godine. Od 2002. godine je redovni profesor Univerziteta u Novom Sadu. Trenutno je prodekan Fakulteta tehničkih nauka. Oblasti interesovanja su energetske pretvarači, obnovljivi izvori energije i kvalitet električne energije.

**RAZVOJ GRAFIČKOG NAMENSKOG JEZIKA ZA ARHIVIRANJE I MIGRACIJU
BAZA PODATAKA****A DEVELOPMENT OF A GRAPHICAL DOMAIN SPECIFIC LANGUAGE FOR
DATABASE BACKUP AND MIGRATION**

Marija Kukić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu, prezentovan je grafički namenski jezik za arhiviranje i migraciju baza podataka kreiranih pod različitim sistemima za upravljanje bazama podataka. Dat je detaljni opis razvoja jezika i metodologija koje su korišćene tokom razvoja. Predstavljeni su meta-model sa osnovnim konceptima jezika, ograničenja jezika, konkretna sintaksa i generator koda. Izvršeno je poređenje razvijenog jezika sa postojećim jezicima slične namene kako bi bile uočene prednosti i mane rešenja i potrebe za unapređenjima.

Ključne reči: Domenski jezik, metamodel, sistemi za upravljanje bazama podataka, generator koda.

Abstract – In this paper we present a graphical domain specific language for database backup and migration with a support for different database management systems. We give a detailed description of the language development and methodologies used during the development. Metamodel with main concepts of the language, language constraints, concrete syntax and code generator are presented. A comparison between the proposed language and other languages with a similar purpose is given for the purpose of future improvements.

Keywords: Domain specific language, metamodel, database management systems, code generator.

1. UVOD

Prilikom razvoja softvera neophodno je ispoštovati korisničke zahteve sistema, smanjiti troškove razvoja i održavanja i dostaviti softver u određenom vremenskom okviru. Da bi navedeni zahtevi bili zadovoljeni neophodno je razviti strategiju i upotrebiti određenu metodologiju kako bi se obezbedile osobine koje karakterišu kvalitetan softver. Jedno od rešenja jeste primena koncepata koje propisuje softversko inženjerstvo zasnovano na modelima, eng. *Model Driven Software Engineering* (MDSE) i posebno disciplina Razvoj softvera zasnovan na modelima, eng. *Model Driven Software Development* (MDS), kao njegov deo. MDSE je inženjerski pristup koji se zasniva na definisanju i upotrebi modela unutar softverskog razvoja [1]. MDS, kao deo pristupa MDSE, predstavlja pristup koji definiše modele kao primarne koncepte koji se ne koriste samo u svrhu dokumentovanja, već u svrhu automatizovanog ili automatskog kreiranja krajnjeg softverskog proizvoda, odnosno generisanja programskog koda.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Ivan Luković, red. prof.

Cilj primene navedene paradigme jeste da se automatizuje proces razvoja softvera kroz sve faze životnog ciklusa.

Razvoj namenskih jezika poštuje koncepte MDSD paradigme. Preduslov korišćenja namenskih jezika, eng. *Domain Specific Language* (DSL) jeste poznavanje koncepata jezika i njihove semantike. Korist primene ovih jezika je u tome što se transformacijama automatski iz semantike modela generiše kod. U ovom radu razvijen je grafički namenski jezik za arhiviranje i migraciju baza podataka kreiranih pod sistemom za upravljanje bazama podataka (SUBP) različitih proizvođača.

Prilikom pada baze podataka postoji rizik da se podaci iz baze podataka izgube. Kako bi bilo obezbeđeno da korisnici budu zaštićeni od gubitka podataka, podatke je potrebno arhivirati. S obzirom na to da SUBP-ovi različitih proizvođača koriste iste koncepte koji definišu osobine arhiviranja, a koriste različite procedure za sprovođenje zadataka koji imaju iste efekte, uočava se potreba za razvojem pristupa koji bi omogućio korisniku da poznavajući osobine arhiviranja specificira proceduru arhiviranja na nezavisan način od izabranog sistema za upravljanje bazama podataka. Namenskim jezikom opisanim u ovom radu korisnik specificira osobine koje definišu način i vrstu arhiviranja, a automatski se generišu skripte koje odgovaraju konkretnim sistemima za upravljanje bazama podataka različitih proizvođača. Jedan podsistem ovog grafičkog jezika namenjen je za podršku migracije baze podataka. Migracija podataka je složen proces, sastavljen od niza aktivnosti, kojima se vrše transformacije nad šemom baze podataka tako da se podaci iz starog sistema očuvavaju i prenose u novi sistem, tako da odgovaraju novim zahtevima sistema [2]. Namenski jezik opisan ovim radom podržava određeni skup transformacija kojima se vrši migracija baze podataka, tako što se vrše izmene šeme baze podataka, kreirane pod jednim konkretnim izabranim SUBP-om. Razvijeni jezik omogućava specificiranje konkretnih scenarija migracije sa ciljem da se korisnik oslobodi ručnog kreiranja skripti za izmene nad šemom. Scenarija su odabrana tako da je očuvanje podataka zagarantovano. Primarni cilj ovog rada je kreiranje novog pristupa koji će omogućiti da korisnik bez poznavanja konkretne sintakse specifičnog sistema za upravljanje bazom podataka generiše skripte za arhiviranje prema željenim parametrima. Pored toga, korisniku je potrebno omogućiti da bez poznavanja konkretne sintakse pod kojom je šema baze podataka kreirana izvrši migraciju šeme prema predefinisanim scenarijima.

Kako bi navedeni cilj bio postignut, neophodno je razviti namenski grafički jezik koji omogućava korisniku da izmodeluje svoje zahteve za arhiviranje i migraciju i da automatski izgeneriše skripte sa naredbama za sisteme za upravljanje bazama podataka izabranih proizvođača: *Oracle*, *Microsoft SQL Server* i *MySQL*. Za dostizanje prethodno definisanog cilja neophodno je izvršiti sledeće zadatke: proučiti i izvršiti analizu postupka arhiviranja i migriranja baze podataka između SUBP-ova različitih proizvođača, kreirati meta-model, definisati skup grafičkih simbola koji predstavljaju konkretnu sintasku i definisati ograničenja za validnost modela.

2. PREGLED PRETHODNIH ISTRAŽIVANJA U OBLASTI

U literaturi se može pronaći značajan broj radova koji se bave temom primene i uticaja principa MDSE na razvoj softvera. Posebno su izdvojeni radovi koji se bave primenom ovih principa u svrhu razvoja generatora SQL koda. S obzirom na to da se ovaj rad bavi razvojem grafičkog jezika, dodatno je dat osvrt na radove koji proučavaju slično okruženje za razvoj konkretne grafičke sintakse.

Upotreba MDSE principa u realizaciji generatora SQL-PL koda na osnovu definisanih OCL izraza predstavljen je u radu [3]. Sličnost ovog rada sa trenutnim istraživanjem je u tome što su i ovim radom opisana preslikavanja izvornih koncepata na ciljne, tako da se generišu skripte za nekoliko SUBP-ova, kao što su *MySQL*, *MariaDB*, *PostgreSQL* i *SQL Server*. Ovim radom takođe je objašnjena pojava da različiti RDBMS implementiraju određene sintaktičke varijacije u standardnu SQL notaciju. Dodatno, ovaj rad naglašava kako je izgradnja modela centralna u dizajnu sistema. Osim toga, naglašava da MDSE principi ispoljavaju pozitivne aspekte sa stanovišta kvaliteta razvoja softvera, kada se izvorni jezik za modeliranje može potpuno povezati sa izabranim ciljnim jezikom.

Prednosti razvoja softvera zasnovanog na modelima, motivacija za potrebom generisanja SQL skripti i koraci implementacije SQL generatora detaljno su opisani u radu [4]. Autori ovog rada bave se razvojem alata za razvoj softvera kojim se dizajnira model nezavisan od tehnologije, na osnovu koga se kao rezultat generiše izvršni programski kod za različita ciljna okruženja i SQL skripte, za različite ciljne RDBMS.

Sa stanovišta sličnih tehnologija i domena istraživanja izdvojen je rad [5] koji opisuje modele baza podataka. U njemu je detaljno prikazan meta-model šeme relacionih baza podataka specifikiran pomoću *Eclipse Modeling Framework* (EMF), čime je specifikiran i meta-model razvijenog jezika. S obzirom na to da je tema ovog rada razvoj grafičkog jezika, rad koji predstavlja sličan razvoj konkretne sintakse je [6]. U ovom radu korišćen je *Sirius* okvir za razvoj grafičkog uređivača domenskog modela, koji pojednostavljuje specifikaciju proizvoda, smanjuje vreme dizajna i brzo povećava ukupnu produktivnost.

3. DOMEN PROBLEMA

Sistemi za upravljanje bazama podataka sadrže mehanizme koji obezbeđuju kreiranje arhive podataka kao vid preventivnog načina za zaštitu od gubitka podataka. Kreiranje arhive naziva se *backup*. *Backup* je kopija podataka iz baze koja se koristi kako bi bilo moguće

rekonstruisati podatke kad je to neophodno. *Backup* podataka moguće je sprovesti u skladu sa potrebama korisnika. Pod tim se podrazumeva da postoji mogućnost odabira strategije, vrste, tipa i režima u kojem se vrši *backup*. Specificiranje strategije, vrste, tipa i režima arhiviranja podržavaju svi izabrani SUBP-ovi različitih proizvođača za koje je arhiviranje razvijenim jezikom podržano.

Arhiviranje *Oracle* baze podataka moguće je sprovesti na tri načina, a to su: koristeći *Recovery Manager* (RMAN), *Enterprise Manager* (EM) ili ručnim kreiranjem skripte. Namenski jezik opisan u ovom radu funkcioniše tako što se grafičkim simbolima zadaju tip, strategija, vrsta i režim arhiviranja nakon čega se automatski generiše RMAN skripta koju treba izvršiti kroz komandnu liniju. Sličnost sa arhiviranjem koristeći EM je to da postoji korisnički interfejs i to što se skripte automatski generišu, a sličnost arhiviranju koristeći RMAN je da skripte treba izvršiti kroz komandnu liniju. Ukoliko korisnik ima potrebu da arhivira *Microsoft SQL Server* bazu podataka, neophodno je da prouči koncepte za korišćenje *SQL Server Agent* i da ručno kodira *T-SQL* skriptove. Korišćenjem grafičkog namenskog jezika opisanog u ovom radu, ove skripte generišu se automatski. Za kreiranje arhive *MySQL* baze podataka postoji više načina od kojih je najzastupljeniji onaj koji je zasnovan na *MySQLDump*-u. *MySQLDump* je uslužni program za *MySQL* i omogućava zadavanje niza opcija za arhiviranje baze podataka. Razvijenim grafičkim jezikom omogućeno je automatsko generisanje naredbi sa željenim parametrima za ovaj program. Koristeći ove načine, moguće je definisati strategiju kreiranja arhive, vrstu arhive, tip arhive i režim u kojem se vrši arhiviranje. Strategija podrazumeva odabir da li se arhivira celokupna baza ili samo deo baze podataka. Vrstom se definiše da li se kopiraju svi podaci ili samo oni koji su se promenili u odnosu na prethodni *backup*. Tipom se definiše na koji način će podaci biti skladišteni i kompresovani. Režim može biti *offline* ili *online*.

Proces migracije podrazumeva bilo koju transformaciju strukture baze podataka koja nastaje kao posledica izmena u strukturi šeme baze podataka. Migracijom se ne transformiše samo šema baze podataka, već se transformišu podaci u bazi podataka, da bi odgovarali novom opisu koji zadaje šema baze podataka. Grafičkim namenskim jezikom opisanim u ovom radu podržano je nekoliko scenarija za promenu nad šemom baze podataka, čime se programeri oslobađaju potrebe za poznavanjem konkretne sintakse i ručne implementacije SQL skriptova. Podržana scenarija su:

- dodavanje nove kolone datog tipa i s mogućnošću opcionog zadavanja predefinisane vrednosti i određenih ograničenja,
- razdvajanje jedne kolone sa postojećim podacima na više kolona sa očuvanim podacima prilagođenim specifikiranim zahtevima,
- spajanje više kolona sa postojećim podacima u jednu kolonu sa očuvanim podacima prilagođenim specifikiranim zahtevima,
- razdvajanje postojeće tabele na više tabela sa definisanim preuzetim kolonama i
- spajanje više tabela prema zajedničkoj koloni, pri čemu se specifikira mogući način spajanja.

Prilikom razvoja namenskog jezika potrebno je utvrditi korisničke zahteve i u skladu sa njima definisati koncepte jezika, tako da se opravda potreba za razvojem takvog alata. Očekivano je da potrebu za ovim akcijama imaju programeri koji koriste podatke i poznaju tehnologiju, ali ne poznaju u dovoljnoj meri koncepte administracije baza podataka i ne žele da se izlažu riziku da poremete strukturu baze podataka, ukoliko bi ručno vršili izmene.

Ključna razlika između ovog alata i alata slične namene u domenu poslova administracije baze podataka je što većina alata obezbeđuje arhiviranje podataka putem samo jednog konkretnog SUBP-a, izabranog proizvođača. Ovim jezikom korisnik može specificirati da se istovremeno formiraju skripte za arhiviranje baza podataka kreiranih pod SUBP-om različitih proizvođača. Osim toga, prednost ovog alata u odnosu na alate namenjene za migraciju podataka jeste u tome što sadrži predefinisani skup scenarija korišćenja u kojima se garantuje da će podaci biti sačuvani i da ručne izmene nisu potrebne.

4. RAZVOJ GRAFIČKOG NAMENSKOG JEZIKA ZA ARHIVIRANJE I MIGRACIJU BAZE PODATAKA

U ovom odeljku detaljno je opisan metamodel na kojem se zasniva konkretna sintaksa, OCL ograničenja koja obezbeđuju pravila za validnost modela, pregled grafičkih simbola sa opisanom semantikom korišćenja, opis korišćene tehnologije u svrhu implementacije i opis generatora koda.

4.1. Metamodel grafičkog namenskog jezika za arhiviranje i migraciju baze podataka

Metamodel biće opisan kroz dva dela: metamodel za definisanje koncepata za arhiviranje baze podataka i metamodel za definisanje koncepata za migraciju baze podataka. Metamodel kreiran je korišćenjem *Eclipse Modeling Framework*-a, koji se zasniva na korišćenju *Ecore* jezika za metamodelovanje.

Korenski element koji sadrži sve ostale elemente je metaklasa *Backup*. Na ovakav način se obezbeđuje, da ukoliko se pristupi jednom elementu, dobijaju se informacije o svim ostalim elementima. Elementi za definisanje parametara za arhiviranje su strategija, tip, vrsta i režim što je predstavljeno metaklasama *Strategija*, *Tip*, *Vrsta* i *Rezim*. Pored toga data je metaklasa *DodatneOsobine* kojom se obezbeđuje da se pored obaveznih parametara zadaju i parametri koji bi arhivu prilagodili korisničkim potrebama. Minimalna specifikacija mora da sadrži strategiju i režim gde se zadaju obavezni parametri za arhiviranje. Ukoliko se vrši parcijalno arhiviranje, neophodno je da se definiše da li se arhivira tabela, datoteka ili tablespace što se opisuje metaklasama *Table*, *File* i *Tablespace*. Za bazu podataka, proizvođač se definiše zadavanjem atributa *subp* čije vrednosti su iz enumeracije *Proizvodjac*. Enumeracija *Proizvodjac* definiše da li će se generisati naredbe za *Oracle*, *MySQL* ili *SQLServer*. Dodatno, moguće je opcionalno specificirati *ZipArhivu* koja obuhvata na kojoj lokaciji se skladišti arhiva, pod kojim nazivom i koje datoteke ona obuhvata. Osim elemenata koji su neophodni kako bi bili opisani procesi arhiviranja, postoje i metaklase koje su neophodne za pravilno definisanje modela. U cilju specifikacije načina povezivanja elemenata, kao i redosleda povezi-

vanja, uvodi se posebna metaklasa *Element*. Metaklasu *Element* nasleđuju *Strategija*, *Tip*, *Vrsta*, *Rezim* i *DodatneOsobine*. Definisanjem elementa je podignut nivo apstrakcije gde svaka od ovih metaklasa predstavlja element sa osobinom redosleda.

Scenarija migracije podržane ovim grafičkim jezikom su dodavanje kolone, spajanje više kolona u jednu, razdvajanje jedne kolone na više, spajanje i razdvajanje tabela po određenom kriterijumu. U skladu sa tim, metamodel sadrži metaklase *RadSaKolonama* i *RadSaTabelama*. *RadSaKolonama* obuhvata attribute neophodne za generisanje skriptova koji sadrže SQL naredbe za dodavanje nove kolone, razdvajanje jedne kolone na više kolona po određenom delimiteru i spajanje više kolona u jednu sa očuvanim podacima. *RadSaTabelama* obuhvata attribute neophodne za generisanje skriptova koji sadrže SQL naredbe za spajanje više tabela u jednu prema zajedničkoj koloni ili razdvajanje jedne tabele i njenih kolona na više tabela.

4.2. Ograničenja grafičkog namenskog jezika za arhiviranje i migraciju baze podataka

Pri kreiranju jezika nije moguće sva potrebna ograničenja implementirati na nivou strukture metamodela, jer jezici za metamodelovanje obuhvataju ograničen skup koncepata za definisanje novog jezika. Implementacija ograničenja obezbeđuje kreiranje pravila koja definišu validan model. U slučaju narušavanja ograničenja, korisnik je obavešten o konkretnoj grešci i onemogućeno je generisanje daljih koraka. Ograničenja su implementirana pomoću OCL-a. OCL je formalni jezik koji se koristi za dopunu strukture metamodela kroz skup tekstualnih pravila sa kojim svaki kreirani model mora biti u skladu. Neka od ograničenja su: postavljanje određenih polja kao obaveznih, definisanje ograničenja jedinstvenosti, ograničenja koja kontrolišu validnost unosa određenog atributa, ograničenja koja kontrolišu redosled i način povezivanja elemenata modela.

4.3. Konkretna sintaksa grafičkog namenskog jezika za arhiviranje i migraciju baze podataka

Razvoj grafičke konkretne sintakse namenskog jezika za arhiviranje i migraciju baze podataka obuhvata više koraka. Najpre je definisan skup simbola. Skup simbola izabran je tako da jezik bude intuitivan za korišćenje i da simboli odgovaraju semantici. Za simbole koje nemaju uobičajnu reprezentaciju izabrana je proizvoljna grafička predstava. Nakon specifikacije simbola i njihove semantike, definisana su pravila kombinovanja i povezivanja pojedinačnih elemenata. Na kraju je svaki element modela preslikan na koncept metamodela. Za razvoj konkretne sintakse korišćen je *Sirius* koji predstavlja *Eclipse* projekat koji omogućava da se kreira radna površina za grafičko modelovanje oslanjajući se na tehnologije *Eclipse Modeling Frameworka*.

4.4. Generator koda

Ideja razvoja ovog jezika je da se podrže specifični, jasno definisani scenariji za koje je smatrano da je potreba najveća, tako da korisnik dobije kompletno izgenerisane skripte za konkretnu situaciju. Prednost je to što ukoliko korisnik koristi jezik u opisanom domenu, neće imati potrebe da dodaje ručno pisani kod. Mana je to da su

scenariji arhiviranja i migracije usko specificirani, tako da ukoliko korisnik ima potrebe za dodatnim funkcionalnostima, morao bi ručno da dodaje kod u skriptove. Generator koda realizovan je pisanjem šablona unutar *.xtend* fajla. Generator obuhvata šablone za generisanje skriptova za arhiviranje i šablone za generisanje skriptova za migraciju baza podataka. Uz svaki kodirani šablon automatski se kreira jedna, ili više kompitabilnih izvršivih *Java* metoda. S obzirom na to da različiti SUBP-ovi imaju različitu sintaksu i postupke za kreiranje arhive, neophodno je za svaki SUBP kreirati po jedan šablon. Rezultat svakog šablona je po jedna skripta, koju je potrebno pokrenuti pod određenim SUBP-om.

5. POREĐENJE SA ALATIMA SLIČNE NAMENE

Poređenje je vršeno sa alatima koji su izabrani tako da kao krajnji proizvod generišu SQL skripte prilagođene za *Oracle*, a to su: *Oracle Enterprise Manager (EM)* i *Oracle Data Integrator (ODI)*. EM predstavlja veb alat za upravljanje bazama podataka koji je ugrađen unutar SUBP *Oracle*. EM obezbeđuje zadavanje parametara za kreiranje arhive, gde se kao rezultat generiše RMAN skripta. Sličnost sa razvijanim jezikom je u tome, što korisnik ukoliko poznaje domen problema i vrednosti parametara koji su neophodni, bez poznavanja konkretnih procedura i sintakse može da arhivira svoje podatke. Razlika je u tome što je korisnički interfejs za zadavanje parametara drugačiji. Prednost EM u ovom segmentu je u tome što korisnik ne mora da proučava pravila jezika za kreiranje modela. Sa druge strane, prednost razvijanog alata je u tome što podržava generisanje skripti za više proizvođača, dok je EM isključivo primenljiv u okviru SUBP *Oracle*.

ODI je alat čije korišćenje ima drugačiju svrhu od razvijanog jezika. Bez obzira na to, razlog zbog čega je ovaj alat izabran za poređenje je u tome što postoje određene sličnosti sa razvijanim jezikom. Sličnosti su:

- ODI predstavlja grafički alat kojim se specificira grafički model sa određenim osobinama,
- namenjen je za sličnu grupu korisnika kao i razvijeni jezik, a to su eksperti domena koji poznaju relacioni model, ali korišćenjem alata su oslobođeni od kodiranja i
- grafičke elemente je potrebno povezati u sekvencu, kako bi se izgenerisala i izvršila SQL skripta koja vrši procese specificirane grafičkim modelom.

6. ZAKLJUČAK

U ovom radu opisan je razvoj namenskog grafičkog jezika za automatsko generisanje skripti za arhiviranje i migraciju za određene proizvođače baza podataka. Razvoj je obuhvatao specifikaciju zahteva, definisanje meta-modela zasnovanog na *Ecore*, implementaciju ograničenja, definisanje konkretne sintakse i šablona za generisanje koda. Jezik obuhvata deo namenjen za administraciju baze podataka i deo namenjen za migraciju baza podataka. Deo za administraciju baza podataka obuhvata zajedničke koncepte sistema za upravljanje bazama podataka različitih proizvođača i time rešenje može da zadovolji uobičajne korisničke potrebe u domenu poslova administracije. U poređenju sa alatima slične namene, prednost je u tome što je generisanje obezbeđeno za više SUBP-ova različitih proizvođača definisanjem samo

jednog modela. U daljem razvoju očekuje se proširenje jezika konceptima kojima bi se pokrile specifičnosti za kreiranje arhive pojedinačnih proizvođača. Proširenje bi obuhvatalo definisanje koncepata u meta-modelu, definisanje i implementaciju ograničenja, definisanje grafičkih simbola i dopunu šablona za generisanje koda.

Deo za migraciju podataka obuhvata predefinisana scenarija migracije baze podataka. Istraživanjem literature ustanovljeno je da su ova scenarija najviše potrebna u praksi. Prednost, u poređenju sa ostalim rešenjima, u tome je što ovim jezikom korisnik kao rezultat dobija izmenjenu bazu podataka sa zasigurno očuvanim podacima u željenoj formi. Mana je u tome što postoji ograničeni skup predefinisanih scenarija.

Nakon poređenja grafičkog namenskog jezika za arhiviranje i migraciju baze podataka sa jezikima slične namene i nakon proučavanja radova koji se bave sličnim temama uočene su potrebe za buduća proširenja. Sledeći korak u razvoju grafičkog jezika za arhiviranje i migraciju baza podataka je proširenje jezika tako da podrži koncepte neophodne za proceduru oporavka. Pored toga, kako bi se poboljšalo korisničko iskustvo, potrebno je određene parametre preuzeti iz baze podataka, čime se izbegavaju greške zadavanja parametara kroz attribute modela. Osim ovog proširenja potrebno je proširiti skup predefinisanih scenarija za migraciju.

7. LITERATURA

- [1] Marco Brambilla, Jordi Cabot, and Manuel Wimmer (2012), *Model-Driven Software Engineering in Practice*. A Publication in the Morgan & Claypool Publishers series SYNTHESIS LECTURES ON SOFTWARE ENGINEERING
- [2] Sava Jelisavčić, Ivan Jelisavčić (2011), Metodologija migracije podataka velikih informacionih sistema, INFOTEH-JAHORINA Vol. 10, Ref. E-I-4, p. 409-413
- [3] Marina Egea, Carolina Dania (2017), SQL-PL4OCL: An Automatic Code Generation from OCL to SQL Procedural Language ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS)
- [4] Ivan Luković, Sonja Ristić, Slavica Aleksić, Aleksandar Popović (2008), An Application of the MDSE Principles in IIS*Case, SIG Model Driven Software Engineering: 3rd Workshop MDSE 08, Serbia
- [5] Sonja Ristić, Slavica Aleksić, Milan Čeliković, Ivan Luković (2013), An EMF Ecore Based Relational DB Schema Meta-Model, ICIT 2013 The 6th International Conference on Information Technology
- [6] Vladimir Vujović, Mirjana Maksimović, Branko Perišić (2014), Sirius: A Rapid Development of DSM Graphical Editor, IEEE 18th International Conference on Intelligent Engineering Systems, Tihany, Hungary

Kratka biografija:



Marija Kukić rođena je u Zrenjaninu 1995. god. Fakultet tehničkih nauka upisala 2014. god. Bečelorski rad iz oblasti Elektrotehnike i računarstva – Računarske nauke i informatika odbranila je 2018. god. Master rad iz iste oblasti odbranila je 2019. god.

GRAFIČKI INTERFEJS EDUKATIVNOG ELEKTROFIZIOLOŠKOG AKVIZICIONOG UREĐAJA**USER INTERFACE OF THE ELECTROPHYSIOLOGICAL DATA ACQUISITION DEVICE FOR EDUCATIONAL PURPOSES***Žaklina Stojšin, Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U okviru ovog rada predstavljena je realizacija grafičkog korisničkog interfejsa koji je razvijen za mikrokontroler ATXMega128A1 u okruženju MikroC pro for AVR. U radu su priloženi rezultati dobijeni upotrebom dizajnirane aplikacije i njihova analiza.

Ključne reči: Grafički korisnički interfejs, mikrokontroler, elektrofiziološki signali

Abstract – This paper represents the realization for designing graphical user interface which is developed for microcontroller ATXMega128A1 in developing platform MikroC pro for AVR. It has results that are given from designed application and it's analysis.

Keywords: Graphical user interface, microcontroller, electrophysiological signals.

1. UVOD

Grafički interfejs predstavlja softversko okruženje koje omogućava korisniku komunikaciju sa računarom na osnovu prethodno definisanih funkcija [1]. Razvoj grafičkog korisničkog interfejsa treba da uvaži karakteristike hardverskog uređaja i dostupne softverske resurse. Pažljivo osmišljen i koncipiran grafički interfejs daje upotpunjen finalni proizvod, a istovremeno kreativni izazov za dizajnera.

Zadatak dizajnera jeste da razume mogućnosti i potrebe korisnika. Potrebno je da interfejs bude jednostavan, lak za učenje, logički organizovan sa uočljivim obeležjima i smernicama. Jednostavnost u vidu razumevanja željene funkcije i pronalaženja željene opcije olakšava upotrebu interfejsa. Veliki izazov dizajniranja interfejsa ogleda se u raznovrsnosti ljudskih sposobnosti, karakteristike ličnosti, kulture i načina rada korisnika [2].

Ekran grafičkog interfejsa treba da odlikuje funkcionalnost i složenost [3]. Uspešno dizajniran ekran treba da pruži korisniku sve podatke u pravilnom redosledu kako bi se određeni zadatak adekvatno obavio. Grafički interfejsi u medicini pružaju mogućnost brzog dijagnostikovanja bolesti.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vojin Ilić, vanred. prof.

Praćenje stanja pacijenta omogućava se van zdravstvenih ustanova. Dodavanjem trodimenzionalnog predstavljanja, zvuka, animacija, video materijala poboljšava se izgled i informativnost grafičkih interfejsa. Sve veća pažnja se pridaje aspektima upotrebljivosti interfejsa. Konstantnim usavršavanjem dizajna teži se ka postizanju krajnjeg produkta koji ima veliku brzinu rada, minimalan broj grešaka i koji pruža zadovoljstvo u radu. Prepoznatljiv stil daje lični pečat dizajniranom grafičkom interfejsu. Krajnji cilj je zadovoljavanje očekivanja korisnika u okviru budžeta.

U okviru rada prikazano je jedno rešenje edukativnog elektrofiziološkog sistema. Realizovan uređaj predstavlja prototip koji je prenosiv i pristupačan korisnicima. Sistem služi za akviziciju EKG i EMG signala.

2. GRAFIČKI INTERFEJSI

Od davnina se sve veći značaj pridaje realizaciji grafičkih interfejsa. U današnje vreme se pažnja pridaje velikoj funkcionalnosti sistema. Teži se ka realizaciji uređaja koji ima visok kvalitet, laku prenosivost, male dimenzije i koji je ujedno jednostavan za upotrebu. U nastavku je prikazan istorijski razvoj grafičkih interfejsa, dat je primer prvog EKG uređaja i prikazani su neki od razvijenih, funkcionalnijih i savremenijih sistema.

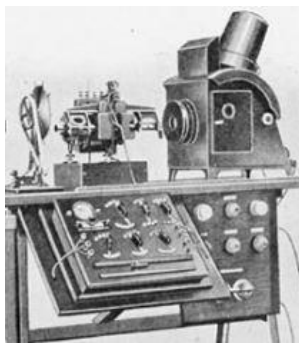
2.1 Telekardiogram

Prvi EKG uređaj je bio veoma robustan. Sastojao se od tanke žice koja se nalazila između polova potkovičastog magneta.

Prilikom prolaska struje kroz žicu, stvaralo se magnetsko polje. Dato polje je izazivalo pomeranje žice. Svetlost koja se odazirala sa žice je stvarala senku na pokretnom fotografskom papiru. Mašina je zahtevala vođeno hlađenje.

Grafički interfejs datog uređaja imao je složenu funkcionalnost, što može da se uoči na slici 1. Medicinsko osoblje je moralo biti obučeno kako bi moglo da upravlja mašinom. S obzirom na njegovu robustnost i težinu, uređaj je bio nepomičan.

Sve karakteristike ukazuju na komplikovanost snimanja EKG signala. Brzina rada uređaja i vreme učenja korisnika bile su male.



Slika 1. Grafički interfejs uređaja

2.3 Holter monitoring uređaj

Holter monitoring uređaj predstavlja prenosivi uređaj koji omogućava snimanje srčanog ritma pacijenta od 24 sata do 48 sati neprestano. Elektrode su postavljene na grudni koš pacijenta i povezane su za prenosivi uređaj putem žica [5]. Grafički interfejs uređaja je veoma jednostavan za upotrebu i razumevanje. Prednost njegove jednostavnosti jeste u mogućnosti upotrebe bez kompleksne obuke korisnika.

Uređaj zbog svoje jednostavnosti i funkcionalnosti se koristi van zdravstvenih ustanova. Posедуje jedan ekran na kom se vide povratne informacije snimanja, odnosno EKG signal. Sadrži dugmiće sa jasnim smernicama za upotrebu, što može da se vidi na slici 2.



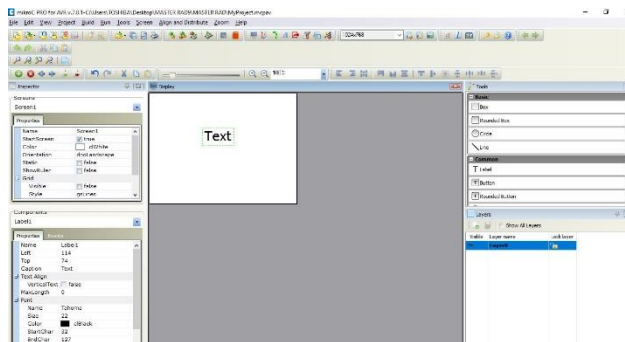
Slika 2. Grafički interfejs uređaja

3. PROGRAMSKO OKRUŽENJE ZA RAZVOJ GRAFIČKOG INTERFEJSA

Za dizajn aplikacije korišćeno je radno okruženje *MikroC PRVO for AVR*, razvijeno od strane kompanije *Mikroelektronika*. Dato okruženje se koristi za programiranje mikrokontrolera u programskom jeziku C. Okruženje pruža mogućnost dizajniranja *Visual TFT aplikacije*. Kako bi se realizovala data mogućnost neophodno je odabrati datu opciju prilikom pravljenja novog projekta.

Nakon željenog odabira, korisniku se pružaju opcije odabira direktorijuma u kojem želi da se nalazi dati projekat i mogućnost imenovanja aplikacije. Nakon izvršenih koraka, generiše se okruženje u kom korisnik kreira kod.

Nakon što je korisnik odabrao *Visual TFT project* kao novi projekat, pored okruženja u kom se kreira kod, može da se pristupi i okruženju u kom se dizajniraju ekrani date aplikacije. Dato radno okruženje može da se vidi na slici 3. Postavljanje objekata na ekran se vrši jednostavnim prevlačenjem iz menija sa alatima.



Slika 3. Visula TFT okruženje

Prozor sa alatima, lociran u gornjem desnom uglu služi za dizajniranje ekrana. Korisnik ima mogućnost da bira između geometrijskih oblika (*Box*, *Rounded Box*, *Circle*), linije (*Line*), labela za pisanje teksta (*Label*), dugmića (*Button*, *Rounded Button*, *Circle Button*), polja za unos slike na ekran aplikacije (*Image*), polja za čekiranje (*CheckBox*, *Radio Button*).

Jednostavnim prevlačenjem dizajner ima mogućnost da postavi objekte u neograničenoj količini. Svaki objekat može da se obriše sa ekrana, u bilo kom trenutku kreiranja aplikacije.

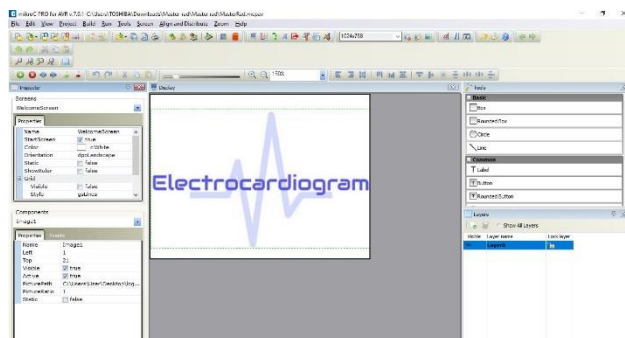
Prozor u gornjem levom uglu okruženja služi za personalizovanje ekrana. Prozor pruža mogućnost odabira pozadinske boje ekrana (*Color*). Ukoliko aplikacija poseduje više od jednog ekrana pruža se mogućnost odabira početnog ekrana (*StartScreen*).

Postoji mogućnost odabira orijentacije ekrana (*Orientation*). Mogućnost biranja da li će ekran biti statičan ili dinamičan se bira sa obeležavanjem kvadratića pod opcijom *Static*. Prikaz lenjira na gornjem i levom rubu ekrana pomaže korisniku da na lakši način napravi željene promere.

Data opcija se aktivira obeležavanjem kvadratića pod opcijom *ShowRuler*. Postoji mogućnost prikaza mreže na ekranu, koja služi samo kao orijentacija za dizajnera i pomaže mu u lakšem postavljanju objekata na ekvidistantne razdaljine. Opcija se aktivira obeležavanjem kvadratića u rubrici *Visible*.

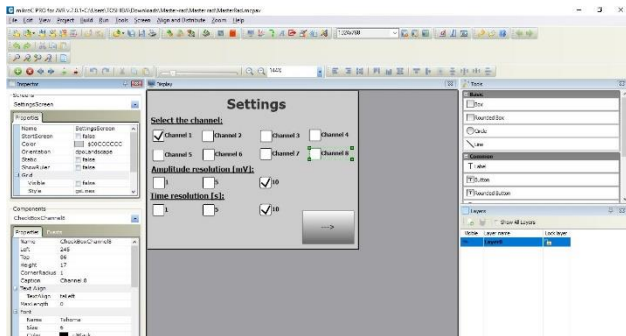
3.1 Grafički interfejs edukativnog elektrofiziološkog sistema

Krajnji izgled dizajnirane aplikacije može da se vidi na slikama 4, 5, 6. Početni ekran dizajnirane aplikacije predstavlja logo aplikacije. Logo aplikacije je uvezena slika formata *.jpg*, što može da se vidi na slici 4.



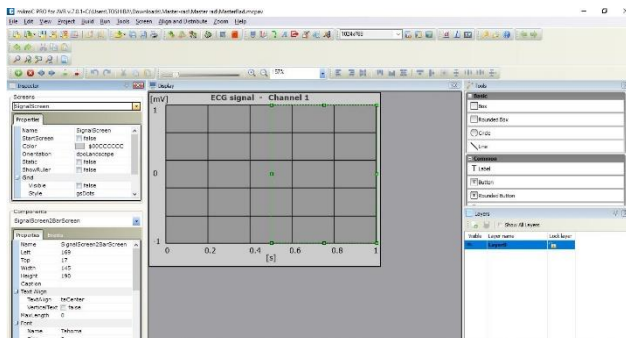
Slika 4. Početni ekran dizajnirane aplikacije

Nakon početnog ekrana se pojavljuje *Settings* ekran koji na sebi ima naslov, tri tekstualne labele, 14 polja za čekiranje i dugme za prelazak na sledeći ekran, što može da se vidi naslici 7. Naslov je tipa **bold**, veličine 16, statičan i neaktivan. Tri labele koje predstavljaju podnaslove (*Select the channel*, *Amplitude resolution [mV]*, *Time resolution [s]*) su tipa underline, veličine 8, statični i neaktivni. Od 14 polja za čekiranje, tri polja su obeležena od početka pojavljivanja ekrana, a ostala polja nisu obeležena. Sva polja su aktivna. Dugme za nastavak na sledeći ekran je tipa *OnClick* i aktivan.



Slika 5. *Settings* ekran dizajnirane aplikacije

Nakon *Settings* ekrana se pojavljuje ekran za prikaz signala. Na datom ekranu se nalazi naslov (*ECG signal – Channel 1*), koji je tipa **bold**, statičan i neaktivan, nalazi se polje na kom se iscrtava signal. Pozadina polja predstavljaju dva dugmeta, koja su spojena bez prostora između njih. Desno dugme je neaktivno, a dugme sa leve strane je aktivno i tipa *OnDown*. Nakon pritiska na dugme sa leve strane, aplikacija vraća korisnika na *Settings* ekran, bez obzira što se iscrtavanje signala nije završilo. Ispod donje ivice polja za iscrtavanje signala nalazi se vremenska osa, koju sačinjavaju labele. Sa leve strane polja za iscrtavanje signala nalazi se amplitudska osa koju takođe sačinjavaju labele.



Slika 6. Ekran za prikazivanje snimljenog signala

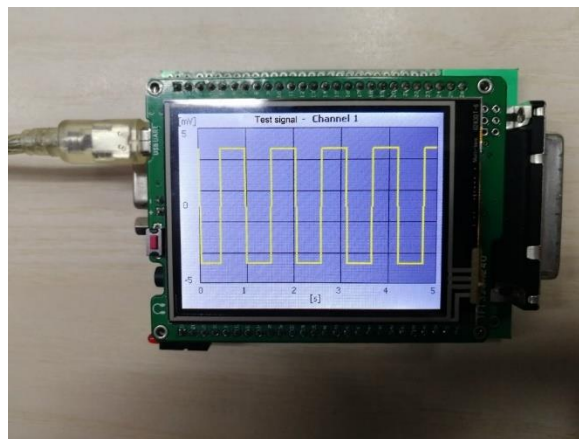
4. VERIFIKACIJA REALIZOVANOG EDUKATIVNOG ELEKTROFIZIOLOŠKOG SISTEMA

Prvobitno je testiran uređaj sa dva tipa generatora:

1. Interni generator test signala integrisanog kola ADS1299
2. Eksterni test signal generator *SIGI II EasyCap*

Prilikom testiranja aplikacije sa internim generatorom test signala dobijena je povorka pravougaonih četvrtki amplitude od -3.75 do +3.75 mV. Na ekranu je prikazan

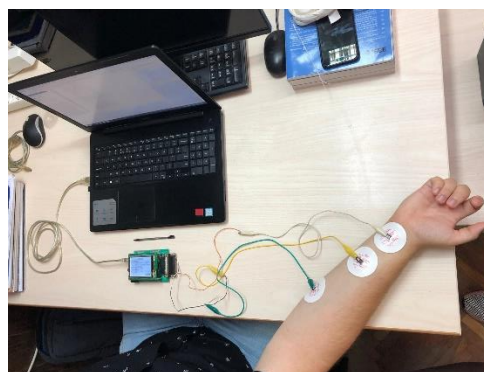
signal na vremenskom intervalu od 5 sekundi. Primer dobijenog signala može da se uoči na slici 7.



Slika 7. Izgled test signala dobijen internim test generatorom

4.1 Rezultati dobijeni snimanjem realnih elektrofizioloških signala

Kako bi snimanje EMG/EKG signala bilo uspešno potrebno je imati mikrokontroler ATXMega128A1 koji ima spušten program za izvršavanje datog zadatka. Mikrokontroler je povezan sa računarom preko *USB* komunikacije. Potrebno je imati tri površinske elektrode koje su sa jedne strane postavljene na ruku/ruke pacijenta, a sa druge strane su povezane žično sa pločicom mikrokontrolera, što se vidi na slici 8.



Slika 8. Proces snimanja EMG signala

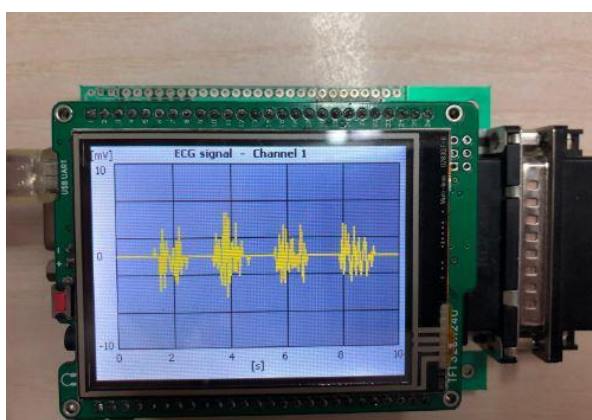
Početni ekran aplikacije je bio dizajnirani logo koji se zadržavao na ekranu 3 sekunde i nakon toga je automatski nestajao sa ekrana.

Nakon početnog ekrana sledi “*Settings*” ekran. Dati ekran se nalazi na mikrokontroleru sve dok korisnik ne pritisne dugme za nastavak koji se nalazio u donjem desnom uglu ekrana. Na datom ekranu se nudi opcija odabira kanala snimanja signala. Korisnik može da izabere jedan od ponuđenih 8 kanala. Takođe ima mogućnost odabira amplitudske rezolucije ekrana koja je prikazana u mV (milivoltima). U zavisnosti od korisničkog odabira amplitude signal se skalira i signal se prikazuje u odabranoj razmeri. Korisniku se nudi opcija biranja amplitude signala od 1mV, 5mV I 10mV. Poslednja opcija koju korisnik može da bira je vremenska rezolucija signala koja se prikazuje u sekundama. U zavisnosti da li je korisnik odabrao 1 sekundu, 5 ili 10, signal se u

prikazuje u odabranoj vremenskoj brzini. Korisnik ima zadatak da obeleži jedan od polja u svakom području. Ukoliko se ne zadovolji dati uslov korisnik neće dobiti povratnu informaciju od mikrokontrolera. Korisnik takođe ne sme da obeleži više od jednog polja

Nakon pritiska na dugme za nastavak pojavljuje se treći ekran koji u zavisnosti od odabranih opcija na "Settings" ekranu prikazuje dati signal. U gornjem uglu ekrana stoji naslov koji prikazuje odabrani kanal snimanja.

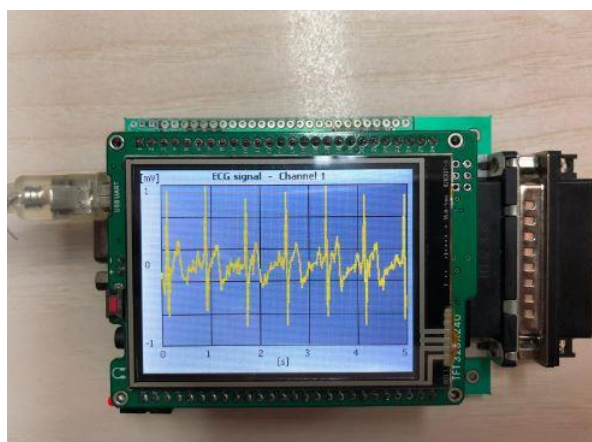
Za snimanje EMG signala korišćena je bipolarna konfiguracija elektroda. Data konfiguracija obuhvata tri elektrode, pozitivnu, negativnu i DRL elektrodu. Na slici 9 je prikazan signal snimljen sa prvog kanala. Vremenska rezolucija prikazivanja signala je 10 sekundi, stoga može da se uoče nekoliko mišićnih kontrakcija prilikom snimanja EMG signala. Amplitudska rezolucija ekrana je od -10 do +10 mV.



Slika 9. EMG signal

Prilikom snimanja EKG signala upotrebljen je prvi bipolarni odvod. Površinske elektrode se postavljaju na levu i desnu ruku. Pozitivna elektroda je postavljena na desnu ruku, a negativna i DRL elektroda su postavljene na levu ruku.

Na slici 10 prikazan je EKG signal. Vremenska rezolucija ekrana je 5 sekundi. Amplitudska rezolucija ekrana je od -1 do +1 mV.



Slika 10. EKG signal

5. ZAKLJUČAK

Mogućnosti dizajniranja grafičkih interfejsa za projektante su neiscrpne. Uticaj interfejsa na korisnike je veoma velik. Razvoj grafičkog interfejsa u medicini pomaže korisniku da se zainteresuje za sopstveno zdravstveno stanje. Posedovanjem prenosivog interfejsa pruža se mogućnost merenja elektrofizioloških signala van zdravstvenih ustanova.

Razvijeni edukativni elektrofiziološki uređaj omogućava korisniku, a pre svega studentima da steknu praktično znanje iz oblasti elektrofiziologije i akvizicionih mernih sistema. Pruža se mogućnost merenja realnih elektrofizioloških signala. Uređaj je jednostavan za upotrebu i prenosiv.

Performanse uređaja se mogu unaprediti uvođenjem bežične komunikacije sa računarom. Mogućnost razvoja merenja impedanse interfejsa elektroda – tkivo bi dodatno unapredilo uređaj. Ukoliko bi se dodalo merenje oksigenisanosti krvi u organizmu tada bi performanse uređaja bile potpune. Kvalitet usluga ima rastući značaj kako za korisnika tako i za proizvođače.

Težnja za sveprisutnim računarskim sistemima i zahtevi tržišta neprekidno povećavaju značaj univerzalnosti i upotrebljivosti interfejsa. Savršenstvo je u jednostavnosti.

6. LITERATURA

- [1] Vikipedija, slobodna enciklopedija, *Grafički korisnički interfejsi*.
- [2] Ben Sheiderman, Catherine Plaisant, *Designing the user interface*, University of Maryland 2005.
- [3] Orawit Thinnukool, *The user's satisfaction of graphic user interface in designing for health care mobile applicaton*, Chiang Mai University 2017.
- [4] Vikipedija, slobodna enciklopedija, *Elektrokardiogram*.

Kratka biografija:



Žaklina Stojšin rođena je u Subotici 1995. godine Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva je odbranila 2019. godine.

Kontakt: zanastojšin@yahoo.com

**ЕЛЕКТРОМАГНЕТНА КОМПАТИБИЛНОСТ ПРЕТВАРАЧА ЕНЕРГЕТСКЕ
ЕЛЕКТРОНИКЕ У ИНДУСТРИЈИ****ELECTROMAGNETIC COMPATIBILITY OD POWER ELECTRONICS IN INDUSTRY**

Горан Мамлић, Веран Васић, Ђура Орос, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду приказани су неки аспекти везани за електромагнетну компатибилност претварача енергетске електронике у индустрији. Стављен је аспект на утицај електромагнетних сметњи, рад фреквентних претварача, генерисање електромагнетних сметњи од стране фреквентних претварача, примену електромагнетске компатибилности у пракси.

Кључне речи: Електромагнетска компатибилност, фреквентни претварачи, електромагнетске сметње.

Abstract – This paper presents some aspects related to the electromagnetic compatibility of power electronics converters in industry. The aspect was put on the influence of electromagnetic interference, operation of frequency converters, generation of electromagnetic interference by frequency converters, application of electromagnetic compatibility in practice.

Keywords: Electromagnetic compatibility, frequency converters, electromagnetic interference.

1. УВОД

Са сталним развојем технологије и повећањем индустријске производње последњих година долази до тога да је потребна све већа ефикасност и сигурност свих структура које учествују у производњи.

Раст електро индустрије довео је до тога да долази до целокупног искоришћења електромагнетног спектра за обављање многих задатака повезаних са различитим гранама индустрије.

Са повећањем сметњи како у индустрији тако и у стамбеном сектору намеће се потреба да се електромагнетне сметње на неки начин елиминишу. Како је немогуће елиминисати у потпуности све врсте сметњи долази до развоја стандарда који описују технике за њихово смањење.

Уводи се појам електромагнетне компатибилности уређаја. Данас приликом производње електронских уређаја мора се водити рачуна да дати уређај у окружењу неће допринети повећању електромагнетних сметњи, као и да окружење не утиче на електрични уређај.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Веран Васић, ред. проф.

2. ЕЛЕКТРОМАГНЕТНА КОМПАТИБИЛНОСТ**2.1. Опште**

Електрична, електромеханичка и електронска опрема мора бити у складу са спецификацијама које имају за циљ да осигурају електромагнетну компатибилност која је према ИЕС (енг. *International Electrical Commission*) објави под називом ИЕС 60050 познатој и као „Electropedia – IECV“ дефинисана као:

- „Способност уређаја, опреме или система да задовољавајуће функционише у свом електромагнетном окружењу без увођења неподношљивих електромагнетних сметњи било ком уређају у том окружењу.“

Према ИЕС 60050 дефиниција електромагнетне сметње је:

- „Сваки електромагнетни феномен који може да погорша карактеристике рада електричног уређаја, опреме или целокупног система, или негативно да утиче на животну околину или инертну материју. Електромагнетна сметња може бити електромагнетна бука, нежељени сигнал или промена карактеристика саме средине.“

У суштини електромагнетна сметња (енг. *EMI - Electromagnetic interference*) која се у радио фреквенцијском опсегу назива и радио – фреквентна интерференција (енг. *RFI - Radio frequency interference*) се састоји од било ког нежељеног, лажног, и / или зраченог сигнала електричног порекла које може изазвати неприхватљиву деградацију карактеристика система или опреме. Ефекти које електромагнетна сметња може да изазове могу да се крећу од мање неугодности до катастрофалних последица. Електромагнетне сметње обухватају распон фреквенције од 0Hz до 400GHz.

2.2. Извори електромагнетских сметњи

Како се густина електромагнетног окружења наставља повећавати због све веће употребе различитих електричних и електронских уређаја, забринутост о ефектима из извора који производе електромагнетне сметње такође се повећава. Електромагнетна компатибилност увек има два комплементарна аспекта.

Свака ситуација некомпатибилности мора имати извор електромагнетних сметњи и пријемник (опрема

или уређај – потрошач) који је подложен овим сметњама. Ако било ко од њих није присутан, нема проблема везаних за електромагнетну компатибилност. Иста опрема може бити извор у једној ситуацији и пријемник у другој. Извори електромагнетних сметњи могу да се класификују као људски створени или природни. Из основа електротехнике знамо да „протицање“ електричне струје кроз електричне проводнике доводи до појаве електромагнетног поља око датог проводника. Створено електромагнетно поље може да изазове сметње уређајима који се нађу у његовој околини. На основу овога може се закључити да сваки електрични или електронски уређај који преноси, дистрибуира, обрађује или на други начин користи било који облик електричне енергије може бити потенцијални извор електромагнетних сметњи ако било који аспект његовог рада генерише електромагнетне сигнале који могу проузроковати деградацију перформанси било које опреме или система који дели исто оруђење.

Како су електромагнетне сметње директно пропорционалне фреквенцији (простиру се као таласи) у електромагнетној компатибилности користи се фреквентни домен за описивање дејства сметњи и њихове компензације. У оваквом начину представљања, периодични сигнали се представљају као сума хармоника. Фреквенција и таласна дужина електромагнетне сметње (таласа) остварују везу према следећој једначини (2.2.1):

$$c = \lambda \cdot f [m/s] \quad (2.2.1)$$

где су:

- c – брзина светлости (ширења таласа), $c = 3 \cdot 10^8 [m/s]$,
- f – фреквенција таласа $[1/s]$, $[Hz]$,
- λ – таласна дужина $[m]$.

У фреквентном домену најчешће се користи следећа једначина за описивање промене величине слабљења:

$$p = 10 \cdot \log \frac{P_1}{P_2} [dB] \quad (2.2.2)$$

где су:

- p – слабљење изражено у децибелима у фреквентном домену,
- P_1 – амплитуда величине на извору (уређају),
- P_2 – амплитуда величине у датој тачки.

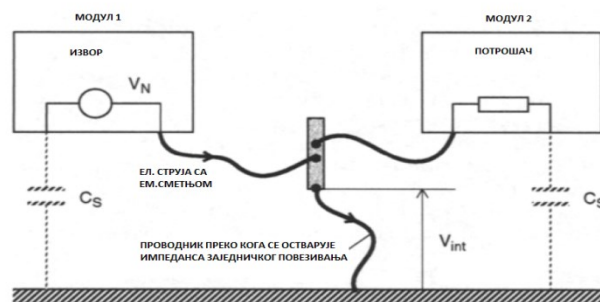
2.3.1. Преношење сметњи путем каблова, сигналних каблова, електроенергетских водова

Најједноставнији начин преноса електромагнетних сметњи је путем директне везе између извора електромагнетне сметње и пријемника. Најбољи пример који описује наведено, је веза преко електричне мреже којом се напајају потрошачи. Електромагнетне сметње присутне на излазима мрежног напајања путем електричних водова, електроенергетских каблова преносе се до мрежних улаза напајања потрошача (пријемника). Сама мрежа снабдевања електричном енергијом обично се сматра пасивним учесником приликом преноса електро-

магнетних сметњи. Поремећаји дуж водова, каблова, могу да се простиру путем њих на следећи начин:

- диференцијални начин спрезања (енг. *differential mode coupling*),
- спрезање преко заједничке импедансе (енг. *common mode coupling*).

Диференцијални начин спрезања тј. преноса електромагнетних сметњи огледа се у томе да се сметње простиру између фазних проводника. Спрезање преко заједничке импедансе тј. пренос електромагнетних сметњи представља пренос путем свих проводника у систему гледано у односу на земљу као референтни вод (заједнички вод). На слици 1 приказан је пренос електромагнетних сметњи преко заједничке импедансе.



Слика 1. Пренос електромагнетних сметњи преко заједничке импедансе.[1]

2.3.2. Преношење сметњи путем електромагнетних поља

Електрични уређаји (потрошач, извор електромагнетне сметње) не морају бити директно спојени да би дошло до преноса сметњи између њих. Као што је већ наведено, електрична струја која „протиче“ кроз проводник, око тог проводника ствара магнетно поље, а ако постоји напон између проводника ствара се и електрично поље.

Свако од ових поља способно је да индукује напон у другом струјном кругу који се налази у домену тих поља. Преношење сметњи (поремећаја) можемо разликовати на два начина:

- индуктивно спрезање – магнетно (енг. *inductive coupling*),
- капацитивно спрезање – електрично (енг. *capacitive coupling*).

Према закону електромагнетне индукције - Фарадејев закон (енг. Michael Faraday) ако се у магнетном пољу нађе проводник који се креће, или се магнетно поље креће око датог проводника на крајевима тог проводника долази до појаве електричног напона тј. индукује се електрични напон на крајевима датог проводника. Ако је струјни круг затворен успоставиће се и електрична струја кроз поменути проводник. Једначина која описује наведену појаву дефинисана је на следећи начин:

$$V_N = -M \cdot \frac{di}{dt} [V] \quad (2.3.2.1)$$

где су:

- V_N – индуковани електрични напон на крајевима посматраног проводника [V],
- M – међусобна индуктивност као последица електромагнетног спрезања [H],
- $\frac{di}{dt}$ – промена електричне струје у времену која ствара магнетно поље.

Индуковани напон на крајевима проводника суперпонира се на већ постојеће напоне у струјном кругу и на тај начин изазива сметње (поремећаје), овај напон не утиче на импедансу потрошача. Међусобна индуктивност одређена је размаком између проводника, дужином датих проводника, њиховом геометријом, као и присуством било ког другог страног поља између њих. Најчешћа ситуација у којој је магнетна спрега значајна је када се неколико различитих струјних кругова преноси заједно у једном кабелу. Допуна магнетној спрези укључује капацитивност која се може појавити између електричних уређаја, електричних уређаја и референтне тачке тј. земље и посредно електричног поља које се појављује између њих. Када се између два електрична уређаја (проводника) појави разлика напона, ствара се електрично поље. Ово поље индукује напон на пријемнику који је сразмеран са:

$$V_N = -C_C \cdot Z_N \cdot \frac{dV_S}{dt} [V] \quad (2.3.2.1)$$

где су:

- V_N - индуковани електрични напон у колу пријемника (на „крајевима“ потрошача) [V],
- Z_N - импеданса пријемника (потрошача) [Ω],
- C_C - капацитивност између електричних уређаја [F],
- V_S - електрични напон извора електромагнетне сметње (поремећаја) [V],

Можемо приметити да у односу на спрезање путем магнетног поља које зависи од фреквенције извора, спрезање путем електричног поља у значајној мери зависи од импедансе потрошача, па према томе електрична кола са великом импедансом су подложна спрезању путем електричног поља, а извор и пријемник морају бити референтни једно у односу на друго.

Пренос сметњи путем индуктивне и капацитивне спреге не може се избећи, али се кућиште уређаја може пројектовати тако да оно не постане ефикасна антена или проводна средина на одређеном опсегу учестаности у средини у којој се може наћи приликом употребе.

2.4. Закони, правилници, стандарди који дефинишу електромагнетну компатибилност

Свака држава доноси законску регулативу која одговара њеним потребама и условима у којима се користи комерцијална и индустријска опрема. Поред наведеног први и основни захтев који се увек ставља је безбедност опреме за рад и њен утицај на човека. Законске регулативе се дефинишу тако да електрична, електронска па и свака друга опрема за рад не доводи до штетних услова за рад човека, утицаја на животну средину.

Законска регулатива Републике Србије ослања се на законе, правилнике и уредбе који се позивају на одређење хармонизоване стандарде који су у складу са Европском и Интернационалном стандардизацијом. Република Србија као потписница чланица за приступање Европској Унији у области електромагнетне компатибилности ослања се на Европску директиву о електромагнетној компатибилности под називом “Directive 2014/30/EU of the European parliament and of the council on the harmonisation of the laws of the Member States relating to electromagnetic compatibility.”

Стандарди одређују методе и границе мерења или нивое испитивања за емисију и имунитет електричне опреме, електричних инсталација и система. Подела европских стандарда на различите категорије олакшава проналажење правила која се односе на одговарајућу опрему. Генерички стандарди неприметно се примењују на сву опрему. Ако је опрема у оквиру специфичних породица производа, стандарди се могу користити за проверу усаглашености опреме са локалним законима. Основи стандарди садрже информације о појавама сметњи и општим методама мерења.

3. ПРЕТВАРАЧИ ЕНЕРГЕТСКЕ ЕЛЕКТРОНИКЕ У ИНДУСТРИЈИ

Механички рад представља основу производње људског друштва. Што је већа количина механичког рада произведена по појединцу то је виши животни стандард посматране земље. Претварање механичке енергије у електричну и обрнуто у највећем броју случајева врше електромотори. Електромотори који су највише заступљени у индустрији су трофазни асинхрони мотори, а то све из разлога што су једноставне конструкције и ниске цене.

Најједноставнији, а и најчешћи начин пуштања трофазних електромотора у рад је директан старт електромотора, тј. пуштање мотора у рад директним довођењем његових намотаја под трофазни мрежни напон. Описани начин пуштања у рад има одређене недостатке који се огледају у великој полазној струји електромотора која може бити и до 8 пута већа од номиналне електричне струје датог електромотора. Како су електронске компоненте временом постале јефтине за израду долази се до решења која не изискују директно пуштање мотора у рад, за производне процесе где је то од битног значаја. Ова решења огледају се у виду електронских уређаја који служе за пуштање електромотора у рад и њихову регулацију. Два уређаја која се користе за ту сврху су:

- „софт стартер“ - служи за „меко“ пуштање електричних мотора у рад,
- фреквентни претварач – служи за управљање и регулацију брзине електромотора.

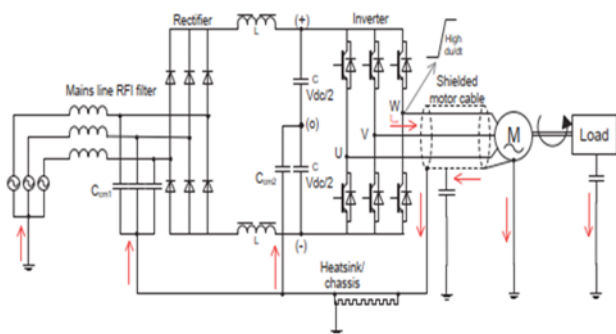
3.2 Фреквентни претварачи

Фреквентни претварачи данас најчешће могу да се сретну у индустријском окружењу. Користе се да узету електричну енергију из мреже претворе у потребан облик енергије за електричне моторе. Фреквентни претварачи врше брзо укључење /

искључење напона (имају велик однос du/dt) у веома кратким временским интервалима. Ово се огледа у прекидању напона од $500 \div 1000V$ за неколико μs у зависности од снаге самог фреквентног претварача. Већина статичких фреквентних претварача који се данас користе у индустрији за регулацију и управљање брзином трофазних електричних мотора су прављени на основу два принципа:

- фреквентни претварачи без међукола,
- фреквентни претварачи са међуколом.

Последица употребе електронских компоненти које раде на различитим учестаностима комутације генерише електромагнетне сметње, а поред наведеног протикање електричне струје кроз фреквентни претварач ствара електромагнетно поље око њега. Ово електромагнетно поље и све сметње које генеришу електронске компоненте могу да доведу до интеракције са другим уређајима у околини фреквентног претварача, па је потребно да се поведе о електромагнетној компатибилности фреквентног претварача у датом окружењу. На слици 2 приказано је ширење сметњи код фреквентног претварача.



Слика 2. Ширење сметњи код фреквентног претварача [4]

Електрична струја која ствара наведену сметњу на неки начин мора да се врати у свој извор из кога је потекла. Ако се користи оклопљени кабл за напајање мотора, а оклоп кабла мотора је повезан са кућиштем мотора, а други крај оплета кабла је повезан са кућиштем фреквентног претварача, у идеалном случају, највећи део струје која изазива сметње вратиће се у једносмерно међуколо из кога је и потекла, путем спреге са заједничком импедансом – земљом као референтном тачком, док ће део струје да се врати назад преко везе паразитних капацитивности. Део електричне струје сметње који се враћа назад у мрежу преко заједничке импедансе је непожељан и стога је потребна употреба филтера да би се утицај сметњи смањио да друге потрошаче у напојној мрежи.

4. ЕЛЕКТРОМАГНЕТНА КОПАТИБИЛНОСТ У ПРАКСИ

Уземљење представља јако битан аспект за неометан рад уређаја и индустријских процеса. Уземљење је битно и са аспекта безбедности јер у случају деградације електричне изолације уређаја, не дође до

могућности да метална маса било ког дела уређаја дође под напон. Безбедност има знатно већи приоритет него електромагнетна компатибилност. Под уземљењем фреквентних претварача подразумева се повезивање металног кућишта фреквентног претварача на референтни потенцијал земље, а поред кућишта и оплета тј. ширма свих каблова који иду ка електричном мотору са којим се управља. Веома је важно да се схвати да је потребно да се оба краја оклопљеног кабла прописно повежу на референтни потенцијал земље, јер ако је само један крај повезан на потенцијал земље, може доћи до проблема са електромагнетним сметњама и ширењем сметњи према слици 2.

5. ЗАКЉУЧАК

На основу свега наведеног види се да електромагнетна компатибилност представља битан аспект који је директно везан за неометан рад електричних уређаја а самим тим и доступност електричне енергије. Како се електрична опрема налази у разним окружењима у којима ради, јављају се разна ограничења и проблеми везани за њен рад. Да би се могло остварити неометан рад електричних уређаја у разним окружењима и доступност електричне енергије потребно је да се испуне одређени захтеви везани за квалитет производа.

Остварити оба ова циља подразумева да се разумеју појаве које се тичу извору сметњи, преноса сметњи са предајника на пријемник. Највећи део везан за електромагнетну компатибилност може се остварити већ при самом пројектовању електричних уређаја. Поред наведеног потребно је посветити и посебну пажњу карактеристикама окружења у коме ће се употребљавати електрични уређаји нпр. фреквентни претварачи.

6. ЛИТЕРАТУРА

- [1] EMC for Systems and Installations, Newnes, Tim Wil Hams & Keith Armstrong, 2000
- [3] Practical Implementation – EMC in Drive Engineering, SEW EURODRIVE, 2013
- [4] Facts Worth Knowing about Frequency Converters, Handbook, VLT® Frequency Converters, Danfoss, 2014

Кратка биографија:



Горан Мамлић рођен је у Новом Саду 1990. год. Дипломски рад на Факултету техничких наука из области Електротехнике и рачунарства - Енергетска електроника и електричне машине одбранио је 2017. год..

контакт: mamranlicgo@gmail.com

CONNECTOR ANDROID APLIKACIJA**CONNECTOR ANDROID APPLICATION**Darian Pudić, Predrag Teodorović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom članku predstavljena je društveno-poslovna softverska platforma Connector. Ova platforma omogućava brzo i jednostavno umrežavanje korisnika u cilju razmene informacija i podataka.

Ključne reči: Android, PHP, server, klijent

Abstract – Social-business software platform Connector is presented in this article. This platform provides quick and easy creation of network between users in order to exchange information and data.

Keywords: Android, PHP, server, client

1. UVOD

Kod većine društveno-poslovnih mreža današnjice akcentat je na tome da pojedinac sam pronađe ljude ili uslugu koji su mu na neki način bitni i od interesa. Da je to zaista tako, potrebno je samo prisetiti se tipičnog procesa zaposlenja, prilikom kojeg čovek i po nekoliko puta može doživeti neuspeh.

Usled navedenih činjenica, ukazala se potreba za razvojem aplikacije (nazvane *Connector* na osnovu njene osnovne svrhe, a to je povezivanje ljudi) koja će, na neki način samostalno, odnosno uz minimalnu asistenciju korisnika, pronaći za njega kontakte od interesa. Jednom uneti parametri pretrage primenjivaće se konstantno, odnosno sve vreme dok je telefon uključen. Usled pomenutih prednosti, *Connector* aplikacija predstavlja izvrstan alat za pronalaženje onoga što korisnik smatra bitnim.

Pronalaženje kontakata od interesa samo je jedan od čitavog niza mogućnosti koje aplikacija pruža. Ona nije fokusirana samo na traženje posla, već može i da posluži za sklapanje prijateljstava, zabavu, ostvarivanje profita, zadovoljstvo, uštedu u vremenu i novcu, promocije, itd.

**2. OSNOVNI TEORIJSKI KONCEPTI
CONNECTOR SISTEMA**

Connector mobilna aplikacija, čije su neke od primena opisane u uvodnom delu, čini samo klijentsku stranu *Connector* sistema. U nastavku će biti izneta nekolicina najbitnijih teorijskih koncepta na kojima sistem počiva.

2.1. Komunikacioni model

U pozadini celokupnog sistema krije se jedan od široko rasprostranjenih i veoma često primenjivanih modela komunikacije, a to je klijent-server komunikacija [1].

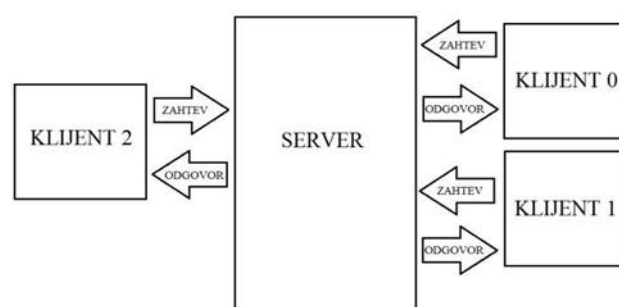
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof dr Predrag Teodorović.

U ovom modelu komunikacije, klijent je onaj uređaj koji zahteva resurs ili informaciju, dok je server onaj uređaj koji klijentu pruža zahtevano.

Proces pružanja zahtevanog resursa ili informacije naziva se servis ili servisiranje.

Grafički prikaz modela klijent-server komunikacije, a ujedno i celokupnog *Connector* sistema, prikazan je na slici 1.



Slika 1. Blok dijagram celokupnog sistema

Funkcionalnost servera opisana je nizom instrukcija koje su definisane PHP skriptom. Ova skripta izvršiće se svaki put kada se klijent obrati serveru.

2.2. O bazama podataka i SQL-u

Veoma bitan teorijski aspekt na kojem počiva rad servera jesu baze podataka. Baza podataka predstavlja strukturiran način za čuvanje i prikaz srodnih informacija na serveru, dok je SQL jezik za rad sa njima.

U slučaju *Connector* sistema, u bazi podataka biće skladišteni podaci o korisnicima sistema, dok će se pomoću SQL-a iz baze izvlačiti podaci o korisnicima od interesa. Prikupljanje podataka od interesa obavlja se prosleđivanjem SQL upita, koji predstavljaju sastavni deo PHP skripte servera.

Kao primer baze podataka, može se uzeti jedna veoma često upotrebljavana – telefonski imenik. To je baza čiji su podaci imena ljudi, njihovi brojevi telefona, kao i adrese na kojima žive.

2.3. Android platforma

Prema autorima [2], Android predstavlja najpopularniju mobilnu platformu na svetu. Ova moćna platforma sadrži sve ono što je neophodno ne bi li se efikasno i uspešno pisao softver za mobilne uređaje – aplikacije. Njen prikaz po slojevima (najniži su najbliže hardveru) dat je na slici ispod:



Slika 2. Apstrakcija softverskih slojeva Android platforme

Kao što se može uočiti sa slike 2, najviši nivo apstrakcije, odnosno sloj najbliži korisniku, zauzimaju aplikacije. Sve ono što se nalazi ispod njih služi programeru prilikom pisanja aplikacija i omogućava da programer u aplikaciju utisne željenu funkcionalnost. Android aplikacija sastoji se od dva glavna dela: opisa njenog izgleda i opisa njene funkcionalnosti. Za izgled, zadužen je XML, dok je funkcionalnost opisana Java programskim jezikom.

XML predstavlja jezik pomoću kojeg se vrši formatiranje podataka. Dakle, on ne predstavlja programski jezik, kao što su recimo C ili Java, jer njime ne može da se implementira neki algoritam.

Takođe, fajl napisan pomoću XML-a ne može da se kompajlira u cilju generisanja izvršnog fajla. Jedina uloga XML-a jeste definisanje seta pravila za označavanje podataka i on se u Android operativnom sistemu koristi za definisanje položaja, dimenzija, kao i rasporeda komponenti u okviru Android mobilne aplikacije.

Programski jezik pomoću kojeg se Android aplikaciji pruža funkcionalnost jeste Java. Java je objektno orijentisani programski jezik, odnosno programski jezik zasnovan na objektima i klasama.

Klasa je element koji enkapsulira svojstva i osobine neke pojave ili pojma iz ljudskog okruženja. Svojstva klase opisana su numerički ili slovno, u vidu polja klase. Polja klase su zapravo varijable ili konstante unutar tela klase.

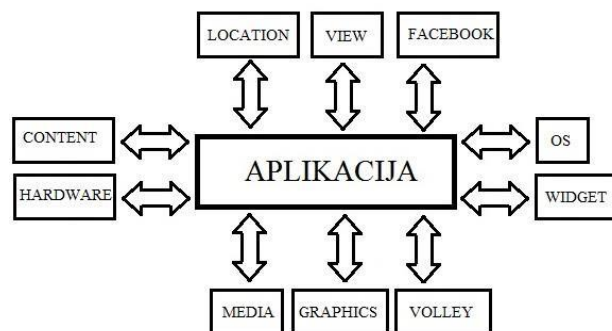
Funkcionalnosti koje se vezuju za klasu opisane su elementima koji se zovu metode. Primer Java klase dat je na slici ispod:



Slika 3. Grafička reprezentacija klase Vozilo

3. OPIS CONNECTOR APLIKACIJE NA VISOKOM NIVOU APSTRAKCIJE

Connector aplikacija predstavlja posrednika između servera i korisnika, te na taj način omogućava korisniku da se obrati serveru u cilju pronalaženja ljudi od interesa. Usled pomenute uloge, neophodna je pravilna i pravovremena upotreba softverskih komponenti koje Android operativni sistem nudi ne bi li olakšao korišćenje dostupnih hardverskih resursa. Connector aplikacija koristi mnoštvo softverskih biblioteka u cilju ostvarenja željene funkcionalnosti, te su samo najbitnije date na slici 4.



Slika 4. Softverske komponente neophodne za funkcionisanje aplikacije

Možda jedna od najneophodnijih softverskih komponenti jeste Widget komponenta. Bez nje, izgradnja čitavog interfejsa aplikacije bila bi nemoguća. Pod *widget*-ima, podrazumevaju se tasteri, prostor za prikaz teksta, slika, *layout*-i, itd. Takođe, u Widget softverskoj komponenti definisana je takozvana Toast poruka, koja služi da korisniku obezbedi povratnu informaciju nakon, recimo, pritiska tastera, a može da se koristi i prilikom debagovanja aplikacije. Isto tako, još jedna praktično neophodna komponenta za proces vizuelizacije jeste View komponenta, koja obezbeđuje prikaz svega pomenutog na ekranu telefona.

Za upravljanje hardverskim resursima, kao što je recimo kamera, na raspolaganju stoji biblioteka Hardware. Pomoću metoda koje su tamo definisane, dizajneru aplikacije otvaraju se mogućnosti korišćenja svih hardverskih resursa koje nudi mobilni uređaj. Softverska podrška za GPS kao hardverski resurs je posebno implementirana u vidu biblioteke Location, gde se nalazi najvažnija polazna tačka za upotrebu ovog resursa – interfejs *LocationListener*.

Što se tiče biblioteka koje omogućavaju rad na internetu, neophodne su Facebook i Volley. Volley biblioteka se koristi ne bi li se uspešno primili i poslali HTTP zahtevi serveru, te bez nje, zamišljena funkcionalnost *Connector* sistema nikako ne bi mogla biti ostvarena. Što se tiče Facebook biblioteke, ona se upotrebljava ne bi li se korisniku omogućilo da se pomoću *Connector* aplikacije uloguje na svoj Facebook nalog i praktično na taj način obezbedi link aplikaciji do svog naloga.

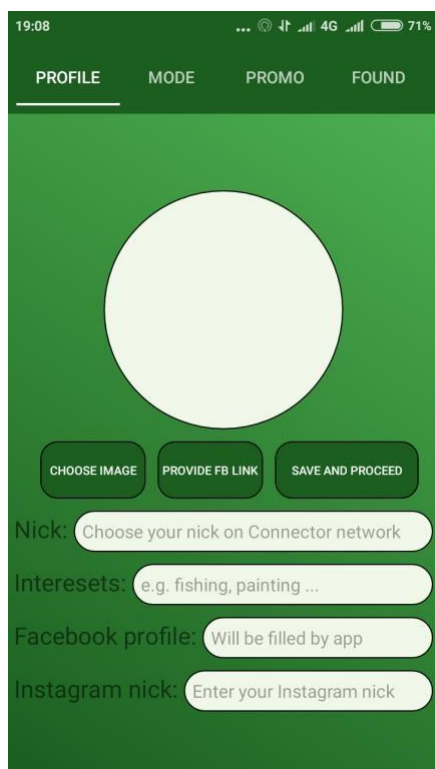
4. INTERFEJS APLIKACIJE

4.1. PROFILE tab

Prvi ili profilni tab (slika 5.), predstavlja prostor u koji će korisnik uneti i uskladištiti osnovne lične podatke. Svako od polja koje je neophodno popuniti sadrži pomoćni test u formi instrukcija, ne bi li korisnik što lakše shvatio šta, i u kojoj formi, treba da unese. Pritiskom na dugme CHOOSE IMAGE, korisnik će pokrenuti proces odabira i postavljanja profilne slike, koja se bira iz galerije. Nakon toga, korisnik bi trebao da u prvo prazno tekstualno polje unese svoj nadimak, odnosno *nick*, koji će koristiti na Connector mreži. Potom, neophodno je uneti lična interesovanja, stručnost ili hobije u polje Interests.

Naredni podatak koji aplikacija zahteva jeste link do korisnikovog Facebook naloga. Korisnik naravno ovo polje ne popunjava sam. Za tu svrhu, obezbeđeno je dugme PROVIDE FB LINK. Pritiskom na ovo dugme, na ekranu se pojavljuje interfejs koji zahteva od korisnika logovanje na Facebook nalog. Nakon logovanja, polje Facebook profile će biti automatski popunjeno.

Poslednje polje čije popunjavanje zahteva aplikacija jeste Instagram *nick*, gde korisnik manuelno unosi svoj Instagram nadimak. Nakon popunjenih podataka, korisnik treba da pritisne dugme SAVE AND PROCEED. Pritisak ovog dugmeta će aktivirati čuvanje unetih podataka i nakon toga proslediti korisnika na naredni tab – MODE.



Slika 5. Prikaz profilnog taba

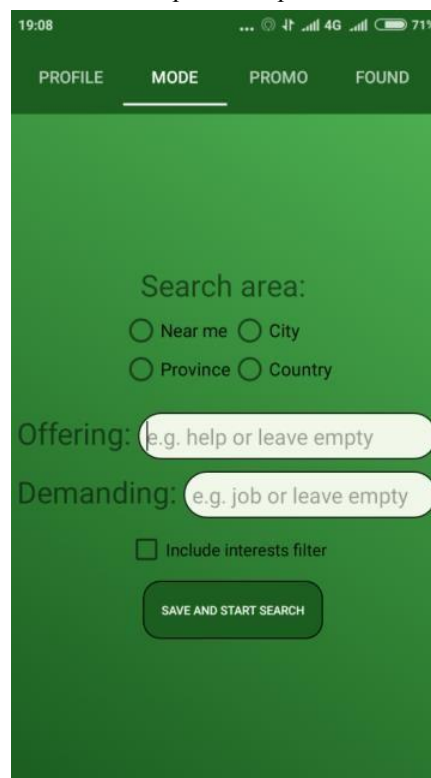
4.2. MODE tab

U ovom tabu (slika 6.) korisnik podešava kriterijume na osnovu kojih će aplikacija da traži kontakte od interesa. Prvi kriterijum koji treba podesiti jeste Search area, odnosno veličina oblasti na kojoj se potencijalno nalaze kontakti od interesa.

Nakon što je odabrao oblast od interesovanja, korisnik prelazi na popunjavanje Offering i Demanding tekstualnih polja. U ova polja korisnik treba unese ono što trenutno nudi (*offering*) ili potražuje (*demanding*).

Nakon Demanding i Offering polja, od korisnika se očekuje da saopšti aplikaciji da li pri pretrazi želi da se u razmatranje uzmu njegova prethodno navedena interesovanja, kao i interesovanja potencijalnih kontakata.

Nakon što je podesio kriterijume pretrage, korisnik treba da pritisne dugme SAVE AND START SEARCH, kojim će sačuvati unete podatke i poslati ih na server.

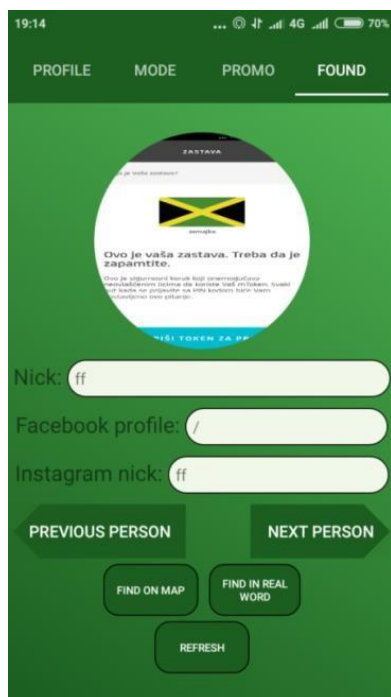


Slika 6. Prikaz MODE taba

4.3. FOUND tab

FOUND tab (slika 7.), služi da korisniku prikaže rezultate poslednje pretrage. Ne bi li se prikaz osvežio najnovijim rezultatima, neophodno je da korisnik pritisne dugme REFRESH. Kroz listu pronađenih osoba korisnik može da prolazi pritiscima na dugmad PREVIOUS PERSON i NEXT PERSON.

Takođe, korisniku je na raspolaganju i opcija lociranja pronađene osobe. Lociranje je implementirano pomoću dva dugmeta – FIND ON MAP i FIND IN REAL WORLD. Pritiskom na prvo dugme, korisnik će aktivirati Google Maps servis, koji će na mapi prikazati poslednju lokaciju pronađene osobe. Pritiskom na drugo dugme aktivira se kamera mobilnog uređaja. Kada korisnik uperi kameru na lokaciju na kojoj je poslednji put bila pronađena osoba, interfejs će aktivirati blinkanje markera, ne bi li korisniku dao do znanja da je pronađena osoba bila na označenom mestu. Ovo svojstvo poznato je pod nazivom „obogaćena stvarnost“.



Slika 7. Prikaz FOUND taba

4.4. PROMO tab

Poslednji, četvrti tab, koji postoji u interfejsu aplikacije jeste PROMO tab i prikazan je na slici 8. Sadržaj ovog taba osvežava se svakih 5 sekundi i prikazuje reklamni materijal kompanija ili pojedinaca koji su se pretplatili na tu uslugu. Pozicija PROMO taba nije slučajno odabrana – on je ciljano smešten između MODE i FOUND taba, ne bi li korisnik bio prinuđen da pređe preko njega, kada poželi da pogleda rezultate pretrage.



Slika 8. Prikaz PROMO taba

5. ZAKLJUČAK

Na osnovu teorijskih fundamenata iznetih na početku rada, osmišljen je inicijalni dizajn celokupnog sistema, kao i njegovo ponašanje na najvišem nivou apstrakcije. Potom, dizajn je tokom procesa implementacije pretočen u dve softverske komponente – serversku i klijentsku. Faza implementacije bila je ujedno i vremenski najzahtevnija faza čitavog projekta, jer je zahtevala veoma velik broj iteracija instalacije prototipa klijentske strane. U finalnim fazama projekta, implementacija je polako prerasla u testiranje i otkrivanje sitnijih grešaka koje su se provukle tokom faze dizajniranja i implementacije.

Uz sve pomenuto, definisano je i korisničko uputstvo za upotrebu *Connector* aplikacije, odnosno klijentske softverske komponente, kao i detaljan opis njegovog interfejsa.

Finalna faza, odnosno faza testiranja, pokazala se kao najmanje zahtevnom fazom celokupnog projekta. Razlog za to jeste činjenica da su većinom sve grube greške otklonjene za vreme faze implementacije, te su u ovoj fazi uočeni samo sitniji propusti, koji ujedno predstavljaju ideje vodilje za dalje faze usavršavanja sistema. Jedno od mogućih poboljšanja svakako jeste odsustvo logovanja korisnika prilikom pokretanja aplikacije.

6. LITERATURA

- [1] Top-Down Network Design, Third Edition, Priscilla Oppenheimer, 2010 Cisco Press
- [2] Head First Android Development, Dawn and David Griffiths 2015 O'Reilly Media

Kratka biografija:



Darian Pudić rođen je u Somboru 1994. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo – Embeded sistemi i algoritmi odbranio je 2017. godine. Oblasti interesovanja su mu kreiranje Android aplikacija, kao i razvoj softvera u programskim jezicima C, Java i PHP.
kontakt: darian.pudic@gmail.com

PROGRAMSKO REŠENJE ZA PRAĆENJE MREŽNOG SAOBRAĆAJA SCADA PROTOKOLA**SOFTWARE SOLUTION FOR MONITORING SCADA NETWORK TRAFFIC***Zvezdana Klašnić, Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je prezentovano programsko rešenje za praćenje mrežnog saobraćaja SCADA protokola. Implementirano programsko rešenje može se koristiti za nadgledanje mrežnog saobraćaja, analizu poruka SCADA protokola, za formiranje i prikazivanje konfiguracije mreže u vidu stabla na osnovu obrađenih poruka, kao i za otkrivanje problema u mreži.

Ključne reči: SCADA, Real-time praćenje, Obrada SCADA poruka, Konfiguracija mreže.

Abstract – The document presents a software solution for monitoring SCADA network traffic. The implemented software solution can be used to monitor network traffic, analyze SCADA protocol messages, to form and display network configuration based on processed messages and to troubleshoot network problems.

Keywords: SCADA, Real-time monitoring, SCADA message processing, Network configuration.

1. UVOD

Računarske mreže susreću se sa brojnim problemima, kao što su različiti napadi na mrežu i greške u konfiguraciji. Zadatak mrežnih administratora je da otkriju i otklone takve probleme kako bi održali mrežu bezbednom, pri čemu im može pomoći analiza paketa. Ona predstavlja proces snimanja i interpretacije podataka koji se razmenjuju kroz mrežu, kako bi se ustanovilo ima li neželjenih dešavanja u mreži. Postoji mnoštvo alata namenjenih za analizu mrežnog saobraćaja. Oni presreću pakete koji se prenose kroz mrežu, uzimaju sirove podatke iz paketa, pokušavaju da detektuju korišćeni protokol i analiziraju informacije koje se prenose. Dobar izbor alata može da unapredi mrežne performanse i da obezbedi sigurnost sistema.

Neki od često korišćenih alata su Wireshark [1], TCPdump [2], NetDecoder [3] i Capsa [4]. Zajedničko za sve ove alate je to što koriste dobro poznate i predefinisane portove za detektovanje korišćenog protokola, što predstavlja njihovu manu. Ne znaju sami da prepoznaju korišćeni aplikativni protokol, što je potrebno za parsiranje poruke, već je za svaki komunikacioni link potrebno ručno uneti kom protokolu pripada. Ova mana bi naročito došla do izražaja u slučaju da je potrebno pratiti hiljade adresa za više različitih protokola, jer bi tada bilo potrebno ručno uneti isto toliko komunikacionih linkova.

NAPOMENA:

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

Navedeni alati nisu eksplicitno namenjeni za praćenje mrežnog saobraćaja SCADA (*Supervisory Control And Data Acquisition*) protokola. Većina pokriva samo najpoznatije SCADA protokole (IEC 104, DNP3 i Modbus), dok bi sve ostale protokole bilo potrebno samostalno implementirati. Ukoliko za određenu mrežu nije poznato koje SCADA protokole koristi *remote unit*, bilo bi potrebno isprobavati protokole i detaljno ih analizirati, da bi se ustanovilo da li je odabran dobar protokol. Navedeni alati ne pokrivaju mogućnost uvida u konfiguraciju telemetrije u mreži.

Sa ciljem otklanjanja prethodno navedenih mana implementirano je programsko rešenje koje je tema ovog rada. Ovo rešenje može se koristiti za *real-time* praćenje mrežnog saobraćaja, analizu poruka SCADA protokola, otklanjanje problema u mreži, kao i za formiranje i prikazivanje konfiguracije mreže u vidu stabla na osnovu obrađenih poruka. Razlikuje se od prethodno navedenih alata po načinu prepoznavanja aplikativnog protokola i parsiranja uhvaćenog paketa. Zbog nepouzdanosti korišćenja predefinisanih portova za detekciju korišćenog protokola, implementirano rešenje u te svrhe koristi proveru preambule, parsiranje aplikativnog zaglavlja, validiranje pojedinih polja protokola, kao i *packet payload* informacija. Još jedna prednost je to što je na osnovu obrađenog mrežnog saobraćaja moguće dobiti hijerarhijski prikaz računarske mreže, gde su prikazani RTU-ovi sa svojim tačkama.

2. TEORIJSKE OSNOVE

SCADA paketi se kroz mrežu šalju preko TCP/IP (*Transmission Control Protocol/Internet Protocol*) infrastrukture. TCP/IP model u mrežnom sloju koristi *Internet Protocol* (IP). IP obezbeđuje fragmentaciju datagrama, ne garantuje prijem paketa, a provera greške je ograničena na proveru kontrolne sume zaglavlja.

Glavna funkcija transportnog sloja je da omogući komunikaciju između programskih aplikacija koje se nalaze na različitim računarima u mreži. TCP/IP model podržava dva transportna protokola:

- *Transmission Control Protocol* (TCP)
- *User Datagram Protocol* (UDP)

U radu je akcenat stavljen na TCP konekcije, jer je za prenos SCADA komunikacije bitno da podaci stignu kompletni i u tačnom redosledu.

Aplikacioni sloj direktno komunicira sa korisničkim aplikacijama i predstavlja njihov interfejs ka mreži. Svaka aplikacija na računaru identifikuje se pomoću porta. Za najčešće korišćene aplikacije uvedena je lista dobro

poznatih portova, za koje postoji dodela porta određenom aplikativnom protokolu. Korišćenjem dobro poznatih portova, detekcija protokola aplikacionog nivoa može biti neuspešna kada pojedini protokoli ne koriste te portove.

Na aplikacionom nivou postoji mnoštvo protokola, međutim biće pomenuti samo pojedini SCADA protokoli, koji su od značaja za ovaj rad. SCADA protokoli su industrijski aplikativni protokoli, pomoću kojih je omogućena komunikacija u SCADA sistemima. SCADA protokoli podržani implementiranim rešenjem su:

- Ferranti VAN-COMM [5]
- DNP3.0 [6]
- IEC 60870-5-101 [7]
- IEC 60870-5-104 [8]

3. IMPLEMENTIRANO PROGRAMSKO REŠENJE

Implementirano rešenje može se koristiti za *real-time* praćenje, analizu poruka SCADA protokola, za formiranje i prikazivanje mrežne konfiguracije u vidu stabla, kao i za otkrivanje problema u mreži. Mrežni saobraćaj se iz sirovog binarnog formata konvertuje u čoveku razumljiv format, čime se olakšava analiza. Obradene poruke SCADA protokola alat prikazuje u tabeli, gde su prikazani osnovni podaci.

Detaljan prikaz aplikativnog dela poruke prikazuje se u *data grid*-u kada se izabere željena poruka u tabeli. Ukoliko se u toku parsiranja poruke ustanovi da se ne radi o poruci SCADA protokola, ta poruka se odbacuje i alat je nigde ne prikazuje. Bitno svojstvo implementiranog rešenja je dobijanje informacija o konfiguraciji mreže tj. o RTU-ovima i njihovim tačkama, samo na osnovu obrađenih poruka. Pružena je i mogućnost filtriranja poruka po određenim parametrima, kao i sortiranje po kolonama, što olakšava analizu. Još jedno od svojstava implementiranog rešenja jeste algoritam, koji je osmišljen da bi iz tabele uklanjao poruke koje su pogrešno protumačene. Implementirano rešenje pruža i prikaz performansi u posebnom prozoru.

3.1. Arhitektura realizovanog rešenja

Osnovne komponente implementiranog rešenja su:

- NetworkListener,
- PacketListener i
- PacketAnalyzer.

NetworkListener komponenta predstavlja glavnu komponentu implementiranog rešenja. Ona pokreće i zaustavlja alat, instancira komponentu PacketListener, koja služi za hvatanje paketa iz mrežnog saobraćaja izabranog mrežnog interfejsa, a takođe instancira i komponentu PacketAnalyzer, pomoću koje se obrađuju uhvaćeni paketi. PacketAnalyzer komponenta instancira komponentu za parsiranje IP i TCP zaglavlja uhvaćenog paketa, parsere za pojedinačne protokole, komponentu za formiranje i prikazivanje mrežne konfiguracije, kao i komponentu za merenje performansi rešenja.

3.2. Implementacija realizovanog rešenja

Opis implementacije programskog rešenja biće podeljen u nekoliko poglavlja, kako bi se za svako svojstvo objasnilo na koji način je implementirano. Programsko rešenje implementirano je u C# programskom jeziku.

3.2.1. Hvatanje paketa

PacketListener komponenta za hvatanje paketa koristi *raw socket*, koji omogućava da se sirovi paketi primljeni na mrežnom sloju direktno prosleđuju *raw* soketima, čime se omogućava korisniku da pristupi i manipuliše zaglavlja i korisničkim podacima protokola nižih slojeva. Hvatanje paketa se vrši u asinhronom režimu sa mrežne kartice koja je izabrana.

3.2.2. Parsiranje paketa

Nakon što su podaci uhvaćeni, potrebno je isparsirati ih. Pošto su za hvatanje paketa korišćeni *raw* soketi, uhvaćeni paketi pored aplikativnog dela poruke mogu da sadrže i mrežno i transportno zaglavlje. Ukoliko paket ne sadrži IP, TCP i aplikativna zaglavlja, taj paket se može odbaciti jer sigurno ne sadrži SCADA aplikativni deo. Sledeći korak je pokušaj parsiranja protokola aplikacionog sloja tj. podržanih SCADA protokola.

3.2.2.1. Ferranti VAN-COMM

Prepoznavanje VAN-COMM poruke započinje proverom preambule. Ukoliko preambula ima vrednost 0xFE ili 0xFD pretpostavlja se da se radi o VAN-COMM poruci. Nakon toga proverava se format poruke, poruka se parsira i pojedina polja se validiraju, kako bi se potvrdilo da je zaista u pitanju VAN-COMM protokol. Jedan VAN-COMM frejm može nositi između 4 i 14 bajta korisničkih informacija, pa se iz tog razloga vrši provera dužine uhvaćenog frejma. Kako VAN-COMM odgovor u sebi ne sadrži originalni kod funkcije, kao ni tip poslatih podataka, bilo je potrebno da se pamti kakav je zahtev poslat na RTU da bi se odgovor na taj zahtev mogao tumačiti.

Odgovor se može sastojati iz više frejmova koje nije moguće tumačiti pojedinačno, pa je bilo potrebno sačekati sve frejmove odgovora, kako bi bilo moguće protumačiti ga u celosti. Početni deo frejma svih zahteva je isti, dok formati glavnog dela frejma sa komandnim parametrima mogu biti različiti, u zavisnosti od toga koji kod funkcije se nalazi u zahtevu.

Nakon što se na osnovu koda funkcije odredi vrsta zahteva, kreira se parser za tu vrstu zahteva i vrši se parsiranje. Na sličan način implementirani su i odgovori na zahteve. Ukoliko se poruka sastoji iz više frejmova, samo prvi frejm sadrži preambulu, adresu RTU-a i statusne bite. Format glavnog dela frejma odgovora određuje se na osnovu koda funkcije iz zapamćenog zahteva. Kreira se parser za određeni format odgovora i vrši se parsiranje poruke. Prilikom parsiranja vrednosti statusnih tačaka nije moguće znati pojedinačne vrednosti tačaka zbog nedostatka znanja o tipu grupe kojoj statusna tačka pripada. Iz tog razloga implementirano je da se isparsira vrednost cele grupe, a ne pojedinačnih statusnih tačaka.

Odgovor na zahtev za skeniranjem svih statusa ne sadrži informacije o tome koja vrsta statusa se nalazi u kojoj grupi, pa rešenje ne zna kako da tretira koju grupu, da li kao *memory*, *latching* ili *momentary* statusne tačke.

Kod parsiranja analognih vrednosti bez znanja o konfiguraciji nije moguće odrediti da li se radi o neoznačenoj ili označenoj vrednosti, pa implementirano rešenje tretira sve analogne vrednosti kao neoznačene.

Kod parsiranja *Raise Lower Immediate Execute* zahteva, vrednosti vremenskih intervala čuvaju se u vidu bajtova, jer bez poznavanja konfiguracije nije moguće tumačiti te vrednosti na osnovu protokolom definisane tabele. Kod

kontrolnih operacija (*Control Select*, *Control Execute* i *Control Immediate Execute*) implementirano rešenje samo čita koji su releji selektovani u poruci, ali zbog nedostatka znanja o konfiguraciji ne može da rastumači koja će od operacija (TRIP, CLOSE ili PULSE) biti izvršena.

3.2.2.2. DNP3.0

Da bi se pretpostavilo da je korišćeni aplikativni protokol DNP3, vrednosti prva dva bajta frejma moraju biti 0x05 i 0x64. Ukoliko je taj uslov ispunjen, proverava se format poruke, vrši se parsiranje i validiraju se polja poruke. DNP3 poruke mogu se sastojati iz više frejmova koje nije moguće tumačiti pojedinačno, pa je bilo potrebno sačekati sve frejmove kako bi se poruka protumačila u celosti. Pomoću pristiglog frejma kreira se segment, odstranjujući zaglavlje sloja veze, i vrši se njegovo parsiranje. Sledeći bitan korak je validacija sekvence segmenta, gde se na osnovu broja sekvence, koji se nalazi u zaglavlju segmenta, proverava da li su segmenti primljeni u dobrom redosledu.

Kada se primi poslednji frejm, spajanjem svih prethodno primljenih frejmova, odnosno segmenata, kreira se fragment i vrši se njegovo parsiranje. Parsiranje fragmenta podrazumeva parsiranje zaglavlja aplikacionog sloja, parsiranje zaglavlja objekata, kao i samih DNP3 objekata. Na osnovu grupe i varijacije dobijenih parsiranjem zaglavlja objekta, određuje se tip DNP3 objekata. Pomoću mapiranja se na osnovu tipa objekta dobija funkcija, pomoću koje se kreira DNP3 objekat. Zatim se vrši parsiranje koje može biti različito, u zavisnosti od toga o kom se DNP3 objektu radi.

3.2.2.3. IEC 101 i IEC 104

Prvi korak prilikom pretpostavke da je aplikativni protokol uhvaćenog paketa IEC 101 podrazumeva proveru preambule, gde su moguće vrednosti 0x68, 0x10 i 0xE5. Na osnovu vrednosti preambule kreira se određeni format frejma i vrši se njegovo parsiranje uz validaciju polja frejma. Značenje bita u kontrolnom polju frejma razlikuje se u zavisnosti od toga da li je komunikacija balansirana ili nije. Zbog nedostatke znanja o konfiguraciji, implementirano rešenje u detaljnom prikazu poruke bite iz kontrolnog polja prikazuje sa nazivima iz oba formata, tako da njihovo značenje nije jasno određeno.

Kod IEC 101 protokola postoje dva formata ASDU poruka. Format je određen poljem SQ. Ukoliko je 0 vrednost polja SQ, ASDU nosi pojedinačne informacione objekte, gde svaki objekat ima svoju koordinatu. U suprotnom, ASDU ima jedan informacioni objekat, koji u sebi nosi sekvencu informacionih elemenata, gde samo prvi element ima koordinatu, a ostale koordinate se računaju na osnovu prve. Na osnovu polja koje određuje tip podataka i vrednosti polja SQ, određuje se koja je vrsta informacionog objekta u pitanju, tačnije kreira se odgovarajući parser za taj objekat i njegove elemente. Parseru se dodaje i odgovarajući deskriptor kvaliteta, kao i vremenska oznaka ukoliko je to potrebno. Zatim se vrši parsiranje i validacija polja. Implementacija IEC 104 parsera je slična implementaciji IEC 101 parsera. Vrednost preambule za pretpostavku IEC 104 protokola mora biti 0x68. IEC 104 za prenos podataka koristi strukturu APDU, koja u sebi pored ASDU-a sadrži i APCI. Postoje tri tipa APCI-a, koja definišu tri APDU formata. I-format pored APCI-a sadrži

i ASDU, koji može imati dva formata. Ovaj problem je rešen na isti način kao što je prethodno opisano za IEC 101. Kod rešenja smešten je u *CommonIEC* projekat, koji koriste parseri oba protokola.

3.2.3. Čuvanje i prikazivanje isparsiranih paketa

Poruke svih podržanih protokola imaju različite formate, međutim programsko rešenje koristi klasu *ProtocolDiagnosticsChannelData* za čuvanje informacija o raznorodnim protokolima. Parser nakon parsiranja popunjava *property*-e prethodno pomenute klase. Isparsirani paket se preko *event*-a dodaje u listu uhvaćenih paketa na *ViewModel*-u, čiji se elementi prikazuju u tabeli.

3.2.4. Brisanje pogrešno protumačenih poruka

Da bi se unapredila analiza poruka osmišljen je pametni algoritam, koji uklanja poruke koje su pogrešno protumačene. Implementirano rešenje čuva informaciju o portovima na koje su stizale SCADA poruke i broj pristiglih poruka, kao i informaciju o ostalim portovima sa brojem neuspešnih pokušaja obrade na njima. Ukoliko se nakon pokušaja parsiranja zaključi da uhvaćena poruka ne pripada SCADA protokolima, poziva se metoda *CheckIsFailedTriesExceededLimit*.

Ova metoda proverava da li su na taj komunikacioni link prethodno stizale SCADA poruke. Ako jesu, proverava se da li je broj neuspešnih obrada poruka veći od 75% ukupnog broja pokušaja obrada na tom komunikacionom linku (uspešni + neuspešni). Ukoliko jeste, pomoću *event*-a se na *ViewModel*-u brišu iz tabele sve poruke koje sadrže tu IP adresu i port. Takođe se i iz konfiguracije briše komunikacioni link i RTU-ovi koji se nalaze pod njim.

3.2.5. Filtriranje uhvaćenih poruka

Programsko rešenje podržava filtriranje radi lakšeg praćenja mrežnog saobraćaja. Parametri po kojima se može filtrirati su imena kolona iz tabele, predstavljena skraćenim stringom, dok su operatori koji se mogu koristiti: `==`, `!=`, `!contains`, `contains`, `and` i `or`. Ispravna forma filtera je: `<ime kolone><operator><vrednost>`.

3.2.6. Prikaz mrežne konfiguracije

Programsko rešenje pravi mrežnu konfiguraciju na osnovu isparsiranih SCADA poruka. U konfiguraciji postoji podela po protokolima, a ispod svakog protokola čuvaju se komunikacioni linkovi. Pod jednim komunikacionim linkom može se nalaziti više RTU-ova sa ulaznim i izlaznim tačkama. Kada se izvrši parsiranje poruke, unapređuje se mrežna konfiguracija ukoliko je to potrebno tj. proverava se da li se protokol, komunikacioni link, RTU i njegove tačke već nalaze u konfiguraciji ili je potrebno dodati ih.

3.2.7. Prikaz performansi procesiranja

Da bi se omogućilo praćenje performansi, koristi se štoperica, koja startuje kada kreće obrada paketa, a zaustavlja se kada se obrada završi.

Na početku obrade uvećava se ukupan broj uhvaćenih paketa. Nakon obrade, ukoliko je uspešno isparsiran SCADA protokol, uvećava se broj SCADA paketa i ukupno vreme SCADA obrade.

U suprotnom se uvećava broj paketa koji nisu SCADA i ukupno vreme neuspešnih obrada. Pamti se i broj neuspešnih pokušaja obrade za svaki komunikacioni link. Prozor za performanse prikazuje sledeće vrednosti:

- ukupan broj svih uhvaćenih paketa,
- broj uhvaćenih SCADA paketa,
- broj ostalih paketa,
- procentualni prikaz uhvaćenih SCADA paketa u odnosu na ukupan broj svih uhvaćenih paketa,
- prosečna vremena obrade SCADA i ostalih paketa, i
- port sa najviše neuspešnih pokušaja obrade.

4. VERIFIKACIJA PROGRAMSKOG REŠENJA

Verifikacija implementiranog programskog rešenja vršena je testiranjem četiri slučaja, koji se razlikuju po broju obrađenih poruka u određenom vremenskom periodu. Postepeno je povećavano opterećenje mreže, odnosno broj SCADA poruka u mreži. U sva četiri slučaja merenje je vršeno 10 minuta nakon pokretanja alata. Tabela 1 prikazuje rezultate testiranja.

Tabela 1: Rezultati testiranja programskog rešenja

Broj rtu-a	890	970	1070	1670
Broj tačaka	18.360	34.360	54.352	174.352
Ukupan broj paketa	391.788	506.554	620.417	1.203.788
SCADA paketi	153.958	204.134	252.270	499.345
Ostali paketi	237.830	302.429	368.147	704.442
Procenat SCADA paketa	39,3%	40,3%	40,7%	41,5%
Prosečno vreme obrade SCADA paketa [μs]	161	199	212	306
Prosečno vreme obrade ostalih paketa [μs]	9	10	11	12
Procenat vremena utrošenog na obradu ostalih paketa	7,9%	7,2%	7,1%	5,2%

U prvom slučaju obrađivano je oko 256 SCADA poruka u sekundi, u drugom slučaju 340, u trećem slučaju 420, a u četvrtom 832 SCADA poruke u sekundi. Sa porastom broja SCADA poruka raste i procenat SCADA poruka u odnosu na ukupan broj uhvaćenih poruka. Može se primetiti i da prosečno vreme obrade SCADA paketa postepeno raste, dok se prosečno vreme obrade ostalih paketa povećava za po jednu mikrosekundu. Procenat vremena utrošenog na obradu ostalih paketa smanjuje se sa porastom uspešnih SCADA obrada.

Ovim testiranjem verifikuje se da implementirano programsko rešenje uspešno uspeva da prepozna SCADA protokole, kao i da obradi i isparsira uhvaćene SCADA pakete, bez potrebe za prethodnim mapiranjem portova i protokola, što je i bio cilj ovog rada.

5. ZAKLJUČAK

Analiza mrežnog saobraćaja predstavlja bitnu stavku prilikom održavanja svake mreže. Ukoliko mrežni administratori primete nepravilnosti u radu mreže, kroz analizu snimljenog saobraćaja može da se otkrije uzrok problema. Cilj ovog rada bio je razvoj programskog rešenja, čiji je zadatak da prati mrežni saobraćaj SCADA protokola i da obrađuje uhvaćene pakete, kako bi se omogućila detaljna analiza mrežnog saobraćaja.

Većina postojećih alata za praćenje mrežnog saobraćaja za detekciju protokola koristi predefinisane portove mapirane na podržane protokole. To nije pouzdana tehnika, a i može oduzeti puno vremena, zbog potrebe za ručnim mapiranjem portova i protokola. Zato implementirano rešenje prepoznaje protokole na osnovu parsiranja poruke i validiranja pojedinih polja protokola. Time je omogućeno praćenje mrežnog saobraćaja bez potrebe za prethodnim poznavanjem mrežne konfiguracije i za mapiranjem portova i protokola.

Prednost u odnosu na većinu postojećih alata jeste i mogućnost dobijanja mrežne konfiguracije na osnovu obrađenih poruka SCADA mrežnog saobraćaja, što je značajno korisniku rešenja ukoliko se radi o njemu prethodno nepoznatom SCADA sistemu. Za razliku od postojećih alata, programsko rešenje eksplicitno je namenjeno za praćenje mrežnog saobraćaja SCADA protokola.

Jedan od mogućih pravaca daljeg razvoja jeste dodavanje parsera za brojne SCADA protokole koji nisu trenutno obuhvaćeni implementiranim rešenjem. Bilo bi korisno unaprediti logiku za detekciju protokola, pamćenjem koji je protokol detektovan na kom portu, da bi se sledeći put kad se uoči poznati port pokušalo prvo sa parsiranjem zapamćenog protokola, kao i pamćenjem neuspešnih detekcija protokola za svaki port, da se sledeći put ne bi gubilo vreme na neuspešne pokušaje. Takođe je poželjno i dodavanje veštačke inteligencije za detekciju protokola.

6. LITERATURA

- [1] Wireshark. <http://www.wireshark.org/> (Pristupljeno 23.07.2019.)
- [2] All about TCPdump. <http://www.tcpdump.org/> (Pristupljeno 23.07.2019.)
- [3] NetDecoder. <http://www.fte.com/docs/usermanuals/-NetDecoderAsyncUM.pdf> (Pristupljeno 23.07.2019.)
- [4] All about Colasoft Capsa. <http://www.colasoft.com> (Pristupljeno 23.07.2019.)
- [5] Appendix B, Ferranti International Controls, *Message Standard: Van-Comm*
- [6] DNP3 SPECIFICATION, Version 2.01, 2007
- [7] G. Clarke, D.Reynders, E. Wright, *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*, Elsevier, 2004.
- [8] P.Matoušek, *Description and analysis of IEC 104 Protocol*, Brno University of Technology, 2017.

Kratka biografija:

Zvezdana Klašnić rođena je u Novom Sadu 1994. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva, smer Primenjeno softversko inženjerstvo, odbranila je 2019. godine.

Kontakt: zklasnic94@gmail.com

РЕАКТИВНО ПРОГРАМИРАЊЕ У ПРОГРАМСКОМ ЈЕЗИКУ JAVA REACTIVE PROGRAMMING IN JAVA PROGRAMMING LANGUAGE

Зоран Лулеција, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У раду је дат преглед парадигме реактивног програмирања у програмском језику Јава, као и анализа одзивности веб апликација развијених на стандардан и реактиван начин. Анализа одзивности је обухватила утицај два фактора, број истовремених HTTP захтева упућених према веб апликацијама, као и трајање обраде захтева код позиваног веб сервиса.

Кључне речи: Јава језик, реактивно програмирање, Project Reactor, анализа одзивности веб апликација

Abstract – This paper gives an overview of the reactive programming paradigm in Java, as well as an analysis of the responsiveness of web applications developed in a standard and reactive manner. Response analysis included the impact of two factors, the number of concurrent HTTP requests sent to web applications, as well as the duration of request processing at the called web service.

Keywords: Java language, reactive programming, Project Reactor, web application responsiveness analysis

1. УВОД

Задатак овог рада јесте преглед парадигме реактивног програмирања у програмском језику Јава, као и анализа одзивности веб апликација развијених на стандардан и реактиван начин. Анализа одзивности обухватила је утицај два фактора, број истовремених HTTP (Hypertext Transfer Protocol) захтева упућених према веб апликацијама, као и трајање обраде захтева код позиваног веб сервиса. Овом анализом извршена је провера утицаја различитих имплементација веб сервера и веб апликација на одзивност истих, у случајевима где постоји потреба за обрадом великог броја паралелних HTTP захтева, и где се у току обраде захтева позивају други веб сервиси.

2. РЕАКТИВНО ПРОГРАМИРАЊЕ

Реактивно програмирање представља парадигму програмирања базирану на токовима података и пропагацији измена [1]. Представља алтернативу парадигми императивног програмирања и решава нека од њених ограничења [2].

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Александар Купусинац, ванр. проф.

Према [3, 4], иницијална идеја настала је приликом развоја реактивне екстензије *ReactiveX* за .NET у компанији *Microsoft*, а креатором исте сматра се Ерик Мејер (енг. *Eric Meier*). Компанија *Netflix* препознала је бенефите исте и допринела развоју реактивне екстензије за Јава програмски језик – *RxJava*. Нешто касније, уследила је стандардизација за програмски језик Јава од стране *Reactive Streams* иницијативе. Овом стандардизацијом дефинисана је спецификација, то јест, скуп интерфејса и правила интеракције за реактивне библиотеке које се извршавају унутар Јава виртуелне машине. Од верзије Јава 9, ови интерфејси су интегрисани директно у Јава програмски језик.

Тренутно, постоји више библиотека које омогућују реактивно програмирање у програмском језику Јава. Према [1, 3, 4], неке од познатијих су *RxJava*, *Project Reactor* и *Akka Streams*. У овом раду коришћена је *Project Reactor* библиотека.

Такође, треба напоменути разлику између реактивних система и реактивног програмирања. Реактивни системи представљају архитектонски дизајн коришћен за изградњу одзивних, робусних и дистрибуираних система базираних на асинхроној размени порука. Реактивно програмирање представља развојни модел базиран на опсервацији токова података, реаговању на промене и њиховој пропагацији [5].

3. АНАЛИЗА ОДЗИВНОСТИ

У овом поглављу представљен је стандардан и реактиван начин развоја веб апликација у програмском језику Јава. Извршена је анализа одзивности веб апликација развијених на ова два начина, и представљени су резултати исте. Анализа одзивности обухватила је утицај два фактора:

- Број истовремено упућених HTTP захтева према тестираним веб апликацијама
- Дужина обраде захтева код позиваног веб сервиса

3.1. Стандардан начин развоја веб апликација

Под појмом стандардан начин развоја веб апликација, мисли се на употребу оквира (енг. *framework*) базираних на Јава сервлетима (енг. *Java servlets*) и *Thread pool* -у. Овај начин развоја веб апликација у програмском језику Јава је већ дуго присутан, а може се рећи и најчешће коришћен. Један од познатијих оквира из овог скупа у програмском језику Јава је *Spring MVC* [2].

Према [2], обрада захтева код стандардно развијених веб апликација започиње позајмљивањем једне нити из скупа доступних нити, то јест *Thread pool* -а.

Позајмљена нит је блокирана за време обраде захтева. По завршеној обради захтева, позајмљена нит се враћа у скуп из којег је узета и може се поново користити.

Последица оваквог начина обраде захтева јесте неефикасно коришћење ресурса и не баш најбоља могућност скалирања веб апликација базираних на овим оквирима.

Неефикасно коришћење ресурса односи се на стање блокираности нити у случајевима кашњења насталих током обраде захтева. Ова кашњења настају услед позива упућених према базама података, веб сервисима, раду са фајловима, и слично. Коначан број нити у *Thread pool* -у и њихова неефикасна употреба, отежава скалирање ових веб апликација у случајевима када је то потребно.

3.2. Реактиван начин развоја веб апликација

Под појмом реактиван начин развоја веб апликација, мисли се на употребу неблокирајућих асинхроних оквира базираних на *Event loop* -у. *Spring WebFlux* представља један такав оквир, и доступан је почев од *Spring 5* верзије [2].

Према [2], неблокирајући асинхрони оквири постижу већу скалабилност уз мање нити. У најчешћем случају, број нити је једнак броју процесорских језгара. Применом *Event loop* -а, ови оквири су у могућности да обраде велики број захтева уз ефикасно трошење ресурса. *Event loop* обрађује све као догађај (енг. *event*), како захтеве корисника, тако и повратне позиве из интензивних операција (позиви према базама података, веб сервисима, и слично).

Извршавање једне интензивне операције започиње њеним иницирањем и регистрањем повратног позива (енг. *callback*) који се активира по завршетку дате операције. За време извршавања ове интензивне операције, *Event loop* паралелно наставља обраду других догађаја.

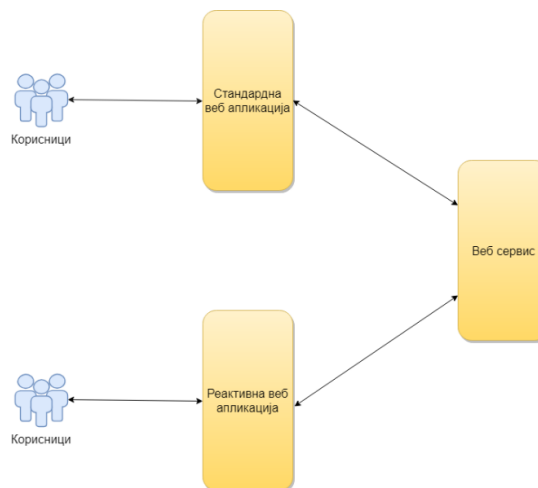
3.3. Тест сценарио

Генерисања *HTML* (*Hypertext Markup Language*) садржаја са прибављеним подацима са других веб сервиса, представља сценарио који се данас може наћи у већини веб апликација.

Тест сценарио коришћен у овој анализи одзивности је само поједностављена верзија оног који можемо видети код веб апликација у реалној употреби.

Дијаграм комуникације коришћен у тест сценарију приказан је на слици 1. Приликом обраде захтева, тестиране веб апликације генеришу случајан број и прослеђују га као параметар позиваном веб сервису. Веб сервис са одређеним кашњењем враћа *JSON* (*JavaScript Object Notation*) репрезентацију тестног садржаја.

Одговор веб сервиса се даље користи у генерисању *HTML* садржаја који се враћа позиваоцима.



Слика 1. Дијаграм комуникације коришћен у тест сценарију

Симулација корисника је изведена употребом *ab*, алата развијеног од стране *Apache* организације, намењеног тестирању *HTTP* сервера и апликација које се налазе на њима. Поменуте веб апликације и веб сервис су развијени помоћу *Spring Boot* -а, пројекта базираних на *Spring* оквиру, који има за циљ бржи и једноставнији начин за постављање, конфигурирање и покретање једноставних веб апликација у програмском језику Јава. Реактивна веб апликација и веб сервис базирани су на *spring-boot-starter-webflux* артефакту, док је стандардна веб апликација базирана на *spring-boot-starter-web* артефакту.

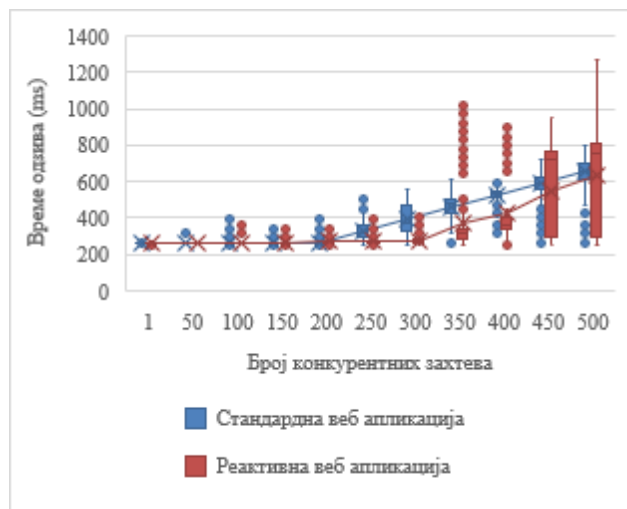
Приликом сваког теста оптерећења *HTTP* захтевима, извршено је 10000 позива према тестираној веб апликацији. Такође, наведени број позива у сваком тесту је распоређен на одређени број симулираних корисника. Број симулираних корисника представља начин на који ће тест бити извршен, то јест вредност која дефинише број позива који се извршавају паралелно. Вредности коришћене за симулацију броја корисника у овом тест сценарију се крећу од 1 до 500, с тим да је прва вредност 1, друга 50, и свака следећа је са кораком 50. Поред ових 11 вредности којима је симулиран одређен број корисника, тестови су узели у разматрање и дужину обраде захтева код позиваног веб сервиса у тест сценарију. Вредности којима су симулиране дужине обраде захтева су 250, 500 и 750 милисекунди.

Анализа одзивности извршена је на рачунару са *Intel Core i5-3210M* процесором, *8GB DDR3* меморије и *500GB HDD* диском. Сваки од тестова из ове анализе одзивности покретан је независно од других тестова.

3.4. Резултати

На слици 2 приказан је график са резултатима тестираних веб апликација за случај када позивани веб сервис има кашњење од 250 милисекунди. Са графика се може приметити пораст просечне дужине обраде захтева код стандардно развијене веб апликације, када је број симулираних корисника који истовремено приступају тестираној страници изнад

200. Такође, видљив је и нешто мањи пораст просечне дужине обраде захтева код реактивно развијене веб апликације, у случајевима где је број конкурентних захтева већи од 300.



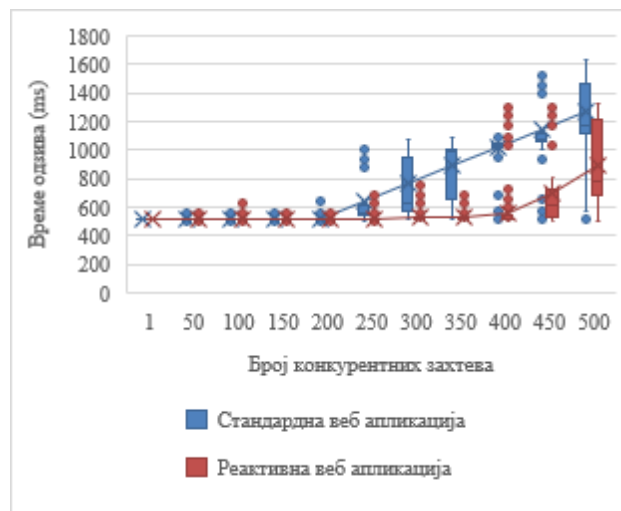
Слика 2. Анализа одзивности стандардне и реактивне веб апликације под различитим оптерећењима у случају када позивани веб сервис има кашњење од 250 милисекунди

Просечно време одзива код стандардно развијене веб апликације, за случај са 500 конкурентних захтева, са кашњењем од 250 милисекунди код позиваног веб сервиса је 653,36 милисекунди, што је за 387,34 милисекунде дужи од просечног времена одзива када је број конкурентних захтева 1. Просечно време одзива код реактивно развијене веб апликације, за случај са 500 конкурентних захтева, са истим кашњењем код позиваног веб сервиса, износи 632,88 милисекунди, што је за 366,98 милисекунди дужи од просечног времена одзива када је број конкурентних захтева 1.

Слика 3 представља график са резултатима тестирања, за случај када позивани веб сервис има кашњење од 500 милисекунди. На овом графику такође можемо приметити пораст просечне дужине обраде захтева код стандардно развијене веб апликације, у случајевима када је број симулираних корисника који истовремено приступају тестираној страници већи од 200. Такође, и реактивно развијена веб апликација има мањи пораст просечне дужине обраде захтева у случајевима где је број симулираних корисника изнад 400. Просечно време одзива код стандардно развијене веб апликације, са 500 конкурентних захтева, са кашњењем од 500 милисекунди код позиваног веб сервиса је 1271,31 милисекунда.

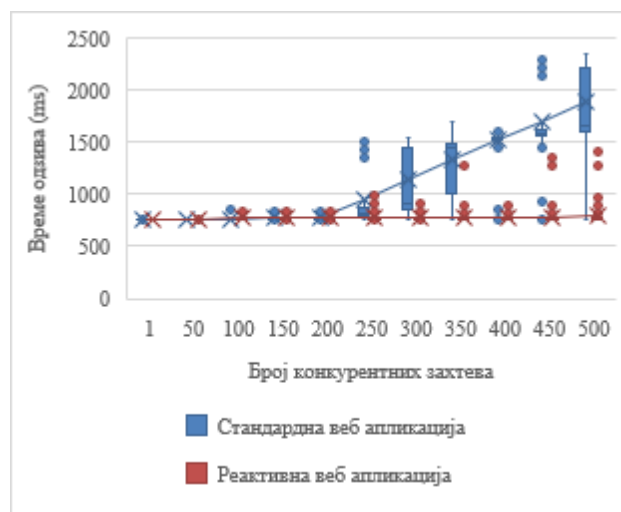
Просечно време одзива исте веб апликације, са истим кашњењем код позиваног веб сервиса и са једним конкурентним захтевом је 516 милисекунди, што је мање за 755,31 милисекунду од претходног случаја. Реактивно развијеној веб апликацији, за случај са 500 конкурентних захтева, са кашњењем од 500 милисекунди код позиваног веб сервиса, у просеку је

неопходно 891,68 милисекунди за обраду захтева. Иста ова веб апликација, за исто кашњење код позиваног веб сервиса, са једним конкурентним захтевом, треба у просеку 516 милисекунди за обраду захтева, што је 375,65 милисекунди мање од случаја са 500 конкурентних захтева.



Слика 3. Анализа одзивности стандардне и реактивне веб апликације под различитим оптерећењима у случају када позивани веб сервис има кашњење од 500 милисекунди

На слици 4 представљени су резултати тестирања поменутих веб апликација, за случај када је кашњење код позиваног веб сервиса 750 милисекунди. И на овом графику је видљив идентичан образац пораста дужине обраде захтева у случајевима веће оптерећености конкурентним захтевима.



Слика 4. Анализа одзивности стандардне и реактивне веб апликације под различитим оптерећењима у случају када позивани веб сервис има кашњење од 750 милисекунди

Просечно време одзива код стандардно развијене веб апликације, са 500 конкурентних захтева, са кашњењем од 750 милисекунди код позиваног веб сервиса износи 1884,11 милисекунди, што је за 1118,2 милисекунде дуже од просечног времена одзива када је број конкурентних захтева 1. Просечно време одзива код реактивно развијене веб апликације, у случају 500 конкурентних захтева, са кашњењем од 750 милисекунди код позиваног веб сервиса износи 786,53 милисекунде, што је за 20,53 милисекунде дуже од просечног времена одзива када је број конкурентних захтева 1.

Анализом добијених резултата у овом тест сценарију, можемо приметити да повећањем броја паралелно упућених *HTTP* захтева према тестираним веб апликацијама, у случајевима где постоје позиви ка другим веб сервисима у току обраде захтева, реактивно развијена веб апликација даје бољу одзивност. То јест, реактивно развијена веб апликација има мања одступања у одзивности, за разлику од стандардно развијене веб апликације. Такође, дужина трајања обраде захтева код позиваних веб сервиса има мањи утицај на одзивност код реактивно развијених веб апликација, него што је то случај са стандардно развијеним веб апликацијама. Треба напоменути, резултати добијени овом анализом су извршени на веб апликацијама и веб сервису са иницијалним *Spring Boot* конфигурацијама, и могли би варирати у одређеној мери да се исте промене. Међутим, и у другачијим околностима, ефикаснија употреба ресурса би била на страни реактивно развијене веб апликације.

4. ЗАКЉУЧАК

Горенаведеном анализом одзивности, потврђена је чињеница да реактивно развијене веб апликације имају конзистентнију одзивност у односу на стандардно развијене веб апликације, у случајевима веће оптерећености *HTTP* захтевима, као и код дужих обрада позива упућених према другим веб сервисима. Треба напоменути, реактиван начин развоја веб апликација не представља замену за стандардан начин развоја веб апликација. Он само решава одређени скуп проблема, који стандардан начин развоја веб апликација није могао да реши због ограничења у својој имплементацији.

Недостатак парадигме реактивног програмирања у програмском језику Јава представља другачији приступ решавању проблема. Овај приступ захтева учење одређених концепата, што је најчешће праћено стрмом кривом учења (енг. *steep learning curve*). Међутим, бенефити који се остварују применом поменуте парадигме, свакако надилазе наведени недостатак.

5. ЛИТЕРАТУРА

- [1] M. Bernhardt, *Reactive Web Applications: Covers Play, Akka, and Reactive Streams*, Manning Publications Co., 2016.
- [2] C. Walls, *Spring in Action, 5th Edition*, Manning Publications Co., 2019.
- [3] T. Nurkiewicz and C. Ben, *Reactive Programming with RxJava: Creating Asynchronous, Event-based Applications*, O'Reilly Media, Inc., 2016.
- [4] Project Reactor - <https://projectreactor.io/docs/core/release/reference/> (приступљено 27.08.2019)
- [5] C. Escoffier, *Building Reactive Microservices in Java*, O'Reilly Media, Inc., 2017.

Кратка биографија:



Зоран Лулеџија рођен је 1987. године у Сарајеву, Република Босна и Херцеговина. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Примењене рачунарске науке и информатика одбранио је 2019. године.

контакт: zoran.luledzija@outlook.com

ANALIZA BEZBEDNOSNIH RIZIKA I MERA ZAŠTITE VEB APLIKACIJA SECURITY RISK ANALYSIS AND WEB APPLICATION SECURITY MEASURES

Bojana Samardžić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je opisan problem bezbednosti veb orijentisanih informacionih sistema. Radi demonstracije načina na koje je moguće ugraditi sigurnosne mehanizme u softversko rešenje, razvijena je aplikacija za kupovinu i prodaju dostupnih artikala iz ponude. Realizovano rešenje oslanja se na radni okvir Spring i MySQL relacionu bazu podataka.

Ključne reči: aplikacija, baza podataka, bezbednost, maliciozni napadač, MySQL, napad, OWASP, pouzdanost, ranjivost, sigurnost, Spring

Abstract – The paper describes the security problem of web-oriented information systems. To demonstrate how security mechanisms can be incorporated into a software solution, an application has been developed to buy and sell available items from the offer. The implemented solution relies on the Spring framework and the MySQL relational database.

Keywords: application, database, security, malicious attacker, MySQL, attack, OWASP, reliability, vulnerability, safeness, Spring

1. UVOD

U domenu informacionih tehnologija najveću vrednost imaju informacije. Vremenom se razvila grana koja je otišla u smeru krađe podataka, podmetanja virusa, prisluškivanja komunikacije između klijentske i serverske strane, ubrizgavanja malicionih skripti koje se izvršavaju na pretraživaču žrtve, kreiranja sadržaja koji može zbuniti interpreter baze podataka i izvršiti neželjenu transakciju, krađe korisničkih kredencijala, izmene prava pristupa, krađe kriptografskih ključeva i sl.

Veb aplikacije su se inkorporirale u veći deo mreže savremenog društva. U značajnoj meri su uticale na promenu dosadašnjih zahteva i očekivanja korisnika. Kako ova vrsta aplikacija radi u realnom vremenu, čim se desi neka izmena, ona postaje vidljiva i utiče na dalji rad sistema.

Cilj rada je upoznavanje osnovnih bezbednosnih koncepata i implementacija istih na primeru aplikacije za kupovinu i prodaju dostupnih artikala iz ponude.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinać, vanr. prof.

2. BEZBEDNOST U INFORMACIONIM SISTEMIMA

Bezbednost informacionih sistema se primarno fokusira na zaštitu računara, mreža i njihovih korisnika. Digitalne pretnje dolaze u svim oblicima i veličinama: krađa privatnih podataka nakon upada u bazu podataka, instalacija zlonamernih softvera na mašini, namerno izazivanje prekida u radu servisa i druge [1].

2.1. Terminologija pretnji

Informaciona sigurnost odnosi se na sprečavanje ili makar smanjenje verovatnoće neovlašćenog pristupa, upotrebe, otkrivanja, ometanja, brisanja/uništavanja, korupcije, modifikacije i devalvacije informacija, a podrazumeva i smanjenje štetnih posledica incidenata. Upravljanje informacionom sigurnošću je proces detaljnog analiziranja i definisanja bezbednosnih kontrola u cilju zaštite informacionih sredstava.

2.1.1. Pretnja i rizik

Sa stanovišta računarske sigurnosti pod pojmom pretnja podrazumeva se potencijalna opasnost koja može iskoristiti ranjivost kako bi narušila bezbednost i samim tim prouzrokovala štetu. Rizik se objašnjava kao gubitak poverljivosti, integriteta i/ili dostupnosti informacija.

2.1.2. Ranjivost

Ranjivost se definiše kao osobina koju poseduje sistem u celosti ili neka njegova komponenta, a koja ostavlja prostor za zloupotrebu.

2.1.3. Resurs

Resursi su objekti od značaja – ako oni nisu raspoloživi ili ako su kompromitovani, sistem neće moći da funkcioniše ispravno.

2.1.4. Subjekat

Termin subjekat (u literaturi se sreće i pojam agent) označava entitet koji ima trenutno aktivnu ulogu u sistemu i može da obavlja akcije koje su mu stavljene na raspolaganje, tj. za koje ima ovlašćenja.

2.1.5. Napad

Realizacija pretnje. Napad je akcija koja se sprovodi sa ciljem da iskoristi detektovane ranjivosti sistema ne bi li se napadač domogao informacija koje može zloupotrebiti. Realizuje se vektorom napada (putanja ili način na koji se ostvaruje cilj).

2.1.6. Napadač

Čovek ili grupa ljudi koji nameravaju da ugroze resurse odabranog preduzeća.

2.1.7. Protivmere

Cilj definisanja bezbednosnih kontrola jeste svođenje eksploatacije ranjivosti sistema na minimum. Protivmere koje se uglavnom sprovode nad veb aplikacijama su heširanje lozinki, kriptovanje podataka u skladištu, upotreba sertifikata izdatih od strane poverljivih sertifikacionih tela, vođenje evidencije o aktivnostima koje se dešavaju u sistemu i druge.

2.2. Trijada informacione sigurnosti – CIA

U središtu sigurnosti informacija nalazi se *CIA* trijada: poverljivost, integritet i dostupnost. Rizici, pretnje i ranjivosti sa kojima se suočavaju softverski sistemi mere se na osnovu njihove potencijalne sposobnosti da kompromituju jedan ili više elemenata trijade.

2.2.1. Poverljivost

Koncept poverljivosti odnosi se na zaštitu podataka od subjekata koji nemaju ovlašćenje da im pristupaju, čime se osigurava to da se potreban nivo tajnosti primenjuje na svim mestima na kojima se vrši spajanje i obrada informacija.

2.2.2. Integritet

Pod pojmom integritet podrazumeva se obezbeđivanje tačnosti i pouzdanosti informacija i sistema, kao i sprečavanje neovlašćenih izmena.

2.2.3. Dostupnost

Termin dostupnost znači da informacije može pregledati i modifikovati svaki entitet koji za to ima dozvolu u odgovarajućem vremenskom roku.

2.3. Osnovni sigurnosni mehanizmi

Baza na kojoj je razvijena politika informacione sigurnosti sačinjena je od tri komponente: mehanizma identifikacije, autentifikacije i autorizacije. Podjednako važan bezbednosni mehanizam je i neporecivost.

2.3.1. Identifikacija

Identifikacija je čin predstavljanja sistemu, tj. tvrdnja ko je neko ili šta.

2.3.2. Autentifikacija

Autentifikacija je postupak utvrđivanja verodostojnosti tvrdnje da je subjekat to za šta se predstavlja.

2.3.3. Autorizacija

Autorizacija se objašnjava kao postupak utvrđivanja i proveravanja koja se ovlašćenja trebaju vezati za kog korisnika. Obavezno joj prethodi uspešna autentifikacija.

2.3.4. Neporecivost

Neporecivost znači da subjekat ne može da porekne da je preduzeo korake koji su doveli do izvršenja jedne ili niza akcija.

3. OWASP TOP 10 NAJKRITIČNIJIH SIGURNOSNIH RIZIKA VEB APLIKACIJA

3.1. OWASP zajednica

Politika *OWASP* zajednice (eng. *Open Web Application Security Project – OWASP*) usmerena je na identifikaciju

pretnji i detekciju ranjivosti koje uslovljavaju njihovu pojavu, na uspostavu bezbednosnih praksi koje su se pokazale kao najdelotvornije, a nudi i upozorenja, ideje, savete, alate i procedure za izgradnju kvalitetnog rešenja.

3.1.1. Sigurnost veb aplikacija

OWASP organizacija kreirala je listu 10 najučestalijih napada na veb aplikacije, koja se periodično ažurira kako bi mogla odgovoriti konstantnoj pojavi novih pretnji [2].

3.1.1.1. Injection

Ideja napada je da eksploatiše interpreter komandi, odnosno parser. Onaj ko sprovodi napad dobija pristup tokovima podataka za koje nema pravo, a ostavlja se mogućnost i da napadač stvori potpuno novi tok podataka koji nije predviđen dizajnom. Najefikasniji način odbrane jeste validacija svih ulaznih podataka.

3.1.1.2. Broken Authentication

Meta ovog napada je autentifikaciona logika. Može se realizovati ukoliko se informacijama vezanim za sesiju upravlja na neadekvatan način usled čega je ugrožena korisnička identifikacija. Sprečavanje ove klase napada vrši se implementacijom multifaktorske autentifikacije.

3.1.1.3. Sensitive Data Exposure

Nastaje kao posledica toga što informacije namenjene isključivo ovlašćenim osobama bivaju slučajno otkrivene neautorizovanim korisnicima u nekritovanom, slabo zaštićenom ili nezaštićenom okruženju. Dobra praksa za izbegavanje napada iz ove grupe jeste identifikacija osetljivih podataka, nakon čega sledi realizacija adekvatnih bezbednosnih mehanizama zaštite.

3.1.1.4. XML External Entities

Servisno orijentisani sistemi osetljivi su na klasu napada koji eksploatišu postupak parsiranja *XML* šeme. Napad je omogućen ukoliko je obrada *XML* ulaza koji sadrži referencu na neki eksterni entitet poverena loše konfigurisanom *XML* parseru. Rešenje ovog problema bila bi validacija po *XML* šemi.

3.1.1.5. Broken Access Control

Grupa napada koja iskorišćava izostanak kontrole pristupa na nivou funkcija i nebezbednih direktnih referenci na objekte. Za uspešno prevazilaženje ove vrste ranjivosti neophodno je vršiti rigorozne kontrole svih permisija koje se dodeljuju subjektima.

3.1.1.6. Security Misconfiguration

Meta napada je čitav sistem. Manipuliše nepravilnom implementacijom kontrolnih mehanizama koji bi aplikaciju trebali da održe bezbednom. Kao posledicu može imati to da se napadač potpuno infiltrira u sistem a da se za to neko vreme ne zna. Najdelotvornija mera odbrane jeste gašenje svega onog što nije neophodno.

3.1.1.7. Cross-Site Scripting

Napad koji eksploatiše i klijentsku i serversku stranu, što može dovesti do krađe kolačića i kompromitovanja osetljivih podataka. Za uspešnu realizaciju potrebno je da maliciozni napadač prvo preuzme kontrolu nad veb čitačem napadnutog subjekta, nakon čega može prinuditi veb čitač žrtve da izvrši bilo kakav podmetnuti *JavaScript*

kod zaobilazeći polisu zajedničkog porekla. Validacija podataka jedan je od načina odbrane od ove klase napada.

3.1.1.8. Insecure Deserialization

Distribucija malicioznih sadržaja vrši se pomoću serijalizovanih objekata. Uspešan napad može uzrokovati kompromitovanje ili brisanje podataka sačuvanih na disku, stvaranje potrebnih uslova za realizaciju *injection* grupe napada i eskalaciju privilegija. Sprečavanje ove klase napada zahteva onemogućavanje deserijalizacije podataka koji su potekli od neproverenih izvora.

3.1.1.9. Using Components with Known Vulnerabilities

Ako se radi sa već gotovim komponentama, bibliotekama i radnim okvirima, mora se obratiti pažnja na ranjivosti koje one unose u sistem. Prevencija napada moguća je ukoliko se vodi precizna evidencija o korišćenim komponentama i ako se one redovno održavaju.

3.1.1.10. Insufficient Logging & Monitoring

Nedovoljno često i detaljno nadgledanje u kombinaciji sa neefikasnom integracijom sa odgovarajućim odgovorima na incident ostavljaju dovoljno prostora napadačima da izvrše napad. Najvažnija bezbednosna kontrola koja se sprovodi sa ciljem smanjenja rizika od ove grupe napada je svakodnevno praćenje saobraćaja koji se razmenjuje i analiziranje dnevnika aktivnosti u aplikaciji.

4. RADNI OKVIR SPRING

Spring je radni okvir otvorenog koda koji pruža sveobuhvatnu infrastrukturnu podršku za razvoj *Java* aplikacija. *SpringBoot* je proširenje *Springa*. Dozvoljava da se više koraka spoji u jedan, zahvaljujući čemu se postupak podešavanja konfiguracije pojednostavljuje.

4.1. Arhitektura veb aplikacije za kupovinu i prodaju artikala iz ponude zasnovane na Spring radnom okviru

Aplikacija namenjena kupovini i prodaji artikala iz ponude organizovana je tako da se može uočiti jasna razlika između modela, prezentacionog sloja, upravljačkog sloja, servisnog sloja i repozitorijuma. Pripadajuće klase svakog od slojeva smeštene su u zasebnim paketima.

5. BAZA PODATAKA

5.1. Relaciona baza podataka

Relaciona baza podataka oslanja se na relacioni model koji je specifičan po tome što podatke organizuje u skup relacija između kojih se uspostavljaju odgovarajuće veze i ograničenja.

5.1.1. MySQL relaciona baza podataka – osnovni koncepti i integracija sa aplikacijom za kupovinu i prodaju artikala iz ponude

Da bi veb aplikacija namenjena kupovini i prodaji artikala iz ponude mogla da koristi podatke sačuvane u *MySQL* relacionoj bazi podataka, neophodno je podesiti konfiguracione parametre u okviru *application.properties* fajla i u *pom.xml* fajl dodati odgovarajuće zavisnosti.

Kreiranje tabela i međurelacionih ograničenja u bazi vrši se na osnovu *Spring* anotacija – sve klase moraju biti

propisno anotirane kako bi se znalo kom sloju pripadaju i na koji način će se uspostaviti veze između njih.

6. BOOKSTORE APLIKACIJA

Bookstore je veb aplikacija primarno namenjena kupovini i prodaji artikala iz ponude. Ideja je da implementirano softversko rešenje omogući rad sa sistemom koji je otporan na neke od poznatih učestalih sigurnosnih napada.

6.1. Veza između korisničkih uloga i raspoloživih funkcionalnosti

U aplikaciji se pravi razlika između korisnika. Oni se klasifikuju na registrovane i neregistrovane posetioce. Registrovani korisnici se potom dele na administratore i na obične registrovane posetioce sajta.

Tabela 1 sadrži pregled informacija o tome koja korisnička uloga može obavljati koje od funkcionalnosti aplikacije.

Tabela 1 Pregled informacija o tome koja korisnička uloga može obavljati koje od funkcionalnosti aplikacije

Funkcionalnost / korisnička uloga	Neregistrovani korisnik	Običan registrovan posetilac sajta	Administrator
slanje poruka	+	+	+
registracija	+	+	+
prijava	-	+	+
promena lozinke	-	+	+
ažuriranje korisničkog naloga	-	+	+
deaktivacija korisničkog naloga	-	+	+
pretraga dostupnih artikala	+	+	+
prikazivanje detaljnih informacija o odabranom artiklu	+	+	+
kupovina artikala	-	+	-
upravljanje nalogima običnih registrovanih posetilaca sajta	-	-	+
dodavanje novih artikala u ponudu	-	-	+
izmena postojećih artikala	-	-	+
brisanje artikala iz ponude	-	-	+

6.2. Sigurnosni mehanizmi implementirani u veb aplikaciji za kupovinu i prodaju artikala iz ponude

Implementirana veb aplikacija razvijena je u skladu sa politikom razvoja bezbednog softverskog proizvoda.

Rešenje je osmišljeno tako da ga je u bilo kom trenutku moguće nadograditi dodatnim sigurnosnim kontrolama, ukoliko se za tim javi potreba.

6.2.1. Sigurna komunikacija preko mreže

Za uspostavu bezbedne komunikacije između klijentske i serverske strane korišćen je *HTTPS* protokol. Na taj način sprečeno je neovlašćeno prisluškivanje saobraćaja na mreži, kao i *man in the middle* napad.

6.2.2. Logovanje aktivnosti

Softversko rešenje realizovano je tako da podržava logovanje svih dešavanja u aplikaciji. Na osnovu istorije aktivnosti moguće je zaključiti kada su nad sistemom počete da se preduzimaju sumnjive radnje, a takođe se garantuje i neporecivost zato što se tačno zna ko je i u kom trenutku izvršio koju operaciju.

6.2.3. Kriptovanje i heširanje korisničkih kredencijala

Podaci u skladištu naročito su ranjive prirode ukoliko se skladište u formi običnog teksta. Detektovani najosetljiviji podaci kojima sistem rukuje su korisnički kredencijali. Oni se u bazi podataka čuvaju u šifrovanom, odnosno heširanom obliku. Za heširanje lozinke korišćena je ugrađena klasa *BcryptPasswordEncoder*. Kriptovanje korisničkog imena vrši se *AES* algoritmom.

6.2.4. Validacija korisničkih unosa

Razvijena veb aplikacija ima proveru korisničkih unosa i na strani klijenta i na strani servera. Validacija podataka vrši se kombinovanom upotrebom liste dozvoljenih vrednosti i liste zabranjenih unosa jer se u praksi ovaj pristup pokazao najefikasnijim.

6.2.5. JSON Web Token

JSON Web Token je mehanizam koji se koristi da omogući da strane koje učestvuju u komunikaciji mogu samostalno i na siguran način razmenjivati podatke u *JSON* formatu. U pitanju je otvoreni standard koji garantuje da su informacije verifikovane i da im se može verovati zahvaljujući tome što su digitalno potpisane.

U aplikaciji za kupovinu i prodaju dostupnih artikala mehanizam *JWT*-a primenjen je u postupku autorizacije, čime je sprečen napad na neporecivost.

6.2.6. Cross-Site Request Forgery

U implementiranoj aplikaciji postoji zaštita od *CSRF* napada, što znači da nije moguće sprovesti napad koji bi naterao autentifikovanog korisnika da inicira operaciju koju ne želi. Zaštita od ove klase napada zahteva anotiranje konfiguracionih klasa na odgovarajući način.

6.2.7. Permisije i sopstvene anotacije

Razvijeni sistem poštuje strogu kontrolu pristupa resursima. Da bi se postigao efekat da svaku od ponuđenih funkcionalnosti aplikacije može da izvrši tačno određena uloga, uvedene su sopstvene permisije i anotacije.

7. ZAKLJUČAK

U radu su detaljno opisani osnovni koncepti informacione sigurnosti, izloženi su najčešći napadi na veb aplikacije i objašnjeno je zbog kojih je sve razloga potrebno raditi na razvoju bezbednih softverskih sistema.

Investiranje u proces razvoja pouzdanih i sigurnih softverskih rešenja dovodi do unapređenja procesa poslovanja zato što se njihova snaga ogleda upravo u mogućnosti da korisnicima daju garanciju da mogu obavljati visoko rizične operacije poput novčanih transakcija i ostavljanja ličnih podataka na uvid administratorskom osoblju bez brige o tome da li će oni u bilo kom trenutku i na bilo koji način biti kompromitovani i/ili čak zloupotrebljeni.

Ukoliko se poslušaju praktični saveti koji podstiču izgradnju bezbednog softvera, ako se za proces implementacije angažuju programeri koji vode računa o tome da je neophodno eliminisati što je moguće veći broj potencijalnih ranjivosti, kvalitet dobijenog rešenja bio bi na izuzetno visokom nivou.

Zahvaljujući tome, značajno bi se smanjila verovatnoća realizacije efikasnih napada i povećalo bi se poverenje koje korisnici imaju u sistem.

8. LITERATURA

- [1] <https://www.computersciencedegreehub.com/faq/what-is-information-systems-security> (pristupljeno u avgustu 2019.)
- [2] [https://www.owasp.org/images/7/72/OWASP_Top_10-2017_\(en\).pdf.pdf](https://www.owasp.org/images/7/72/OWASP_Top_10-2017_(en).pdf.pdf) (pristupljeno u avgustu 2019.)

Kratka biografija:



Bojana Samardžić rođena je 20.07.1995. u Novom Sadu. Završila je osnovnu školu „Jovan Dučić“ u Petrovaradinu. Nakon završene osnovne škole upisuje srednju školu, gimnaziju „Svetozar Marković“ u Novom Sadu. 2014. godine upisuje „Fakultet tehničkih nauka“ u Novom Sadu, smer *Računarstvo i automatika*. Školske 2016/17. godine opredeljuje se za usmerenje *Primenjene računarske nauke i informatika*. Školske 2017/18. godine upisuje modul *Internet i elektronsko poslovanje*. Zvanje *diplomirani inženjer elektrotehnike i računarstva* dobija 29.08.2018. Školske 2018/19. godine upisuje master akademske studije, smer *Primenjene računarske nauke i informatika – Elektronsko poslovanje*. Zvanje *saradnik u nastavi* na Departmanu za računarstvo i automatiku na Fakultetu tehničkih nauka u Novom Sadu dobija 09.11.2018.

kontakt: bojana.samardzic@uns.ac.rs

PREDLOG ARHITEKTURE CI/CD PIPELINE-A KORIŠĆENJEM JENKINS I DOCKER ALATA**THE PROPOSAL OF CI/CD PIPELINE USING JENKINS AND DOCKER**Milena Počuča, Srđan Popov, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu su detaljno analizirana dva najpopularnija alata za pravljenje CI/CD pipeline-a. Infrastruktura je postavljena uz akcenat na Docker na kome su pokrenuti svi servisi potrebni za CI/CD. Za primer aplikacije na kojoj je pipeline implementiran uzeta je SpringBoot i AngularJS aplikacija podsistema za fakturisanje.

Ključne reči: CI/CD, Docker, Jenkins, SpringBoot, AngularJS, pipeline, DevOps

Abstract – The paper analyzes the two most popular tools for implementing CI/CD pipeline. The infrastructure is set up with the accent on Docker on which all the services needed for this pipeline are configured and run. The application on which the pipeline was implemented is a SpringBoot and AngularJS application subsystem for invoices.

Keywords: CI/CD, Docker, Jenkins, SpringBoot, AngularJS, pipeline, DevOps

1. UVOD

CI/CD predstavlja skup CI (*continuous integration*) i CD (*continuous delivery*) paradigmi koji je našao široku primenu u industriji. CI predstavlja pristup u kome se programeri podstiču da implementiraju male izmene na kodu što je frekventnije moguće. CD se nadovezuje na CI i automatizuje proces isporuke aplikacija na izabrana okruženja. CI/CD pipeline u radu je implementiran korišćenjem Jenkins i Docker alata. CI/CD pipeline-om bavi se DevOps.

Cilj rada je detaljno upoznavanje sa Jenkins i Docker alatima uz demonstraciju načina implementacije infrastrukture za CI/CD, kao i uz osvrt na već postojeće implementacije pipeline-a.

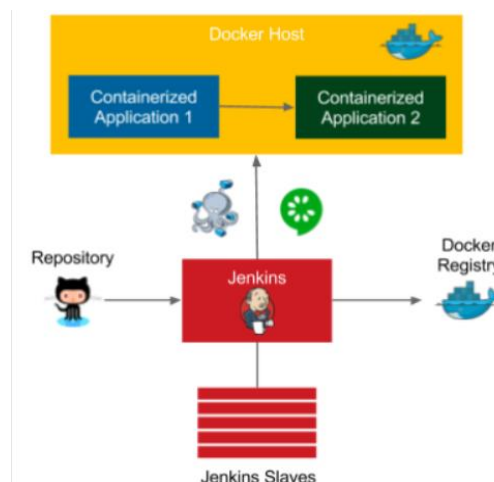
2. INFRASTRUKTURA SISTEMA

Jezgro sistema predstavlja Docker mašina koja u sebi ima pokrenute kontejnere svih alata koji su potrebni za funkcionisanje aplikacije. Na Docker-u su startovani na lokalnom računaru Jenkins kao CI server, GitLab kao sistem za upravljanje git repozitorijumom, Minio namenjen čuvanju biblioteka kreiranih od strane Jenkins-a, MySQL Server za upravljanje bazom podataka, SonarQube za proveru kvaliteta koda, Redis za konfiguraciju Docker registra.

NAPOMENA:

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

Na slici 1 prikazan je primer arhitekture sistema koji se bazira na Docker-u.



Slika 1. Prikaz arhitekture bazirane na Docker-u

3. DEVOPS

DevOps predstavlja spoj tehnologija koje objedinjuju razvoj softvera (*Dev*) i IT operacija (*Ops*) kako bi skratili životni ciklus razvoja softvera.

Ciljevi DevOps paradigme su učestalije isporučivanje koda, skraćivanje vremena potrebnog da se stigne do produkcionog rešenja kao i vremena koje protekne od kreiranja zahteva do njihovog ispunjenja, snižavanje stope otkaza novijih verzija proizvoda, brže srednje vreme oporavka sistema.

4. JENKINS

Jenkins je samostalan, javni automatizacioni server koji se može koristiti da automatizuje razne vrste zadataka povezane sa izgradnjom, testiranjem, isporukom ili *deployment*-om softvera. Jedna od najvećih prednosti ovog servera je velika proširivost koja omogućava da se njegove funkcionalnosti prošire instalacijom *plugin*-ova.

4.1. Pipeline

Jenkins pipeline je skup *plugin*-ova koji omogućava implementaciju i integraciju *continuous delivery pipeline*-ova u Jenkins. *Jenkinsfile* komitovan na repozitorijum omogućava veliki broj prednosti jer je verzionisan i u svakom trenutku su sve izmene na fajlu vidljive.

4.2. Zašto pipeline

Podržava širok skup slučajeva od jednostavih *continuous integration* do obimnih CD pipeline-ova. Sadrži brojne funkcionalnosti, kao što su implementacija pipeline-a u kodu, pauziranje, svestranost, proširivost.

4.3. Rečnik pojmova

Osnovni pojmovi u Jenkins-u su: *agent*, *artifact*, *build*, *core*, *downstream*, *executor*, *fingerprint*, *folder*, *item*, *job*, *label*, *master*, *node*, *project*, *pipeline*, *plugin*, *stage*, *step*, *trigger* [4].

5. SINTAKSA PIPELINE-A

Od verzije 2.5 plugin-a za *pipeline*, postoje dve sintakse za njen opis koje će u nastavku biti objašnjene.

5.1. Deklarativni pipeline

Ova sintaksa je prilično nov dodatak *Jenkins pipeline*-a koji predstavlja dosta jednostavniji način definisanja. Svaki pipeline ovog tipa mora biti obuhvaćen u okviru *pipeline* bloka.

5.1.1. Sekcije

Obično su sastavljene iz jedne ili više direktiva.

5.1.1.1 Agent

Sekcija *agent* određuje gde će se čitav *pipeline*, ili određena faza, izvršavati na *Jenkins* okruženju, u zavisnosti od toga gde je sekcija *agent* smeštena. Mora se definisati na najvišem nivou unutar *pipeline* bloka, ali se može definisati i unutar *stage* bloka. Sadrži parametre koji se mogu primeniti na najviši nivo *pipeline* bloka, ili unutar svake *stage* direktive. Mogući parametri su *any*, *none*, *label*, *node*, *docker*, *dockerfile*, *kubernetes* [4].

5.1.1.2. Post

Ova sekcija sadrži jedan ili više dodatnih koraka koji se pokreću nakon završetka *run* dela *pipeline*-a ili *stage*-a. Podržava brojne *post-condition* blokove koji dozvoljavaju izvršavanje koraka unutar svakog uslova. Mogući uslovi su: *always*, *changed*, *fixed*, *regression*, *aborted*, *failure*, *success*, *unstable*, *unsuccessful*, *cleanup* [4].

5.1.1.3. Stages

Ova sekcija sadrži sekvencu jedne ili više *stages* direktiva i označava sekciju gde je najveći deo posla u *pipeline*-u smešten. Preporučuje se da ova sekcija sadrži bar jednu *stage* direktivu za svaki deo CD procesa – *Build*, *Test*, *Deploy* [4].

5.1.1.4. Steps

Sekcija *steps* definiše skup jednog ili više koraka koji se mogu izvršavati u datoj *stage* direktivi. Sekcija je obavezna, ne zahteva parametre i dozvoljena je unutar svakog *stage* bloka.

5.1.2. Direktive

5.1.2.1. environment

Ova direktiva predstavlja sekvencu ključ-vrednost parova koji će biti definisani kao varijable okruženja za sve korake, ili za korake specifične za neki *stage*, u zavisnosti od toga gde se ova direktiva nalazi u *pipeline*-u. Podržani tipovi kredencijala: *Secret Text*, *Secret File*, *Username and Password*, *SSH with Private key* [4].

5.1.2.2. options

Direktiva *options* omogućuje konfiguraciju opcija karakterističnih za *pipeline* unutar samog *pipeline*-a. Dostupne opcije za *pipeline*: *buildDiscarder*, *checkoutTo-Subdirectory*, *disableConcurrentBuilds*, *disableResume*, *newContainerPerStage*, *overrideIndexTriggers*, *quiet-Period*, *retry*, *skipDefaultCheckout*, *skipStagesAfter-Unstable*, *timeout*, *timestamps*, *parallelsAlwaysFailFast*, Opcije koje se odno-

se na *stage* predstavljaju deo opcija za *pipeline*: *-skipDefaultCheckout*, *timeout*, *retry*, *timestamps* [4].

5.1.2.3. parameters

Direktiva *parameters* obezbeđuje listu parametara koji bi korisnik trebalo da navede kada pokreće izvršavanje *pipeline*-a. Tipovi parametara su: *string*, *text*, *booleanParam*, *choice*, *password*.

5.1.2.4. triggers

Direktiva *triggers* definiše automatski način na koji bi *pipeline* trebalo da se ponovo trigeruje. Vrste trigera su: *cron*, *pollSCM*, *upstream* [4].

5.1.2.5. Jenkins cron sintaksa

Jenkins cron sintaksa prati sintaksu *cron utility*-ja sa sitnim razlikama. Svaka linija se sastoji iz 5 delova koji su odvojeni TAB-om ili razmakom. U sintaksi se specificiraju 'M H DOM MONTH DOW', gde M predstavlja minut u satu, H sat u danu, DOM dan u mesecu, MONTH mesec u godini, a DOW dan u nedelji.

5.2.1.5. stage

Direktiva *stage* se nalazi u okviru *stages* sekcije. Sadrži *steps* sekciju, opcionu *agent* sekciju ili druge direktive specifične za *stage*. U praksi, najveći deo realnog posla *pipeline*-a se odvija unutar jedne ili više *stage* direktiva. Potrebno je definisati bar jednu *stage* direktivu koja će imati jedan obavezni parameter, a to je ime *stage*-a.

5.2.1.6. tools

Ova sekcija definiše alate koji se automatski instaliraju i dodaju na PATH. Ignoriše se ukoliko se specificira *agent none*. Dozvoljena je unutar *pipeline* ili *stage* bloka. Podržani alati su *maven*, *jdk* i *gradle*.

5.2.1.7 when

Direktiva *when* dozvoljava *pipeline*-u da odluči da li *stage* treba da se izvrši u zavisnosti od datog uslova. Unapred dostupni uslovi su: *branch*, *buildingTag*, *changelog*, *changeset*, *changeRequest*, *environment*, *equals*, *expression*, *tag*, *not*, *allOf*, *anyOf*, *triggeredBy* [4].

5.1.5. Steps

Deklarativni *pipeline* koriste sve korake koji su dokumentovani u [Pipeline Steps reference](#) dokumentaciji i sadrže obilnu listu koraka.

5.2. Skriptovani pipeline

Za razliku od deklarativnog, skriptovani *pipeline* je DSL opšte namene izgrađen pomoću *Groovy*-ja. Većinu funkcionalnosti koje omogućuje *Groovy* jezik moguće je koristiti u skriptovanom *pipeline*-u. Najvažniji aspekt ovog *pipeline*-a je kontrola toka.

5.3. Deklarativni naspram skriptovanog pipeline-a

Oba *pipeline*-a su fundamentalno ista u *pipeline* podsistemu. Razlikuju se ipak, u sintaksi i fleksibilnosti. Deklarativni *pipeline* ograničava šta je dostupno korisniku sa striktnijom i predefinisanim strukturom, što ga čini idealnim izborom za jednostavniji CD *pipeline*. Skriptovani *pipeline* obezbeđuje mali broj ograničenja, može se reći da su to ograničenja definisana u samom *Groovy* jeziku, ne u *pipeline*-u, što ga čini idealnim izborom za iskusnije korisnike i korisnike koji imaju kompleksne zahteve. Deklarativni *pipeline* promovise deklarativni model programiranja, dok skriptovani *pipeline* promovise imperativni [4].

6. DOCKER

Docker je alat koji je dizajniran da olakša kreiranje, *deploy*-ovanje i pokretanje aplikacija korišćenjem kontejnera. *Docker, Inc.* je osnovan 2010. godine, a javnosti je predstavljen 2013.

6.1. Kontejneri i virtuelne mašine

Docker kontejneri se mogu porediti sa virtuelnim mašinama. I jedni i drugi imaju sličan cilj, a to je izolacija aplikacije i njenih zavisnosti u samoodrživu jedinicu koja se može svuda pokrenuti. Virtuelna mašina ima svoj virtuelni operativni sistem u kome hipervizor igra esencijalnu ulogu tako što joj obezbeđuje platformu za upravljanje i izvršavanje *guest* operativnog sistema.

Za razliku od virtuelne mašine, kontejner obezbeđuje virtuelizaciju na nivou operativnog sistema. Kontejnerizacija je takođe vrsta virtualizacije s prednošću da je efikasnija jer nema *guest* operativnog sistema, već koristi *host*-ov operativni sistem.

6.2. Docker Engine

Docker engine predstavlja srž *Dockera*. To je aplikacija koja se instalira na *host* mašinu. Funkcioniše po principu klijent-server arhitekture. Glavne komponente ove arhitekture su: server koji predstavlja program koji se dugo izvršava zvani *daemon* proces, CLI koji predstavlja klijenta, REST API preko koga CLI i *daemon* komuniciraju [5].

6.3. Arhitektura Docker-a

Docker koristi klijent-server arhitekturu u kojoj *Docker* klijent komunicira sa *Docker daemon*-om koji ustvari radi najveći posao - izgradnju, pokretanje i distribuciju *Docker* kontejnera.

6.4. Slojevi Docker-a

Docker image se kreira iz serije slojeva (eng. *layers*) gde svaki sloj predstavlja instrukciju *image*-a u *Dockerfile*-u. Svi slojevi sem poslednjeg sloja imaju samo pristup čitanja, dok se na poslednjem sloju može i pisati.

6.5. Dockerfile

Docker automatski pravi *image* čitanjem instrukcija iz *Dockerfile*-a. Ovaj fajl je tekstualni dokument koje korisnik može da koristi da bi napravio *image*. Komanda *docker build* pravi *image*.

Dostupne instrukcije koje se zadaju su:

FROM (inicijalizuje *build* i postavlja bazni *image* za ostale instrukcije),

RUN (izvršava komande u novom sloju koji je iznad *image*-a),

EXPOSE (obaveštava *Docker*-a da kontejner sluša na određenom portu),

ENV (postavlja varijablu na zadatu vrednost),

ADD (kopira nove fajlove iz zadatog izvora i dodaje ih na fajl sistem *image*-a),

COPY (kopira nove fajlove iz izvora i dodaje ih na fajl sistem kontejnera),

VOLUME (kreira pristupnu tačku sa zadatim imenom i pamti je kao eksterni volumen *host*-a ili drugih kontejnera),

USER (postavlja korisnikovo ime i grupu korisnika koji će se koristiti za pokretanje *image*-a) [5].

6.6. Docker compose

Compose je alat za definisanje i pokretanje *Docker* aplikacija sa više kontejnera. Koristi *yaml* fajl za konfigurisanje servisa aplikacije. To je proces koji se sastoji iz tri kora-

ka: definisanje okruženja aplikacije korišćenjem *Dockerfile*-a, definisanje servisa koji prave aplikaciju u *docker-compose.yml* fajlu i pokretanje *docker-compose up* komande kako bi se sve to pokrenulo u izolovanom okruženju [5].

7. ALATI ZA IZGRADNJU CI/CD PIPELINE-A

CI/CD *pipeline* se može postaviti korišćenjem većeg broja alata. Ovo predstavlja trenutno najpopularniji trend u razvoju softvera.

7.1. Jenkins

Ovo je najpopularnije rešenje CI/CD *pipeline*-a koje je besplatno.

7.2. Microsoft VSTS

VSTS nije samo servis za CI/CD, već i repozitotijum za kod i planiranje projekata. VSTS je pravljen za *Visual Studio*, ali se može integrisati i sa drugim alatima. Besplatan je za timove do 5 članova.

7.3. Bamboo (Atlassian Dev Tool)

Bamboo je CI server koji je *Atlassian*-ov proizvod, podržava *Docker*, ima probni period od 30 dana, nakon čega se naplaćuje.

7.4. GitLab

GitLab predstavlja skup alata za upravljanje skoro svim aspektima životnog ciklusa razvoja softvera. Dostupan je kao besplatan *Community* i *Enterprise* edicija koja može biti *host*-ovana ili se nalaziti na mašini.

7.5. Codeship

U odnosu na druge alate koji pružaju UI za naprednu konfiguraciju, *Codeship* je uglavnom baziran na skriptama. Posедуje *Basic* i *Pro* verzije koje se naplaćuju.

7.6. Codefresh

Codefresh svaki korak u *pipeline*-u izvršava na posebnom kontejneru. Besplatna verzija omogućuje 3 korisnika mesečno, *Pro* verzija 10, dok *Enterprise* omogućuje neograničen broj.

7.7. TeamCity

TeamCity je CI server koji ima ugrađenu integraciju sa *Docker*-om. Dostupan je besplatno sa ograničenom licencijom, a postoji i verzija koja se plaća.

7.8. Travis CI

Travis CI je *host*-ovani servis za *GitHub* projekte. Besplatan je za javne projekte i omogućuje pretplatu za privatne projekte.

7.9. GoCD

GoCD je besplatan softver koji se instalira lokalno. Ima i plugin za *docker*, besplatan je uz opciju plaćanja podrške.

7.10. CircleCI

CircleCI je drugačiji od ostalih alata jer podržava *build*-ovanje, testiranje i *deploy*-ovanje iOS i macOS projekata korišćenjem macOS virtuelne mašine, mada se može *build*-ovati i korišćenjem *Linux* virtuelne mašine. Cena *CircleCI*-a zavisi od kontejnera i platforme. *Linux* kontejner je besplatan, ali macOS kontejneri i lokalna instalacija se plaćaju.

8. OSVRT NA POSTOJEĆE IMPLEMENTACIJE CI/CD PIPELINE-A

8.1. Automatizacija CI/CD pipeline-a za projekat baziran na agilnom pristupu

Ciljevi ovog rada su razmevanje CI/CD pipeline-a analizom postojećih metoda, pronalaženje odgovarajuće metode za merenje performansi postojećeg sistema, predlog mehanizma za izvršavanje testova opterećenja korišćenjem saobraćaja u produkciji sa minimalnim posledicama na sistem. U radu su od alata korišćeni *Git* kao sistem za kontrolu verzija, *Nexus* kao repozitorijum izvršivih fajlova i *Jenkins* kao CI server. Zaključeno je da je CI/CD pipeline poboljšao agilan proces isporučivanja i poboljšao je produktivnost sistema [1].

8.2. CI/CD za HPC korišćenjem Jenkins-a i Singularity-ja

U radu je opisano kako je *Research Computing* ogranak Bolder (eng. *Boulder*) Univerziteta u Koloradu primenio CI/CD paradigmu na *RMACC Summit supercomputer* kako bi obezbedio sistem inženjerima i istraživačima da iskoriste prednosti CI/CD paradigme.

Kao CI server, korišćen je *Jenkins* koji je povezan sa *Singularity*-jem koji predstavlja rešenje za kontejnerizaciju. Uključivanjem CI/CD paradigme u proces rada HPC-a je povećalo potencijal za isporuku visoko kvalitetnih i pouzdanih softvera [2].

8.3. Implementacija CI principa u maloj kompaniji: studija slučaja

Kao motivacioni faktori za razvoj ove paradigme u kompaniji, identifikovani su poboljšanje kvaliteta softvera, veća pouzdanost *release-a*, kraći *time-to-market* i poboljšanje produktivnosti programera. Centralizovano rešenje, kao što je *GitLab* je primećeno kao najprimerenije rešenje jer on spaja sve potrebne servise. Zaključeno je da je CI/CD paradigma donela poboljšanje rada kompanije, ali da je potrebno nastaviti sa radom na istoj, tako što će preostale repozitorijume migrirati na novi sistem i početi što pre sa testiranjem [3].

9. IMPLEMENTACIJA REŠENJA

Čitav sistem počiva na *Docker*-u. U njemu su pokrenuti *GitLab*, *Jenkins*, *Redis*, *SonarQube*, *Minio* i *MySQL*. Sistem radi tako što kada se kod postavi na repozitorijum, *Jenkins pipeline* registruje da je došlo do izmene repozitorijuma, povlači novi kod i pokreće *build* aplikacije. Nakon što se *build* završi, startuju se testovi i ukoliko se svi uspešno izvrše, jar koji je napravljen se postavlja na *Minio* i šalje se u *Docker* koji kreira novi kontejner za aplikaciju. *SonarQube* vrši analizu novog koda uz proveru pokrivenosti koda testovima. Kako *Docker* ponovo pokreće kontejner aplikacije sa novonastalim izmenama, njih je moguće odmah testirati.

10. ZAKLJUČAK

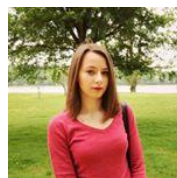
U radu je prikazano kako se konfiguriše CI/CD pipeline za potrebe manjeg projekta uz detaljna objašnjenja dva glavna korišćena alata – *Docker* i *Jenkins*. Neosporivo je da je CI/CD paradigma izuzetno značajna za razvoj IT kompanija sa svim prednostima koje donosi.

Postoji veliki broj alata za implementaciju ove paradigme, pa treba dobro proučiti kada koji iskoristiti, *Jenkins* predstavlja vodeći alat danas sa svim svojim pluginovima i novim deklarativnim pipeline-om. *Docker* je takođe vodeći alat jer ima podršku za *Kubernetes* koji upravlja kontejnerima na distribuiranim sistemima. Dalji razvoj arhitekture bi uključio *Kubernetes* i distribuciju sistema.

11. LITERATURA

- [1] S.A.I.B.S. Arachici, Indika Perera, „Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management“, Department of Computer Science and Engineering, University of Moratuwa, Moratuwa, Sri Lanka
- [2] Zebula Sampedro, Aaron Holt, Thomas Hauser, „Continuous Integration and Delivery for HPC: Using Singularity and Jenkins“, University of Colorado Boulder, Research Computing, Boulder, Colorado
- [3] Kim Rejström, „Implementing Continuous Integration in a Small Company: A Case Study“, Aalto University, School of Science
- [4] <https://jenkins.io/doc/> (pristupljeno u septembru 2019.)
- [5] <https://docs.docker.com> (pristupljeno u septembru 2019.)

Kratka biografija:



Milena Počuca, rođena je 27.08.1994. u Kraljevu. 2013. godine upisuje „Fakultet tehničkih nauka“ u Novom Sadu, smer „Računarstvo i automatika“. 2017. godine dobija zvanje Diplomirani inženjer elektrotehnike i računarstva. 2017/18. godine upisuje master akademske studije, smer „Primenjene računarске nauke i informatika – Elektronsko poslovanje“. 01.03.2018. godine dobija zvanje Saradnika u nastavi na Departmanu za računarstvo i automatiku na Fakultetu tehničkih nauka u Novom Sadu. kontakt: minapocuca@gmail.com

ANALIZA ALGORITAMA ZA VEKTORSKU REPREZENTACIJU TEKSTA**ALGORITHMS FOR VECTOR REPRESENTATION OF TEXT**Milan Stanković, *Fakultet tehničkih nauka, Novi Sad***Oblast – Računarstvo i automatika**

Kratak sadržaj – Obrada teksta odnosno NLP (*Natural language processing*) predstavlja atraktivnu oblast u oblasti računarske inteligencije. U ovom radu će biti predstavljeno više načina za pretvaranje teksta u vektore. Postoje dve velike kategorije, prva se zasniva na statističkim metodama a druga na korišćenju neuronskih mreža. Prednost vektorske predstave je mogućnost da se takvi vektori porede ili dalje lakše koriste u nekoj od metoda mašinskog učenja. Vektorska reprezentacija održava deo semantike, tako da predstavlja pogodnu metodu za reprezentaciju i poređenje kompleksnijeg teksta.

Ključne reči: *enkapsulacija, vektorsko predstavljavanje reči, glove, word2vec, skipgram, cbow, duboko mašinsko učenje*

Abstract – *Natural language processing (NLP) is an attractive area of computing intelligence. This paper will introduce several ways to convert text to vectors. There are two major categories, the first based on statistical methods and the second using neural networks. The advantage of vector representation is the ability to compare or further use such vectors in one of the machine learning methods. Vector representation maintains some semantics, so it is a convenient method for representing and comparing more complex text.*

Keywords: *encapsulation, word vectors, glove, word2vec, skipgram, cbow, deeplearning*

1. UVOD

Postoji mnogo načina za predstavljavanje teksta u dokumentu kao i njegovu analizu. Trenutno jedan od najpopularnijih i uspešnijih načina za reprezentaciju teksta je enkapsulacija. Cilj enkapsulacije je da se tekst pretvori u oblik koji mašina, odnosno neuronske mreže ili algoritmi za mašinsko učenje, mogu lakše da koriste. Tekst, za razliku od numeričkih podataka, nije moguće direktno analizirati gledajući samo ASCII vrednosti karaktera. Retko kada ima smisla gledati samo pojedinačne karaktere, jer bez da vidimo koji se nalaze ispred i iza njih ne možemo dobiti celu predstavu o tekstu. Većina algoritama za transformaciju teksta rade po principu pretvaranja teksta u vektorski oblik, češće višedimenzioni.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinać, vanr. prof.

Ako ima potrebe samo za prebacivanje u vektorski oblik, osnovni algoritmi to rade time što svakoj jedinstvenoj reči dodeljuju numeričku vrednost, a onda se ceo tekst može predstaviti kao niz brojeva. Ovom osnovnom transformacijom sami brojevi samo zamenjuju reči, ali ne doprinose razumevanju te reči. Na osnovu dobijenog niza moguće je dobiti malo konteksta za sve reči koje se nalaze u tekstualnom korpusu. Čistom analizom frekvencije pojave nekih reči potencijalno možemo dobiti koje su reči značajne u tom tekstu. Međutim, u praksi se ovakvi jednostavni pristupi nisu pokazali kao previše korisni.

2. OSNOVNI ALGORITMI ZA ENKAPSULACIJU

Svakodnevni govor, ali i pisani tekstovi, se sastoje od dosta veznika, rečica, ali i uzrečica koje pisac koristi. Reč tog karaktera bi se pojavljivala kao najznačajnije, dok su suštinski retko kada bitne. Takvom analizom ne dobijamo gotovo nikakve korisne informacije. Moguće je izbaciti sve reči koje su kraće od zadate dužine ili, još bolje, da imamo korpus veznika i ostalih reči koji ne nose veliki semantički značaj. Prilikom analize teksta njih ne uzimati u obzir. Ovakvi pristupi nam, nažalost, rešavaju samo deo problema.

Drugi veliki problem kod ovakve predstave teksta je činjenica da reči koje su u množini ili u različitim padežima, vremenima, superlativi ili sinonimi bi bili svi predstavljeni kao različiti brojevi i, samim time, kao u potpunosti različite reči. Reči “knjiga”, “knjige”, “knjizi” nikako ne bi bile povezane, iako se u suštini svode na istu stvar. Postoje algoritmi koji rade lematizaciju (odbijanje prefiksa i sufiksa) i stemovanje (svođenje reči na njen koren) ili oni koji porede reči na osnovu broja koraka koje je potrebno obaviti kako bi se jedna reč pretvorila u drugu. Ovakvi pristupi često mogu da rade dobro, čak i kada ima gramatičkih grešaka. Njihovom upotrebom se dobija dosta bolja predstava teksta, ali se gubi dosta konteksta i ne rešavaju se problemi žargona, sinonima. Ako uzmemo u obzir reč “radio” ona može imati skroz različito značenje u “radio sam na tom projektu” naspram “uključio sam radio”. U prvom obliku je glagol koji predstavlja predikat u rečenici, dok je u drugom imenica koja predstavlja objekat. Rešenje ovih problema zahtevaju dosta kompleksnije algoritme. Kako je govor, a samim time i tekst, kompleksan sa dosta nijansi i razlika između jezika, kultura, govornih područja, još uvek nije napravljen savršen sistem za reprezentaciju i analizu teksta.

Češći način za prebacivanje teksta u neku numeričku reprezentaciju je da se svakoj reči dodeli vektor. Osnovni oblik ovakve transformacije, se naziva *one-hot encoding*

(kodiranje). Kako bi se ono odradilo potrebno je znati broj jedinstvenih reči u tekstu koji se analizira. Recimo za rečenicu “Ja volim NLP i ja obožavam skijanje !”. Dužina **one-hot** vektora bi bila šest. Prvo je potrebno odraditi osnovno procesiranje teksta. Reč “JA” se u rečenici nalazi u dva oblika “Ja” i „ja“, kako bi se izbegle greške koje mogu uslediti zbog velikih i malih slova, najčešće se sva slova pretvaraju u velika. Uz to se otklanjaju i svi specijalni karakteri kao što su navodnici, tačke, upitnici... Ova konverzija povremeno može uvesti nove greške. Zatim, dodeljujemo svakoj reči vrednosti koje su nula ili jedan po principu da će svi vektori biti jedinstveni i da će svaki vektor imati samo jednu jedinicu i ostale vrednosti će biti nule. Na kraju se dobijaju vektori koji su prikazani u tabeli br. 1. Ovom tehnikom je za isti ulaz moguće dobiti različite vektore za iste reči. Na ovaj način imamo mogućnost da predstavimo bilo koju reč u tekstu kao vektor, ali taj vektor, nažalost, ne sadrži nikakvu semantiku.

JA	=	[	1,	0,	0,	0,	0,	0	]
VOLIM	=	[	0,	1,	0,	0,	0,	0	]
NLP	=	[	0,	0,	1,	0,	0,	0	]
I	=	[	0,	0,	0,	1,	0,	0	]
OBOZAVAM	=	[	0,	0,	0,	0,	1,	0	]
SKIJANJE	=	[	0,	0,	0,	0,	0,	1	]

Tabela 1. Prikaz one-hot kodiranja

Postoje i kompleksniji oblici pretvaranja reči u vektor, koji uzimaju u obzir kontekst, odnosno odnos reči jednih prema drugima. Jedan od načina da se odradi ovakva reprezentacija je da se napravi matrica uzajamnog pojavljivanja. Na slici 1. se može videti ovakva matrica na primeru rečenice “Ja volim NLP i ja obožavam skijanje !”.

U slučaju da se u rečenici, odnosno tekstu, reči nalaze jedna do druge u tabeli na mestu preseka se vrednost postavlja na jedan. Osnovni primer ove tabele može imati samo vrednosti jedan ili nula. Bolji algoritmi popunjavaju tabelu tako što povećavaju vrednost za jedan, za svako ponavljanje susedstva. Zbog različite frekvencije reči, često se i drugi način predstave reči na kraju skalira na domen od [0..1] ili [-1..1]. Skaliranje pomaže pri daljim analizama time što smanjuje varijabilnost.

	JA	VOLIM	NLP	I	OBOZAVAM	SKIJANJE
JA	0	1	0	1	1	0
VOLIM	1	0	0	1	0	0
NLP	1	0	0	1	0	0
I	1	0	1	0	0	0
OBOZAVAM	1	0	0	0	0	1
SKIJANJE	0	0	0	0	1	0

Slika 1. Primer matrice pojavljivanja

Na osnovu matrice moguće je kreirati vektore za svaku reč. Za primer prikazan iznad bi se kreirali vektori kao što je prikazano u tabeli ispod.

Za razliku od samog teksta, vektore je moguće porediti. Preko Euklidskog rastojanja moguće je odrediti koliko su dva vektora blizu jedan drugom.

Rastojanje možemo predstaviti kao sličnost dve reči, odnosno što su vektori bliži to su dve reči sličnije. Što je

veći tekstualni korpus na osnovu koje se generišu ove matrice to je veća tačnost.

JA	=	[	0,	1,	0,	1,	1,	0	]
VOLIM	=	[	1,	0,	1,	0,	0,	0	]
NLP	=	[	1,	0,	0,	1,	0,	0	]
I	=	[	1,	0,	1,	0,	0,	0	]
OBOZAVAM	=	[	1,	0,	0,	0,	0,	1	]
SKIJANJE	=	[	0,	0,	0,	0,	1,	0	]

Tabela 2. Prikaz vektora

Postoje modeli zasnovani na neuronskim mrežama obučavaju se na ogromnom korpusu teksta kako bi što bolje reprezentovali reči u nadi da će se u njihovom vektorskom obliku nekako održati sama semantika i rešiti svi prethodno napomenuti problemi. Noviji probabilistički modeli, u određenim zadacima, su pokazali sličnu tačnost kao modeli zasnovani na neuronskim mrežama. Njihova velika prednost je da nemaju potrebu da se obučavaju.

Budući da su na kraju dobijeni vektori, osim poređenja, moguće je raditi i ostale matematičke operacije. Dobro obučeni modeli su pokazali da postoji održanje nekog vida konteksta i da je moguće izvući zanimljive zaključke. Na slici 2. se vidi da je rastojanje između reči muškarac (man) i žena (woman) isto kao rastojanje između reči kralj (king) i kraljica (queen), ovo održanje rastojanja između reči različitog roda se isticalo u mnogim primerima. Isto važi i za rastojanje između reči u različitim vremenima ili udaljenost između glavnog grada i države u kojoj se nalazi. Uz dodatne algoritme, moguće je čak vektorskim sabiranjem dve reči dobijati treću. Recimo reč **apple** (jabuka) i **purple** (ljubičasta) daju reč **plum** (šljiva).



Slika 2. Poređenje održanje rastojanja

3. GLOVE MODELI

Radovi kao što je “GloVe: Global Vectors for Word Representaton”[1] prikazali su da je moguće reprezentovati reči kao n-dimenzione vektore koji sadrže neki vid semantike same reči. **Glove** algoritam je zasnovan na probabilističkim teorijama umesto na korišćenju **LSTM**-a ili drugih neuronskih mreža, kao u **word2vec** i ostalim algoritmima za reprezentaciju reči sa vektorima. Korišćene su matrice međusobnih pojava gde vrednost u matrici predstavlja koliko se puta jedna reč nalazi u kontekstu druge.

Kako bi se kreirao vektor za svaku reč se gleda u kojim se kontekstima javlja i dodeljuju se vrednosti reči koje su blizu, sa tim da je uveden faktor opadanja na osnovu toga koliko je daleka jedna reč od druge.

Glove model ima procentualnu tačnost od 82% u kontekstu semantike. U testovima za sličnost reči je korišćena kosinusna sličnost za evaluaciju vektora, i dobijeni rezultat variraju u zavisnost skupa podataka za testiranje koji se koristi (za četiri od pet skupova je tačnost oko 70%, dok je za poslednji značajno manja sa 40%).

Ovaj model ima veliku prednost naspram algoritama zasnovanim na neuronskim mrežama iz razloga što se mora obučavati i često pokazuje veću tačnost naspram njih, ako je tekstualni korpus mali. Postoje već i unapred odrađeni **Glove** modeli koji su kreirani nad ogromnim korpusima teksta. Kada je pametnije koristiti pretreniran model i eventualno ga dotrenirati, a kada je potrebno kreirati nov model je pitanje koje muči većinu ljudi koji se bave analizom teksta.

Na ovo pitanje ne postoji univerzalan odgovor. Odgovor zavisi od korišćenog teksta. Što je tekst koji se treba analizirati sličniji tekstu koji je korišćen u kreiranju pretreniranog modela, to više ima smisla koristiti pretreniran model uz eventualne izmene ili dopune.

Sa druge strane, veoma zavisi i od veličine tekstualnog korpusa, odnosno komputacione moći računara na kojem je potrebno kreirati novi model.

4. WORD2VEC MODEL

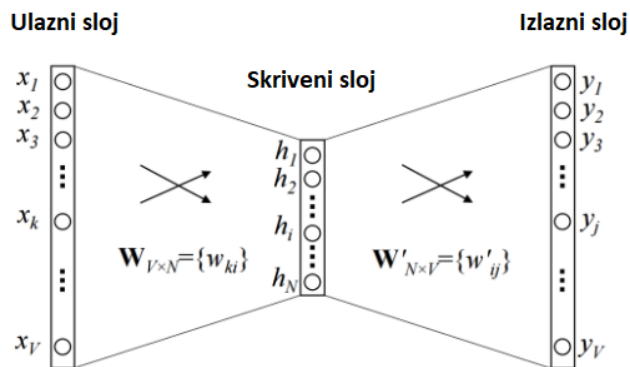
Word2Vec model (reč u vektor) je model zasnovan na neuronskim mrežama koji vrši transformaciju reči u njihov vektorski oblik. U prethodnim primerima je prikazano da je moguće od bilo kojeg teksta kreirati tekst, proces sam po sebi nije složen. Međutim održanje konteksta, odnosno suštine teksta je deo u kojem **word2vec** pokazuje dobre rezultate. On je trenutno jedan od najčešće upotrebljivanih metoda za ovaj posao, budući da u opštem slučaju daje najbolje rezultate bez previše obučavanja. Vektore je moguće dobiti pomoću dve metode, obe uključuju neuronske mreže : **Skip Gram** i **Common Bag Of Words (CBOW)**. Obe metode su sadržane u **word2vec** modelu, korisnik samostalno bira koju će verziju koristiti.

4.1 CBOW model

Ovaj model uzima u obzir kontekst svih reči kao ulaz i pokušava da predvidi reč koja najviše odgovara kontekstu. Ako je recimo unos u neuronsku mrežu reč "dobar" želimo da nam se predvidi da je sledeća reč "dan". Za reč "dan" se vrši **one-hot** kodiranje. Kao ulaz CBOW mreže šaljemo **one-hot** vektor, a kao izlaz očekujemo vektor reči "dobar". Računa se greška i menjaju se težine neuronske mreže. U toku predviđanja ciljane reči dobijamo pravu vektorsku reprezentaciju ciljane reči. Suštinski skriveni sloj neuronske mreže sadrži vektorske reprezentacije reči. Tačnost zavisi od konteksta, odnosno od broja reči koje uzimamo pre ciljane reči.

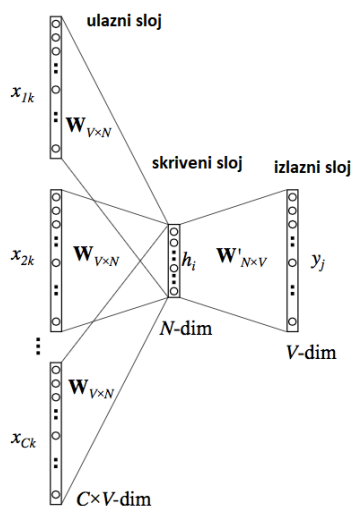
Na slici 3. možemo videti ilustraciju CBOW mreže koja ima kontekst dužine jedan. Ulazna ili kontekstna reč je **one-hot** kodiran vektor veličine V . Skriveni sloj sadrži N neurona, a izlaz je opet vektor dužine V . Izlazi se dobijaju kao izlazne vrednosti softmax funkcije. Neuroni

skrivenog sloja samo kopiraju zbir ulaza pomnožen sa težinama u sledeći sloj. Ne koristi se aktivaciona funkcija poput sigmoida, tanh ili ReLU. Jedina nelinearnost se dobija korišćenjem **softmax**-a u izlaznom sloju. Umesto jedne ulazne reči za kontekst možemo koristiti njih više.



Slika 3. CBOW model sa kontekstom dužine jedan

Model prikazan na slici 4. sadrži reči iz konteksta C . Kada se **W_{VN}** koristi za izračunavanje vrednosti težina skrivenih slojeva, uzima se prosek za svih C ulaznih kontekstnih reči.



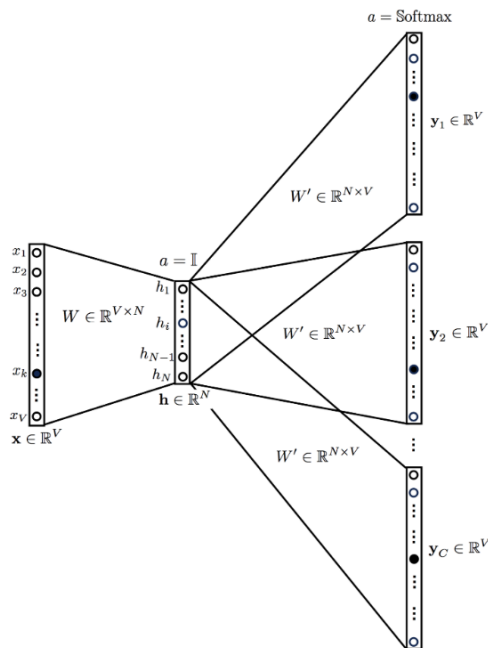
Slika 4. CBOW mreža sa ulazom dužine C

Upotrebom ove mreže možemo generisati vektorski prikaz reči korišćenjem konteksta. Postoji još jedan način da se učini isto. Moguće je koristiti ciljnu reč (reč čiju reprezentaciju tražimo), kako bi predvideli kontekst i tokom tog procesa proizvesti vektorske reprezentacije. Druga varijanta se zove **Skip-Gram** model.

4.2 Skip-Gram model

Skip-Gram model je, u suštini, samo obrnut CBOW model, slika 5. Izgleda kao invertovan višekontekstni CBOW model. U mrežu se unosi ciljna reč, a model daje C distribuciju verovatnoće. Za svaku poziciju u kontekstu dobijamo C verovatnoće podele V verovatnoća, po jednu za svaku reč. U oba slučaja mreža koristi propagaciju unazad za obučavanje.

Oba modela imaju svoje prednosti i mane. **Skip-Gram** model se pokazao kao bolji u slučaju da se radi sa malom količinom podataka, kao i u slučaju gde imamo retke reči. S druge strane, CBOV je brži i bolje se pokazao za predstavljanje češćih reči.



Slika 5. Skip-Gram model

U radu "Distributed Representations of Words and Phrases and their Compositionality" [2] su predstavljene osnove **word2vec** modela.

Oni su za obuku **Skip-gram** modela koristili veliki skup podataka koji se sastoji od različitih članaka. Kreiran je interni Google-ov skup podataka sa milijardu reči. Sve reči koje su se pojavile manje od pet puta su izbačene iz rečnika i kreiran je skup sa vokabularom veličine 692 hiljada reči. Trenirajući **Skip-gram** model na ovim skupom dobijena je tačnost od 72% kada su pogledu održanja konteksta.

5. ZAKLJUČAK

U ovom radu su predstavljene metode za enapsulaciju teksta, odnosno njegovo vektorsko predstavljanje. Trenutno ne postoji najbolje rešenje za ovaj problem ali se algoritmi razvijaju u dva pravca. Rešavaju se koristeći probabilističke modele kao i neuronske, najčešće rekurventne, mreže. Činjenica da word2vec kao i glove model zadržava deo semantike predstavlja veliku prednost pri analizi tekstova.

Zbog svog vektorskog oblika, vektori koji reprezentuju rečenice koje sadrže gramatičke i pravopisne greške se nalaze blizu vektora takvih rečenica sa ispravljenim greškama. Oba modela imaju svoje prednosti i mane tako da odabir zavisi prvenstveno od samog skupa podataka sa kojim se radi. Word2vec model daje bolje rezultate nad većim skupovima podataka pod uslovom da se omogući dovoljno vremena za treiranje modela.

Za dalje istraživanje, unapređivanje vektorizacije teksta upotrebom novih transformer modela bi predstavljalo zanimljivu temu koja bi potencijalno dovela do značajno boljih rezultata.

6. LITERATURA

- [1] Quoc Le, Tomas Mikolov, "GloVe: Global Vectors for Word Representation", Computer Science Department, Stanford University, Stanford, CA 94305, 2014.
- [2] Quoc Le, Tomas Mikolov, "Distributed Representations of Sentences and Documents", Google Inc, 1600 Amphitheatre Parkway, Mountain View CA 94043, May 2014.
- [3] Mikolov, Tomas & Chen, Kai & Corrado, G.s & Dean, Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. 2013.

Kratka biografija:



Milan Stanković rođen je u Subotici 1995. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva Neuroevolucija u Java programskom jeziku odbranio je 2018.god.
kontakt: milan.s@uns.ac.rs

PROCES RAZVOJA GENERIČKE POSLOVNE APLIKACIJE DEVELOPMENT OF GENERIC BUSINESS APPLICATION

Ilija Grbić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U današnje vreme sve više jača tendencija digitalizacije podataka iz različitih domena kako bi se olakšala njihova pretraga i brz pristup. Upotreba informacionih sistema postala je opšteprihvaćena u oblastima obrazovanja, medicine i privrede. Državna uprava i pravni sistem u Republici Srbiji postepeno primenjuju informacione sisteme u svakodnevnom poslovanju što se dalje odražava i na advokate. Advokatske kancelarije trenutno prvenstveno rade sa fizičkim ili eventualno skeniranim dokumentima, dok je primena namenskih softverskih alata slabo prisutna. Cilj ovog rada jeste prikaz procesa razvoja delom generičke aplikacije za evidentiranje i upravljanje pre svega advokatskih predmeta, ali i ostalih podataka neophodnih u radu advokatske kancelarije.

Gljučne reči: *Objektno orijentisano programiranje, veb aplikacije, Objektno orijentisani šabloni, Softver za pravnike*

Abstract – In modern times it is more and more common for data from varying domains to be stored in digital format for faster search and access. Usage of information systems is becoming defacto standard in education, medicine and economy. State Administration and legal system in the Republic of Serbia gradually apply information systems in everyday use, which in turn reflects on lawyers. Law firms currently mainly use physical documents and sometimes scanned documents, while application of domain specific software is quite rare. The goal of this paper is to summarize the development of partly generic application for case management and other necessary lawyer data.

Keywords: *Object oriented programming, Web applications, Object oriented patterns, Legal software*

1. UVOD

Upravljanje podacima i njihova evidencija predstavlja jedan od glavnih izazova sa kojima se advokati sadašnjice susreću. Svake godine se u sudove u Republici Srbiji predaju stotine hiljada predmeta, konkretno u prvih pet najvećih sudova primljeno je preko 400.000 [1]. Iako nijedan advokat ne brine ni blizu o tolikoj količini predmeta, jasno je da je pri radu sa tolikim podacima korisno primeniti tehnike iz oblasti velikih podataka (eng. *big data*) i nauka o podacima (eng. *data science*).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinac, vanredni prof.

Ove tehnike se primenjuju kada postoje prikupljeni podaci u velikoj količini, što dodatno daje na značaju sistemima za evidenciju podataka.

Prilikom postavljanja zahteva za razvoj aplikacija, pored komunikacije sa potencijalnim korisnicima, neophodno je analizirati postojeća rešenja u datoj oblasti. U oblasti pravnih sistema, uže namenjenih advokatima postoji mnoštvo stranih i domaćih rešenja. Najkorištenija rešenja na stranom tržištu su *PracticePanther* i *Clio* koja nude mnoštvo funkcionalnosti poput personalizovanog rada, automatske naplate, planiranja događaja, evidencije predmeta itd. Na domaćem tržištu korišćenje ovakvog namenskog softvera nije u velikoj meri zastupljeno. Advokatske kuće i dalje koriste softver drugih namena, poput *Paragraf Lex* koji je pre svega namenjen za pretragu baze podataka zakona.

2. RAZMATRANI ZAHTEVI

Razmatrane zahteve ćemo podeliti u dve grupe: esencijalni i opcioni. Podela je kreirana pre svega da bi se olakšala implementacija i što pre dobio prototip aplikacije koji podržava samo esencijalne mogućnosti.

Esencijalni zahtevi su:

1. Generička evidencija advokatskih predmeta
2. Registracija i prijava korisnika
3. Upravljanje korisničkim profilom
4. Evidencija kontakata
5. Otpremanje dokumenata
6. Interaktivno planiranje aktivnosti.

Opcioni zahtevi su:

1. Generisanje izveštaja za naplatu
2. Evidencija sudova
3. Brza pretraga dokumenata
4. Konverzija skeniranih dokumenata u tekst.

Generička evidencija predmeta je potrebna jer tipovi predmeta koji mogu da se pamte u sistemu nisu uvek isti. Metapodaci za različite predmete mogu biti značajno drugačiji tako da je jedan od glavnih problema kreiranje sistema koji može pamtititi različite vrste predmeta kao i njihove metapodatke. Cilj je da broj tipova predmeta i njihovi metapodaci ne budu predefinisani, već da mogu generički da se podese za odgovarajući region ili konkretne potrebe nekog korisnika.

Kako bi se obezbedila personalizovano radno iskustvo na aplikaciji, registracija i prijava korisnika je neophodna. Takođe ovo potencijalno može biti korisno i u slučaju kada je neophodno ograničiti pristup podacima u sistemu određenim korisnicima.

Potrebno je omogućiti generisanje izveštaja za naplatu na osnovu podataka u sistemu. Ovi izveštaji bi trebalo da se koriste dalje za naplatu usluga u skladu sa definisanim cenovnicima. Bitan detalj za ovaj slučaj korišćenja jeste omogućavanje personalizovanja šablona za kreiranje izveštaja u skladu sa zahtevima pojedinačnog korisnika.

3. KORIŠTENE TEHNOLOGIJE

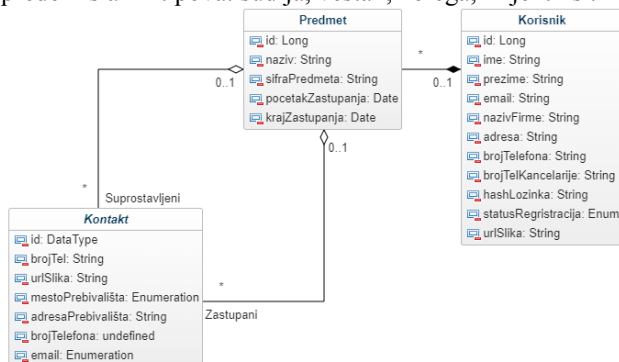
Sam softverski proizvod je realizovan kao *web* aplikacija. Svaka *web* aplikacija se sastoji od *backend*-a i *frontend*-a.

Backend je realizovan u *Java* programskom jeziku uz upotrebu *Spring frameworka*. Za čuvanje podataka se koristi relacionalna baza podataka. *Backend* komunicira sa ostatkom sistema putem HTTP zahteva kreiranih po REST arhitektonskom šablonu. Za generisanje izveštaja se koristi *JasperReports* biblioteka koja omogućava generisanje izveštaja na osnovu predefinisanih XML šablona [2]. Ovi šabloni su značajni jer omogućavaju personalizovano kreiranje šablona za na osnovu potreba pojedinačne advokatske kuće. *Elasticsearch* se koristi za indeksiranje dokumenata koje omogućava brzu pretragu otpremljenih tekstualnih dokumenata ne samo po nazivu već i po njihovom sadržaju [3].

Frontend je realizovan kao klijentska jednostranična *Angular* aplikacija (eng. *single-page application*). *Angular* predstavlja jedan od najpopularnijih *framework*-a, pored *ReactJS* i *VueJS*, za kreiranje *web* frontenda [4].

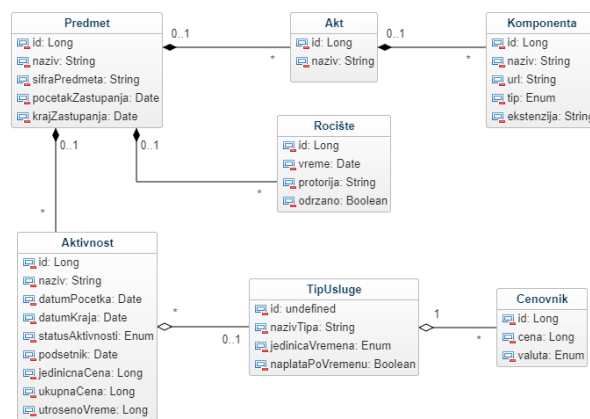
4. GENERIČKI PREDMET

Advokatski predmet je modelovan tako da svaki korisnik sistema ima listu predmeta na kojima radi. Sam predmet ima dalje sve relevantne podatke poput zastupane i suprostavljene strane, sudija angažovanih na slučaju, statusa predmeta, akata, ročišta i aktivnosti. Na slici 1. može se videti deo ovih metapodataka. Klasa *Kontakt* reprezentuje podatke o osobi od značaja u advokatskom sistemu i može imati proizvoljan tip, dok su neki od predefinisanih tipova: sudija, veštak, kolega, klijent i sl.



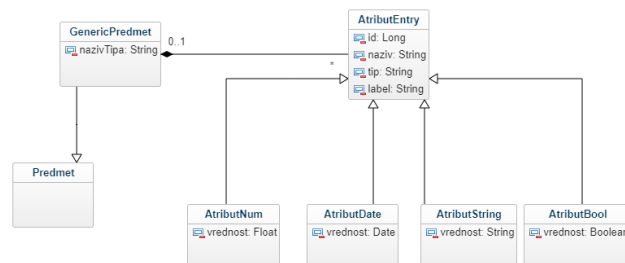
Slika 1. Deo klas modela predmeta

Aktivnosti predstavljaju jednu od najbitnijih delova sistema jer omogućavaju evidenciju svih bitnih podataka vezanih za naplatu advokatskih usluga. Aktivnost može imati proizvoljan tip kako bi se omogućila fleksibilnost korisnicima prilikom njihovog definisanja. Ovo je modelovano klasom *TipUsluge*. Omogućeno je i definisanje cena aktivnosti u zavisnosti od specifičnosti konkretnog predmeta, koje omogućava klasa *Cenovnik*.



Slika 2. Klas model veze predmeta, akata, ročišta i aktivnosti

Predmet pored pored prethodno navedenih metapodataka može imati dodatne metapodatke koji mogu biti različiti u zavisnosti od tipa predmeta [5]. Ovaj problem bi se mogao rešiti pomoću nasleđivanja gde bi se za svaki tip predmeta definisali dodatna neophodna polja. Međutim, ovakav pristup ima nekoliko problema, najočiglednije za svaki novi tip predmeta koji treba podržati morao bi se menjati izvorni kod i implementirati odgovarajuću klasu. Pored toga što sudski pravilnik može da se promeni, ukoliko želimo da podržimo upotrebljivost aplikacije i na stranom tržištu tj. u drugim pravnim sistemima, nepraktično je implementirati sve moguće predmete. Ovaj problem može biti rešen implementacijom generičke klase koja bi opisala bilo koji predmet, uz odgovarajuća ograničenja.



Slika 3. Klas model generičkog predmeta

Generički predmet pored nasleđenih polja iz klase *Predmet* ima listu dodatnih atributa. Svaki od ovih atributa ima naziv atributa, jedinstven na nivou jednog predmeta, labelu za prikaz na korisničkom interfejsu, tip podatka koji se čuva, kao i polje koje govori da li da se atribut prikazuje u skraćenom prikazu na korisničkom interfejsu.

Do sad je definisan samo način da se generički predmet sačuva, ali ne i način da se kreira. Prvi problem koji se nameće je definisanje metapodataka za sve predmete koje treba čuvati unutar sistema. Ovo je moguće rešiti ubacivanjem tih podataka u bazu podataka, međutim, prosečni korisnik, pa čak i administrator sistema, ne bi trebao da komunicira sa sistemom na ovom nivou. Kako bi se ovaj problem rešio definisanje podataka o podržanim predmetima je realizovano pomoću konfiguracionog fajla. Za format konfiguracionog fajla je odabran JSON, pre svega zbog lake ljudske čitljivosti fajla.


```

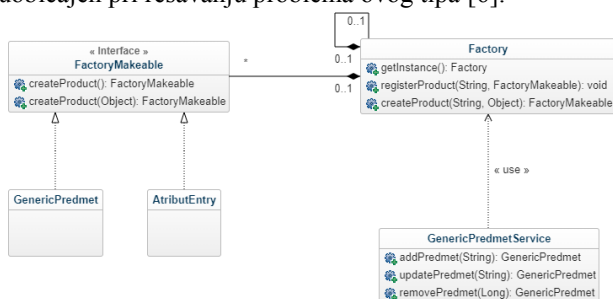
{
  "predmeti": [
    {
      "tip": "Privredni",
      "atributi": [
        {
          "naziv": "kazna",
          "label": "Kazna",
          "prikazi": false,
          "atr_tip": "String"
        },
        {
          "naziv": "vrstaOdluke",
          "label": "Vrsta odluke",
          "prikazi": true,
          "atr_tip": "Enum",
          "enum_values": [
            "USVOJEN",
            "ODBIJEN"
          ]
        }
      ]
    }
  ],
}

```

Slika 4. Primer konfiguracionog fajla za jednostavan tip predmeta

Konfiguracioni fajl se sadrži od liste predmeta koje korisnik želi da podrži gde svaki predmet ima naziv tipa i atribut koji su za njega neophodni. Polje *atr_tip* je obavezno i definiše tip atributa. Podržani tipovi su podržani su numerički (poput *Float* ili *Long*), datum (*Date*), proizvoljan tekst (*String*), logički tip (*Boolean*) i nabrojivi tip (*Enum*). Jedino za nabrojivi tip su potrebni dodatni metapodaci tj. lista svih opcija (*enum_values* lista). Sistem učitava konfiguracioni fajl i na osnovu njega zna koji tipovi predmeta su podržani kao i njihovu strukturu.

Kreiranje objekata klase *GenericPredmet* zahteva i generisanje objekata klase *AtributEntry* (slika 3.) u realnom vremenu. Kako bi se kreiranje što elegantnije realizovalo, potrebna je primena *Factory* šablona koji je uobičajen pri rešavanju problema ovog tipa [6].



Slika 5. Klas model fabrika šablona za kreiranje generičkih predmeta

5. KORISNIČKI INTERFEJS

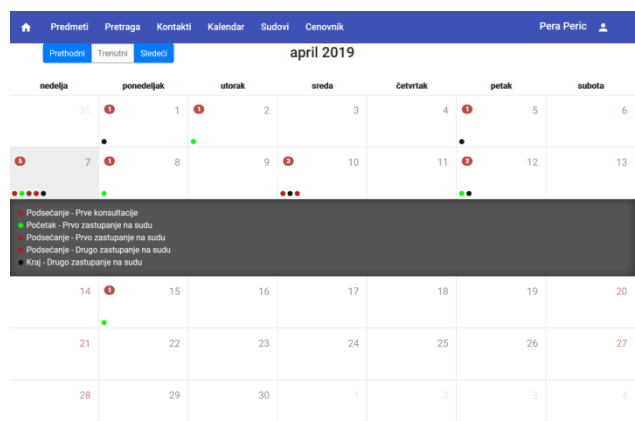
Bitan zahtev prilikom kreiranja korisničkog interfejsa je mogućnost da on funkcioniše na različitim uređajima. *Angular framework* podržava korišćenje *AngularMaterial* komponenti koje pružaju dobru podršku za prikaz kako na mobilnim uređajima poput telefona i tableta tako i na personalnim računarima.

Prilikom unosa predmeta pored podrazumevanih polja za svaki predmet potrebno je uneti i dodatna polja definisana u tipu generičkog predmeta.

Slika 6. Prikaz dela korisničkog interfejsa za unos predmeta

Na slici 6. se može videti primer izgleda ovakvog korisničkog interfejsa. Nakon što korisnik izabere tip predmeta, sa desne strane se prikazu izgenerisanja polja za unos specifičnih atributa.

Jedna od najbitnijih funkcionalnosti je prikaz aktivnosti u nekom vremenskom periodu na lak i intuitivan način. Ovo je realizovano prikazivanjem aktivnosti u formi kalendara. Svaka aktivnost ima tri bitne vremenske odrednice: početak, kraj i vreme podsećanja. Svako od ovih vremena je prikazano u kalendaru odgovarajućom bojom: početak zelenom, kraj crnom i podsećanje crvenom (slika 7.). Korisnik svako od vremena može *drag-n-drop* interakcijom da promeni. Aktivnosti kojima je vreme podsećanja prošlo se prikazuju korisniku u posebnoj listi notifikacija za podsećanje.



Slika 7. Prikaz aktivnosti u kalendaru

6. ZAKLJUČAK

Najkomplikovaniji deo prilikom procesa razvoja poslovnog sistema jeste projektovanje modela podataka. Da bi se ovaj korak kvalitetno realizovao, potrebno je prikupiti veliku količinu domenskog znanja kako bi se kreirala arhitektura koja pokriva sve slučajeve korišćenja. Sve veći broj predmeta, a samim time informacija i dokumenata, advokatima otežava posao. Čuvanje podataka u čistom tekstu na papiru usporava svakodnevni rad jer papir onemogućava jaku povezanost između dokumenata kao i laku pretragu.

Prednosti namenskog poslovnog sistema za advokate su samim tim neizmerne. Implementiran softver je od početka projektovan imajući u vidu pre svega advokatske kancelarije u Republici Srbiji i time ima veliku prednost s

obzirom na manjak konkurencije u oblasti. Uz dodatne konsultacije sa advokatima i naknadnu doradu, softver bi mogao da postane komercijalno rešenje za ove probleme. Očigledna mesta za unapređenja ovog softvera su napredno pretvaranje slika u tekst ili automatsko popunjavanje formi na osnovu postojećih papirnih dokumenata.

6. LITERATURA

- [1] <https://www.mpravde.gov.rs/cron/files/O/2018/O2018Y00.html> (pristupljeno u septembru 2019.)
- [2] <https://en.wikipedia.org/wiki/JasperReports> (pristupljeno u septembru 2019.)
- [3] <https://www.elastic.co/what-is/elasticsearch> (pristupljeno u septembru 2019.)
- [4] <https://existek.com/blog/top-front-end-frameworks-2019/> (pristupljeno u septembru 2019.)
- [5] <https://www.paragraf.rs/propisi/sudskiposlovnik-2.htm> (pristupljeno u septembru 2019.)
- [6] <https://www.oodeesign.com/factory-pattern.html> (pristupljeno u septembru 2019.)

Kratka biografija:



Ilija Grbić rođen je u Novom Sadu 1996. god
Diplomski rad na Fakultetu tehničkih nauka iz
oblasti Elektrotehnike i računarstva –
Biblioteka za neuronske mreže u
programskom jeziku C++ odbranio je
2018.god. kontakt: ilijagrbić@uns.ac.rs

ANALIZA I OBRADA TEKSTA POMOĆU RAZLIČITIH MODELA TEMA TOPIC MODELS FOR TEXT ANALYSIS

Olivera Hrnjaković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad opisuje trenutne mogućnosti i ograničenja postojećih algoritama za izdvajanje tema iz teksta. Dat je teorijski prikaz popularnih modela tema uz sve neophodne korake analize i obrade teksta koji se izvršavaju pre slanja podataka na ulaz modela. Praktičan deo rada je izdvajanje tema iz pitanja sa sajta Stack overflow. Upoređeni su LSA, PLSA i LDA pristup, a evaluacija modela je izvršena određivanjem koherentnosti tema odgovarajućim merama, imenovanjem tema i analizom njihove vizuelizacije u prostoru. Kako modeli tema unapred zahtevaju navođenje broja tema koje će biti izdvojene iz teksta, deo rada posećen je optimizaciji hiperparametara. Izabrani model za modelovanje tema jeste LDA sa 6 tema. Da bi se dobila numerička procena performansi modela 30 pitanja je ručno označeno imenima dobijenih tema i simuliran je klasifikacioni model. Ova pitanja su korišćena kao test skup podataka u kreiranom LDA klasifikacionom modelu. Postignuta je uspešnost od 77% tačnosti.

Ključne reči: modelovanje tema, analiza teksta, LDA

Abstract – This paper describes the current capabilities and limitations of existing topic modeling algorithms. A theoretical overview of popular topic models was given, along with all the necessary analysis and text processing steps that should be performed on the input data. The practical part of the paper is to extract topics from questions from the Stack overflow site. LSA, PLSA and LDA approaches were used and evaluated using coherence, perplexity, naming techniques and topic visualization in space. To get the best performance for topic modeling, we estimated the best topic number. The results showed that the best model is LDA and the best topic number is 6. In order to obtain a numerical evaluation of the model performance, 30 questions were manually annotated with the names of the topics acquired. In that way, we simulated a classification model. These questions were used as a test data set in the created LDA classification model. The accuracy of the classification model was 77%.

Keywords: Topic modeling, text analysis, LDA

1. UVOD

Ekspanzija podataka i nemogućnost njegove obrade na tradicionalan način dovela je do porasta količine neobrađenih podataka čime se potencijalno gube informacije. Kao rešenje ovog problema pojavile su se tehnike veštačke inteligencije koje mogu da rade sa podacima brže od čoveka. Nestruktuirani podaci u vidu teksta predstavljaju većinu neobrađenih podataka. Oblast veštačke inteligencije koja radi sa analizom teksta na osnovu jezika naziva se obrada prirodnog jezika. Modelovanje tema je deo obrade prirodnog jezika i jedan je način za određivanje značenja teksta. Poznavanje semantike teksta pruža mogućnosti za upoređivanje tekstova i dalju primenu u sistemima za preporuku, komunikaciju, prevodjenje i slično. U ovom radu biće opisane trenutne mogućnosti postojećih algoritama za izdvajanje tema iz teksta.

U narednoj sekciji izložen je teorijski prikaz obrade prirodnog jezika i modelovanja tema. Treća sekcija sadrži opis problema koji je obrađivan u radu i pregled postojeće literature. Četvrta sekcija sadrži metodologiju predloženog rešenja. U petoj sekciji objašnjena je eksperimentalna evaluacija rezultata, dok poslednja sekcija zaključuje rad i predlaže pravce potencijalnog daljeg istraživanja.

2. OBRADA PRIRODNOG JEZIKA I MODELOVANJE TEMA

Računari su davno prevazišli čoveka u mnogim veštinama poput matematičkog računa i memorisanja podataka. U nekim oblastima dosegli su jednakost sa čovekom, ali i dalje je izazov napraviti sistem koji bi mogao da nadmaši čoveka u rešavanju problema koji iziskuju kreativnost i reagovanje na do sada neviđene probleme na način svojstven čoveku.

Primer ovakvog sistema jeste sistem koji radi sa jezikom. Jezik je “živa” materija, menja se zajedno sa ljudima, deli se u dijalekte, vremenom se pojavljuju nove reči, izrazi i idiomi, a postojeći se gube. Ovim problemima bavi se obrada prirodnog jezika.

Modeli tema (engl. *Topic models*) su klasa probablističkih latentnih promenljivih modela za tekstualne podatke koji služe da predstavljaju tekstualne podatke u obliku raspodele tema [1]. Ovi modeli kreiraju neku vrstu sumarizacije ili filtriranja teksta u formatu koji se može interpretirati kao tema (engl. *topic*). Modeli tema spadaju u nenadgledani oblik učenja i predstavljaju neku vrstu klasterovanja “sličnih” reči. Dakle, formiraju se apstraktne teme koje u stvari predstavljaju grupu povezanih reči.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinać, van. prof.

Postoje dve vrste modela tema:

- metode zasnovane na sadržaju teksta i
- prediktivne metode.

Razlika između ovih metoda jeste u tome što metode koje su bazirane na sadržaju teksta rade na osnovu statistike i prebrojavanja. Sračunava se koliko često se neka reč pojavljuje u kombinaciji sa susednom rečju u velikom tekstualnom korpusu, a potom se taj broj mapira na vektor koji predstavlja svaku reč.

Prediktivni modeli direktno predviđaju reč na osnovu njoj susednih reči koristeći naučene vektore za ugradnju. U ovom radu izloženi su algoritmi koji analiziraju sadržaj teksta. Izloženi su LSA, PLSA i LDA koji su trenutno aktuelni algoritmi u oblasti modelovanja tema. Pored njih opisan je i *Deep learning* pristup koji predstavlja osnovu za budući razvoj ovih algoritama.

3. OPIS PROBLEMA

Stack overflow je sajt na kome ljudi koji se bave programiranjem mogu da postavljaju pitanja i daju odgovore. Sajt posećuje veliki broj korisnika, što znači da se na njemu svakodnevno postavlja veliki broj pitanja i daje veliki broj odgovora. Svako pitanje ima naslov u kome skraćeno treba objasniti njegovu suštinu i sam tekst pitanja koji sadrži detaljno objašnjenje pitanja, često uz primere i delove koda. Cilj sajta jeste da na što brži način omogući korisnicima da reše problem. Zato je neophodno da pitanja budu semantički povezana i grupisana sa sebi međusobno sličnim pitanjima.

Postojanje sistema koji obrađuje tekstove u naslovu i tekstu komentara i na osnovu toga povezuje slična pitanja veoma je važno za ovakav sajt. Analiziranje sadržaja pitanja je značajno, jer će opisi sličnih problema u vidu značenja tekstova biti slični. U ovom radu biće prezentovan sistem koji izvlači teme iz teksta i na taj način predstavlja njegovo značenje. Ovakva reprezentacija teksta mogla bi poslužiti kao deo sistema za preporuku ili deo sistema koji eliminiše duplikate tj. upućuje korisnike na slična pitanja.

3.1. Opis skupa podataka

Korišćen je skup podataka *Python Questions from Stack Overflow* [2] preuzet sa sajta *Kaggle*. Ukupan broj pitanja u skupu podataka je 607282, a za potrebe ovog projekta iskorišćen je deo skupa podataka od 100000 pitanja. Tekstovi pitanja pisani su na engleskom jeziku. Skup podataka obuhvata 3 fajla, fajl sa pitanjima, sa odgovorima i sa tagovima. Za potrebe projekta korišćen je fajl sa pitanjima, fajl sa tagovima je eksperimentalno korišćen radi kreiranja potencijalne metrike za evaluaciju, dok fajl sa odgovorima nije korišćen. Korišćeni su atributi: *Title* (Naslov pitanja) i *Body* (Telo odnosno tekst pitanja) koji su spojeni u jedan atribut *Text* koji je dalje korišćen u radu.

3.2. Pregled postojeće literature

Veštačka inteligencija je jedna od najpopularnijih oblasti u kompjuterskim naukama u današnje vreme. Jedan od pokazatelja njene popularnosti je velik broj objavljene

stručne literature, naučnih radova i članaka. Modelovanje tema je dostiglo visok nivo, a pristupi koji se koriste se konstantno unapređuju radi poboljšanja performansi.

Tehnike probabilističkog modelovanja tema za formiranje sistema za preporuku naučnih radova iskorišćene su u [3]. Korišćen je skup podataka sa sajta *CiteULike* koji je sadržao veliki broj citata naučnih radova. Za svaki rad izdvojeni su naslov i abstrakt koji su potom transformisani u odgovarajući oblik da bi bili pogodan ulaz u model. Na osnovu ovoga formirano je obeležje koje je korišćeno u ovom radu kao osnovno obeležje na osnovu koga je vršena analiza. Obrada tekstualnog sadržaja sa sajta *Stack overflow* bila je tema rada [4]. Ideja rada bila je izdvojiti i grupisati članke i diskusije na osnovu tema koje se javljaju u sadržaju.

Za ovo je korišćen LDA kao mehanizam modelovanja tema. Pre nego što je primenjen LDA, urađeno je preprocesiranje teksta. Najpre je iz teksta odbačeno sve što nije rečenica, odnosno izbačeni su delovi koda. Delove koda je jednostavno izbaciti iz teksta zato što su na sajtu označeni html tagom `<code>`.

Takođe, svi drugi HTML tagovi koji opisuju izgled sadržaja na sajtu su izbačeni. Nakon toga obavljeno je uklanjanje najčešćih engleskih reči poput "a", "the" "is", koje ne utiču na izvlačenje tema iz teksta. Kako je reč o istom skupu podataka, preprocesiranje u ovom radu je urađeno po ugledu na dati rad.

Analiza teksta i modelovanje tema rađeno je u radu [5] koji se bavi poređenjem LDA i LSA modela kako bi se formirao automatski sistem za preporuku filmova. Ovaj rad korišćen je zbog upoređivanja tehnika modelovanja tema i metode evaluacija modela. Sistem za preporuku zamišljen je da funkcioniše bez bilo kakve informacije o korisniku, odnosno u pitanju je sistem za preporuku baziran na sadržaju filmova. Metod evaluacije bio je eksperimentalna ručna evaluacija: 30 korisnika je ocenilo rad sistema za preporuku. Rezultati su pokazali da je LSA bio uspešniji u davanju preporuka od LDA.

4. METODOLOGIJA

Tekst je najpre pripremljen za ulaz u modele preprocesiranjem. Algoritmi za modelovanje tema očekuju ulazne tekstualne podatke u tačno određenom formatu. Kao ulaz u većinu modela potrebno je proslediti rečnik i *bag-of-words* zbirku reči. Prvi korak preprocesiranja bila je tokenizacija. Tokenizacija je proces dobijanja tokena iz reči. Za dobijanje tokena korišćene su funkcije iz biblioteke *gensim*. Nakon što je dobijen token nad njim su izvršeni stemovanje i lematizacija kako bi se dobio koren reči i model ne bi bio zavisn od vrste reči. Nakon stemovanja i lematizacije formirani su rečnik i *bag-of-words* model (tekstualni korpus). Za ove korake u preprocesiranju takođe postoji podrška u *gensim*-u za engleski jezik.

Iako je često teško odrediti optimalan broj tema kada se formiraju modeli tema, postoji nekoliko metoda koje služe kao smernice u ovom poslu. Čovek najtačnije određuje koliko su dobijene teme povezane sa tekstom od koga smo pošli, ali bi za ovakav posao bilo potrebno angažovanje

velikog broja ljudi, a posao bi bio izuzetno vremenski zahtevan. Zbog toga je gotovo nemoguće odrediti uspešnost modela na ovaj način. Korišćene su *coherence* [6] i *perplexity* [7] mere, čijom kombinacijom je određeno da je 6 optimalan broj tema za LSA i LDA modele. Rezultati dobijeni merom nedoumice (*perplexity*) se vide u tabeli 1. Bolji model ima manju meru nedoumice, a najmanja mera postignuta je za 6 i 7 tema, za TFIDF korpus.

Tabela 1. Vrednost mere nedoumice za LDA model

	5 tema	6 tema	7 tema
Običan korpus	-7.030	-7.025	-7.021
TFIDF korpus	-7.28	-7.32	-7.32

Da bi se donela konačna odluka o optimalnom broju tema nad LDA i LSA modelom ispitana je mera koherencije koja je pokazala da je 6 optimalan broj tema za oba modela. Za PLSA model ne postoji kompletna implementacija zbog čega nije moguće ispitati uspešnost modela u odnosu na ove mere.

Korišćen je običan tekstualni korpus ali i korpus dobijen TFIDF metodom, a na kraju su upoređeni dobijeni rezultati. TFID ili tf-idf je metoda koja se koristi u pretraživanju informacija kako bi se utvrdila važnost reči za dokument u odnosu na neki tekstualni korpus [8].

Upoređena su 3 modela: LSA, PLSA i LDA. Za kreiranje LSA i LDA modela korišćen je paket *models* iz *gensim* biblioteke. Model se kreira tako što se konstruiše *LsiModel* ili *LdaModel* objekat kojima se prosleđuju parametri od kojih je minimalno proslediti tekstualni korpus, rečnik i broj tema.

Kada je reč o kreiranju PLSA modela, koriste se dva pristupa. Prvi pristup koristi latentne promenljive i verovatnoću i algoritam maksimizacije očekivane verodostojnosti - odnosno EM algoritam. Mane ovog pristupa su sporo izvršavanje i teško rukovanje sa modelom kada je reč o obradi novih dokumenata [9]. Drugi pristup koristi faktorizaciju matrice, odnosno NMF algoritam (nenegativnu faktorizaciju matrice).

Ovaj način simulacije PLSA je znatno brži u odnosu na model latentnih promenljivih i jednostavniji je za implementaciju s obzirom da postoji gotova implementacija u biblioteci *sklearn*.

5. ANALIZA REZULTATA

Kao glavni metod evaluacije modela odabrana je ručna analiza tema, a u obzir su uzeti i rezultati dobijeni na osnovu mera nedoumice i koherentnosti [10]. Svaka tema prikazana je preko svojih 10 najfrekventnijih reči. Najpre su definisane metode imenovanja tema na osnovu kojih su teme imenovane [11]. U modelima tema počeli smo od formiranja latentnog sloja kako bi dobili upotrebljiviji model. Međutim u fazi imenovanja tema i evaluacije modela cilj je razbiti u delove latentni sloj kako bi shvatili njegovo značenje. Reči koje se javljaju u pronađenim temama treba objediniti tako da opišu semantiku tih tema. LDA i LSA su posebno pogodni za

ovakav pristup pošto su u temama reči navedene zajedno sa svojom frekvencijom ponavljanja čime je predstavljen značaj svake reči u temi. Analizom dobijenih tema metodom PLSA pokazano je da se reči iz tema ponavljaju u više tema i da skup reči ne predstavlja celinu koju je moguće opisati ili imenovati.

Najbolji model na osnovu mera nedoumice i koherencije pokazao se LDA model. Analizom reči u temama za modele LSA i LDA zaključeno je da je za LDA model moguće imenovati 5 tema, a za LSA 4.

Zbog toga je LDA model proglašen za najbolji model za dati skup podataka. Prikaz dobijenih imenovanih tema vidi se na slici 1.

	Stringovi i tekst	Kolekcije	Objektno programiranje	Veb programiranje	Topic 5	Fajl sistem i komandna linija
0	string	list	class	django	datetim	subprocess
1	charact	valu	method	file	pyqt	command
2	encod	function	decor	instal	typeerror	script
3	unicod	string	object	error	slice	stdout
4	regex	file	attribut	script	twitter	popen
5	text	array	instanc	server	excel	file
6	match	dictionari	function	user	player	scrapi
7	lxml	code	inherit	work	oauth	shell
8	express	number	subclass	code	sud	cython
9	html	lik	defin	window	chart	swig

Slika 1. Imenovane teme za najbolji model (LDA)

Konačna evaluacija je izvršena ručno, analizom nekoliko nasumično izabranih dokumenata odosno pitanja iz test skupa podataka. Za test skup podataka je izdvojeno 10000 pitanja. Ručno je labelirano 30 dokumenata imenima tema za LDA model. Svaki dokument označen je sa najmanje jednom a najviše šest oznaka. Korišćena je metrika koja je brojala koliko označenih tema je model pogodio. Dakle to znači da su u obzir uzeti samo *true positives*, odnosno model nije "kažnjavan" za pogrešno izdvojene teme. Rezultati LDA modela pokazali su 30 pogodjenih labela od ukupnog 39 označenih, što znači da je model za ovaj uzorak imao uspešnost od 77%.

6. ZAKLJUČAK

U ovom radu su predstavljeni osnovni koncepti modelovanja tema kroz teorijski i praktični prikaz. Definirano je modelovanje tema kroz obradu prirodnog jezika, objašnjene su matematičke osnove na kojima se ova oblast zasniva kao i aktuelno stanje u ovoj oblasti i potencijalni pravci budućeg razvoja. Poseban akcenat stavljen je na transformaciju teksta, eksploratornu analizu i vizuelizaciju podataka, jer su to koraci koji prave razliku između dobrih i loših modela. poređeni su LSA, PLSA i LDA. Korišćeni skup podataka su pitanja sa sajta *Stack overflow*.

Za određivanje performansi modela kombinovano je nekoliko metoda: mere koherentnosti i nedoumice, vizuelizacija tema u prostoru i analiza koherentnosti ručnim imenovanjem tema. Utvrđeno je da su korišćenjem LDA modela dobijene najbolje teme.

Da bi se dobila numerička procena performansi modela 30 pitanja je ručno označeno imenima dobijenih tema i simuliran je klasifikacioni model.

Ova pitanja su korišćena kao test skup podataka u simuliranom LDA klasifikacionom modelu. Postignuta je uspešnost od 77% tačnosti.

Praktično rešenje predstavljeno u ovom radu je polazna tačka u rešavanju opisanog problema. Da bi se ovaj predlog rešenja unapredio najpre bi trebalo za obučavanje modela koristiti veću količinu podataka. Tačnije bilo bi potrebno pronaći optimalnu količinu podataka za dati problem.

Još jedno unapređenje rešenja mogli bi biti dobijeno ako se broj reči ili karaktera u pitanjima ograniči nekim brojem. Ovako nešto moglo bi se uraditi tehnikama za sumarizaciju teksta i time bi se potencijalno izbalansirao skup podataka. Konačno neophodno bi bilo unaprediti metriku za evaluaciju modela kako bi se modeli mogli egzaktnije uporediti. To bi moglo biti postignuto ručnim označavanjem pitanja imenima tema, modifikacijom tagova koji već postoje za pitanja ili kombinacijom ove dve metode.

7. LITERATURA

- [1] Topic model. In Wikipedia, The Free Encyclopedia. Retrieved August, 2019, from https://en.wikipedia.org/wiki/Topic_model
- [2] Python Questions from Stack Overflow Retrieved from <https://www.kaggle.com/stackoverflow/pythonquestions>
- [3] Wang, Chong, and David M. Blei. "Collaborative topic modeling for recommending scientific articles." Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011.
- [4] Barua, Anton, Stephen W. Thomas, and Ahmed E. Hassan. "What are developers talking about? an analysis of topics and trends in stack overflow." Empirical Software Engineering 19.3 (2014): 619-654.
- [5] Bergamaschi, Sonia, Laura Po, and Serena Sorrentino. "Comparing Topic Models for a Movie Recommendation System." WEBIST (2). 2014.
- [6] Mimno, David, et al. "Optimizing semantic coherence in topic models." Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, 2011.
- [7] Perplexity To Evaluate Topic Models Retrieved from <http://qpleple.com/perplexity-to-evaluate-topic-models/>
- [8] tf-idf. In Wikipedia, The Free Encyclopedia. Retrieved August, 2019, from <https://en.wikipedia.org/wiki/Tf%E2%80%93idf>
- [9] Source code. <https://github.com/laserwave/plsa/blob/master/plsa.py>
- [10] Evaluate Topic Models: Latent Dirichlet Allocation (LDA) Retrieved from <https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0>
- [11] Binkley, David, et al. "Understanding LDA in source code analysis." Proceedings of the 22nd international conference on program comprehension. ACM, 2014.

Kratka biografija:



Olivera Hrnjaković rođena je u Novom Sadu 1995. god. Osnovne studije je upisala 2014. godine na Fakultet tehničkih nauka, odsek Računarstvo i automatika. Osnovne studije je završila 2018. godine, nakon čega upisuje master akademske studije na Fakultetu tehničkih nauka, smer Inteligentni sistemi.

VEB APLIKACIJA BAZE RETKIH BOLESTI NA SRPSKOM JEZIKU
WEB APPLICATION OF A DATABASE OF RARE DISEASES IN SERBIANIvana Tešanović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Predmet ovog rada jeste implementacija web aplikacije koji omogućava pristup informacijama o retkim bolestima na srpskom jeziku, kao i implementacija alata koji ubrzavaju i olakšavaju proces prevođenja. Rad je podeljen na više poglavlja, počevši od istorije u medicini, preko trenutnog stanja u svetu po pitanju retkih bolesti do opisa krajnjeg rešenja i tehnologija kojima se služilo da se dođe do istog.

Ključne reči: Retke bolesti, OrphaNet, Baza retkih bolesti, Umbraco, C#, MSSQL

Abstract – Subject of this work is an implementation of a web application which gives possibility of accessing information about rare diseases in serbian language as well as an implementation of tools that speed up the translation process and make it more easier. The paper is divided into multiple parts, starting with the history of medicine through rare diseases today until the final solution and the technologies that made it happen.

Keywords: Rare diseases, OrphaNet, Database of rare diseases, Umbraco, C#, MSSQL

1. UVOD

Dostupnost informacija o retkim bolestima nikada nije bila na zavidnom nivou. Čak i sad, u XXI veku, postoje mnogi ljudi koji još uvek nemaju ni načina da stignu do istih. Napravljen je određen broj rešenja koji delimično zadovoljavaju potrebe, ali samo na engleskom ili drugim globalno rasprostranjenim jezicima, dok na srpskom jeziku nije postojao nikakav izvor informacija.

Iz tih razloga, i mnogih drugih, javila se ideja za kreiranjem web sajta koji će moći da sadrži sve trenutno dostupne informacije o retkim bolestima na srpskom jeziku.

Sam rad je podeljen na više poglavlja. Prvi deo rada fokusiran je na motivaciju za pokretanje ovakvog projekta, kao i na to kakvo je trenutno stanje u svetu i koja rešenja već postoje. Takođe, napravljen je kratak uvod o tome šta su retke bolesti i kako i koliko je ljudi njima pogođeno. Nakon toga, dolazi se do dela koji opisuje šta je korišćeno od tehnologija i alata i na koji način su olakšali implementaciju web sajta i ostalih delova projekta.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinac, red. prof.

2. RETKE BOLESTI I REŠENJE PROBLEMA

Kako se korača ka novom milenijumu, postaje vrlo jasno da biomedicinske nauke ulaze u najzanimljiviju fazu svog razvoja. Paradoks tome, medicinska praksa prolazi kroz fazu nesigurnosti kako u razvijenim tako i u zemljama u razvoju [1].

Međutim, zdravstvena nega je grana koja se konstantno razvija. Kako nove tehnologije i mogućnosti podstiču sposobnosti doktora i istraživača, pacijenti dobijaju nova lečenja, praćenja i lekove.

2.1. O retkim bolestima

U Sjedinjenim Američkim Državama, retka bolest je stanje koje pogađa 1 u 1250 osoba. Ova definicija je kreirana od strane Kongresa u *Orphan Drug Act*-u 1983. S druge strane, u Evropskoj Uniji bolest se definiše kao retka ako pogađa 1 u 2000 osoba. Trenutno je otkriveno 6-7 hiljada retkih bolesti dok je broj obolelih procenjen na 300 miliona.

Do danas, uzrok velikog broja retkih bolesti ostaje nepoznat. Veoma dugo vremena je polje retkih bolesti bio nepoznat teren i nisu postojala odgovarajuća istraživanja koja su se time bavila. Čak ni danas nisu na zadovoljavajućem nivou. Upravo zbog manjka naučnog i medicinskog znanja, mnogim pacijentima nisu dijagnostikovane bolesti i kao takvi najviše pate u primanju odgovarajuće podrške i lečenja. Za mnoge retke bolesti ne postoji lek, međutim, sa odgovarajućim lečenjem i medicinskom negom, može da se poboljša kvalitet života pacijenata, kao i da im se produži životni vek.

Pokretač projekta, udruženje „Život“, pronalazi inspiraciju za njegov razvoj u postojećem rešenju - web sajtu *OrphaNet*-a.

2.2. O projektu

Projekat je implementiran kao web aplikacija i podeljen je na više delova. Neki od delova pripadaju samoj aplikaciji, dok su drugi eksterni i ne pripadaju aplikaciji. Pod internim delovima podrazumevaju se web sajt, program za prevođenje i forum, dok su eksterni delovi *Crawler*, *DB Initializer* i program za import i eksport excel fajlova.

Web sajt je dostupan svima i nije potreban nikakav vid autentifikacije da bi mu korisnik pristupio. Deo sajta koji zahteva autentifikaciju i autorizaciju jeste program za prevođenje i validaciju prevoda i jedan deo foruma. Od uloga na web sajtu razlikujemo prevodioce i validatore, koji, respektivno, prevode retke bolesti i validiraju prevode.

Crawler predstavlja deo projekta koji je zadužen za prikupljanje podataka sa engleske verzije ovog sajta tj. *OrphaNet*-a. *DB Initializer* opisuje program koji ubacuje preuzete podatke u bazu podataka. Program za import i eksport radi sa *excel* fajlovima na način da izvlači podatke iz *excel* fajlova i vrši njihovo importovanje u bazu podataka i obrnuto, izvlači podatke iz baze podataka radi eksportovanja u *excel*.

2.3. O rešenju

OrphaNet [2] poseduje sopstvenu bazu podataka, trenutno najveću o retkim bolestima na internetu, koja je dostupna svima. S tim u vezi, implementiran je *Crawler* koji je preuzeo sve od informacija direktno sa *web* stranica, a zatim i *DB Initializer* koji je te informacije uneo u bazu podataka.

Podaci o retkim bolestima koji su rezultat rada *Crawler*-a smeštaju se u *json* fajl tako da je svaka retka bolest odvojena odgovarajućim tagovima i može se identifikovati *Orpha* brojem. *Json* fajl, kao ulazni parametar u *DB Initializer*, biva transformisan u *sql* skriptu koja se automatski izvršava nad bazom podataka. Na taj način podaci bivaju unešeni i spremni za korišćenje od strane aplikacije.

Korisnik *web* sajta ima potpunu slobodu da se kreće kroz njega, da pretražuje retke bolesti, kao i da čita novine o samom udruženju pa i generalnom razvoju u svetu retkih bolesti.

Jedan od najbitnijih delova sajta, pretraga, može se pronaći na dva odvojena mesta na sajtu i obuhvata brzu i proširenu pretragu. Brza pretraga vrši pretraživanje po nazivu bolesti, odnosno unešeni pojam mora da bude sadržan u nazivu ili nazivu sinonima retke bolesti da bi bio kandidat za rezultat.

Odlaskom na stranicu za proširenu pretragu, proširuje se lista atributa za pretragu (*Slika 1*) pa se tako može pretraživati i po *Orpha* broju, *ICD-10*-u i *OMIM*-u. U slučaju ova tri atributa, unešeni pojam mora biti identičan kao u bazi podataka da bi se retka bolest našla u rezultatima.

Slika 1. Forma proširene pretrage

Takođe, postoji mogućnost i pretrage po prvom karakteru u nazivu ili nazivu sinonima retke bolesti, na koju se može otići klikom na „Abecedna lista bolesti“ u formi pretrage na Slici 1.

Drugi podjednako bitan deo sajta predstavlja program za prevođenje. Da bi mu se pristupilo, neophodno je da korisnik bude ulogovan na sistem i da pripada bar jednoj od dveju korisničkih grupa: prevodiocima ili validatorima. Kao što sama reč kaže, prevodioci se bave prevodom retkih bolesti, dok validatori proveravaju

podnešene prevode i u skladu sa tim ih odobravaju ili odbijaju.

Klikom na link za prevođenje, odlazi se na stranicu koja sadrži formu za prevođenje. Forma je podeljena na dva dela (*Slika 2*). Levi deo sadrži *input* polja u koje će se upisati prevod na srpskom jeziku onih podataka koji se nalaze u desnom delu.

Slika 2. Jedan deo forme za prevođenje

Retka bolest koja se učitava u formu mora imati inicijalno stanje, odnosno isto koje je dobila prilikom upisa u bazu. Prilikom podnošenja prevoda, podaci se upisuju u bazu kao kandidati za validaciju. Logovanjem na sistem, validator ima pristup stranici za validaciju. Izgled stranice je u potpunosti isti kao i kod prevođenja, s tim što u levom delu forme su učitani prethodno podnešeni prevodi. Validator ima mogućnost da, pregledom odgovarajućeg prevoda, izmeni neke od podataka i na kraju prihvati ili odbije prevod. Ukoliko se prevod prihvata, retka bolest postaje kandidat rezultatima za pretragu odnosno sigurna je za prikaz na *web* sajtu.

Forum *web* sajta se bavi komunikacijom između korisnika foruma i administratora sajta. Takođe, na ovaj način korisnici mogu i između sebe da komuniciraju i razmenjuju iskustva i mišljenja. U osnovi foruma i *web* sajta nalazi se isti cilj a to je pružanje korisnicima najviše moguće kvalitetnih i ispravnih informacija, koje do sada nisu bile dostupne na srpskom jeziku. Prednost kod foruma jeste što su korisnici u direktnoj komunikaciji, što znači da mogu postavljati konkretna pitanja kao i diskutovati o specifičnim temama.

Još neke od stvari na koje se forum fokusira jesu i novosti u vezi sa zakonima, regulativama kao i dostignućima u polju retkih bolesti. Neophodno je da korisnik bude registrovan i ulogovan na forumu da bi obavljao pomenute akcije, dok za sam pristup i pregled dostupnih informacija autentifikacija nije potrebna. Još jedna od mogućnosti jeste pretraživanje, koje se vrši na nivou celog foruma, odnosno na nivou svih korisnika, diskusija, kategorija i oznaka.

Program za import i eksport napravljen je kao alternativna programu za prevođenje. Nemogućnost korišćenja interneta u pojedinim situacijama dovela je do toga da se pronađe drugi način kojim će moći da se postigne isti cilj. Na kraju se ispostavlja da je metoda eksportovanja u fajlove efikasnija, jer je rad sa *excel*-om rasprostranjeniji od korišćenja *web* sajta. Mogućnost za ubacivanje više retkih bolesti u jedan *excel* fajl je dodato da bi se uštedelo na vremenu i da bi se olakšao sam proces prevođenja. Retke bolesti unutar jednog fajla se razdvajaju upotrebom *sheet*-ova, gde je svaka retka bolest je opisana unutar jednog *sheet*-a.

Entitet retke bolesti koji zadovoljava uslove za eksport njenih podataka zahteva nepromenjenost u podacima odnosno inicijalno stanje pri ubacivanju u bazu. Kao prvi korak u ubacivanju podataka u *excel* fajl jeste inicijalizacija istog, upisivanjem naziva redova i kolona. Od kolona postoje *Fields*, *Serbian* i *English*. *Fields* kolona opisuje nazive atributa koji se pridružuju retkoj bolesti, uključujući i sinonime i sadržaj. Druge dve kolone opisuju na kom su jeziku podaci koji će se u njihovim ćelijama renderovati. Eksportom se ubacuju vrednost u *English* kolonu, čitanjem odgovarajućih podataka iz baze i mapiranjem na attribute u *Fields* koloni.

Ovako popunjen fajl (*Slika 3*), šalje se stručnjacima na prevođenje. Kada osoba odgovorna za prevod unese sve potrebne podatke u Serbian kolonu svih fajlova, oni se učitavaju u program i vrši se importovanje u bazu.

Fields	English	Serbian
Pun naziv bolesti	Duchenne muscular dystrophy	Dišeno mišićna distrofija
Skraćeni naziv bolesti	-	-
Definicija	Duchenne muscular dystrophy (DMD) is a neuromuscular disease characterized by rapidly progressive muscle weakness and wasting due to degeneration of skeletal, smooth and cardiac muscle.	Dišeno mišićna distrofija (DMD) je neuromuskularno oboljenje, koje se karakteriše rapidno progresivnom slabošću mišića i hipotrofijom istih, zbog degeneracije poprečnoprugaste, glatke i muskulature srca.
Sinonimi	Severe dystrophinopathy, Duchenne type DMD	Teška distrofinopatija, Dišenov tip, DMD
Kategorija	-	-
Potkategorija	-	-
Naziv bolesti na stranom jeziku	-	-
Orpha broj	98896	98896
Učestalost	1-9 / 100 000	1-9 / 100 000
Nasledivanje	X-linked recessive	X vezano, recesivno
Period početka bolesti	-	-
ICD-10	G71.0	G71.0
OMIM	310200	310200
UMLS	C0013264	C0013264
GARD	6291	6291
MedDRA	10013801	10013801
Tekstualni opis	Onset occurs in early childhood, and affected boys may show a delay in motor milestones or global developmental delay. Children with DMD generally never achieve the ability to run or jump. The condition progresses rapidly and the child develops a waddling gait and a positive Gower's sign. Climbing stairs becomes difficult and the child begins to fall frequently. Loss of independent ambulation occurs between the ages of 6 and 13 years, the average being 9.5 years in non-steroid treated patients. Once ambulation is lost, joint contractures and scoliosis develop rapidly. Cardiomyopathy and respiratory failure are the cause of death in young adulthood.	Nastaje u ranom detinjstvu, i zahvata dečake, kod kojih se registruje zaostatak u motornom ili globalnom razvoju. Deca sa DMD skoro nikada ne mogu da trče ili skaču. Stanje se brzo pogoršava i dete razvija gegajući hod sa pozitivnim Gowerovim znakom. Penjanje uz stepenice postaje značajan problem, i dete često i pada. Gubitak samostalnog hoda/pokretljivosti javlja se između 6-te do 13-te godine, sa prosečnom vrednošću od 9.5 godina, ukoliko se pacijent ne leči kortikosteroidima. Kada se sposobnost hoda izgubi, javljaju se kontrakture zglobova i skolioza. Kardiomiopatija i respiratorna insuficijencija su česti uzrok smrti u ranom odraslom dobu.

Slika 3. *Excel fajl spreman za importovanje*

3. ZAKLJUČAK

Zahvaljujući ovom projektu, ljudi oboleli od retkih bolesti, njihove porodice pa i svi ostali koji žele da se informišu mogu to da urade odlaskom na *web* sajt. Moguće je pretraživati retke bolesti po nazivu pa i drugim atributima, pratiti najnovije vesti u svetu retkih bolesti, pa i komunicirati sa ostatkom zajednice preko foruma. Prevodiocima je proces prevođenja sa engleskog jezika umnogome olakšan, korišćenjem programa za prevođenje implementiranog unutar *web* sajta ili upisivanjem prevoda u eksportovane *excel* fajlove.

Ciljevi koji su do sada dostignuti jeste prevod preko 1000 retkih bolesti, s tim da se veliki broj informacija prevodi u svakom trenutku.

Načini za poboljšanje trenutnog rešenja obuhvataju efikasniju biblioteku za program za import i eksport, kao i proširenje programa za prevođenje dodavanjem mogućnosti za izmenu prihvaćenog prevoda. Što se tiče unapređenja po pitanju tehnološkog razvoja, moguće je u budućnosti prilagoditi aplikaciju za korišćenje na različitim veličinama i rezolucijama ekrana, kao i implementirati aplikaciju za smart uređaje.

4. LITERATURA

- [1] Dean T Jamison, Joel G Breman, Anthony R Measham, George Alleyne, Mariam Claeson, David B Evans, Prabhat Jha, Anne Mills, and Philip Musgrove, Disease Control Priorities in Developing Countries, 2nd edition, 2006
- [2] Orphanet: an online rare disease and orphan drug data base. © INSERM 1999. Available on <http://www.orpha.net>

Kratka biografija:



Ivana Tešanović rođena je u Novom Sadu 1993. god. Završivši osnovne studije na Fakultetu tehničkih nauka, upisala je master studije, smer Elektronsko poslovanje, i odbranila master rad 2019. godine. Kontakt: inna0507@gmail.com

**PRIMENA KVIZA DOMENSKOG ZNANJA UZ POMOĆ SPHERO ROBOTA U
OBRAZOVANJU****APPLICATION OF THE DOMAIN KNOWLEDGE QUIZ WITH THE HELP OF SPHERO
ROBOT IN EDUCATION**

Dejan Stojkić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**1. UVOD**

Kratak sadržaj – Postoje brojna istraživanja koja govore o tome kako deca u mlađim uzrastima dosta brže uče kroz igru. Tako da smo mi putem Sprk+ robota i desktop igre pokušali da kviz koji je vezan za domenska znanja koja se uče u školi učinimo što zanimljivijim. Istraživačke studije u razvojnoj i kognitivnoj psihologiji ukazuju na to da su uticaji na životnu sredinu važni tokom ranih godina razvoja i da nedostatak neophodnih stimulacija u vezi sa djetetovim razvojem dovodi do toga da dijete ne dostigne svoj puni potencijal. U ovom radu su opisane dvije aplikacije koje su u formatu kviza. Prva aplikacija predstavlja kviz koji poseduje određenu grupu pitanja na osnovu koga robot sphero (SPRK+) vrši kretanje. Na osnovu tačnih odgovora iz date aplikacije vrši se pomeranje robota putem sphero EDU aplikacije. U Sphero EDU aplikaciji smo kreirali programe upotrebom javascript programskog jezika. Robot se kreće po ručno napravljenoj mapi koju smo prilagodili određenom domenu znanja. Druga aplikacija predstavlja desktop igru u formatu kviza razvijena putem WPF i OpenGL gde imamo virtualnu mapu na kojoj se kreće 3D objekat na osnovu odgovora iz date igre.

Ključne reči: Savremene obrazovne tehnologije, Kviz, Sphero SPRK+

Abstract – There is a wealth of research showing that children at a younger age learn much faster through play. So, through Sprk + robots and a desktop game, we tried to make the domain knowledge quiz we learn at school as interesting as possible. Research studies in developmental and cognitive psychology indicate that environmental impacts are important during the early years of development and that the lack of necessary stimuli related to the child's development causes the child to reach his or her full potential. This paper describes two applications that are in quiz format. The first application is a quiz that has a specific set of questions on the basis of which the robot sphero (SPRK +) moves. So, based on the correct answers from a given application, the robot is moved through a sphero EDU application. In the Sphero EDU application, we have created programs using the javascript programming language. The robot moves around a handmade folder that we customize to a specific domain of knowledge. The second application is a desktop game in quiz format developed through WPF and OpenGL where we have a virtual map on which a 3D object moves based on the answers from a game.

Keywords: Modern Educational Technologies, Quiz, Sphero SPRK +

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinac, van. prof.

Sa razvojem tehnologije dolazimo do mogućnosti razvoja mnogih grana industrije ali i obrazovanja. U vreme kada deca od ranih godina imaju dodira sa računarom i pametnim uređajima i sama rasprostranjenost i dostupnost interneta teraju roditelje da neprestano brinu o upotrebi i načinu korišćenja kompjutera od strane dece. Danas je sve veća i veća potreba za primenom tehnologije i olakšica koje tehnologija omogućava prilikom rešavanja određenih problema. Tako sam razvoj modernog doba ima i prednosti i mane. Da bi iskoristili prednosti modernog doba i što više zainteresovali decu za učenjem pokušavamo upotrebiti prednosti tehnologije i računara za efikasnije i zanimljivije izvođenja nastave.

Tako da ovo rešenje u obliku kviza na čije odgovore imamo posledicu kretanja robota i posebnu interaktivnu igru u formatu kviza može biti interesantno za razne oblasti i predmete koji se izučavaju u školama. Putem ovakvih kvizova deca se više interesuju i stvaraju želju za takmičenjem ko će imati više tačnih odgovora i ko će biti brži. Aplikacija, odnosno kviz, je razvijen u React [6] okruženju sa četiri pitanja po pet ponuđenih odgovora od kojih je samo jedan tačan. Kviz se sastoji iz četiri grupe pitanja. Prva grupa pitanja predstavlja laka pitanja ako su uspešno predena onda je to ocjena dva. Druga grupa pitanja predstavlja malo teža pitanja za ocjenu tri. Treća grupa pitanja su još teža pitanja za ocjenu četiri i poslednja grupa pitanja koja su najteža a za ocjenu pet. Pitanja iz svih grupa su vezana za isti domen znanja samo su poredana po grupama i po težini kako bi pomogli nastavniku/učitelju pri ocenjivanju.

2. MOTIVACIJA

Postoje brojna istraživanja koja govore o tome kako deca u mlađim uzrastima dosta brže uče kroz igru. Tako da vremenom na svetskom tržištu imamo sve više i više igračaka koje u suštini služe za učenje. Tako da smo mi putem Sprk+ robota pokušali da kviz koji je vezan za domenska znanja koja se uče u školi učinimo što zanimljivijim. Istraživačke studije u razvojnoj i kognitivnoj psihologiji ukazuju na to da su uticaji na životnu sredinu važni tokom ranih godina razvoja i da nedostatak neophodnih stimulacija vezanih za djetetov razvoj dovodi do toga da dijete ne dostigne svoj puni potencijal. Dakle, obrazovanje u ranim danima djetinjstva je od velikog značaja za mnoge aspekte djetetovog razvoja, a istraživači predlažu naučno obrazovanje treba početi u ranim godinama školovanja [1]. Postoji nekoliko

razloga za početak podučavanja nauke tokom ranog detinjstva. Prvo, deca imaju prirodnu tendenciju u posmatranju i razmišljanju o prirodi [2].

Mala deca su motivisana da istraže svet oko sebe i rana naučna iskustva mogu iskoristiti ovu sklonost. Razvojno odgovarajuće angažovanje, sa kvalitetom iskustva učenja nauke je od vitalnog značaja za pomoć deci da razumeju svet, prikupe, organizuju informacije, primenjuju, testiraju ideje i razvijaju pozitivne stavove prema nauci. Kvalitetna iskustva učenja nauke pružaju solidnu osnovu za kasniji razvoj naučnih koncepata sa kojima će se deca susresti tokom njihovog akademskog života [3]. Ova fondacija pomaže studentima da konstruišu razumevanje ključnih naučnih koncepata i omogućava budućnost učenje više apstraktnih ideja (Reynolds & Walberg, 1991). Angažovanje naučnih iskustava omogućava razvoj i naučno razmišljanje.

Podrška deci dok razvijaju naučno razmišljanje tokom ranog detinjstva mogu lako dovesti do toga da prenose svoje sposobnosti razmišljanja na druge akademske domene koji mogu značajno uticati na njihova akademska dostignuća i njihov smisao samoeфикаsnosti [3].

3. SPHERO

Sphero je sferna robot igračka koju je dizajnirala Sphero kompanija. To je bela kugla umotana u polikarbonatnu plastiku, sposobna da se okrene, menja boje, pokreće programe i kontroliše se pametnim telefonom ili tabletom. Sphero SPRK+ i drugi roboti koje je razvio Sphero koriste iste metode za komunikaciju i navigaciju, tako da programi razvijeni za jednog robota obično rade na svim Sphero robotima. Sphero roboti mogu da se kotrljaju za datu brzinu u određenom vremenu kod SPRK postoje dve led diode koje svetle u bilo kojoj zadatoj boji.

Unutrašnji senzori Sphero robota uključuju IMU i lokator. Omogućeno je bluetooth povezivanje sa uređajima kao što su iPads, iPhone, Android telefoni i tableti. Sphero je pokrenuo SPRK obrazovanje kako bi podržao nastavnike, roditelje i decu koristeći Sphero da nauče programiranje, robotiku i matematiku na zabavan, interaktivan način [4]. Sphero SPRK aplikacija također uključuje OVAL, programski jezik zasnovan na C-u. Deca mogu da nauče da programiraju robota sa blokovima, a zatim evoluiraju do kodiranja pomoću OVAL-a, što može pomoći u upoznavanju sa načinima rada C, C++, Python, Ruby i drugih jezika.

Sphero EDU SPRK aplikacija omogućava korištenje OVAL-a za programiranje Sphero robota. Drugi koriste blokove za kreiranje i definiranje koraka. Blokovi predstavljaju ugrađene komande koje mi u suštini slažemo po želji i robot ih izvršava u nizu jednu za drugom.

Mi nažalost nismo uspjeli da iskoristimo javascript biblioteku i njen API kako bi uspostavili vezu sa sprk+ robotom, pokušali smo na svim operativnim sistemima ali za verziju sprk+ nije moguće na taj način konektovati se zbog veće zaštite i zatvorenosti samog uređaja. Zbog hardverskih (tehničkih) problema koristili smo Sphero EDU aplikaciju koja nas je ograničavala za par funkcionalnosti. Zbog hardverskih problema i ograničenosti Sphero Edu aplikacije nismo mogli u real-time režimu da zadajemo naredbe koje proizilaze iz kviza.

Jedini način je bio da nakon što korisnik odgovori na sva pitanja pošaljemo date informacije Sphero EDU aplikaciji putem email-a sa datim informacijama ispitivaču koji bi u zavisnosti od dobijenih informacija pokrenuo određeni modul.



Slika 3.1: Sphero SPRK+ robot

4. IMPLEMENTACIJA KVIZA SA ROBOTOM

Izrada same aplikacije kviza zasniva se na kreiranju 4 grupe pitanja koja raspodeljena po težini. Za svako pitanje ponuđeno je pet odgovora od kojih je samo jedan tačan. Omogućeno je da se kviz može prilagoditi bilo kom domenu znanja što utiče na generičnost kviza i olakšava upotrebu nastavnicima i profesorima. Aplikacije je pre svega namenjena deci uzrasta osnovne škole od 5. do 8. razreda tako da imamo četiri grupe pitanja. Prva grupa pitanja je vezana za biljke. Kandidat prolazi kroz različite podoblasti ove oblasti i pokazuje svoje znanje. U drugoj oblasti su pitanja vezana za životinje. Treća grupa je vezana za izučavanje čoveka pa su pitanja prilagođena tom domenu. Četvrta grupa, koja je namenjena 8. razredu.

Ovaj deo aplikacije je napisan u react, javascript okruženju. React nam pruža veliku slobodu prilikom konstrukcije ovakvog rešenja i uz pomoć njega uspevamo da na jednostavan način kreiramo kviz sa svim reaktivnim komponentama. Pitanja su predstavljena u obliku json fajla gde je moguće izmeniti oblast i pitanja po potrebi. Tako da je u budućnosti moguća promena domena pitanja i samim tim rešenje postaje dinamičnije.

Ovaj deo aplikacije je napisan u react, javascript okruženju. React nam pruža veliku slobodu prilikom konstrukcije ovakvog rešenja i uz pomoć njega uspevamo da na jednostavan način kreiramo kviz sa svim reaktivnim komponentama. Pitanja su predstavljena u obliku json fajla gde je moguće izmeniti oblast i pitanja po potrebi. Tako da je u budućnosti moguća promena domena pitanja i samim tim rešenje postaje dinamičnije. Drugi deo implementacije je vezan za programiranje robota odnosno kretanja robota i dodavanje raznih specijalnih efekata i zvukova u zavisnosti od kombinacije odgovora.

Zbog problema sa hardverom odnosno nemogućnosti povezivanja u real-time režimu. Napravili smo neophodan broj modula odnosno odvojenih programa u Sphero Edu aplikaciji koji se pokreću u zavisnosti od krajnjeg ishoda. Postoji 15 modula, 15 JavaScript skripti koje su vezane za sve moguće ishode.

Jedna skripta predstavlja ishod u kome je učenik odgovorio samo na prvo pitanje tačno. Svaka skripta se pokreće u odnosu na broj tačnih odgovora i u zavisnosti od rednog broja tačnih odgovora.

Obuhvaćene su sve mogućnosti čak i da učenik ne da ni jedan tačan odgovor. Neophodno je pre pokretanja programa pravilno kalibrirati robota da bi potpuno ispravno izvršio predviđenu skriptu. U osnovi svake skripte se nalaze naredbe kako će se robot kretati po mogućim poljima na mapi.

Dobrodošli u sferu kviz!
Izaberite grupu pitanja

5. Razred 6. Razred 7. Razred 8. Razred

1. Ko ne opirašuje biljke?

Covek
Vetar
Murva
Kolibri
Pčele

2. Koja je funkcija korena?

Da drži biljku uspravno
Da proizvodi hranu
Da upija hranjive materije iz zemlje
Da vrši fotosintezu
Da oslobađa ugljenik dioksida

3. Čime se bavi botanika?

Proučavanjem životinja
Proučavanjem mikroorganizama
Proučavanjem gljiva
Proučavanjem biljaka
Proučavanjem čoveka

4. Ko ne pripada porodici ruža?

Jagoda
Jabuka
Šljiva
Kruška
Paradajz

Završi kviz.

Slika 4.1: Kviz sa ponuđenim pitanjima

Na početku svakog modula (skripte) imamo uvodnu euforičnu melodiju koja dočarava početak avanture odnosno kviza i na kraju melodiju koja opisuje kraj avanture date runde koja može biti sretna ili tužna melodija u zavisnosti od tačnih odgovora. Ako su svi odgovori tačni onda je sretna melodija na kraju u suprotnom je tužna.

Nakon uvodne melodije čuje se efekat poletanja aviona sa piste pa se robot kreće ako je tačan odgovor i postepeno menja boju iz plave u zelenu ako je tačan odgovor i čuje se zvuk koji se obično koristi kod kvizova kako bi dočarali tačan odgovor i na kraju svakog pitanja ide akcija koja na engleskom govori da li je tačan odgovor.

Ako je netačan odgovor robot ide u jamu ili ne pomera se ako je već u njoj, boja postepeno prelazi iz trenutne boje u crvenu boju i čuje se zvuk koji se obično koristi da bi opisali netačan odgovor, na kraju poslednja akcija vezana za to pitanje jeste glas koji na engleskom govori da je odgovor na dato (prvo, drugo itd..) pitanje ne tačan.

Treći segment implementacije vezan je za kreiranje same mape po kojoj se robot kreće. Na hamer dimenzije 100x70cm smo prvo odredili na kom mestu će se nalaziti start polje iz kog se robot kreće nakon ispravnog kalibrisanja. Zatim smo odredili polja koja predstavljaju prvo, drugo, treće i četvrto pitanje. Ako je odgovor tačan robot ide na dato polje u suprotnom se vraća u start odnosno u jamu.



Slika 4.2: Ručno napravljena mape

5. IMPLEMENTACIJA IGRE U FORMATU KVIZA

Ovaj kviz predstavlja igru gde postoje dve vrste korisnika profesori i učenici. Profesori imaju dodatne mogućnosti u odnosu na učenike. Prilikom prijavljivanja na aplikaciju na stranici za logovanje bira se vrsta korisnika.

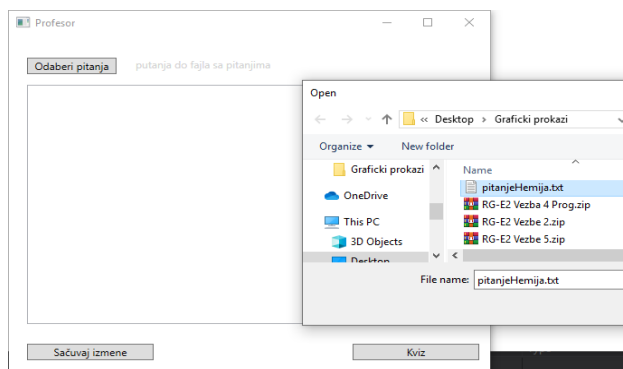
Dobrodošli

Profesor

Ucenik

Slika 5.1

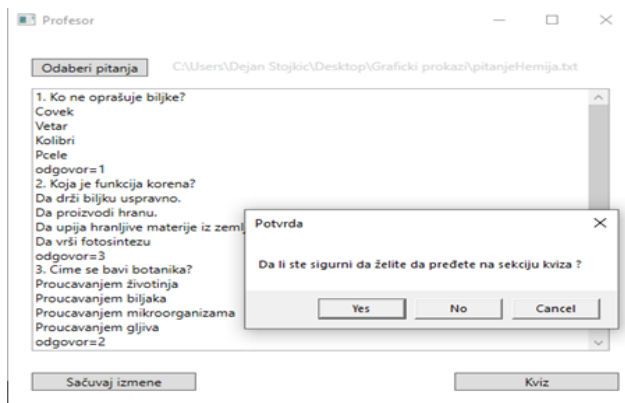
Logovanje je jednostavno realizovano trudeći se da bude slično kao logovanje na sistemu, nešto što je većini poznato. Naravno da bi se ulogovali korisnici koristi svoje ime i prezime sa šifrom. Šifre su heširane putem Bcrypt algoritma [5]. To nam omogućava sigurnost u slučaju da neko provali u bazu podataka.



Slika 5.2

Postoji mogućnost da se uveze fajl sa pitanjima da bi olakšali profesoru posao (Slika 5.2). Ako je fajl koji je učitao kao na slici 5.3, isparsiran po pravilima profesor će dobiti poruku da je sve u redu i da će moći da pristupi kvizu. Na osnovu (Slike 5.3) vidi se da u fajlu moramo staviti i koji je tačan odgovor. Moguće je ručno promeniti

u fajlu odgovor kada se učita. Brojevi nam govore koji redni broj ponuđenog odgovora je tačan. Dok u pozadini se fajl parsira uz proveru tako da nije moguće snimiti fajl ako profesor zaboravi prilikom pravljenja pitanja i ponuđenih odgovora da napiše koji je tačan odgovor. Neophodno je da imamo postavljeno pitanje sa upitnim znakom “?” i rednim brojem pitanja.



Slika 5.3

Naravno u ovom režimu na većinu akcija imamo dodatne dijaloge za potvrdu akcija, zbog moguće nepažnje, uz naravno nedozvoljavanje prelaska u narednu sekciju ako parsiranje fajla nije u potpunosti ispoštovalo pravila. Svako pitanje poseduje četiri ponuđena odgovora. Gde korisnik ima parvo da odabere koji je tačan odgovor, dodatna opcija je takava da ako je kviz u režimu učenja korisniku se neće postaviti naredno pitanje sve dok ne da tačan odgovor na postavljeno pitanje. Ako se kviz ne nalzi u tom režimu korisnik bez obzira na ispravnost odgovora dobija naredno pitanje. Nakon tačnog odgovora vrši se kretanje figure zbog domenskog znanja figura koja se kreće po mapi zavisi od domena u ovo slučaju na pitanje iz biologije 3D objekat koji služi za kretanje je drvo umesto robota (slika 5.4).



Slika 5.4

6. ZAKLJUČAK

Zadatak projekta jeste konstrukcija kviza domenskog znanja uz pomoć Sphero SPRK+ robota i pandam njemu konstrukcija desktop igre. Deci je potrebno kvalitetno naučno iskustvo tokom njihovog ranog djetinjstva.

Nauka i pismenost pruža sistematski, obrazovni okvir, zasnovan na standardima kao što su nastavni plan, program i kvalitetni nastavnici. Ovaj program takođe efikasno integriše tekst, ilustracije i dijagrame instrukcija zasnovana na istraživanju.

Cilj ove aplikacije jeste učenje određene oblasti kroz igru, gde će učenik videti posledice svojih odgovora u realnom svetu. Aplikacija pomaže profesoru i učeniku jer profesor može lakše i objektivnije da odredi znanje učenika odnosno da ih ocjeni za prikazano znanje. Nakon primene u osnovnoj školi zaključak je da su učenici vidno zainteresovaniji i motivisaniji za učenje. Kao što smo videli kroz brojna istraživanja svetski poznatih psihologa da je motivacija najveći problem u mladim uzrastima i da ona najviše koči intelektualno razvijanje deteta. Možemo reći da je projekat uprkos tehničkim problemima ipak uspeo u realnoj primeni i dao pozitivne rezultate.

7. LITERATURA

- [1] Eshach & Fried 2005; Watters, Diezmann, Grieshaber, & Davis, 2000 „Enhancing Science Education for Young Children: A Contemporary Initiative“ from, https://www.researchgate.net/publication/27464158_Enhancing_Science_Education_for_Young_Children_A_Contemporary_Initiative
- [2] Eshach, H., & Fried MN; „Should science be taught in early childhood“; (2005)
- [3] Kuhn & Pearsall; Kuhn & Schauble, & GarciaMilla, „Contemporary Perspectives and Research on Early Childhood Education“; (2000)
- [4] Sphero robot: <https://sdk.sphero.com/sphero-robot-basics/what-is-sphero> (pristupljeno u julu 2019.)
- [5] Bcrypt algoritam, <https://thehackernews.com/2014/04/securing-passwords-with-bcrypt-hashing.html> (pristupljeno u julu 2019.)
- [6] Wikipedia contributors. (2018, March 26).React. In Wikipedia, The Free Encyclopedia. Retrieved 20:07, Jun 13, 2019, from [https://sr.wikipedia.org/wiki/React_\(библиотека_за_JavaScript\)#cite_note-7](https://sr.wikipedia.org/wiki/React_(библиотека_за_JavaScript)#cite_note-7)

Kratka biografija:



Dejan Stojkić rođen je u Zvorniku 1995. god. upisuje Gimnaziju Petar Kočić u Zvorniku završava 2014. godine kao dobitnik Vukove diplome. Fakultet tehničkih nauka u Novom Sadu je upisao 2014. godine. Stipendista softverske kompanije Naovis tokom studija. Po završetku osnovnih studija 2018. upisuje master na smeru Inteligentni sistemi i istovremeno radi na FTN-u kao saradnik u nastavi i kao profesor u srednjoj školi Jovan Jovanović Zmaj u Novom Sadu.

**VIŠEKRITERIJUMSKO ODLUČIVANJE U PLANIRANJU AUTOMATIZACIJE
ELEKTRODISTRIBUTIVNIH MREŽA****MULTIPLE-CRITERIA DECISION MAKING IN POWER DISTRIBUTION NETWORK
AUTOMATION PLANNING**Miloš Vulin, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je prikazan pristup za određivanje optimalnog plana automatizacije (optimalnog broja i lokacije uređaja za automatizaciju) u elektro-distributivnim mrežama u kojima se pouzdanost meri korišćenjem indeksa SAIDI i SAIFI. Prikazani pristup određuje najbolji plan automatizacije korišćenjem alata za višekriterijumsku optimizaciju. U prvom koraku se određuje skup nedominantnih rešenja (planova automatizacije), a zatim se korišćenjem alata za višekriterijumsko odlučivanje bira najbolji plan iz dobijenog skupa.

Ključne reči: Pouzdanost elektrodistributivnih mreža, automatizacija, višekriterijumsko odlučivanje

Abstract – This paper presents the approach for optimal network automation planning (obtaining the optimal number and location of automation devices) in the networks where reliability is measured by using indices SAIDI and SAIFI. Presented approach defines the best automation plan by employing the tools for multiple criteria optimization. First, the set of non-dominated solutions (automation plans) is determined and then the best plan in the set is obtained by using the tool for multiple criteria decision making.

Keywords: Reliability of power distribution networks, automation, multiple criteria decision making

1. UVOD

Jedna od glavnih karakteristika distributivnog sistema jeste zadovoljenje što većeg nivoa pouzdanosti. Pouzdanost napajanja se danas postiže primenom različitih vidova distributivne automatike i softverskih alata. Pouzdanost distributivnog sistema zavisi od pouzdanosti njegovih elemenata. Elementi distributivnog sistema (rasklopna oprema, transformatori, sabirnice, vodovi...) mogu se u analizama modelovati na različite načine, a najčešće se opisuju sa dva stanja [1]:

- stanje ispravnog rada i
- stanje otkaza.

Pouzdanost mere svi operatori distributivnog sistema u celom svetu. Jedan od načina merenja je preko neekonomskih i ekonomskih pokazatelja pouzdanosti. Danas jedna od najčešće primenjivanih tehnika za unapređenje pouzdanosti je automatizacija distributivne mreže, koja

se i razmatra u ovom radu. Za određivanje optimalnog nivoa automatizacije u ovome radu je korišćena jedan od energetske funkcije razvijena u ADMS (Advanced Distribution Management System) softveru (verzija 3.8.2). Kako problem postaje problem višekriterijumske optimizacije pored navedene funkcionalnosti za razmatranje neekonomskih pokazatelja pouzdanosti (SAIDI, SAIFI, ENS) se koristi i metoda višekriterijumskog odlučivanja. Kriterijumske funkcije ovih pokazatelja se sastoje od više različitih kriterijuma, tj. kriterijumi nemaju iste jedinice. Za izbor najboljeg scenarija automatizacije korišćena je metoda višekriterijumskog odlučivanja TOPSIS.

**2. POKAZATELJI POUZDANOSTI U
DISTRIBUTIVNIM MREŽAMA**

Pokazatelji pouzdanosti koji će biti obrađeni u ovom radu su SAIFI i SAIDI. Indeks prosečne učestanosti prekida isporuke električne energije - SAIFI (System Average Interruption Frequency Index), predstavlja srednji broj trajnih prekida napajanja potrošača u definisanom vremenskom periodu i definisan je sledećom relacijom [2]:

$$SAIFI = \frac{\text{ukupan broj prekida napajanja potrošača}}{\text{ukupan broj usluženih potrošača}} = \frac{\sum \lambda_i N_i}{N_T}, \quad (1)$$

gde su:

λ_i – intezitet otkaza i-tog potrošačkog čvora,
 N_i – broj korisnika pogođenih prekidom isporuke električne energije usled dugotrajnog prekida i ,
 N_T – ukupan broj korisnika.

Indeks prosečnog trajanja prekida isporuke električne energije u definisanom vremenskom periodu, izražava se u minutima ili časovima i definisan je sledećom relacijom [2]:

$$SAIDI = \frac{\text{suma trajanja prekida pojedinih potrošača}}{\text{ukupan broj usluženih potrošača}} = \frac{\sum r_i \lambda_i N_i}{N_T} \quad (2)$$

gde su:

r_i – vreme potrebno za restauraciju napajanja usled prekida i ,
 λ_i – intezitet otkaza i-tog potrošačkog čvora,
 N_i – broj korisnika pogođenih prekidom isporuke električne energije usled dugotrajnog prekida i ,
 N_T – ukupan broj korisnika.

Većom zastupljenošću uređaja za automatizaciju (sekcionalizera, indikatora prolaska struje kvara) obezbeđuje se smanjenje broja korisnika pogođenih kvarom i vremena trajanja kvara. Time se unapređuju i gore prikazani pokazatelji pouzdanosti. Određivanje optimalnog nivoa automatizacije je složen, nelinearan, optimizacioni problem sa

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Popović, doc.

ograničenjima. Da bi se postiglo opšte poboljšanje nije dovoljno razmatrati samo jednu funkciju cilja, dakle postavlja se i rešava višekriterijumska optimizaciona funkcija (problem) [3].

Najvažnije pitanje je kako postići visoki stepen pouzdanosti distributivnog sistema uz minimalne kapitalne troškove (investicije) odnosno naći kompromis između pouzdanosti i potrebnih ulaganja zarad postizanja iste. To obuhvata utvrđivanje optimalnog broja i lokacija postavljanja kontrolnih i zaštitnih uređaja u distributivnom sistemu kao i efikasno i sigurno planiranje razvoja distributivne mreže [3].

3. FORMULACIJA PROBLEMA

Cilj ovog rada je da odredi optimalan broj i lokaciju sekcionera i indikatora prolaska struje kvara koje bi trebalo ugraditi u mrežu u cilju postizanja najboljih neekonomskih pokazatelja pouzdanosti. U ovom slučaju, pošto se razmatraju neekonomski pokazatelji pouzdanosti (SAIDI i SAIFI), optimizacioni problem postaje problem višekriterijumske optimizacije. Za rešavanje ovakvog problema je u poglavlju 3.1 prikaza pristup za dobijanje skupa nedominantnih rešenja tj. skupa nedominantnih planova automatizacije. Izbor najboljeg plana automatizacije iz dobijenog skupa je urađen korišćenjem metode višekriterijumskog odlučivanja TOPSIS, koja je prikazana u poglavlju 3.2.

3.1 Matematički model

Pristup koji je obrađen u ovom radu definiše optimalnu strategiju za automatizaciju elektrodistributivnih mreža ukoliko se pouzdanost određuje na sledeći način:

- U situaciji kada se uspešnost distributivne kompanije meri samo na osnovu pokazatelja pouzdanosti SAIFI i SAIDI cilj planiranja automatizacije distributivne mreže se može formulisati na sledeći način:

$$F1 = \min \{ SAIFI, SAIDI, \text{Trošak automatizacije} \} \quad (4)$$

Kriterijumska funkcija (4) predstavljaju višekriterijumsku formulaciju problema planiranja automatizacije distributivne mreže. U (4) su razmatrani pojedinačni kriterijumi koji su izraženi (mereni) u različitim jedinicima i zbog toga se za rešavanje ovoga problema koristi ekvivalentni indeks pouzdanosti [4] i pristup za rešavanje višekriterijumskog problema optimizacije zasnovan na metodi ograničenja u prostoru kriterijumske funkcije [5]. Ekvivalentni pokazatelj pouzdanosti, SAFDI, je definisan na sledeći način:

$$SAFDI = t_{SAIFI} \cdot \frac{SAIFI}{SAIFI_c - SAIFI_{maxB}} + t_{SAIDI} \cdot \frac{SAIDI}{SAIDI_c - SAIDI_{maxB}} \quad (5)$$

gde je:

t_{SAIFI} , t_{SAIDI} – težinski faktori kojima se uvažava značaj (težina) pojedinačnog indeksa,
 $SAIFI_c$, $SAIDI_c$ – vrednosti indeksa u mreži sa tekućim nivoom automatizacije,
 $SAIFI_{maxB}$, $SAIDI_{maxB}$ – vrednosti indeksa pouzdanosti koji bi se imali kada bi se maksimalno dozvoljeni budžet potrošio na njihovo unapređenje. Ove vrednosti se dobijaju rešavanjem (4) za pojedinačni indeks i

maksimalni budžet. Kao alternativo rešenje se može uzeti da su ove vrednosti jednake nuli (0).

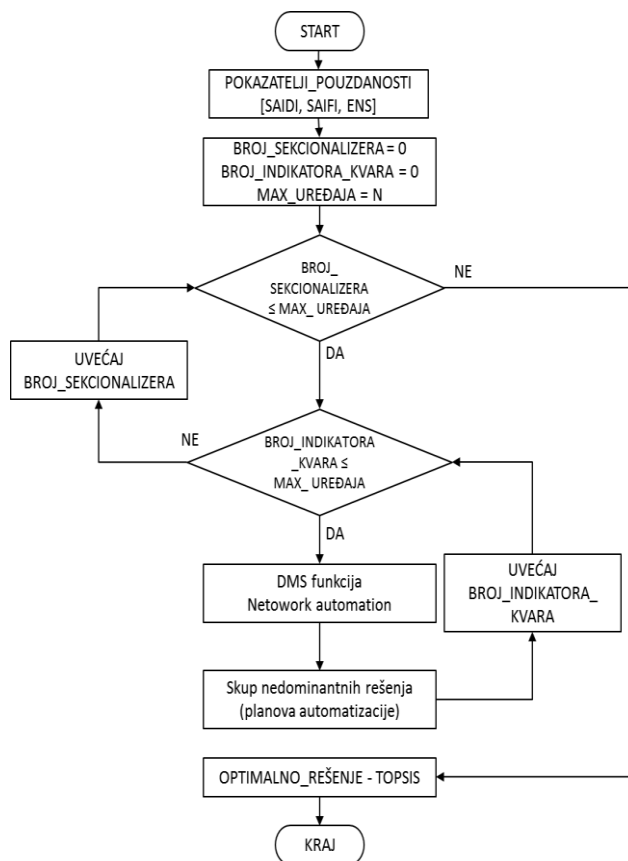
Na osnovu prethodnog i korišćenjem metode ograničenja u prostoru kriterijumske funkcije [5] se višekriterijumski problem (4) prevodi u sledeći niz jednokriterijumskih optimizacionih problema:

$$F1 = \min SAFDI \quad (6)$$

Uz ograničenja:

$$\text{Trošak automatizacije} \leq \text{BUDŽET}_T \quad (7)$$

Variranjem budžetskih ograničenja (7) u (6), odnosno variranjem broja uređaja za automatizaciju koji su na raspolaganju za ugradnju, može se dobiti skup nedominantnih rešenja problema (4). Na ovaj način se dobija skup različitih planova automatizacije koji unapređuju SAIDI i SAIFI, a određivanje najboljeg među njima se vrši primenom metode za višekriterijumsko odlučivanje TOPSIS, prikazane u poglavlju 3.2. Detaljnu formulaciju problema u terminima mešovitog celobrojnog linernog programiranja (MILP) nije moguće dati u radu zbog ograničenog prostora. Detaljna formulacija je prikazana u master radu [7].



Slika 1. Blok-dijagram višekriterijumskog odlučivanja

Generisanje skupa nedominantnih rešenja korišćenjem napred navednog pristupa je detaljnije prikazano blok dijagramom na slici 1.

Nakon što je generisan skup nedominantnih rešenja potrebno je metodom višekriterijumskog odlučivanja pronaći najbolje rešenje. U nastavku će biti objašnjeno ukratko kako korišćenjem metode TOPSIS dolazimo do najbolje rešenja

3.2 Metoda TOPSIS

Metoda TOPSIS (Technique for Order Preference by Similariity to Ideal Solution) je zasnovana na konceptu da izabrana alternativa treba da ima najkraću Euklidovu udaljenost od "idealnog" rešenja i najdalju Euklidovu udaljenost od "anti-idealnog" rešenja, pa se na taj način uvode dve referentne tačke: "idealno rešenje" i "anti-idealno rešenje" [5]. Osnovni koraci algoritma za pronalaženje najbolje rešenja korišćenjem metode TOPSIS su sledeći:

1. Transformacija raznorodnih kriterijumskih funkcija
2. Određivanje normalizovanih vrednosti kriterijumskih funkcija
3. Određivanje "idealnog" i "anti-idealnog rešenja"
4. Određivanje mera odstupanja od "idealnog rešenja" i "anti-idealnog rešenja"
5. Određivanje relativne blizine "idealnom rešenju" C_j^*
6. Formiranje rang-liste alternativa.

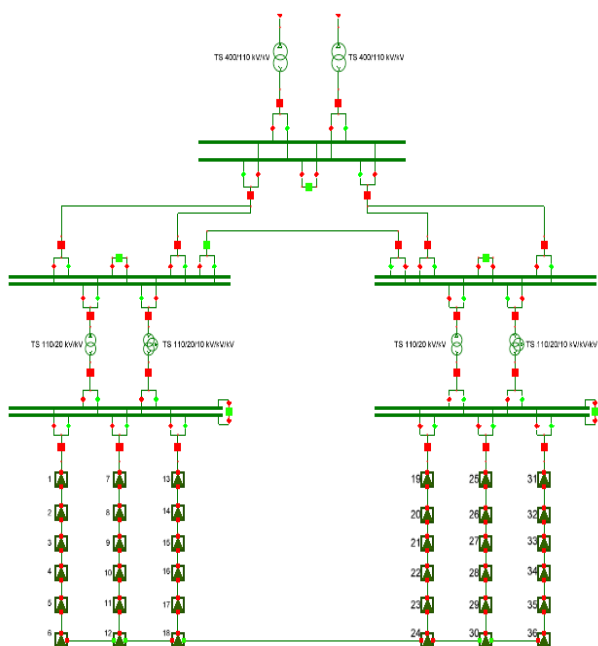
Na kraju prikazani algoritam uvek pruža jedinstveno kompromisno rešenje tako što se formiranjem rang-liste svih alternativa na osnovu vrednosti mere C_j^* bira najbolja alternativa sa najvećom vrednošću C_j^* . Mera C_j^* proračunava relativnu blizinu „idealnom rešenju“ svake alternative. Jasno je da za $C_j^*=1$ dobijeno „idealno rešenje“, a za $C_j^*=0$ dobijeno „anti-idealno rešenje“, pa je jasno da je najbolja alternativa bliža „idealnom rešenju što je C_j^* bliže 1.“

4. PRIMER PRIMENE

U nastavku je opisana test srednjenaponska (SN) distributivna mreža i dati su rezultati proračuna, odnosno određivanja optimalnog nivoa automatizacije sa stanovišta neekonomskih pokazatelja pouzdanosti SAIDI i SAIFI. Poračuni su rađeni na način opisan u poglavlju 3.1 a od opreme za automatizaciju se koriste sekcionalizeri i indikatori prolaska struje kvara

4.1 Opis test distributivne mreže

Test distributivna SN mreža koja je korišćena za potrebe proračuna je prikazana na slici 2.



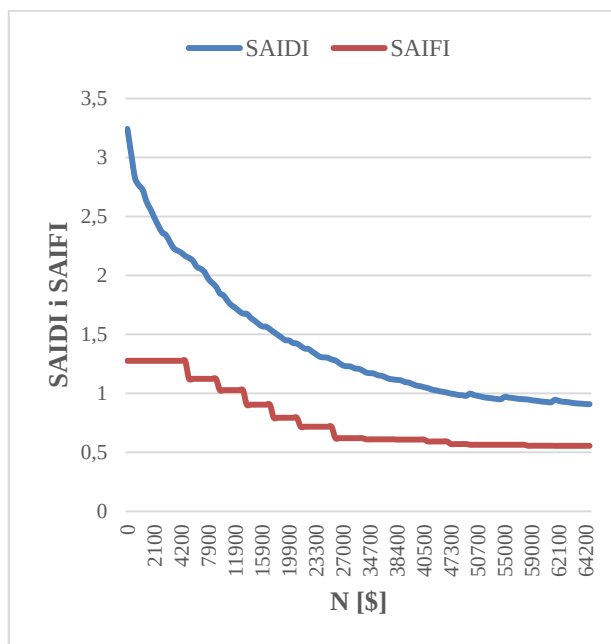
Slika 2. Test mreža

Ona je potencijalno upetljana, sastoji se od dve transformatorske stanice VN/SN, SN dela mreže i potrošačkih čvorova. SN deo mreže se sastoji od šest izvoda i na svakom od izvoda se nalazi po šest potrošačkih čvorova. Potrošački čvorovi na svakom izvodu predstavljaju transformatorske stanice SN/NN sa različitim brojem i tipovima potrošača. Inteziteti kvarova svih sekcija kao i dužine svih sekcija su različite za datu test mrežu i njihove verednosti su date u [7].

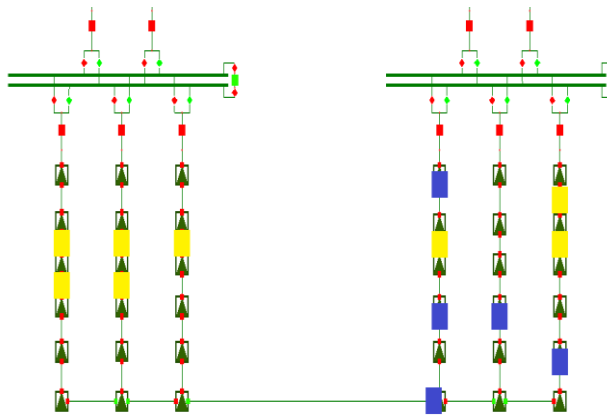
4.2 Određivanje optimalnog plana automatizacije za test mrežu

Primenom pristupa prikazanog u poglavlju 3.1 su dobijeni različiti planovi automatizacije razmatrane mreže. Oni se međusobno razlikuju kako po broju tako i po mestu ugradnje sekcionalizera i indikatora prolaska struje kvara. Za svaki od dobijenih planova se znaju investicioni troškovi i vrednosti pokazatelja pouzdanosti SAIDI i SAIFI. Oni su prikazani na slici 3. Na slici se može videti da indeksi pouzdanosti SAIDI i SAIFI opadaju sa porastom broja ugrađenih uređaja za automatizaciju, odnosno sa porastom investicija.

Da bi se među dobijenim planovima izabrao najbolji(optimalan) plan automatizacije koristi se metoda



Slika 3. Promena indeksa SAIDI i SAIFI sa povećanjem broja sekcionalizera i indikatora kvara



Slika 4. Optimalan broj i lokacija ugrađenih sekcionalizera i indikatora prolaska struje kvara

višekriterijumsko odlučivanje TOPSIS. Ovde treba imati na umu da je svaki od planova potrebno vrednovati na osnovu raznorodnih kriterijuma. Ti kriterijumi su:

1. Ukupni trošak ugradnje uređaja [€]
2. Indeks prosečnog trajanje otkaza sistema – SAIDI [h/god]
3. Indeks prosečne učestanosti otkaza sistema – SAIFI [1/god]

Smatra se da svaki od navedenih kriterijuma ima isti značaj, odnosno istu težinu.

Korišćenjem metode TOPSIS urađeno je rangiranje dobijenih planova automatizacije. Rezultati rangiranja za najboljih šest planova su prikazani u Tabeli 1. U ovoj tabeli, na primer, plan 5/7 označava da je to plan u kome je ugrađeno 5 sekcionalizera i 7 indikatora prolaska struje kvara. Iz tabele se vidi da je najbolji plan onaj sa 5 ugrađenih sekcionalizera i 8 indikatora prolaska struje kvara. Ovaj plan, odnosno broj i lokaciju indikatora prolaska struje kvara i sekcionalizera predloženih ovim planom su prikazani na slici 4. Žutom bojom su označeni indikatori prolaska struje kvara, dok su plavom bojom označeni sekcionalizeri.

Vrednost koju su imali indeksi SAIDI i SAIFI bez uređaja za automatizaciju je 3.242 i 1.276, respektivno, a sa opremom za automatizaciju koju definiše najbolji plan automatizacije ovi indeksi su 1.38 i 0.717, respektivno, uz investiciju od 22400 €.

Tabela 1. Rang lista planova automatizacije

Plan	C_j^*	Pozicija na rang-listi
5/7	0.727648	2
5/8	0.727897	1
5/9	0.725859	5
5/10	0.726246	4
5/11	0.712409	6
5/12	0.726361	3

5. ZAKLJUČAK

U radu je prikazan pristup u za određivanje optimalnog plana automatizacije u elektrodistributivnim mrežama u kojima se koriste pokazatelji SAIFI i SAIDI za merenje pouzdanosti. U ovakvim slučajevima problem planiranja automatizacije postaje višekriterijumski problem.

Za rešavanje ovakvog problema je korišćen algoritam zasnovan na mešovitom celobrojnom linearnom programiranju [7] i pristup za rešavanje višekriterijumskog problema optimizacije zasnovan na metodi ograničenja u prostoru kriterijumske funkcije. Na taj način se dobija skup nedominatnih planova automatizacije. Izbor najbolje plana automatizacije iz tako dobijenog skupa je urađen korišćenjem metode za višekriterijumsko odlučivanje TOPSIS.

Pristup predložen u ovom radu može da bude koristan alat donosicima odluke za određivanje najboljeg plana automatizacije u elektrodistributivnim kompanijama u kojima se koriste pokazatelji SAIDI i SAIFI za merenje pouzdanosti.

6. LITERATURA

- [1] M.D. Nimrihter: *Elektrodistributivni sistemi*, Fakultet tehničkih nauka, Novi Sad, 2009.
- [2] IEEE Guide for Electric Power Distribution Reliability Indices 2012.
- [3] M.S. Stojanovic, D.S. Tasic, A.T. Ristic: *Optimal Allocation of Distribution Automation Devices in Medium Voltage Network*, Faculty of Electronic Engineering, University of Nis.
- [4] L. Wang, Ch. Singh: Reliability-constrained optimum placement of reclosers and distributed generators in distribution networks using an ant colony system algorithm, IEEE Transaction on Systemes, Man, and Cybernetics – Part C, Vol. 38., No. 6, 2008, pp. 757-764.
- [5] S.Opricović: *Višekriterijumska optimizacija sistema u građevinarstvu*, Građevinski fakultet Univerziteta u Beogradu, Beograd, 1998.
- [6] Ching-Lai Hwang, Kwangsun Yoon: *Multiple Attribute Decision Making- Methods and Applications*; Springer-Verlag, Berlin, 1981.
- [7] M.Vulin: *Višekriterijumsko odlučivanje u planiranju automatizacije elektrodistributivnih mreža*, master rad, Novi Sad, 2019.

Kratka biografija:



Miloš Vulin rođen je u Bajinoj Bašti 1992. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranio je 2019.god.

OSNOVI CLOUD TEHNOLOGIJE SA PRIMERIMA PRIMENE**CLOUD TECHNOLOGY PRINCIPLES AND EXAMPLES**Daniela Kotur, Željens Trpovski, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu prezentovana je osnovna ideja cloud computing tehnologije, ključne karakteristike, modeli, kao i problem sigurnosti. Od mnogobrojnih rješenja koja cloud nudi izdvaja se disaster recovery funkcionalnost koja je detaljno objašnjena. U okviru rada, predstavljena je i realizacija i izvršeno testiranje i analiza date funkcionalnosti.

Ključne reči: Cloud, virtualizacija, virtualni data centar, modeli, sigurnost, DR funkcionalnost

Abstract – This paper describes the basic idea of the cloud computing technology, its key features, models, and security problem. From the number of solutions of this technology, disaster recovery functionality is described as the most significant. Within the paper is also the realization, testing and analysis of the DR functionality.

Keywords: Cloud, virtualization, virtual data center, models, security, DR functionality

1. UVOD

Sa ubrzanim razvojem tehnika za obradu i memorisanje informacija, kao i razvojem Interneta, računarski resursi postaju sve snažniji, dostupniji i jeftiniji više nego ikada. Trend razvoja informacionih i komunikacionih tehnologija (ICT) omogućio je uvođenje i realizaciju novog okruženja koje se naziva *Cloud computing*.

Cloud computing tehnologija omogućava organizacijama da rade u opsegu i koriste agilne sisteme koji im daju konkurentnu prednost. Međutim, od velikog značaja je da organizacije imaju potpuno povjerenje u sigurnost cloud tehnologije i da svi podaci, sistemi i aplikacije budu zaštićeni od krađe podataka, korupcije i brisanja.

Postojanje sistema i procedura za oporavak IT procesa i poslovanja postala je neizostavna funkcija IT sistema. Razvojem virtuelnih tehnologija, kao i *disaster recovery* funkcionalnosti, IT rješenja danas postaju lako primjenljiva i finansijski pristupačna.

Cilj ovog rada jeste da se kroz realizaciju i testiranje *disaster recovery* funkcionalnosti koje predstavlja jedno od cloud rješenja ukaže na važnost bezbjednosti u cloud-u, ali i drugim komunikaciono-informacionim tehnologijama.

NAPOMENA:

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

2. O CLOUD-U

Cloud predstavlja skup hardvera, mreža, memorija i interfejsa koji zajedno pružaju uslugu korisniku, upravljaju resursima i definišu životni ciklus aplikacije. Zadatak cloud tehnologije jeste da korisniku u svakom trenutku, uz prisustvo Internet mreže, dostavi zahtjevane podatke bez potrebe da korisnik poznaje fizičku lokaciju sistema sa koje mu se pruža servis. Dakle, omogućava isporuku IT resursa i servisa na daljinu putem Interneta, telefonske ili privatne mreže.

U skladu sa NIST definicijom [1], postoje tri različita područja koja definišu *cloud computing* tehnologiju:

- a) ključne karakteristike,
- b) modeli pružanja usluga i
- c) modeli implementacije.

2.1. Ključne karakteristike

Postoji pet ključnih karakteristika koje pokazuju odnos i razlike *cloud computing* sistema u odnosu na tradicionalni pristup u računarstvu:

- a) Pružanje usluga na zahtjev korisnika (engl. *On-demand self-service*) - korisnik može samostalno odabrati i pokrenuti računarske resurse, kao i birati vrijeme opsluživanja i mrežni prostor za čuvanje podataka bez potrebe za interakcijom sa elementima pojedinog davaoca usluge.
- b) Širok mrežni pristup (engl. *Broad network access*) - dostupne su usluge putem mreže kojima se pristupa korištenjem standardnih mehanizama.
- c) Alokacija resursa (engl. *Resource pooling*) - korisnik nema kontrolu i znanje o tačnom mjestu upotrebljenih resursa, ali ga može odrediti na većem nivou apstrakcije.
- d) Brza elastičnost (engl. *Rapid elasticity*) - *Cloud computing* nudi korisnicima usluge koje mogu biti brzo i elastično pokrenute kako bi se ostvarilo proporcionalno povećanje ili smanjenje resursa kada oni više nisu potrebni.
- e) Mjerljiva usluga (engl. *Measured service*) - *Cloud* sistemi automatski kontrolišu i optimizuju upotrebu resursa u zavisnosti od potrebe korisnika i tipa usluge koja se traži. Ove usluge su mjerljive i njihovo korišćenje je transparentno, kako za provajdera, tako i za klijente, što je veoma važno sa finansijskog aspekta.

2.2. Modeli pružanja usluga

Cloud computing koristi model pružanja usluga poznat kao SPI (eng. *Software Platform Infrastructure*) i označava tri grupe servisa, a to su:

- a) Softver kao servis – SaaS (engl. *Software as a Service*) - odnosi se na softverske usluge u *cloud-u*. Sastoji se u tome da korisnik iznajmljuje softver od proizvođača koji se nalazi u *data* centru i pruža pristup sistemu preko Interneta na bazi pretplate. Korisnik može da pristupi servisu sa bilo kog uređaja.
- b) Platforma kao servis – PaaS (engl. *Platform as a Service*) - predstavlja kompletno okruženje za razvoj i implementaciju u *cloud-u* sa resursima koji omogućavaju isporučivanje aplikacija, od onih jednostavnih na *cloud-u* do onih sofisticiranih poslovnih. Potrebni resursi se kupuju od provajdera usluga po principu *pay-as-you-go*, a pristup se vrši preko bezbjedne internet veze [2].
- c) Infrastruktura kao servis – IaaS (engl. *Infrastructure as a Service*) - predstavlja koncept u kome se vrši isporučivanje virtuelnih resursa preko interneta, uključujući procesorsku snagu, prostor na disku i memoriju pružajući sigurnost i skalabilnost koju garantuje provajder usluge. Kod ovog modela, korisnik plaća usluge koje koristi po principu *pay-per-use*, a tarifiranje se obavlja iznajmljivanjem po satima korišćenja ili na mjesečnoj bazi.

2.3. Modeli implementacije

U zavisnosti od tipa vlasništva i upravljanja nad IT infrastrukturom *cloud computing-a* razlikujemo sljedeće razvojne modele:

- a) javni oblak (engl. *public cloud*) – predstavlja model kod koga provajderi usluga pružaju svoje resurse kao javno dostupne. Ovakav model prepušten je kompaniji koja nudi *cloud computing* usluge, a usluge su dostupne korisnicima preko zajedničke infrastrukture.
- b) privatni oblak (engl. *private cloud*) - formira se za korišćenje samo jednog korisnika, obezbjeđujući maksimalnu kontrolu nad podacima, bezbjednost, pouzdanost i kvalitet usluga. Može da ga izgradi sama kompanija koja ga koristi ili neki spoljni provajder.
- c) hibridni oblak (engl. *hybrid cloud*) - kombinuje dva ili više različitih oblaka (privatni, zajednički ili javni) tehnologijom koja omogućava razmjenu podataka i aplikacija između njih.
- d) zajednički oblak (engl. *community cloud*) - predstavlja model u kojem nekoliko organizacija dijeli strukturu oblaka. Upravljanje *cloud-om* može vršiti organizacija ili neko treće lice (provajder usluga).

3. SIGURNOSNI PROBLEMI I RIZICI

Glavni problem *cloud computing* tehnologije predstavlja problem sigurnosti i bezbjednosti. Prilikom iznajmljivanja usluga, svaki korisnik očekuje od strane provajdera usluga u *cloud-u* sveukupnu bezbjednost koja uključuje ponudu više različitih rješenja, kao i korišćenje adekvatnih algoritama za enkripciju i kontrolu. Očekuje se da se obezbijedi povjerljivost, integritet i dostupnost podataka.

Cloud, prema [3], u pogledu sigurnosti nudi brojne prednosti u koje se ubrajaju:

- Centralizovana sigurnost
- Smanjeni troškovi
- Smanjena administracija

- Pouzdanost

Takođe, provajderi usluga nude i *Security as a Service* (SECaaS) uslugu koja upravlja sigurnošću podataka korisnika. Ova usluga predstavlja poslovni model u kojem provajder integriše svoje usluge bezbjednosti u korporativnu infrastrukturu na osnovu pretplate.

Pod SECaaS uslugom podrazumijevaju se usluge kao što su upravljanje identitetom i pristupom, zaštita od gubitaka podataka, *web* sigurnost, *e-mail* sigurnost, procjena bezbjednosti, zaštita od upada, enkripcija, kontinuitet poslovanja i *disaster recovery*, kao i bezbjednost informacija i upravljanje događajima.

4. DISASTER RECOVERY FUNKCIONALNOST

U složenom okruženju kao što je *cloud* moguća je pojava grešaka, kao i otkaz opreme, kako zbog ljudskog faktora, tako i zbog sistemskih otkaza i prirodnih činilaca. Iz tog razloga, uspostavljaju se mehanizmi za oporavak od katastrofe (engl. *Disaster Recovery* - DR) koji se odnose na skup politika, alata i procedura koji omogućavaju brz oporavak *cloud* sistema i infrastrukture, a sve sa ciljem da se obezbijedi kontinuitet poslovanja i obavljanja poslovnih aktivnosti. DR predstavlja jedan od najvažnijih, ali i najzahtjevnijih poslova u oblasti informacionih tehnologija.

Ova tehnologija omogućava visok stepen automatizacije u procesu migriranja servisa i virtuelnih mašina sa primarne na rezervnu lokaciju, kao i povratak servisa na primarnu lokaciju [4]. Koristi se virtuelizacija radi pojednostavljenja planiranja i testiranja postupaka za nastavak poslovanja u slučaju katastrofe. Virtuelizacijom se cijeli server, uključujući operativni sistem, aplikacije i podatke, kapsulira u jedan softverski paket ili virtuelni server koji se zatim kopira ili pravi rezervna kopija u rezervni *data* centar i upisuje u virtuelnu mašinu za nekoliko minuta [5].

Proces prenosa i sinhronizacije podataka između više računarskih *data* centara ili sajtova na *cloud-u* naziva se replikacija. Replikacija se vrši automatizovano i u pozadini i transparentna je za korisnike *cloud* sistema.

U svakom *cloud* okruženju postoje dvije vrste replikacije:

- a) sinhrona
- b) asinhrona

U slučaju sinhronne replikacije, upisivanje novih podataka vrši se simultano na dvije lokacije kako bi se obezbijedila sinhronizacija podataka na udaljenom sajtu sa podacima na glavnom sajtu u realnom vremenu.

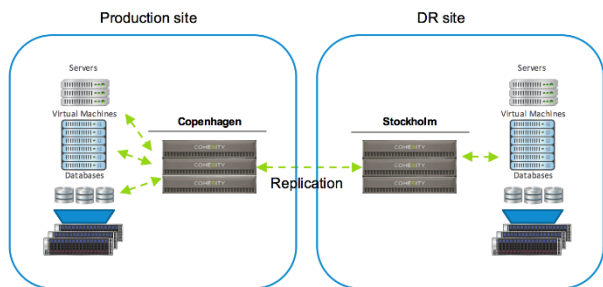
S druge strane, kod asinhronne replikacije, novi podaci upisuju se prvo na primarni sajt, da bi zatim izvršili replikaciju na sekundarni, ali po unaprijed definisanom rasporedu ili u vremenskom periodu kada je opterećenje *cloud* sistema manje.

Način na koji se vrši replikacija prikazan je na slici 1.

4.1. Realizacija funkcionalnosti

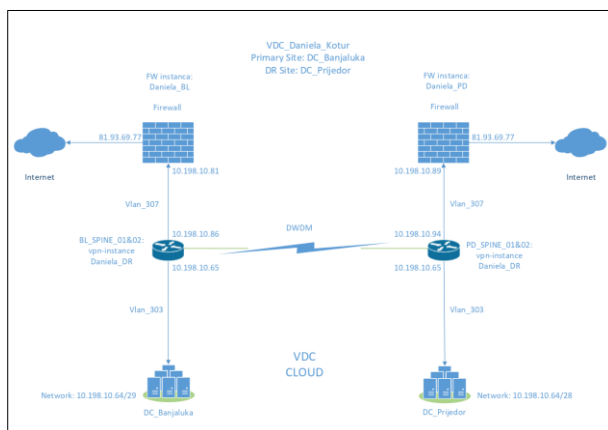
Prvi korak ka realizaciji *Disaster Recovery* funkcionalnosti jeste pravljenje logičke šeme povezivanja uređaja i dodjeljivanja mrežnih adresa, odnosno izrada šeme VDC-a (engl. *Virtual Data Center*). Realizacija ove funkcional-

nosti vršila se u zgradi Mtel-a, telekom operatera u Bosni i Hercegovini.



Slika 1. Replikacija između produkcionog i DR sajta (primarne i rezervne lokacije) [6]

Oprema se nalazi u dva data centra, jednom u Banjaluci, drugom u Prijedoru. DC_Banjaluka predstavlja primarni sajt, dok se DC_Prijedor odnosi na DR sajt. Oba centra rade nezavisno jedan od drugog, ali mogu, po potrebi, da obezbijede *backup* za određene klase servisa ili određene grupe uređaja. Ova dva centra, kao i cjelokupna logička šema prikazana su na slici 2.



Slika 2. Logička šema realizacije VDC-a

Za potrebe ovog rada, korisniku je od strane administratora dodjeljeno četiri virtualna procesora, 16GB radne memorije, 100GB prostora na *storage-u*, kao i četiri virtualne mašine. Ista kvota definiše se za obe zone. Za pravilno odvijanje saobraćaja neophodno je kreirati eksternu mrežu pri čemu svaka mreža koja se fizički kreira postaje virtualna. Na osnovu kreiranog IP plana, VDC-u se dodjeljuje virtualna mreža pod nazivom VLAN_303.

Korisnik sam na svom nalogu kreira onoliko broj virtualnih mašina (engl. *virtual machines* – VMs) koliko mu je neophodno. U ovom radu, kreirane su po dvije virtualne mašine u svakom data centru, jedna sa *Windows*, a druga sa *Linux* operativnim sistemom. Virtualne mašine u drugoj zoni, odnosno u Prijedoru, kreiraju se tek kad se aplicira za DR, odnosno pošalje zahtjev, nakon čega se pojavljuju mašine koje automatski dobijaju sufiks *_dr* pored imena. Takođe, podešeno je da se radi *backup* virtualnih mašina svaki dan 7 dana, nakon čega će kopije biti izbrisane. Moguće je podesiti i trajno čuvanje kopija, ali postoji opasnost od zagađenja, odnosno popunjavanja memorije za skladištenje.

4.1.1. Aplikacija eReplication

Konfiguracioni proces DR-a obavlja se u *eReplication* korisničkom interfejsu i sastoji se od sljedećih modula:

- DR resurs (engl. *DR Resource*)
- Zaštita (engl. *Protection*)
- Oporavak (engl. *Recovery*)

Pod modulom *DR Resource* podrazumijeva se kreiranje sajta na osnovu DR režima i dodaju se resursi povezani sa zaštićenim objektom. U ovom radu, organizacija Mtel predstavlja sajt koji je prethodno bio definisan.

U modulu *Protection* kreira se kolekcija zaštićenih grupa specifičnih za tip aplikacije, a obavlja se preko *FusionSphere-a*, koji ima funkciju supervizora prilikom izvršavanja replikacije. Kolekciju zaštićenih grupa čine mašine koje rade sinhrono i asinhrono.

Posljednji modul konfiguracionog DR procesa predstavlja *Recovery* modul, u okviru kojeg se mogu odabrati sljedeće opcije:

- planirana migracija (engl. *Planned Migration*)
- prinudni oporavak (engl. *Fault Recovery*)
- ponovna zaštita (engl. *Reprotection*)

Izborom opcije planirane migracije izvršiče se replikacija virtualnih mašina smještene na primarnom sajtu na virtualne mašine smještene na DR sajtu, uz minimalan rizik od gubitka podataka. U tom slučaju, virtualne mašine na primarnom sajtu postaju *backup* za one na DR sajtu. Ukoliko dođe do greške tokom izvršavanja, migracija može da se zaustavi i otkloni greška.

Pod *fault recovery* podrazumijeva se da je došlo do neplanirane migracije, odnosno do pada sistema u jednom data centru zbog čega se vrši replikacija na virtualne mašine u drugom data centru, ali ne i *backup* s obzirom da mašine za *backup* nisu u funkciji. Migracija na ovaj način se ne će zaustaviti ako se u toku rada dese bilo kakve greške.

Reprotection se mora izvršiti nakon svake planirane migracije ili prinudnog oporavka radi sinhronizacije u obrnutom smjeru.

4.2. Testiranje funkcionalnosti

Zadatak ovog rada jeste da se testira *Recovery* modul, odnosno da prilikom planiranog ili prinudnog pada sistema na jednoj lokaciji dođe do automatskog pokretanja sistema na drugoj lokaciji i izvršavanja replikacije virtualnih mašina.

Početno stanje sistema prikazano je na slici 3.

Name	Status	IP	Specification	Availability Zone	Tag	Function	User
Daniela_Windows_2	Running	10.198.10.69	CPU: 1CPUs Memory: 2GB	5500V3_BL		Production Cloud Host	zoran_ors
Daniela_CentOS_2	Running	10.198.10.68	CPU: 1CPUs Memory: 1GB	5500V3_BL		Production Cloud Host	zoran_ors
Daniela_Windows_dr	Running	10.198.10.67	CPU: 1CPUs Memory: 2GB	5500V3_PD		DR Cloud Host	Daniela
Daniela_CentOS_dr	Running	10.198.10.66	CPU: 1CPUs Memory: 1GB	5500V3_PD		DR Cloud Host	Daniela

Name	Status	IP	Specification	Availability Zone	Tag	Function	User
Daniela_Windows_2_dr	Stopped	10.198.10.69	CPU: 1CPUs Memory: 2GB	5500V3_PD		DR Cloud Host	Daniela
Daniela_CentOS_2_dr	Stopped	10.198.10.68	CPU: 1CPUs Memory: 1GB	5500V3_PD		DR Cloud Host	Daniela
Daniela_Windows_dr	Stopped	10.198.10.67	CPU: 1CPUs Memory: 2GB	5500V3_PD		DR Cloud Host	Daniela
Daniela_CentOS_dr	Stopped	10.198.10.66	CPU: 1CPUs Memory: 1GB	5500V3_PD		DR Cloud Host	Daniela

Slika 3. Početno stanje virtualnih mašina u dvije zone

Virtualne mašine u Banjaluci (primarni sajt) imaju ulogu produkcije i one su aktivne, dok su virtualne mašine u Prijedoru neaktivne i imaju ulogu DR mašina. Svaka od njih posjeduje svoj status, svoju IP adresu, kao i

specifikaciju. Korisnik je virtualne mašine koje vrše asinhronu replikaciju označio sa brojem 2, radi lakšeg snalaženja.

Prilikom testiranja izvršena je prvo planirana migracija virtualnih mašina koje rade sinhrono u već navedenom *Recovery* modulu, nakon čega se dešava sljedeće: produkcione mašine u Banjaluci se gase, postaju neaktivne, dok se na drugoj lokaciji automatski pale, odnosno postaju aktivne, što se može vidjeti na slici 4. Kao što je već navedeno, mašine u Banjaluci koje rade sinhrono preuzimaju DR ulogu, i postaju *backup* za mašine na drugoj lokaciji - u Prijedoru. Nakon toga, obavezno se bira opcija *Reprotection* radi sinhronizacije.

Na isti način izvršeno je testiranje virtualnih mašina koje rade na asinhron način.

Name	Status	IP	Specification	Availability Zone	Tag	Function	User
Daniela_Windows_2	Running	10.198.10.69	CPU: 1CPUs Memory: 2GB	5500V3_BL		Production Cloud Host	zoran_org
Daniela_CentOS_2	Running	10.198.10.68	CPU: 1CPUs Memory: 1GB	5500V3_BL		Production Cloud Host	zoran_org
Daniela_Windows	Stopped	10.198.10.67	CPU: 1CPUs Memory: 2GB	5500V3_BL		DR Cloud Host	Daniela
Daniela_CentOS	Stopped	10.198.10.66	CPU: 1CPUs Memory: 1GB	5500V3_BL		DR Cloud Host	Daniela

Name	Status	IP	Specification	Availability Zone	Tag	Function	User
Daniela_Windows_2_dr	Stopped	10.198.10.69	CPU: 1CPUs Memory: 2GB	5500V3_PD		DR Cloud Host	Daniela
Daniela_CentOS_2_dr	Stopped	10.198.10.68	CPU: 1CPUs Memory: 1GB	5500V3_PD		DR Cloud Host	Daniela
Daniela_Windows_dr	Running	10.198.10.67	CPU: 1CPUs Memory: 2GB	5500V3_PD		Production Cloud Host	Daniela
Daniela_CentOS_dr	Running	10.198.10.66	CPU: 1CPUs Memory: 1GB	5500V3_PD		Production Cloud Host	Daniela

Slika 4. Planirana migracija virtualnih mašina koje rade sinhrono

Za posljednji slučaj testiranja *Recovery* modula, uzet je *fault recovery* koji je već prethodno objašnjen. Stanje sistema vraćeno je na prvobitno kada su sve virtualne mašine u glavnoj, produkcionalnoj zoni aktivne, a u drugoj, DR neaktivne. Testiranjem ovog slučaja (slika 5), dolazi do gašenja svih mašina (sinhronih i asinhronih) u glavnoj zoni, a automatskog paljenja u DR zoni.

Name	Status	IP	Specification	Availability Zone	Tag	Function	User
Daniela_Windows_2	Stopped	10.198.10.69	CPU: 1CPUs Memory: 2GB	5500V3_BL		Production Cloud Host	zoran_org
Daniela_CentOS_2	Stopped	10.198.10.68	CPU: 1CPUs Memory: 1GB	5500V3_BL		Production Cloud Host	zoran_org
Daniela_Windows	Stopped	10.198.10.67	CPU: 1CPUs Memory: 2GB	5500V3_BL		Production Cloud Host	Daniela
Daniela_CentOS	Stopped	10.198.10.66	CPU: 1CPUs Memory: 1GB	5500V3_BL		Production Cloud Host	Daniela

Name	Status	IP	Specification	Availability Zone	Tag	Function	User
Daniela_Windows_2_dr	Running	10.198.10.69	CPU: 1CPUs Memory: 2GB	5500V3_PD		Production Cloud Host	Daniela
Daniela_CentOS_2_dr	Running	10.198.10.68	CPU: 1CPUs Memory: 1GB	5500V3_PD		Production Cloud Host	Daniela
Daniela_Windows_dr	Running	10.198.10.67	CPU: 1CPUs Memory: 2GB	5500V3_PD		Production Cloud Host	Daniela
Daniela_CentOS_dr	Running	10.198.10.66	CPU: 1CPUs Memory: 1GB	5500V3_PD		Production Cloud Host	Daniela

Slika 5. Prinudni oporavak virtualnih mašina

Ovaj slučaj razlikuje se od prethodnih iz razloga što ne postoji *backup* za mašine koje trenutno rade, odnosno nijedna virtualna mašina ne preuzima DR ulogu, što se može detaljnije vidjeti na slici. Kao i nakon svake uspješno obavljene migracije, vrši se *Reprotection*.

5. ZAKLJUČAK

Cloud computing tehnologija predstavlja nov koncept u računarstvu koji se intenzivno razvija još od 2006. godine. Danas je moguće koristiti ovu tehnologiju na način na koji se koriste komunalne usluge – računarski resursi se plaćaju prema potrošnji.

DR usluga u *cloud-u* daje mogućnost potpune verifikacije i testiranja rješenja za oporavak, kao i dodatnu mogućnost provjere na zahtjev. Naplaćivanje se vrši prema obimu angažovanih resursa, pa rješenje postaje ekonomična usluga, potpuno usklađena sa stvarnim potrebama korisnika. Potpuna nezavisnost od hardvera daje mogućnost oporavka servisa na drugoj infrastrukturi, čak i u slučaju uništenja infrastrukture korisnika. Na ovaj način obezbjeđuje se kontinuitet poslovanja svake kompanije.

U ovom radu izvršeno je testiranje DR funkcionalnosti koje može poslužiti kao dokaz da rješenje sa sigurnošću funkcioniše i da takav način predstavlja najisplativiji, ali i najbezbedniji način korištenja informacione i komunikacione tehnologije. Takođe, predstavljena je i *cloud computing* tehnologija koja se rapidno razvija posljednjih nekoliko godina i koja donosi sa sobom velike promjene u organizaciji informaciono-komunikacionog sistema.

6. LITERATURA

- [1] Peter Mell, Timothy Grance, „The NIST Definition of Cloud Computing”, National Institute of Standards and Technology, Gaithersburg, 2011.
- [2] <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/>, pristupljeno u avgustu 2019.
- [3] <https://www.forcepoint.com/cyber-edu/cloud-security>, pristupljeno u septembru 2019.
- [4] <https://coming.rs/kontinuitet-poslovanja/oporavak-u-slucaju-katastrofe/>, pristupljeno u septembru 2019.
- [5] <https://www.otava.com/reference/the-benefits-of-disaster-recovery-in-cloud-computing/>, pristupljeno u septembru 2019.
- [6] <http://www.vfrank.org/2017/02/08/disaster-recovery-and-site-redundancy-with-cohesity/>, pristupljeno u septembru 2019.

Kratka biografija:



Daniela Kotur rođena je u Prijedoru 1995. god. Osnovne akademske studije na Fakultetu tehničkih nauka završila je 2018. god. Master studije iz oblasti Elektrotehnike i računarstva – Telekomunikacioni sistemi upisala je školske 2018/2019, a master rad odbranila 2019. god.



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

ПРОШИРЕЊЕ MONGODB БАЗЕ ПОДАТАКА ОТВОРЕНОГ КОДА ЗА ИНДЕКСИРАЊЕ И ПРЕТРАЖИВАЊЕ СРПСКИХ ТЕКСТОВА**AN EXTENSION OF THE MONGODB OPEN SOURCE DATABASE FOR INDEXING AND SEARCHING SERBIAN TEXTS**

Стефан Петковић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – Рад описује имплементацију проширења, MongoDB базе података отвореног кода, за подршку индексирања и претраживања српских текстова писаних на оба писма.

Кључне речи: Дигитални документи, индексирање, претрага, претпроцесирање, стеминг, Snowball, MongoDB

Abstract – This paper describes the implementation of the extension for the open-source MongoDB database that adds support for indexing and searching of Serbian texts written in both alphabets.

Keywords: Digital documents, indexing, search, preprocessing, stemming, Snowball, MongoDB

1. УВОД

Људи су одувек имали потребу да забележе одређене важне догађаје, стечена искуства и знање и тако сакупљене информације пренесу наредним генерацијама. Сlike на зидовима пећине Ласко, записи о кретању звезда, медицинске књиге које описују анатомију људског тела – све оно што су људи научили и открили се бележи и преноси [1]. Свака писана или цртана репрезентација мисли или неког догађаја која нам пружа одређене информације, доказе или се користи као службени записник сматра се документом [2].

Данас када кажемо документ можемо мислити на две ствари:

1. традиционални папирни документ и
2. дигитални документ [3].

Дигитални документ је заправо рачунарски обрађена информација којом се рукује као основном јединицом обраде. Примери дигиталних докумената су:

- текстуални дигитални документи (текстуални описи или поруке)
- графички документи (слике, цртежи, дијаграми, графикони)
- структурирани документи – *Hypertext Markup Language, Extensible Markup Language* [3].

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Драган Ивановић, ванр. проф.

MongoDB база података омогућава индексирање и претрагу садржаја складиштених у текстуалним пољима докумената. Тренутна верзија ове базе података подржава индексирање и претрагу текстуалних садржаја на данском, енглеском, италијанском, мађарском, немачком, норвешком, португалском, румунском, руском, турском, финском, француском, холандском, шведском и шпанском језику.

Тема мастер рада јесте да прикаже опис имплементације проширења функционалности текстуалне претраге. Проширење описано у овом раду имплементирано је за потребе претраге текстуалних садржаја на српском језику, али је на исти начин могуће додати подршку и за друге језике.

2. ТЕОРИЈСКЕ ОСНОВЕ

Системи за проналажење информација нам омогућују да пронађемо оно што нам је од значаја у мору информација унутар великих колекција дигиталних докумената. Своју потребу за информацијама корисници систему представљају задавањем одговарајућег упита. Задати упит се прво обради, а затим се на основу обрађеног упита врши претрага колекције. Обрада упита и претрага колекције, за информацијама од значаја, представљају процес претраживања. Како би се обезбедило ефикасно претраживање неопходно је извршити припрему постојећих докумената у колекцији, а такође и сваког новог документа приликом додавања. Ова припрема представља процес индексирања.

Приликом претраге великих колекција дигиталних докумената не захтева се од корисника да буде упознат са колекцијом коју претражује, нити се захтева прецизно дефинисање критеријума претраге (подразумева се непрецизност). Као резултат претраге добија се листа докумената која често поред релевантних резултата садржи и оне који то нису. Системи за проналажење информација теже да пронађу све релевантне документе, а да се при томе број нерелевантних докумената сведе на минимум.

Текстуални дигитални документи се састоје од речи које представљају одређени број знакова у тексту. Токен представља инстанцу речи која се појављује у документу, а једна класа еквиваленције речи представља терм. Ако се речи након примене одређених правила за претпроцесирање текста сведе на исти низ знакова, онда припадају истој класи еквиваленције [3]. На пример следећа реченица:

Миш уз пушку, миш низ пушку.

садржи шест токена, док се број термова зна тек након примене правила за претпроцесирање текста. Јасно нам је да трећи и шести токен („пушку”) представљају један терм, али први („Миш”) и четврти токен („миш”) могу припадати истој класи еквиваленције само ако се примени правило пребацивања свих слова у мала слова.

Претпроцесирање текста представља примену одређених правила на текстуалне садржаје који обезбеђују флексибилност претраге [3]. Иста правила се примењују и на документе и на упите, а сам квалитет претраге у великој мери зависи од ових правила.

Лематизација представља редукцију различитих граматичких облика речи (лица, родови, падежи, времена, итд.) на њихов базни речнички облик, односно лему. Стеминг је груби хеуристички процес који одсеца крајеве речи са циљем да се постигне резултат што је могуће сличнији ономе који се постиже правилном лематизацијом базираном на лингвистичком знању [3]. Резултат стеминга (стем) представља одређени низ карактера који није нужно лема, односно корен улазне речи, а не мора чак ни постојати као реч у оквиру језика. Важно је једино да употреба стеминга допринесе повећању перформанси система.

3. ТЕХНОЛОГИЈЕ КОРИШЋЕНЕ ЗА ИМПЛЕМЕНТАЦИЈУ

MongoDB јесте документ оријентисана база података отвореног кода. Сваки рекорд складиштен у **MongoDB** бази података представља један BSON документ (*Binary JSON – JavaScript Object Notation*) кога чине парови поља и њихових вредности.

MongoDB складишти документе у колекцијама (*collections*), а колекције у базама података (*databases*). Колекције се могу упоредити са табелама у релационим базама података.

Базе података и колекције у **MongoDB** не морају се експлицитно креирати већ ће оне бити креиране приликом првог складиштења докумената у њима – имплицитно креирање. Због ове чињенице могуће је изабрати непостојећу базу података и у нову колекцију убаци документ. Убацивањем документа у колекцију креирају се и нова база података и нова колекција, уколико нису већ постојале.

Основне операције за рад са документима обухватају операције за креирање (додавање), измену, брисање и претрагу, односно читање докумената. Све наведене операције се извршавају над једном колекцијом. Атомичност (*atomicity*) операција за манипулацију документима (креирање, измена и брисање) остварена је на нивоу једног документа, чак и онда када се операција врши над документом који садржи више других докумената. Уколико једна операција врши модификацију већег броја независних докумената модификација сваког документа понаособ је атомична, али операција у целини није.

Приликом коришћења операција за претрагу докумената потребно је задати одговарајући критеријум пре-

траге, односно филтер. Уколико се филтер изостави, односно уколико се проследи празан филтер резултујућа листа ће садржати све документе колекције.

Ефикасније извршавање операција за претрагу постиже се коришћењем индекса. Уколико се над колекцијом не креирају индекси претрага захтева проверу и поређење сваког документа у колекцији са задатим упитом (филтером).

Индекси представљају специјалне структуре података које чувају мали део скупа података у облику који се лакше претражује. У **MongoDB** индекси представљају Б-Стабла у којима се чувају сортиране вредности једног или више поља.

Snowball је прост језик за обраду ниски (*string*) креиран по узор на **SNOBOL** (*String Oriented and Symbolic Language*).

Основни типови података којима **Snowball** рукује јесу ниске, означени цели бројеви (*signed integers*) и Булов тип података (*booleans*). Називи могу садржати слова, цифре и доњу црту, али морају почињати словом. Везују се за типове података, рутине (*routines*), групације (*groupings*) и споља видљиве интерфејсе (*externals*). Сви називи морају бити декларисани.

Рутине представљају секвенце једне или више наредби. Након дефинисања, рутине се могу позвати једноставним навођењем њиховог имена.

Ток извршавања **Snowball** програма контролише се имплицитном употребом сигнала, уместо експлицитне употребе различитих наредби за контролу тока (*if, else, break*, итд.). Све наредбе у оквиру **Snowball** језика резултују позитивним (*true*) или негативним (*false*) сигнаlima. Уколико наредба резултује негативним сигналом наредбе које следе након ње се игноришу, а у супротном даље извршавање наредби се наставља.

Snowball програм се састоји од декларација након којих следе дефиниције групација и рутина. Декларисане променљиве доступне су из било које тачке програма и постоје све до краја његовог извршавања. Позив програма остварује се путем **Snowball API** (*Application Programming Interface*) кроз његове споља видљиве интерфејсе (*externals*). **Snowball API** дефинише појам тренутне ниске (*current string*) над којом споља видљиви интерфејси врше обраду. Вредност тренутне ниске, након извршене обраде, јесте резултат **Snowball** програма.

4. СТЕМЕР ЗА СРПСКИ ЈЕЗИК

Српски језик припада породици индоевропских језика, тачније словенској групи, а јужнословенској подгрупи језика. Због чињенице да је српски језик флективан језик његов стемер садржи далеко већи број правила од стемера за језике који то нису. Као основа стемера за српски језик коришћена су правила за стеминг хрватског језика, аутора Николе Љубешића и Ивана Панчића [4]. Постојећа правила су модификована и допуњена за потребе стеминга српског језика. За опис алгоритма коришћен је **Snowball** језик.

Главне рутине алгоритма за стеминг српског језика су:

Транслитерација (пресловљавање) ћирилице у латиницу – Српски језик користи два писма, ћирилицу и латиницу, а како је алгоритам за стеминг развијен са циљем претраживања информација, доступност информација на оба писма је од великог значаја.

Припрема (*prelude*) – Српски језик, поред два писма, користи и два дијалекта: екавски и ијекавски. На пример, речи сенка, сјенка и сијенка имају исто значење али су написане различитим дијалектом. Наведене речи потребно је сврстати под исту класу еквиваленције (терм), што се постиже трансформацијом екавице у ијекавицу. Такође, све је чешће у употреби неправилна комбинација латиничних слова „d” и „j” уместо слова „đ”, па је и токене као на пример „Djoković” и „Đoković” потребно третирати као исти терм.

Обележавање региона (*mark regions*) – Алгоритам за стеминг српског језика користи један регион *R1* којим се одређује да ли ће стеминг бити примењен или не. Уколико најдужи суфикс из листе упада у регион *R1*, односно уколико не прелази тај регион, стеминг ће бити примењен. Регион *R1* може бити:

1. Регион након првог самогласника уколико се изван њега налазе најмање два слова, у супротном је то регион након првог сугласника који прати самогласник.
2. Регион након првог слова „p” уколико се изван њега налазе најмање два слова, у супротном је то регион након било којег првог слова које прати слово „p”.

Поред региона *R1* постоји и тестна рутина *R2* којом се проверава постојање слова са дијакритицима. Одређена правила за стеминг се примењују само уколико тестна рутина резултује позитивним сигналом, односно уколико реч не садржи слова са дијакритицима.

Морфолошке промене – Представљају последњи корак пре примене стеминга и извршавају се са циљем добијања истих стемова за различите облике речи. На пример, за следеће речи потребно је генерисати исте стемове:

1. „правилан” - мушки род, једнина
2. „правилна” - женски род, једнина
3. „правилно” - средњи род, једнина

Како би добили исти стем за претходно наведене речи, прву реч („правилан”) је потребно из једине мушког рода трансформисати у множину мушког рода („правилни”), а затим применити правила за стеминг.

Стеминг – Процес стеминга подељен је у два корака, односно две рутине. Примарна рутина садржи највећи број правила, док секундарна садржи само неколицину. Секундарна рутина се примењује само у случају када примарна рутина резултује негативним сигналом, односно не изврши стеминг речи.

Евалуација стемера за српски језик извршена је употребом програма аутора Петра Дамјановића [5]. Резултат програма представљају прецизност, поврат, тачност и Ф-мера (*F-measure*) које означавају колико добро одређени стемер ради над колекцијом текстуалних докумената. Резултати евалуације стемера за српски језик, добијени употребом овог програма:

Precision : 0.9468657878535981
Recall : 0.8793230936500129
F-measure : 0.9118453871663618

5. ТЕКСТУАЛНА ПРЕТРАГА У ОКВИРУ MONGODB

MongoDB омогућава претрагу садржаја складиштених у текстуалним пољима докумената коришћењем *\$text* оператора.

Како би се текстуална претрага могла користити, над колекцијом докумената је потребно дефинисати текстуални индекс. Текстуални индекс чува по један запис за сваку класу еквиваленције сваког индексiranог поља у оквиру сваког документа унутар колекције. Креирање текстуалног индекса, над колекцијом докумената, постиже се извршавањем операције:

```
db.collection.createIndex({fieldName:"text"})
```

где *fieldName* представља назив поља чији садржај се индексира, а „*text*” представља текстуални литерал којим се специфицира текстуални тип индекса.

Потпроцес токенизације и претпроцесирања се, у зависности од одабране верзије текстуалног индекса, обавља у оквиру *moveNext* методе једне од класа: *BasicFTSTokenizer* или *UnicodeFTSTokenizer*. Наведена метода проналази наредни токен унутар текстуалног садржаја, а затим врши његово претпроцесирање. Претпроцесирање токена, за све верзије текстуалних индекса, укључује филтрирање стоп речи и стеминг. Новину у трећој верзији текстуалних индекса представља брисање дијакритика након стеминга, што резултује индексима који нису осетљиви на слова са и без истих. Листа стоп речи, као и алгоритам за стеминг одређују се на основу изабраног језика приликом креирања индекса.

Први корак имплементације проширења представља регистрацију новог језика. Језици су представљени *BasicFTSLanguage* и *UnicodeFTSLanguage* класама, које поседују методе за креирање инстанци *FTSTokenizer* класа. Макро директивом:

```
MONGO_FTS_LANGUAGE_DECL(Serbian,"serbian","rs")
```

региструје се нови (у овом случају српски) језик за другу и трећу верзију текстуалног индекса, док се макро директивама:

```
MONGO_FTS_LANGUAGE_DECLARE(languageSerbianV1,  
"serbian",TEXT_INDEX_VERSION_1);
```

и

```
MONGO_FTS_LANGUAGE_DECLARE(languageSerV1, "ser",  
TEXT_INDEX_VERSION_1);
```

региструје нови језик за прву верзију текстуалног индекса.

Након регистрације језика потребно је додати нову листу стоп речи за дати језик. Листе стоп речи чувају се у текстуалним датотекама где сваки ред представља једну реч. Назив датотеке задаје се у формату:

```
stop_words_nazivJezika.txt
```

Креирану листу потребно је регистровати у оквиру скрипте за изградњу *FTS* модула.

Наредни корак представља додавање алгорита за стеминг новорегистрованог језика, уколико одговарајући не постоји у оквиру *MongoDB*. Како је алгоритам за стеминг описан *Snowball* језиком, пре додавања исти је потребно превести у *ANSI C*. Након преводијења нови алгоритам треба дефинисати у оквиру *libstemmer_c* библиотеке коју користи *MongoDB*.

Последњи корак представља проширење *codepointRemoveDiacritics* функције. Наведена функција уклања дијакритике, односно трансформише слова са дијакритиком у њима одговарајућа слова без дијакритика. Овај корак није обавезан када се сва потребна правила већ налазе у оквиру *codepointRemoveDiacritics* функције, међутим у случају проширења за српски језик недостајала су правила за трансформацију слова „Ђ” и „ђ” у њима одговарајућа слова без дијакритика „D” и „d”.

6. РЕЗУЛТАТИ И ДИСКУСИЈА

Након имплементације проширења, како би се исто могло испробати, потребно је покренути процедуру за изградњу *MongoDB* базе података (сервера и клијента).

По покретању сервера и клијента, а пре претраге текстуалних садржаја, потребно је креирати колекцију докумената и над њом текстуални индекс за српски језик.

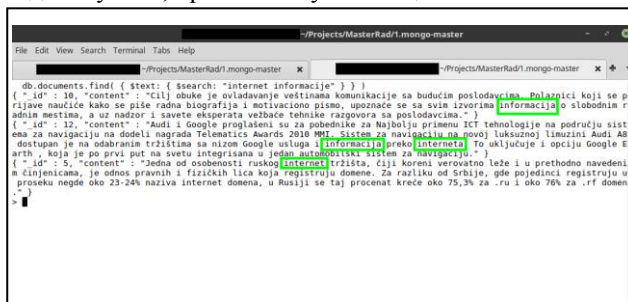
Претрага текстуалног садржаја, на српском језику, у оквиру претходно креиране колекције обавља се операцијом:

```
db.documents.find({$text:{$search:query}})
```

где *query* представља задати упит. На пример, уколико је потребно пронаћи сва документа на српском језику која садрже неку од речи “интернет” или “информације” (у било ком облику), упит се задаје на следећи начин:

```
db.documents.find({$text:{$search:"internet informacije"}})
```

Резултати текстуалне претраге, на основу претходно задатог упита, приказани су на слици 1.



Слика 1. Документи који садрже неку од речи: „интернет” или „информације”.

Проширење текстуалне претраге описано у овом раду представља само један начин на који се одређене ствари могу имплементирати. Процеси транслите-

рације, трансформације ијекавице у екавицу и трансформације комбинације слова „d” и „j” у слово „đ” не морају бити део стеминг алгорита већ се могу имплементирати у оквиру *moveNext* методе.

7. ЗАКЉУЧАК

Циљ рада је био да прикаже опис имплементације проширења функционалности текстуалне претраге доступне у оквиру *MongoDB* базе података отвореног кода. Проширење омогућава индексирање и претрагу текстуалних садржаја на српском језику, без обзира на писмо (ћирилица, латиница) или дијалект (екавица, ијекавица) којим је садржај писан.

Простора за даљи развој решења свакако има. Побољшање рутине за означавање региона, смањење броја правила у оквиру алгорита за стеминг и ревизија листе стоп речи представљају неке од могућих даљих унапређења описаног проширења.

8. ЛИТЕРАТУРА

[1] <https://academichelp.net/samples/academics/essays/expository/passing-on-knowledge.html> (датум приступа: 2019-08-12)

[2] <http://people.ischool.berkeley.edu/~buckland/digdoc.html> (датум приступа: 2019-08-12)

[3] Ивановић Д., Милосављевић Б. (2015), Управљање дигиталним документима, ISBN: 978-86-7892-690-7, Факултет техничких наука, Нови Сад, Србија

[4] <http://nlp.ffzg.hr/resources/tools/stemmer-for-croatian/> (датум приступа: 2019-08-12)

[5] <https://github.com/pedam91/Stemmer-evaluation> (датум приступа: 2019-08-12)

Кратка биографија:



Стефан Петковић је рођен у Сремској Митровици, држава Република Србија. Основну школу „Јован Јова-новић Змај” завршио је 2007. године у Сремској Митровици. Средњу економску школу „9. Мај” у Сремској Митровици завршио је 2011. године. Исте године уписао се на Факултет техничких наука у Новом Саду, одсек Рачунарство и аутоматика. Школске 2013/2014. године уписао се на смер Рачунарске науке и информатика. Положио је све испите предвиђене планом и програмом. Дипломирао је 2016. године, након чега уписује мастер студије на истом смеру, са усмерењем на Електронско пословање. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Примењене рачунарске науке о-дбранио је 2019. године.

DETEKCIJA STRESA KOD PILIĆA TOVLJENIKA NA OSNOVU ANALIZE ZVUKA STRESS DETECTION IN BROILER CHICKENS THROUGH SOUND ANALYSIS

Nina Maljković, Nikša Jakovljević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – OBRADA SIGNALA

Kratak sadržaj – Ovaj rad daje prikaz sistema za detekciju stresa kod pilića tovljenika na osnovu analize zvuka njihovog oglašavanja. Skup obeležja na osnovu kojih ovaj sistem vrši prepoznavanje čine: energija, snaga, kvadratna sredina, džiter, šimer, prosečna visina zvuka, odnos harmonik-šum, izlazi iz mel-filtar banke i mel-frekvencijski kepsralni koeficijenti. Kao klasifikator iskorišćen je klasifikator na bazi vektora nosača. Tačnost sistema na nivou frejma od 50 ms varira od 61 do 88%, u zavisnosti od starosti tovljenika i izbora obeležja.

Ključne reči: tovnji pilići, detekcija stresa, analiza zvuka, SVM

Abstract – This paper presents a system for stress detection in broiler chickens through analysis of their sounds. The feature set used in the system is composed of energy, power, root mean square, jitter, shimmer, average pitch, harmonic-noise ratio, outputs of mel filter bank and mel-frequency cepstral coefficients. Support vector machine is used as a classifier. System accuracy on the 50 ms long frame level varies from 61 to 88 %, depending on how old broilers are and selected features.

Keywords: broiler chickens, stress detection, sound analysis, SVM

1. UVOD

U poslednjih nekoliko godina razvijen je koncept *Precision Livestock Farming*, koji se bavi monitoringom procesa uzgoja domaćih životinja i upravljanjem ovim procesom [1]. Koncept je osmišljen kao koristan alat za pomoć farmerima u ostvarivanju što efikasnijeg uzgoja. Monitoring podrazumeva da se sve varijable koje opisuju stanje životinja kontinuirano prate i u tu svrhu se koriste razne vrste senzora: kamere, senzori temperature i vlažnosti vazduha, senzori protoka [2].

Primena mikrofona kao senzora je naročito interesantna jer zvuk sadrži mnoštvo korisnih informacija o stanju životinja. Snimanje zvuka mikrofonom je beskontaktno i nezavisno od osvetljenja. Takođe, jednim mikrofonom je moguć monitoring velike grupe životinja, što doprinosi ekonomičnosti snimanja.

Pilići, čuveni po svom oglašavanju, predstavljaju odlične kandidate za monitoring mikrofonom. Mnogi iskusni farmeri u stanju su da slušanjem pilića prepoznaju kada

nešto nije u redu sa jatom i da razlikuju pijukanje zdravih i zadovoljnih pilića od pijukanja bolesnih ili uznemirenih. Razvijanje sistema koji bi vršio monitoring jata pomoću mikrofona i detekciju narušavanja normalnog stanja u jatu bilo bi od izuzetne koristi farmerima jer bi takav sistem mogao blagovremeno da ih upozori na situacije koje ugrožavaju blagostanje životinja [3].

Stres je jedan od faktora koji mogu narušiti blagostanje životinja. Na komercijalnim živinarskim farmama prisutno je mnogo faktora koji stresno deluju na piliće. Određene procedure redovnog održavanja (na primer, čišćenje prostorija) predstavljaju potencijalni stresor, a i promene uslova okoline (na primer, temperature vazduha) mogu stresno delovati na piliće [4]. Prolongirani stres štetno deluje i na koke nosilje i na piliće tovljenike i može dovesti do značajnih gubitaka u procesu uzgoja. Zbog toga je bitno da se pojava stresa u jatu blagovremeno detektuje i da se stresori koji su do te pojave doveli što pre uklone. Monitoring jata mikrofonom je dobro rešenje ovog postavljenog zadatka.

U ovom radu predstavljen je neinvazivan sistem za detekciju stresa kod pilića tovljenika starih između 1 i 5 nedelja. Iz audio-snimaka oglašavanja pilića ekstrahovana su određena vremenska i spektralna obeležja, na osnovu kojih je vršena klasifikacija stanja u dve klase – normalno i stresno stanje. Klasifikacija je vršena pomoću metode vektora nosača (engl. *Support vector machine*, SVM), a uspešnost sistema ocenjena je računanjem njegove osetljivosti, specifičnosti i tačnosti.

2. BAZA PODATAKA

Baza audio-snimaka koja je korišćena za obuku i testiranje sistema snimana je na živinarskoj farmi. Za snimanje je korišćen mikrofonski niz ReSpeaker 4 Mic Array for Raspberry Pi [5], koji čine 4 mikrofona SPU0414HR5H-SB bazirana na SiSonic™ MEMS tehnologiji. Baza se sastoji od 85 audio-snimaka dužine od po 10 minuta. Audio signal oglašavanja pilića tovljenika uzorkovan je na 16 kHz sa po 16 bita po uzorku i zapisan je u stereo mp3 formatu.

Snimci su podeljeni u kategorije po nedelji starosti pilića i po njihovom psihološkom stanju. Podela u zavisnosti od starosti pilića je uslovljena činjenicom da se visina zvuka kojim se oglašavaju pilići značajno menja tokom vremena (u prve 3 nedelje života opadne sa 3.5 kHz na 1.5 kHz) [6]. U zavisnosti od psihološkog stanja razlikujemo: normalno stanje (NS), stanje stresa usled povišene temperature vazduha (SS1), stanje stresa zbog procedura redovnog održavanja objekta (SS2) i stanje stresa usled hvatanja i iznošenja pilića iz objekta (SS3). Raspodela snimaka po kategorijama prikazana je u tabeli 1.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Nikša Jakovljević, docent

U inicijalnoj verziji sistema bilo je bitno detektovati samo stresno stanje (ali ne i njegov uzrok) i stoga su stresna stanja objedinjena u jednu klasu. Na audio-snimcima je, pored pijukanja pilića, prisutna i buka koja potiče od ventilatora i hranilice, koja je delimično maskirala oglašavanje samih pilića. Međutim, ovakva buka je neizbežna na savremenim živinarskim farmama i sistem obučen na „idealnim“ snimcima, bez ovog vida buke, bi u realnim uslovima imao izuzetno loše performanse.

Tabela 1. *Raspodela snimaka po starosnim i psihološkim kategorijama. Psihološke kategorije su: NS – normalno stanje, SS – stanje stresa prouzrokovano: SS1 – povišenom temperaturom vazduha, SS2 – redovnim održavanjem i SS3 – hvatanjem i iznošenjem pilića iz objekta.*

Starost/stanje	NS	SS1	SS2	SS3
Nedelja 1	2	1		
Nedelja 2	2	1	5	
Nedelja 3	2	1	20	
Nedelja 4+	2	2	40	7
Ukupno	8	5	65	7

3. PREDOBRADA SIGNALA

Stereo snimci su konvertovani u mono snimke izdvajanjem samo levog kanala. Izvršena je normalizacija signala oduzimanjem srednje vrednosti i deljenjem maksimalnom amplitudom. Svaki segment propušten je kroz pojasni filter određenih graničnih frekvencija. Filtriranje je urađeno u cilju izdvajanja frekvencijskog opsega u kojem se nalazi pijukanje, a ujedno i da bi se u što većoj meri potisnula buka koja potiče od mašina. Frekvencijski opsezi od interesa određeni su na osnovu rada [6], koji opisuje promenu frekvencije oglašavanja pilića sa njihovim starenjem. Za sva filtriranja korišćen je FIR filter dužine 265 tačaka sa Hamingovom prozorskom funkcijom. Granične frekvencije filtera za svaku nedelju starosti pilića prikazane su u tabeli 2.

Tabela 2. *Granične frekvencije filtera za svaku nedelju starosti pilića*

Starost	Donja granična frekvencija	Gornja granična frekvencija
Nedelja 1	2000 Hz	3800 Hz
Nedelja 2	2000 Hz	3800 Hz
Nedelja 3	900 Hz	3800 Hz
Nedelja 4+	900 Hz	3800 Hz

4. KORIŠĆENA OBELEŽJA

Signal se deli na segmente (frejmove) dužine 50ms, tako da se uzastopni međusobno preklapaju po 50%. Za svaki od ovih segmenata se izdvajaju obeležja. Radi bolje preglednosti korišćena obeležja moguće je podeliti u zavisnosti od domena u kom se izdvajaju na: i) vremenska i ii) spektralna (frekvencijska) obeležja.

4.1. Vremenska obeležja

Grupu vremenskih obeležja čine sledeća obeležja: energija signala, snaga signala, kvadratna sredina signala (engl. *root mean square*, RMS), džiter (engl. *jitter*), šimer (engl. *shimmer*), visina zvuka (engl. *pitch*) i odnos harmonik-šum (engl. *harmonic-to-noise ratio*, HNR). Sličan set obeležja korišćen je u radu [4], koji se bavi detekcijom i klasifikacijom stresa kod koka nosilja.

Energija segmenta signala [7] između trenutaka N_1 i N_2 definiše se kao:

$$E = \sum_{n=N_1}^{N_2} x^2(n), \quad (1)$$

gde su $x(n)$ vrednosti signala u trenucima n .

Prosečna energija po jednom uzorku [7], koja se još naziva i snaga signala, računa se kao:

$$P = \frac{1}{N_2 - N_1 + 1} \sum_{n=N_1}^{N_2} x^2(n). \quad (2)$$

RMS [7] se računa po formuli:

$$RMS = \sqrt{\frac{1}{N_2 - N_1 + 1} \sum_{n=N_1}^{N_2} x^2(n)}. \quad (3)$$

Džiter je obeležje koje opisuje varijacije trajanja osnovne periode audio signala, a time i varijacije osnovne učestanosti. Računa se kao:

$$Jitter = \frac{1}{N - 1} \sum_{n=1}^{N-1} (T_n - T_{n-1}), \quad (4)$$

gde je N broj osnovnih perioda signala, a T_n trajanje n -te osnovne periode [7].

Šimer opisuje varijacije amplitude između uzastopnih osnovnih perioda audio signala. Računa se po formuli:

$$Shimmer = \frac{1}{N - 1} \sum_{n=1}^{N-1} 20 \cdot \log_{10} \left(\frac{A_n}{A_{n-1}} \right), \quad (5)$$

gde je A_n razlika između maksimalne i minimalne vrednosti signala unutar n -te osnovne periode [7].

Visina zvuka je termin pod kojim se podrazumeva percipirana tonalnost, odnosno, frekvencija zvuka. U ovom radu, visina zvuka izjednačena je sa osnovnom učestanošću zvuka.

HNR se definiše kao odnos energije harmonijskih (periodičnih) komponenti signala i energije komponenti koje potiču od šuma. Za računanje HNR korišćena je metoda zasnovana na autokorelaciji, odnosno:

$$HNR = 10 \cdot \log_{10} \left(\frac{AC(T)}{AC(0) - AC(T)} \right), \quad (6)$$

gde je $AC(t)$ autokorelacija signala za pomeraj t , a T prosečna osnovna perioda na posmatranom segmentu [7]. Izdvajanje džitera, šimera i HNR-a podrazumeva da je audio signal podeljen na osnovne periode. Za detekciju osnovne periode u ovom radu korišćena je metoda autokorelacije signala u ciljanom opsegu osnovnih perioda [7].

U slučaju džitera, šimera, visine zvuka i HNR-a, ceo segment je najpre *upsample*-ovan 3 puta i zatim je nad njim izvršena podela na osnovne periode, da bi se dobila finija vremenska rezolucija.

4.2. Spektralna obeležja

Grupu spektralnih obeležja čine: energije odziva mel filtera banke, mel-frekvencijski kepsralni koeficijenti (engl. *Mel-frequency cepstral coefficients*, MFCC) i formanti.

Spektralna obeležja korišćena su u radovima koji se bave analizom zvuka oglašavanja živine [3, 8], a takođe i u radovima koji se bave klasifikacijom vrsta ptica na osnovu njihove pesme [9, 10].

Ekstrakcija energije odziva mel filter banke za svaki frejm odvija se kroz sledeća 3 koraka [11]:

1. Množenje uzoraka u frejmu Hamingovom prozorskom funkcijom;
2. Računanje amplitudskog spektra primenom brze Furijeove transformacije u 1024 tačke;
3. Propuštanje izračunatog amplitudskog spektra kroz filter banku koju čini 13 međusobno preklapljenih trougaonih pojasnih filtera iste širine, ravnomerno raspoređenih na mel frekvencijskoj skali u opsegu od interesa.

Logaritmovanjem energije odziva mel filter banke i primenom diskretne kosinusne transformacije nad njima dobijaju se MFCC (u ovoj studiji njih 13).

Formanti se definišu kao maksimumi obvojnice spektra signala. Karakterišu se pozicijom na frekvencijskoj osi i širinom. Za ekstrakciju formanta u ovom radu korišćena je metoda zasnovana na pronalaženju koeficijenta linearnog prediktivnog modela vokalnog trakta, pri čemu su posmatrana samo 2 formanta [7].

5. SVM KLASIFIKATOR

SVM je binarni klasifikator koji klasifikuje linearno separabilne uzorke u prostoru obeležja pronalaženjem hiperravni koja ih najbolje razdvaja [12]. Takva hiperravan maksimizuje marginu – najmanje rastojanje od hiperravni do bilo kog uzorka. U slučajevima kada uzorci nisu linearno razdvojivi, primenjuje se nelinearno preslikavanje uzoraka u višedimenzionalni prostor i konstrukcija optimalne hiperravni razdvajanja u tom višedimenzionalnom prostoru. Pri klasifikaciji se u tom slučaju, radi lakšeg izračunavanja, koriste kernel funkcije. U ovom radu korišćen je SVM klasifikator sa linearnim, polinomijalnim i radijalnim kernelom. Najbolji rezultati dobijeni su sa radijalnim kernelom koji je dat izrazom:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \sum_{k=1}^d (x_{ik} - x_{jk})^2}, \quad (7)$$

gde su $\mathbf{x}_i$ i $\mathbf{x}_j$ uzorci u prostoru obeležja, d je dimenzionalnost prostora, a γ je slobodni parametar kernela koji je pri razvoju sistema u ovom radu postavljen na 2^{-8} .

6. REALIZACIJA SISTEMA

Sistem je realizovan u python okruženju. Pri ekstrakciji odziva mel filter banke i MFCCs korišćena je *librosa* biblioteka [13], dok je za realizaciju SVM klasifikatora korišćena *scikit-learn* biblioteka [14].

U cilju efikasnije obrade, polazni audio-snimci oglašavanja pilića, trajanja od po 10 minuta, su podeljeni na segmente dužine od po 1 minut.

Obuka i testiranje SVM klasifikatora vršeni su unakrsnom 5-fold validacijom, za svaku nedelju starosti posebno. U tabeli 3 prikazan je raspored trening i test jednogminutnih segmenata u svakoj nedelji starosti. Skraćenicom NS označeni su segmenti normalnog stanja, dok su segmenti stresnog stanja označeni skraćenicom SS.

Kako obuka tako i testiranje modela vršeno je na nivou jednog frejma od 50 ms. Na ovaj način je postignuto da se sa relativno malom bazom koja je na raspolaganju,

prilično kvalitetno procene parametri modela. Sličan pristup smo našli u radu [4] koji nam je predstavljao polaznu tačku.

Sa druge strane, odlučivanje se može vršiti i na segmentima dužim od 50 ms, tako što će klasifikator za svaki pojedinačni frejm od 50 ms doneti odluku o klasi, a onda će se većinskim glasanjem ili nekim drugim metodom doneti odluka o celom dužem segmentu.

Ovakvo odlučivanje je moguće, jer se očekuje da uznemirenost pilića traje duže od 50 ms, i da je potrebno alarmirati uzgajivača ukoliko uznemirenost traje barem minut.

Ovakvo odlučivanje nismo mogli da primenimo pošto smo u delu baze koji je odvojen za testove imali samo po 4 segmenta od po minut, što bi značajno smanjilo rezoluciju kojom određujemo tačnost sistema.

Tabela 3. Podela 1-minutnih segmenata na trening i test skup za svaku nedelju starosti u jednom foldu

Starost	Trening NS	Test NS	Trening SS	Test SS
Nedelja 1	16	4	8	2
Nedelja 2	16	4	20	40
Nedelja 3	16	4	20	190
Nedelja 4+	16	4	20	280

Zbog nebalansiranog test skupa, tačnost sistema nije računata kao broj ispravno klasifikovanih frejmova podeljen ukupnim brojem frejmova, već su analizirane osetljivost i specifičnost sistema.

Osetljivost se definiše kao:

$$OS = \frac{TP}{TP + FN} \cdot 100\%, \quad (8)$$

odnosno, kao odnos pravih pozitivna TP i zbira pravih pozitivna i lažnih negativna FN . U slučaju ovog sistema, klasa stresnog stanja je pozitivna klasa, a klasa normalnog stanja je negativna klasa. Specifičnost se računa kao:

$$SP = \frac{TN}{TN + FP} \cdot 100\%, \quad (9)$$

odnosno, kao odnos pravih negativna TN i zbira pravih negativna i lažnih pozitivna FP . Tačnost sistema izračunata je kao aritmetička sredina osetljivosti i specifičnosti, odnosno:

$$TA = \frac{OS + SP}{2}. \quad (10)$$

Isprobana je klasifikacija na osnovu svih prethodno nabrojanih obeležja, a takođe i na osnovu grupe od 7 obeležja koja su se u radu [4] pokazala kao najrelevantnija: odziv poslednjeg filtra u mel filter banci, oba formanta, RMS, visina zvuka, džiter i šimer.

7. REZULTATI

Osetljivost (OS), specifičnost (SP) i tačnost (TA) sistema za svaku od starosnih grupa prikazani su u tabeli 4. Ove vrednosti dobijene su kao prosek vrednosti osetljivosti, specifičnosti i tačnosti u svih 5 foldova.

Postignute performanse sistema su zadovoljavajuće, pogotovo kad se uzme u obzir činjenica da su računane na

nivou frejma. Za sistem koji je obučen za drugu nedelju starosti smo uradili analizu na nivou jednogminutnog segmenta i dobili smo i osetljivost i tačnost 100%, iako je prosečna tačnost svega 67.98%.

Kao što se iz priloženih rezultata u tabeli 4 može videti, tačnost u velikoj meri zavisi od starosti pilića. U slučaju sistema prilagođenog za prepoznavanje zvukova pilića starih 4 nedelje i više, sistem je postigao prilično loše rezultate. Jednominutni segmenti koji su pogrešno klasifikovani su poslani na ponovnu proveru veterinaru. Inicijalni rezultati dela poslatih snimaka ukazuju da je došlo do greške pri labeliranju baze audio-snimaka. Naime, inicijalni segmenti su bili trajanja 10 minuta, tako da se moglo desiti da se na početku i kraju pojedinih fajlova stresnog stanja pojave zvuci koji odgovaraju oglašavanju pilića u normalnom stanju. Normalizacija audio signala bi takođe mogla imati loš uticaj na tačnost sistema, jer određena vremenska obeležja značajno zavise od amplitude signala (npr. energija i snaga). Grupa od 7 odabranih obeležja ipak nije postigla dovoljno dobre rezultate da bi se sistem redukovao samo na ta obeležja. Kao nastavak istraživanja planirano je ispitivanje performansi obeležja van ove grupe, naročito mel frekvencijskih kepralnih koeficijenata, koji se u literaturi najčešće koriste.

Tabela 4. Osetljivost, specifičnost i tačnost sistema za svaku nedelju starosti na osnovu svih obeležja i na osnovu grupe od 7 odabranih obeležja

Starost	Obeležja	OS [%]	SP [%]	TA [%]
Nedelja 1	sva	76.82	74.63	75.72
	odabrana	72.23	64.15	68.19
Nedelja 2	sva	68.14	67.83	67.98
	odabrana	57.28	78.91	68.10
Nedelja 3	sva	86.91	90.78	88.85
	odabrana	82.49	83.49	82.99
Nedelja 4+	sva	55.98	68.79	62.38
	odabrana	51.08	72.98	62.03

8. ZAKLJUČAK

Sistem razvijen u ovom radu ima potencijal da postane koristan alat za monitoring pilića tovljenika i automatsku detekciju njihovog stresa. Naravno, za primenu sistema na živinarskoj farmi neophodna je korektno labelirana i znatno veća baza audio-snimaka, kao i uvođenje odlučivanja na nivou segmenata trajanja nekoliko desetina sekundi, što će biti predmet daljeg istraživanja.

9. ZAHVALNICA

Autori bi želeli da se zahvale kompaniji DunavNet iz Novog Sada koja im je ustupila zvučnu bazu sa snimcima tovnih pilića koji su korišćeni u ovoj studiji.

10. LITERATURA

- [1] D. Berckmans, "Automatic on-line monitoring of animals by precision livestock farming", *International Society for Animal Hygiene*, Saint Malo, 2004.
- [2] D. Berckmans *et al.*, "Animal Sound... Talks! Real-time Sound Analysis for Health Monitoring in Livestock", *International Symposium on Animal Environment and Welfare*, Chongqing, China, October 2015.

- [3] M. Rizwan *et al.*, "Identifying rale sounds in chickens using audio signals for early disease detection in poultry", *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Washington, DC, pp. 55-59, December 2016.
- [4] J. Lee *et al.*, "Stress Detection and Classification of Laying Hens by Sound Analysis", *Asian-Australasian Journal of Animal Sciences*, Vol. 28, No. 4, pp. 592-598, April 2015.
- [5] https://respeaker.io/4_mic_array/ (pristupljeno u septembru 2019.)
- [6] I. Fontana *et al.*, "Sound analysis to model weight of broiler chickens", *Poultry Science*, September 2017.
- [7] F. Eyben, "Real-time Speech and Music Classification by Large Audio Feature Space Extraction", *Springer Theses*, DOI 10.1007/978-3-319-27299-3.
- [8] M. Lin, S. Zhong, L. Lin, "Chicken Sound Recognition using Anti-noise Mel Frequency Cepstral Coefficients", *2015 Third International Conference on Robot, Vision and Signal Processing*, Kaohsiung, Taiwan, pp. 224-227, November 2015.
- [9] J. Cai *et al.*, "Sensor Network for the Monitoring of Ecosystem: Bird Species Recognition", *ISSNIP 2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Melbourne, Australia, pp. 293-298, December 2007.
- [10] F. Briggs, X. Fern, R. Raich, "Technical Report (Not Peer Reviewed): Acoustic Classification of Bird Species from Syllables: an Empirical Study", 2011.
- [11] V. Delić, PPT prezentacije i materijali sa predavanja za predmet Govorne tehnologije, Potral katedre za telekomunikacije i obradu signala (www.ktios.net), Fakultet tehničkih nauka, Univerzitet u Novom Sadu, 2018.
- [12] M. Sečujski, PPT prezentacije i materijali sa predavanja za predmet Prepoznavanje oblika, Potral katedre za telekomunikacije i obradu signala (www.ktios.net), Fakultet tehničkih nauka, Univerzitet u Novom Sadu, 2017.
- [13] B. McFee *et al.*, "librosa: Audio and Music Signal Analysis in Python", *14th Python in Science Conference (SciPy 2015)*, Austin, Texas, pp. 18-24, July 2015.
- [14] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830, October 2011.

Kratka biografija:



Nina Maljković rođena je u Sremskoj Mitrovici 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Obrada signala odbraniće 2019. god. kontakt: maljkovic.nina@gmail.com



Nikša Jakovljević rođen je u Sarajevu 1979 godine. Doktorirao je na Fakultetu tehničkih nauka 2014. Oblasti interesovanja su mu obrada signala i mašinsko učenje. kontakt: jakovnik@uns.ac.rs

RAZVOJ I ANALIZA ANDROID APLIKACIJE PET SHOP**DEVELOPMENT AND ANALYSIS OF ANDROID APPLICATION PET SHOP**

Smiljana Tedić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – U radu je opisana izrada android aplikacije za podršku rada prodavnica kućnih ljubimaca. Modeliranje je dato preko UML dijagrama, a aplikacija je implementirana korišćenjem Java programskog jezika i Android studio razvojnog okruženja.

Ključne reči: Java, Android

Abstract – The work describes the development of android application for development of pet shop store. Modeling is given via UML diagrams and application is implemented using Java programming language and Android studio programming environment.

Keywords: Java, Android

1. UVOD

Master rad se bavi izradom aplikacije za prodaju kućnih ljubimaca korišćenjem Android operativnog sistema. Rad je podeljen na par velikih celina. Rad se sastoji iz opisa postojećih rešenja za ovaj tip aplikacije, detaljnog opisa izabrane tehnologije, objašnjava se način funkcionisanja Android operativnog sistema, spominju se i ostale tehnologije koje su korišćene u projektu. Posebna - poglavlja posvećena su specifikaciji zahteva, specifikaciji dizajna sistema, implementaciji same aplikacije i prikazu rada aplikacije. Prikazani su delovi koda pomoću kojih se detaljnije objašnjava proces izrade aplikacije.

2. ELEKTRONSKA TRGOVINA

Online trgovina je postala najprofitabilniji oblik trgovine, čemu doprinosi jednostavnost i niski troškovi poslovanja u poređenju sa klasičnim vidom trgovine. Prodaja putem interneta znatno smanjuje troškove i poboljšava kontrolu prodaje, a i kupcu omogućava personalizovani i detaljni opis svake ponude kao i ocene drugih korisnika i kupaca. Online prodavnice pružaju mogućnost da se posao prenese na nova tržišta, a uz pomoć interakcije preko interneta i uz poštovanje unapred definisanih konvencija o privatnosti korisničkih podataka svaki prodavac je u mogućnosti da prikupi detaljne informacije o potrebama svake pojedinačne mušterije i automatski prilagodi svoje proizvode ili usluge tako da zadovolje te potrebe. Prodaja putem interneta znatno smanjuje troškove i poboljšava kontrolu prodaje, a i kupcu omogućava personalizovani pristup.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinać, vanr. prof.

3. ANDROID OPERATIVNI SISTEM

Jedan od najzastupljenijih operativnih sistema današnjice svakako je Android operativni sistem, sa brojem korisnika koji neprestano raste. Razvijen od strane Andorid Inc kompanije, a trenutno u vlasništvu korporacije Google. Zasniva se na Linux jezgri i predstavlja paket programske podrške za mobilne uređaje. Andorid razvija native aplikacije, aplikacije za specifične platforme koje se direktno instaliraju na uređaj, a imaju bolje performanse od klasičnih web aplikacija i u većini slučajeva nije im neophodna internet veza za rad. Android je operativni sistem otvorenog koda dostupan pod Apache licencom koja dopušta slobodnu izmenu i distribuciju programske podrške od strane proizvođača uređaja, telekomunikacionih operatera i programera.

3.1. Android aplikacije

Android aplikacije [2] su unificirane tj., izolovane, svaka se izvršava u jednom procesu u jednoj Java virtuelnoj mašini. Pišu se u Java programskom jeziku uz korišćenje Android SDK. Integrisano razvojno okruženje (IDE) koje se koristi za izradu aplikacija je Android Studio. Postoji posebna platforma preko koje se vrši distribucija android aplikacija, ta platforma se zove Google Play Store i dostupna je na svakom android mobilnom uređaju. Osnovne komponente svake android aplikacije su aktivnosti, servisi, dobavljači sadržaja i prijemnici poruka.

4. JAVA

Java je programski jezik visokog nivoa koji je razvila kompanija Sun Microsystems. Prvobitno se zvao OAK. Oak je bio neuspešan pa mu 1995. Sun menja ime u Java i modifikuje jezik da iskoristi razvoj web-a. Karakteristike Jave:

- Jednostavnost – razvijen je sa malim brojem jezičkih konstrukcija što omogućava programerima da jezik brzo nauče. Eliminise nekoliko jezičkih karakteristika dostupnih u C/C++ koje su povezane sa lošom programskom praksom ili retko korišćene: goto naredba, strukture, preklapanje operatora, višestruko nasleđivanje i pokazivače.
- Objektno-orijentisan – podržava konstrukciju programa koji se sastoji od kolekcije objekata koji sarađuju. Ovi objekti imaju jedinstven identitet, enkapsuliraju attribute i operacije i instance su klase povezane sa nasleđivanjem i polimorfizmom.

- Distribuiranost – dizajniran je da podrži različite nivoe mrežne povezanosti.
- Interpretovan – Java program se kompajlira u bajtkod koji je interpretiran od strane Java Runtime Environment (JRE).
- Snažan – dizajniran je da eliminiše određene tipove programskih grešaka. Java je tipiziran programski jezik što dozvoljava dodatno proveravanje grešaka u trenutku kompajliranja. Ne podržava memorijske pokazivače što eliminiše mogućnost da se podaci kompromituju. Ima automatski upravljač memorijom (*garbage collection*) koji eliminiše curenje memorije i druge probleme povezane sa dinamičkom memorijom.
- Siguran – dizajniran je da bude siguran u mrežnom okruženju. JRE koristi bajtkod proces provere da bi osigurao da kod učitao preko mreže ne narušava sigurnosna ograničenja Jave.

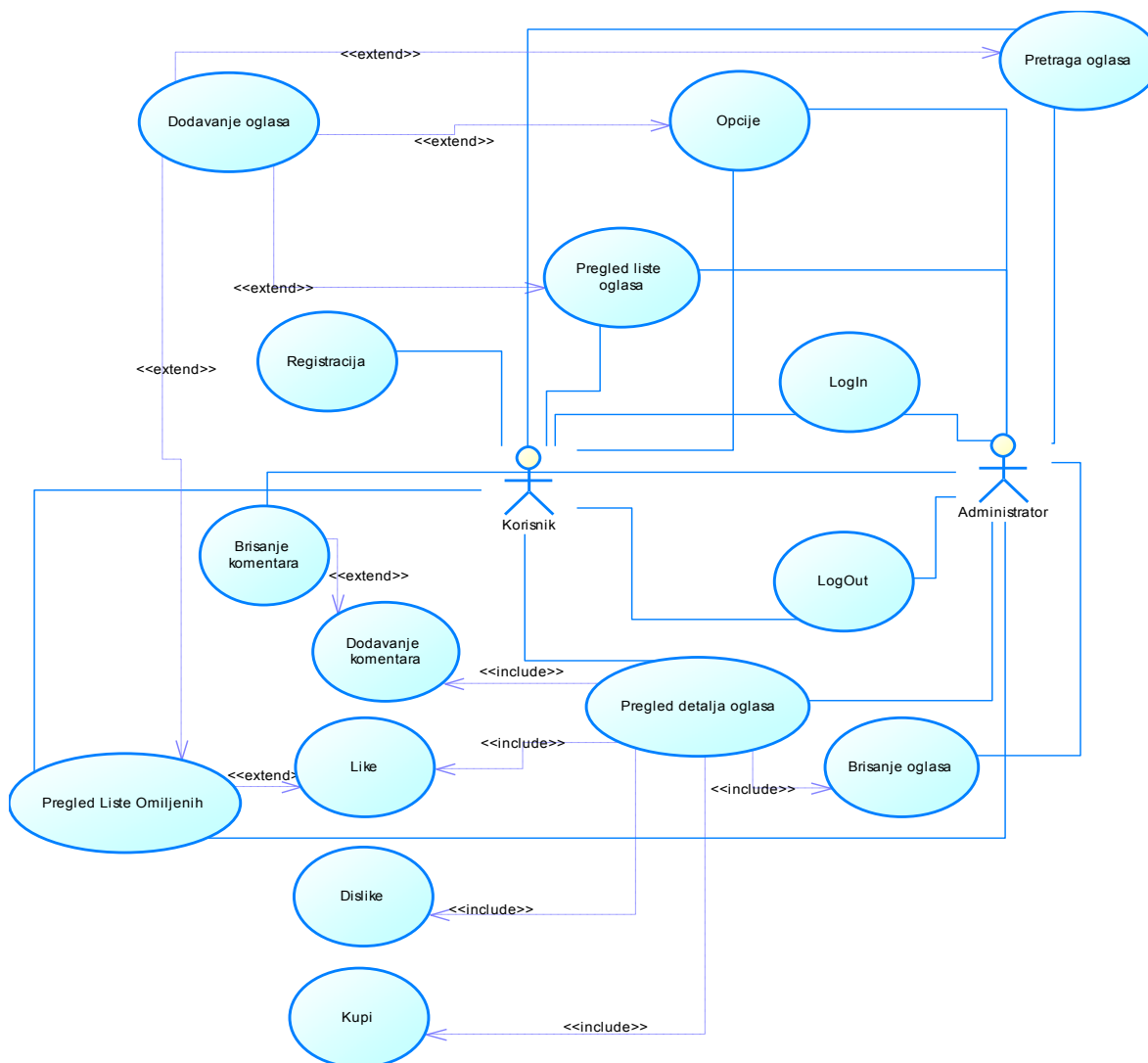
5. SPECIFIKACIJA ZAHTEVA I DIZAJNA

Funkcionalni zahtevi aplikacije specificirani su u obliku slučajeva korišćenja. Dijagramom slučajeva korišćenja

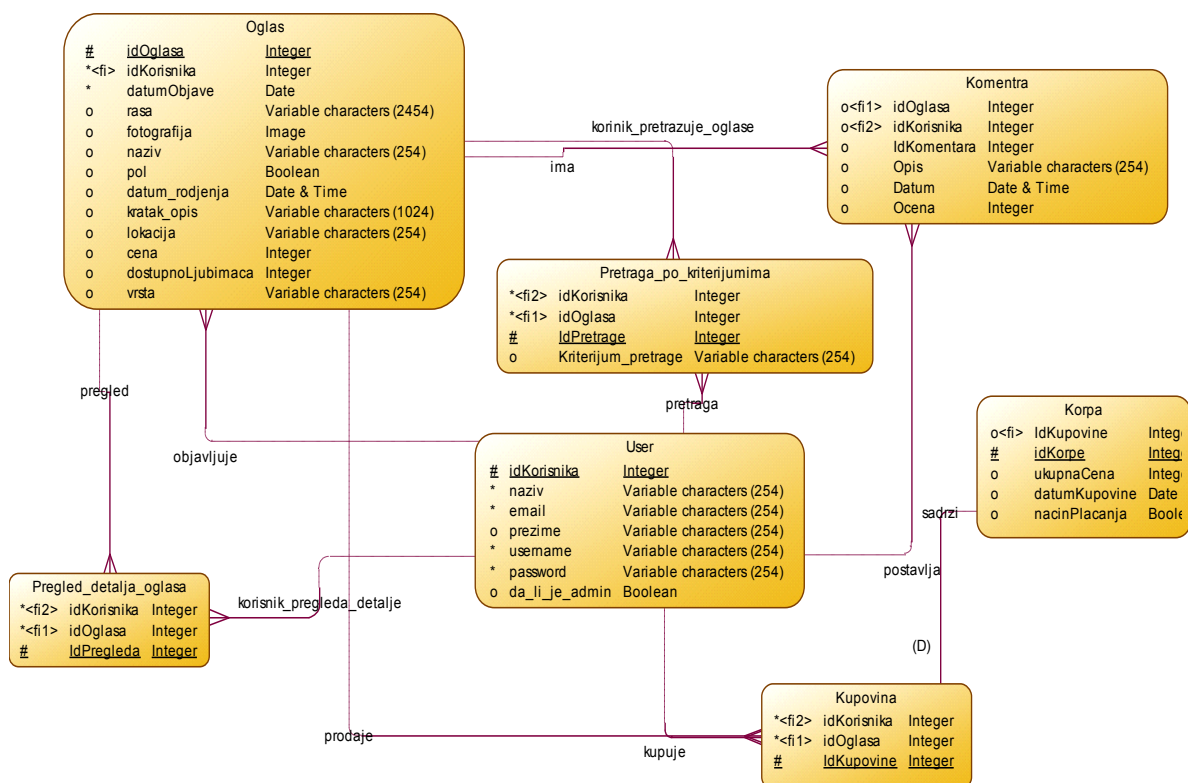
(slika 1.) predstavljene su sve funkcionalnosti koje softver treba da ponudi i podeljene su prema korisniku koji ih inicira [1].

Dijagrame prati tabelarni opis koji obuhvata spisak preduslova, koraka izvršenja, izuzetaka i garancije uspeha ili neuspeha u post uslovima. Za svaki funkcionalni zahtev definisan je prioritet i u skladu sa tim izrađeno je aplikativno rešenje. Prioritet jedan predstavlja esencijalni zahtev odnosno zahtev najvišeg prioriteta i samo za takve zahteve u radu je dat dijagramski i tekstualni opis.

Modeliranje podataka obuhvata određivanje osnovnih objekata sistema, identifikovanje atributa, identifikaciju ključeva za svaki objekat i povezivanje objekata (objekat od koga se uspostavlja veza zove se “roditelj” ili domen, a objekat ka kome je uspostavljena veza zove se “dete” ili ko-domen). Na taj način se kreira logički model podataka (slika 2.), sledeći korak je tehnička analiza koja obuhvata proveru da li su svi atributi dobro postavljeni. Završni korak jeste prevođenje logičkog modela podataka u fizički model podataka.



Slika 1. Use-case dijagram PetShop aplikacije



Slika 2. Logički model podataka sistema

6. IMPLEMETACIJA APLIKATIVNOG REŠENJA I PRIKAZ RADA APLIKACIJE

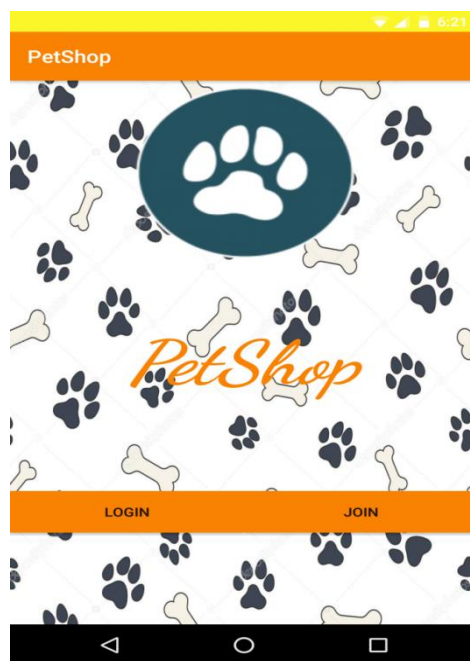
Programski kod aplikacije sastoji se iz koda napisanog u Java programskom jeziku koji je zadužen za glavne funkcionalnosti aplikacije i koda napisanog u XML formatu koji opisuje korisnički interfejs aplikacije. Manipulacija sa podacima je izvršena uz pomoć ugrađenog sistema za upravljanje bazom podataka koji se naziva SQLite [2].

U radu se nalazi primer koda za AdroidManifest.xml datoteku koja je obavezna u svakom android projektu. Priloženi su delovi koda klase koje prati prikaz korisničkog interfejsa u vidu fotografija ekrana mobilnog uređaja u kojima su odrađene sledeće funkcionalnosti:

- registracija i prijava korisnika na sistem (slika 3.),
- dodavanja novog oglasa,
- pregled liste svih oglasa,
- pregled detalja o svakom pojedinačnom oglasu (slika 4.) koji se nalazi u bazi,
- pregled omiljenih oglasa korisnika koji je prijavljen na sistem, ukoliko takvi oglasi postoje u bazi podataka za tog korisnika,
- pretraga svih oglasa koji se nalaze u bazi podataka po definisanim kriterijumima,
- prikazane su i funkcionalnosti kupovine tačno određenog oglasa,
- mogućnost postavljanja komentara,

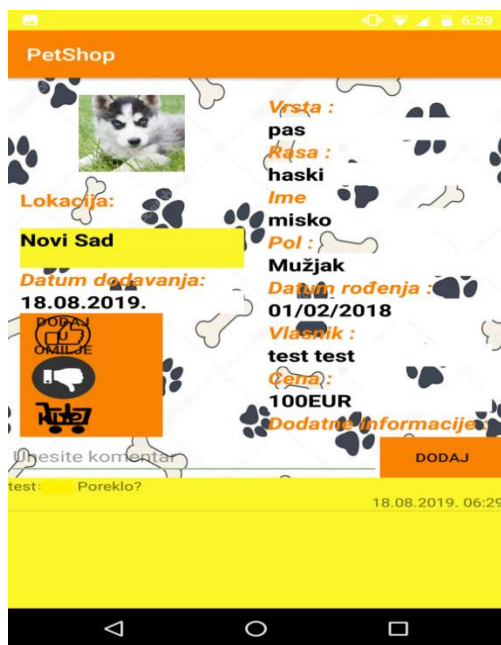
- pregled liste svih postojećih komentara u sistemu,
- kao i davanja like-a ili dislike-a oglasu.

Administrator sistema ima omogućenu funkcionalnost brisanja oglasa i komentara iz liste ponuđenih.



Slika 3. Početna stranica aplikacije

Pet Shop aplikacija koristi *LocationManager* klasu koja pruža pristup uslugama lokacije sistema preko koje se određuje lokacija mobilnog uređaja u kojem je instalirana aplikacija, ukoliko za to aplikacija ima potrebna prava pristupa. Prikazan je i rad sa Google Maps Api-em koji omogućava korišćenje google maps servisa na Android platformi.



Slika 4. Pregled detalja oglasa

7. ZAKLJUČAK

Ovaj master rad prikazuje razvoj aplikacije uz pomoć Android frameworka. Razvoj aplikacije počeo je od nule preko odabira tehnologije koja će se koristiti, izrade dijagrama slučajeva korišćenja, modelovanja baze podataka i implementacije do završetka aplikacije.

Android Studio se pokazao kao veoma produktivno i korisno razvojno okruženje, što i dokazuje njegova popularnost među programerima koji ga koriste. U pitanju je veoma dobar način za razvoj aplikacija koji koriste brojne organizacije i institucije za kreiranje snažnih i održivih aplikacija.

Android operativni sistem se pokazao kao dobar izbor tehnologije ukoliko se žele razvijati male i srednje aplikacije koje se ne planiraju dalje razvijati i unapređivati i ukoliko se želi brza izrada aplikacija, ali je namenjen i za razvoj kompleksnih aplikacija, zatim onih koje treba da budu održavane i proširive, ili koje zahtevaju sigurnost i mogu se sporije razvijati. Android operativni sistem pokreće čak 70% mobilnih uređaja što je uticalo na to da aplikacije kreirane za ovu platformu postanu vodeće na tržištu. Razlog popularnosti ove platforme leži u tome što omogućava kvalitetnu interakciju sa korisnikom, fleksibilna je i modularna, a moguće je koristiti i bez pristupa internetu.

8. LITERATURA

- [1] UVOD U OBJEDINJENI JEZIK MODELIRANJA, Novi Sad, 1999. Ivana Stanojević, Dušan Surla
- [2] Predavanja iz predmeta Programiranje mobilnih aplikacija, Novi Sad, 2018/2019. god, dr Stevan Gostojić

Kratka biografija:



Smiljana Tedić rođena je 23.3.1995. godine u Novom Sadu. Diplomirala na Fakultetu tehničkih nauka u Novom Sadu 19. 9. 2018. godine i stekla zvanje Diplomirani inženjer elektrotehnike i računarstva. Master akademske studije upisala je na Fakultetu tehničkih nauka u Novom Sadu 2018. godine.

**REGULACIJA TEMPERATURE KGH SISTEMA UPOTREBOM PI-LIKE FUZZY
REGULATORA I GA****TEMPERATURE REGULATION OF HVAC SYSTEM USING PI-LIKE FUZZY
CONTROLLER AND GA**

Saša Maksimović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Osnovni zadatak ovog rada jeste regulacija temperature KGH sistema, a da se pri tom ne naruše zahtevani radni uslovi. Korišćen je PI-like fuzzy regulator, koji upravlja ventilom za dovod hladne vode na izmenjivač toplote KGH sistema. Izvršena je optimizacija fuzzy tabele pravila pomoću genetskih algoritama.

Ključne reči: PI-like fuzzy regulator, GA, KGH sistemi

Abstract – The main goal of this paper is the temperature regulation of HVAC system, without influence on the required operating conditions. PI-like fuzzy controller was used, that controls valve for the flow rate of cold water through the heat exchanger of HVAC system. Optimization of fuzzy rules table using genetic algorithms was performed.

Keywords: PI-like fuzzy Controller, GA, HVAC system

1. UVOD

Sa značajnim porastom potrošnje energije u zgradama, strategije štednje energije postale su prioritet u mnogim državama. Kategorije zgrada i sistemi klimatizacije, grejanja i hlađenja – KGH sistemi su ključni elementi koji utiču na potrošnju energije u zgradama, skoro 50%. Stoga je razvoj i implementacija efikasnih kontrolnih tehnika KGH sistema od primarnog značaja. Tačnije, sa smanjenom cenom obrade, skladištenja podataka i generalno komunikacija proteklih godina, dizajn i implementacija kompleksnijih upravljačkih tehnika postali su izvodljivi [1].

2. ENERGETSKA EFIKASNOST

Pod pojmom unapređenja energetske efikasnosti u zgradarstvu podrazumeva se kontinuirani i širok opseg delatnosti kojima je krajnji cilj smanjenje potrošnje svih vrsta energije uz iste ili bolje uslove u objektu.

Pojam energetska efikasnost najčešće se susreće u dva moguća značenja, od kojih se jedno odnosi na uređaje a drugo na mere i ponašanja. Pod energetska efikasnim uređajem smatra se onaj koji ima veliki stepen korisnog dejstva. Energetska efikasnost u zgradarstvu podrazumeva efikasno korišćenje energije uz primenu optimalnih mera čiji je cilj: smanjenje potrošnje energije uz finansijsku

NAPOMENA:

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

uštedu za krajnjeg korisnika, ugodniji i kvalitetniji boravak u zgradi, smanjenje troškova održavanja i produženje životnog veka zgrada, doprinos zaštiti okoline i smanjenju emisija štetnih gasova kao i globalnim klimatskim promenama [2].

3. KGH SISTEMI

Sistemi klimatizacije, grejanja i hlađenja (KGH) su osmišljeni da zadovolje zahteve udobnosti ljudi i procesa. Glavna misija KGH sistema je da zadovolji toplotnu udobnost ljudi podešavanjem i promenom spoljašnjih vazдушnih uslova u željene uslove nastanjene zgrade.

Osnovne komponente KGH sistema su: mešaona komora i kontrola spoljašnjeg vazduha, vazdušni filter, ventilator dovodnog vazduha, izduvni ili ventilatori oduška i izlazna grana vazduha, spoljašnji dovod vazduha, kanali za distribuciju vazduha, terminalni uređaji, sistem povratnog vazduha, izmenjivači toplote, samostalna jedinica za grejanje ili hlađenje, rashladni toranj, parni kotao, kontrola, čiler i oprema za vlaženje i odvlaživanje.

Osnovna podela KGH sistema je na centralizovane sisteme i decentralizovane, odnosno lokalne. Tip sistema zavisi od postavljanja lokacije primarne opreme. Koja može biti na jednom mestu, centralizovana, koja će upravljati celom zgradom kao jedinica. Ili decentralizovana, na više lokacija, koja će odvojeno upravljati po zonama zgrade.

Centralizovani KGH sistem može upravljati jednom ili više termalnih zona. Njegova glavna oprema se nalazi izvan zona, na odgovarajućoj centralnoj lokaciji, bilo unutar, na krovu ili pored zgrade. Unutar svake zone potrebne su kontrolne jedinice kao što su termostati [3].

4. FUZZY LOGIKA

Fuzzy logika u užem smislu predstavlja viševrednosnu logiku. U klasičnoj, bivalentnoj logici svaki iskaz može biti samo tačan ili netačan. U ljudskom govoru postoji puno iskaza koji nemaju vrednost strogo tačno ili strogo netačno. Da bi se opisali ovakvi iskazi proširena je klasična logika. Tako je nastala fuzzy logika.

U širem smislu fuzzy logika je zapravo teorija fuzzy skupova. Teorija fuzzy skupova je teorija koja proučava objekte kod kojih nije strogo definisana granica pripadnosti nekoj klasi, odnosno objekat pripada određenoj klasi sa nekim stepenom [4].

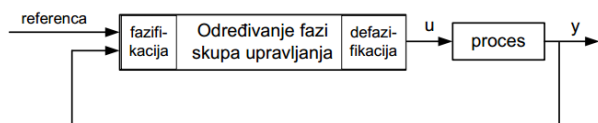
Osnovni element za predstavljanje i obradu nepreciznosti u fuzzy tehnologijama je fuzzy skup. Fuzzy skup predstavlja skup elemenata sa sličnim svojstvima. Dok, diskretan (klasičan) skup predstavlja skup elemenata sa istim svojstvima. U diskretnom skupu svaki element pripada tom skupu sa stepenom pripadnosti jedan. Dok, u fuzzy skupu svaki element pripada tom skupu u izvesnom stepenu [5].

4.1. Struktura fuzzy regulatora

Fuzzy regulator se formira kao skup pravila kojima se izražava veza između registrovanih vrednosti na izlazu sistema i upravljanja koje treba da promeni te izlaze, odnosno da ih dovede do željenog ishoda.

Jedina raspoloživa informacija o procesu je realni fizički signal koji je dobijen merenjem. Njegovo korišćenje od strane fuzzy regulatora zahteva da se izvrši fazifikacija tog signala. Konačno i svi trenutno raspoloživi izvršni organi konstruisani su tako da očekuju konkretni fizički signal na svom ulazu, što znači da je neophodno da se na osnovu dobijenog izlaznog fuzzy skupa odredi veličina upravljačkog signala kojim će se pobuditi izvršni organ, odnosno da se izvrši njegova defazifikacija.

Algoritam fuzzy regulatora radi na osnovu specificiranih pravila. Za svaki primljeni izmereni signal algoritam realizuje sledeća tri koraka: fazifikacija ulaznih signala, određivanje fuzzy skupa upravljanja i defazifikacija. Struktura fuzzy regulatora prikazana je na slici 2 [6].



Slika 2. Struktura fuzzy regulatora.

5. GENETSKI ALGORITMI (GA)

GA spadaju u grupu stohastičkih optimizacionih metoda koje ne koriste derivacije – izvode funkcija. Bazirani su na konceptu prirodne selekcije i evolucije.

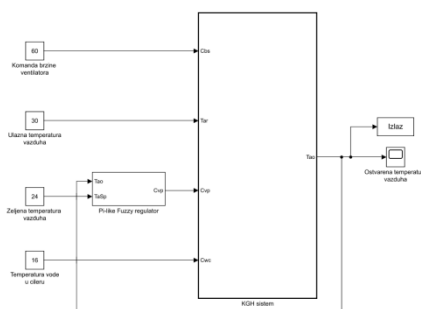
Karakteristike genetskih algoritama: Predstavljaju proceduru paralelnog pretraživanja pogodnu za primenu u sistemima paralelnog procesiranja. Podjednako su primenljivi u rešavanju kontinualnih i diskretnih optimizacionih problema. Predstavljaju stohastičke metode, manje podložne zapadanju u lokalni minimum. Poseduju veoma velik stepen prilagodljivosti, čime se olakšava njihova primena u okviru problema strukturne i parametarske identifikacije složenih modela kao što su neuronske mreže i/ili fuzzy sistemi zaključivanja. Svaku tačku parametarskog ili prostora rešenja predstavljaju preko binarnog niza koji se zove hromozom. Svakom hromozomu je pridružen određen stepen prilagođenosti, tako da se određivanje vrednosti kriterijumske funkcije svodi na procenu njene vrednosti u jednoj tački. Umesto jedne izolovane tačke GA često koriste set tačaka koji se naziva populacija, i koji evoluira u pravcu što boljeg zadovoljenja kriterijumske funkcije. U svakoj generaciji GA primenom genetskih operatora proizvodi novu populaciju. Genetski operatori su selekcija, ukrštanje i mutacija. Članovi populacije koji u većoj meri zadovoljavaju kriterijumsku funkciju imaju veću šansu da prežive i da učestvuju u budućim ukrštanjima. Nakon

većeg broja generacija, populacija sadrži članove sa većim stepenom prilagođenja. Često se navode kao primer optimizacije bazirane na populaciji, gde se popravljaju performanse cele populacije, a ne samo određenih jedinki.

Osnovni elementi genetskih algoritama su: kodiranje, prilagođenost (stepen pogodnosti), selekcija (odabir roditelja), ukrštanje, mutacija i elitizam [5].

6. SOFTVERSKO REŠENJE: MODEL VAZDUH – VODA KGH SISTEMA

U ovom radu korišćen je programski paket *Matlab*. Zbog nedostupnosti realnog sistema kreiran je simulacioni model pomoću alata *Simulink*. Pomoću alata *FuzzyLogic Toolbox* kreirani su *PI-like fuzzy* regulatori. Optimizacija tabele pravila izvršena je pomoću alata *Optimization Tool*. Na slici 3 prikazan je simulacioni model vazduh – voda KGH sistema u kom se vrši regulacija temperature vazduha upotrebom *PI-like fuzzy* regulatora [7].



Slika 3. Simulacioni model vazduh – voda KGH sistema.

6.1. Projektovanje fuzzy regulatora

Za upravljanje regulacionim ventilom dolazne hladne vode projektovana su dva tipa regulatora, mamdani i sugeno. Napravljeno je nekoliko eksperimenata sa različitim tabelama pravila (3x3 i 5x5), koje su prikazane u tabelama 1 i 2, respektivno. Cilj je bio poređenje performansi ova dva tipa. Na kraju, urađen je još jedan eksperiment u kom je izvršena optimizacija tabele pravila fuzzy regulatora tipa mamdani 3x3 pomoću GA.

Za ulazne (e – greška i Δe – izvod greške) i izlazne promenljive (upravljanje) korišćene su funkcije pripadnosti: N – negativno, Z – nula, P – pozitivno, NB – negativno veliki, NS – negativno mali, PS – pozitivno mali i PB – pozitivno veliki.

Tabela 1. Tabela pravila 3x3.

Δe e	N	Z	P
N	NB	NS	Z
Z	NS	Z	PS
P	Z	PS	PB

Tabela 2. Tabela pravila 5x5.

e Δe	NB	NS	Z	PS	PB
NB	NB	NB	NS	NS	Z
NS	NB	NS	NS	Z	PS
Z	NS	NS	Z	PS	PS
PS	NS	Z	PS	PS	PB
PB	Z	PS	PS	PB	PB

7. REZULTATI I DISKUSIJA

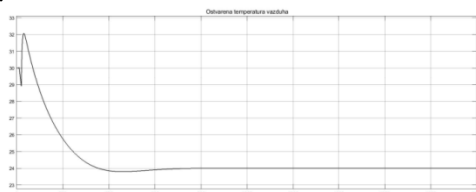
Da bi simulacija bila moguća ulazni signali KGH sistema su predefinisani i uneseni kao konstantne vrednosti. U svim eksperimentima korišćeni su identični ulazni signali, da bi se mogle porediti performanse regulatora. Ulazni signali prikazani su u tabeli 3.

Tabela 3. Vrednosti ulaznih signala KGH sistema.

Komanda brzine ventilatora [%]	60
Ulazna temperatura vazduha [°C]	30
Željena temperatura vazduha [°C]	24
Temperatura vode u čileru [°C]	16

7.1. Mamdani 3x3

U ovom eksperimentu korišćen je mamdani 3x3 fuzzy regulator. Izlazni signal iz KGH sistema prikazan je na slici 4.



Slika 4. Izlazni signal KGH sistema, mamdani 3x3.

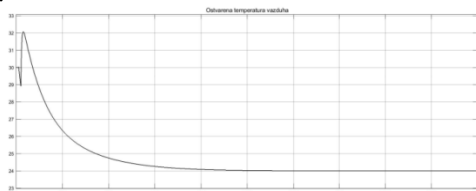
Performanse odziva prikazane su u tabeli 4.

Tabela 4. Performanse odziva KGH sistema, mamdani 3x3.

Preskok [%]	33,6499
Vreme uspona [s]	330,0442
Vreme smirenja [s]	679,2774
Ess – Greška u ustaljenom stanju	1,8495e-05

7.2. Sugeno 3x3

U ovom eksperimentu korišćen je sugeno 3x3 fuzzy regulator. Izlazni signal iz KGH sistema prikazan je na slici 5.



Slika 5. Izlazni signal KGH sistema, sugeno 3x3.

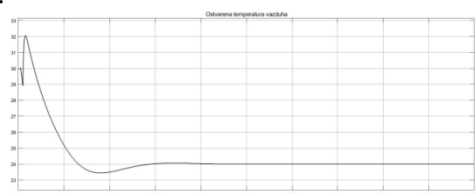
Performanse odziva prikazane su u tabeli 5.

Tabela 5. Performanse odziva KGH sistema, sugeno 3x3.

Preskok [%]	33,6591
Vreme uspona [s]	528,3194
Vreme smirenja [s]	865,4929
Ess – Greška u ustaljenom stanju	1,1320e-04

7.3. Mamdani 5x5

U ovom eksperimentu korišćen je mamdani 5x5 fuzzy regulator. Izlazni signal iz KGH sistema prikazan je na slici 6.



Slika 6. Izlazni signal KGH sistema, mamdani 5x5.

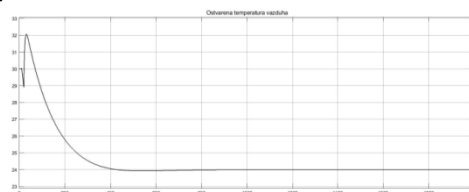
Performanse odziva prikazane su u tabeli 6.

Tabela 6. Performanse odziva KGH sistema, mamdani 5x5.

Preskok [%]	33,6407
Vreme uspona [s]	270,9196
Vreme smirenja [s]	649,8043
Ess – Greška u ustaljenom stanju	1,0616e-05

7.4. Sugeno 5x5

U ovom eksperimentu korišćen je sugeno 5x5 fuzzy regulator. Izlazni signal iz KGH sistema prikazan je na slici 7.



Slika 7. Izlazni signal KGH sistema, sugeno 5x5.

Performanse odziva prikazane su u tabeli 7.

Tabela 7. Performanse odziva KGH sistema, sugeno 5x5.

Preskok [%]	33,6501
Vreme uspona [s]	349,6044
Vreme smirenja [s]	481,3783
Ess – Greška u ustaljenom stanju	1,8336e-05

7.5. Poređenje performansi odziva KGH sistema

Na osnovu dobijenih rezultata, može se zaključiti da su regulatori 5x5, stabilniji od regulatora 3x3. Imaju kraće vreme uspona i vreme smirenja. Takođe, imaju manju grešku u ustaljenom stanju, dok im je preskok isti.

Pri poređenju regulatora po tipu, može se zaključiti da je mamdani 3x3 stabilniji od sugeno 3x3. Ima kraće vreme uspona i vreme smirenja. Takođe, ima manju grešku u ustaljenom stanju, dok im je preskok isti. Ali, kod tipa regulatora 5x5, sugeno ima značajno manje vreme smirenja, dok su mu vreme uspona i greška u ustaljenom stanju neznatno veće. Može se zaključiti da je sugeno 5x5 ipak stabilniji od mamdani 5x5.

7.5. Optimizacija tabele pravila fuzzy regulatora pomoću GA

Za optimizaciju tabele pravila pomoću GA odabran je mamdani 3x3. Urađena su dva eksperimenta, u cilju poređenja performansi. Razlikuju se samo u kriterijumu zaustavljanja. U prvom eksperimentu podešen je kriterijum zaustavljanja na deset generacija. Dok je u drugom eksperimentu podešen na dvadeset generacija.

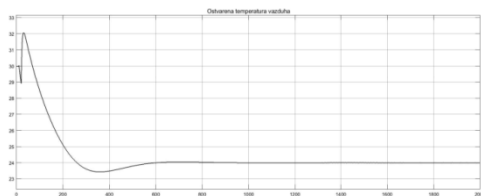
7.5.1. Mamdani 3x3 GA10

Dobijena tabela pravila mamdani 3x3 fuzzy regulatora pomoću GA nakon deset generacija prikazana je u tabeli 8.

Tabela 8. Tabela pravila mamdani 3x3 – GA10.

Δe_e	N	Z	P
N	NS	NB	NS
Z	PB	Z	NB
P	Z	PB	NS

Izlazni signal iz KGH sistema prikazan je na slici 8.



Slika 8. Izlazni signal KGH sistema, mamdani 3x3 – GA10.

Performanse odziva prikazane su u tabeli 9.

Tabela 9. Performanse odziva KGH sistema, mamdani 3x3 – GA10.

Preskok [%]	33,6350
Vreme uspona [s]	265,5720
Vreme smirenja [s]	661,2735
Ess – Greška u ustaljenom stanju	1,1209e-04

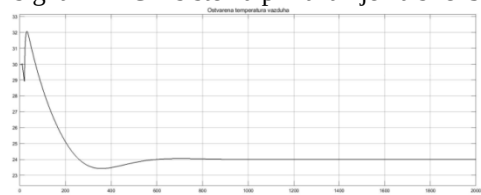
7.5.2. Mamdani 3x3 GA20

Dobijena tabela pravila mamdani 3x3 fuzzy regulatora pomoću GA nakon dvadeset generacija prikazana je u tabeli 10.

Tabela 10. Tabela pravila mamdani 3x3 – GA20.

Δe_e	N	Z	P
N	NS	NB	Z
Z	Z	Z	NS
P	PS	PB	NB

Izlazni signal iz KGH sistema prikazan je na slici 9.



Slika 9. Izlazni signal KGH sistema, mamdani 3x3 – GA20.

Performanse odziva prikazane su u tabeli 11.

Tabela 11. Performanse odziva KGH sistema, mamdani 3x3 – GA20.

Preskok [%]	33,6383
Vreme uspona [s]	262,0181
Vreme smirenja [s]	657,4461
Ess – Greška u ustaljenom stanju	2,3334e-04

7.5.2. Poređenje performansi odziva KGH sistema nakon optimizacije tabele pravila

Na osnovu dobijenih rezultata iz prvog eksperimenta, može se zaključiti da je regulator nakon deset generacija stabilniji od onog pre optimizacije. Vreme uspona i vreme smirenja mu je manje. Greška u ustaljenom stanju je veća. Dok im je preskok isti.

Na osnovu dobijenih rezultata iz drugog eksperimenta, može se zaključiti da je regulator nakon dvadeset generacija još stabilniji u odnosu na regulator nakon deset generacija. Vreme uspona i vreme smirenja mu je manje. Greška u ustaljenom stanju je veća, dok im je preskok isti.

8. ZAKLJUČAK

Regulacija temperature zasnovana na *PI-like fuzzy* regulatorima i GA razvijena je da bi se optimizovala regulacija temperature KGH sistema. Projektovano je više različitih regulatora kako bi se uporedile performanse odziva i izabrao najefikasniji za upravljanje simulacionim modelom KGH sistema.

Optimizacijom tabele pravila *PI-like fuzzy* regulatora pomoću GA su dobijene još bolje performanse odziva KGH sistema.

Na osnovu dobijenih rezultata može se doći do zaključka da postoji mogućnost uštede električne energije, jer se upotrebom *PI-like fuzzy* regulatora i GA povećava energetska efikasnost sistema. Može se istaći da je ovakav način upravljanja KGH sistemima izuzetno pogodan i finansijski isplativ i da ima smisla razmišljati o upotrebi ovakvih regulatora kada se želi ostvariti optimalno upravljanje KGH sistemima.

9. LITERATURA

- [1] Afram, F. Janabi-Sharifi: „Theory and applications of HVAC control systems - A review of model predictive control (MPC)“, Building and Environment 72 (2014), pages 343-355
- [2] Z. Živković, M. Vasić, A. Kuzmanović: „Predlog mera za finansiranje energetske efikasnosti u zgradarstvu u Srbiji“, Građevinska knjiga, Beograd, 2011
- [3] Shaimaa Seyam (November 5th 2018). Types of HVAC Systems, HVAC System, Mohsen Sheikholeslami Kandelousi, IntechOpen
- [4] I. Micić: „Fazi sistemi“, Prirodno-matematički fakultet, Univerzitet u Nišu, 2015
- [5] F. Kulić: „Inteligentni upravljački sistemi“, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, 2014
- [6] S. Turajlić: „Osnovni principi Fazi upravljanja procesima“, Elektrotehnički fakultet, Univerzitet u Beogradu, 2014
- [7] M. Anderson, M. Buehner, P. Young, D. Hittle, C. Anderson, J. Tu, D. Hodgson: „An experimental system for advanced heating, ventilating and air conditioning (HVAC) control“, Energy and Buildings 39 (2007) 136–147

Kratka biografija:



Saša Maksimović rođen je 30.06.1989. godine u Šapcu. 2008. godine upisuje Fakultet tehničkih nauka, univerziteta u Novom Sadu, oblast: Elektrotehnika i računarstvo, studijski program: Računarstvo i automatika, usmerenje: Automatika i upravljanje sistemima. Osnovne akademske studije završava 2013. godine i nastavlja sa obrazovanjem upisom na master akademske studije istog usmerenja.

SMART CITY – NOVI SAD MODEL PRAĆENJA JAVNOG SAOBRAĆAJA**SMART CITY – NOVI SAD MODEL FOR TRACKING THE PUBLIC TRANSPORT**Ivan Jančić, Srđan Popov, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je predložen model implementacije sistema koji krajnjim korisnicima omogućava praćenje GPS lokacije autobusa u realnom vremenu na području Novog Sada.

Ključne reči: Internet stvari, Pametni grad, Novi Sad, sistem za praćenje GPS lokacije

Abstract – The paper presents the model for implementation of GPS tracking system that would enable users to track the real time location of buses in the city of Novi Sad.

Keywords: Internet of Things, Smart City, Novi Sad, GPS tracking system

1. UVOD

Internet stvari (engl. Internet of Things, skraćeno IoT) predstavlja trend koji nam zahvaljujući eksponencijalnom razvoju novih tehnologija iz dana u dan poboljšava i olakšava svakodnevni život kako u socijalnom, tako i u tehničkom i ekonomskom aspektu. Uzimajući u obzir da se broj uređaja koji su povezani na internet neprestano povećava kao i da je izrada elektronskih delova nikad jeftinija primena IoT tehnologija i koncepata je praktično neograničena. Jedna od mnogih primena IoT koja ima veliki uticaj na život čoveka jesu pametni gradovi (Smart cities). Ako uzmemo u obzir da ljudska populacija konstanto raste i da će do 2050. godine 66% svetskog stanovništva živeti u gradovima dolazimo do zaključka da su pametni gradovi jedan od koncepata koji će najviše uticati na kvalitet života čoveka [1].

U ovom radu će biti predstavljena studija slučaja o tome kako bi se moglo implementirati praćenje GPS lokacije autobusa kao sredstva javnog prevoza grada Novog Sada. Razmatra se implementacija sistema koji krajnjim korisnicima omogućava praćenje GPS lokacije autobusa, analiziraju se troškovi i predlažu se moguća unapređenja sistema u budućnosti.

2. INTERNET STVARI**2.1. Definicija**

Internet stvari (engl. Internet of Things, skraćeno IoT) predstavlja međumrežavanje fizičkih objekata, vozila (što se odnosi i na „povezane uređaje” i „pametne uređaje”), zgrada i drugih stvari sa ugrađenom elektronikom,

softverom, senzorima i konektivnošću koji omogućavaju objektima da razmenjuju podatke sa proizvođačem, operaterom i/ili drugim povezanim uređajima.

Godine 2013. Global Standards Initiative on Internet of Things (IoT-GSI) definisala je IoT kao „globalnu infrastrukturu informatičkog društva koja omogućava napredne usluge (fizičkim i virtualnim) umrežavanjem stvari, pritom se zasnivajući na postojećim i interoperabilnim informacionim i komunikacionim tehnologijama u razvoju”.

U tu svrhu, termin „stvar” predstavlja „predmet fizičkog sveta (fizičkih stvari) informacija ili reč (virtualne stvari), koji je moguće identifikovati i koji može da bude integrisan u komunikacionim mrežama” [2].

2.2. Arhitektura

Arhitektura se sastoji iz nekoliko različitih slojeva tehnologija koje su međusobno povezane i sarađuju u cilju obezbeđivanja skalabilnosti, modularnosti i konfiguracije uređaja u različitim scenarijima korišćenja. Slojeve tehnologija možemo, na visokom nivou, svrstati u sledeće:

1. **Nivo uređaja (senzori i aktuatori)** – Senzori se koriste za konverziju prikupljenih informacija iz spoljašnjeg sveta u podatke pogodne za analizu i obradu. Aktuatori imaju mogućnost da utiču na sredinu u zavisnosti od različitih faktora (održavanje temperature u sobi, gašenje svetla...).
2. **Mrežni nivo** – Ovaj nivo je zadužen za adresiranje i rutiranje paketa podataka. Paketi se putem Interneta prenose na specificiranu destinaciju - IP adresu. IPv4 je do skoro bio standardni protokol koji koristi 32 bita za adresu što dovodi do ukupnog broja od 2^{32} adresa. Međutim usled limitiranog adresnog prostora i nemogućnosti da se nosi sa skalabilnosti IoT-a, novi IPv6 protokol je zaživeo. Glavni novitet je 128 bita za adresu što dovodi do ukupnog broja od 2^{128} adresa, a to omogućava adresiranje milijardi novih IoT uređaja.
3. **Nivo procesiranja informacija** – Skladištenje, procesuiranje i analiza velike količine podataka. Koriste se različiti alati za analitiku kao i tehnike mašinskog učenja i veštačke inteligencije.
4. **Aplikativni nivo** – Predstavlja interfejs između uređaja i Interneta. Pruža servise i definiše protokle za razmenu poruka na aplikativnom nivou. Implementiran je u vidu aplikacije na korisničkom uređaju.

NAPOMENA:

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

2.3. Izazovi

1. **Interoperabilnost** - Predstavlja mogućnost sistema ili komponenti sistema da komuniciraju međusobno nezavisno od njihovog proizvođača ili tehničke specifikacije. Uzimajući u obzir da je IoT tržište još uvek u ranoj fazi razvoja postoji dosta podeljenosti između brendova što dovodi do neuniformnosti u standardima za komunikaciju, formatima podataka, protokolima itd. S obzirom da je filozofija interneta kreiranje sveta u kome milioni uređaja međusobno komuniciraju na najbolji mogući način ovo je jedan od najbitnijih izazova sa kojima se IoT susreće. Ovaj problem se rešava dogovorom o uvođenju jedinstvenih standarda i protokola kojih će se proizvođači pridržavati.
2. **Bezbednost** - Podaci koji se prenose između uređaja u IoT mreži mogu biti jako poverljivi (finansije, medicinski podaci...) što dovodi do potrebe za visokim nivoom bezbednosti. S obzirom na to da uređaji nisu previše kompleksni, oni često nemaju dovoljno kompjuterskih resursa za implementaciju složenijih sistema za bezbednost i enkripciju. Proizvođači na sigurnost mogu uticati na sledeće načine:
 - Redovno podsećanje korisnika o važnosti update-ovanja njihovih uređaja i softvera koji koriste.
 - Insistiranjem na kompleksnom sistemu za korisničke šifre.
 - Ukidanjem mogućnosti udaljenog pristupa uređaju ukoliko nije neophodno za osnovne funkcije.
 - Striktna kontrola pristupa za različite API-je.
 - Zaštita cloud centara u kojima se čuvaju podaci od DDoS i drugih hakerskih napada.
3. **Skalabilnost** - Predstavlja mogućnost uređaja da se prilagodi povećanoj količini posla do koje će neminovno doći ukoliko je proizvod uspešan na tržištu. Kako se proizvod razvija i raste, tako i platforma na kojoj se on zasniva mora biti u mogućnosti da se skalira bez kompromitovanja integriteta, sigurnosti, pouzdanosti i funkcionalnosti proizvoda. Organizacija se mora suočiti sa izazovom upravljanja, monitoringa i obezbeđivanja skladišta podataka uzimajući u obzir da će skup podataka koje uređaj skuplja, a koji se kasnije analiziraju, (dodavanje više servera i kapaciteta) umesto da se unapređuje trenutna infrastruktura.
4. **Pouzdanost** - Uzimajući u obzir primene u oblastima kao što su zdravstvo, transport, logistika gde ne sme dolaziti do prekida rada čak i kada dolazi do neočekivanih grešaka, pouzdanost i stabilnost igraju veliku ulogu u razvoju IoT uređaja. Proizvođači moraju implementirati različita rešenja za oporavak od nepredviđenih situacija, dok uređaji moraju imati visoku toleranciju na greške kako bi celokupna platforma bila robusna.

3. PAMETNI GRADOVI

Pametni gradovi su zaživeli kao koncept u kome se IoT rešenja primenjuju na nivou grada i služe da olakšaju, i u

krajnjem slučaju što je više moguće poboljšaju životni standard svojih građana, ali i da modernizuju infrastrukturu i servise (javni prevoz, zdravstvo...) koji postoje u gradu. Senzori raspoređeni širom grada prate različite parametre i skupljaju podatke koji se mogu procesirati, analizirati i kasnije iskoristiti u cilju optimizacije u raznim sektorima. Analiza trenutno implementiranih rešenja pokazuje poboljšanje kvaliteta života u gradovima za 10-30%, a isto tako i govori da su čak i gradovi koji su trenutno najrazvijeniji u ovoj sferi tek na početku primene punog potencijala koji imaju IoT rešenja [3].

3.1. Primene

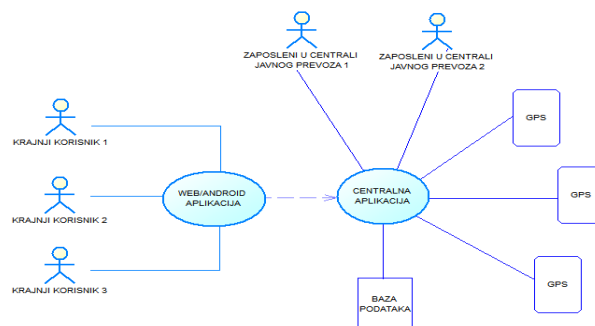
1. **Pametni građani** - Povezivanje građana sa servisima, informacijama i robom koju žele, onda kada oni to žele koristeći tehnologiju koju gotovo svaki građanin poseduje – pametni telefon. Ova veza ide i u suprotnom smeru, odnosno i grad se na ovaj način povezuje sa njegovim građanima koji mogu na različite načine da pruže uvid u probleme koje je potrebno rešiti kako bi se životni standard poboljšao, okolina očuvala i ljudi bili generalno srećniji u svom gradu.
2. **Pametna mreža/potrošnja energije/komunalije** - Jedna od glavnih stavki za poboljšanje kvaliteta života u gradu sigurno predstavlja efikasno i ekološki prihvatljivo korišćenje energetske resursa. Zajedno sa rastom populacije raste i za potreba za resursima. Neka od rešenja koja se nameću predstavljaju električne automobile, pametne merače potrošnje energije u domovima, optimizacija iskorišćenja obnovljivih izvora, solarni paneli... Sva ova rešenja se na nivou grada povezuju u jednu pametnu mrežu koja analizira enormne količine podataka iz mnogih izvora i daje uvid u protok energije na nivou grada i samim tim daje odgovore na pitanja gde je on najotimpalniji, gde je najdostupniji, gde je potrebna optimizacija... Analizom podataka se takođe mogu predvideti periodi intenzivne potrošnje, a mreža se može prilagoditi tako da bude spremna da efikasno odreaguje u takvim periodima.
3. **Pametno upravljanje gradom** - Podrazumeva efikasnije obavljanje administrativnih poslova, težnju da se preko Interneta završavaju administrativne procedure kao i pružanje uvida građanima u cirkulaciju novca kojim grad raspolaže. Pored ovih osnovnih, ljudi koji upravljaju gradom mogu raditi na raznim projektima koji bi promovisali koncepte pametnog grada. Edukacija starijih ljudi o tome kako mogu koristiti novije tehnologije spada u jedan od takvih projekata. Takođe organizovanje raznih hakatona može doprineti razvoju pa čak i implementaciji nekih ideja koje bi poboljšale život građana. Saradnja i razmena znanja sa drugim gradovima koji su uspešno implementirali koncepte pametnog grada je takođe od ključnog značaja jer se unapred mogu izbeći problemi na koje su oni već nailazili prilikom implementacije, kao i izbeći rešenja koja nisu donela pozitivne rezultate.
4. **Pametna zdravstvena zaštita** - Tehnologija je na zdravstvo imala veliki uticaj u današnjem vremenu. Inovacije se vide na svakom koraku, od farmaceutske industrije do komplikovanih hirurških zahvata,

moderna tehnologija je znatno uticala na poboljšanje medicinskih usluga. Tehnologija veštačke inteligencije može putem analize obrazaca može uvideti rizike od razvika bolesti kod pacijenata mnogo pre nego što sam lekar to isto može zaključiti. Robotizovani uređaji olakšavaju život osobama sa invaliditetom, a takođe i usled svoje preciznosti čine nezamenljiv alat u osetljivim operacijama. Sami građani mogu koristiti aplikacije i online platforme za zakazivanje pregleda i time mogu redukovati nepotrebnu papirologiju koja se stvara u zdravstvenim centrima.

5. **Pametni stambeni i poslovni objekti** - Pametne zgrade uvode veliki stepen automatizacije u objekte u kojima ljudi žive. Mnogi senzori se koriste za praćenje raznih uticaja iz okoline, pa se tako temperatura u objektu može automatski regulisati, roletne na prozorima reagovati na količinu svetlosti, senzori za opasnost na detekciju vatre i dima... Takođe implementacija pametnih pratioca potrošnje omogućava stanarima da imaju precizan uvid u to kada i koliko najviše troše električnu energiju, vodu... Stanari na taj način mogu personalizovati podešavanja svog doma tako da se na primer dok su na poslu, sva potrošnja u stanu svede na minimum.
6. **Pametni transport** - Jedan od najvećih problema svetskih metropola jesu upravo gužve u saobraćaju, nedostatak parking mesta, neefikasan javni prevoz... Koncepti pametnog grada u oblasti javnog prevoza omogućuju građanima uvid u lokaciju vozila, vremena kašnjenja, vesti o radovima na putevima i samim tim omogućavaju efikasniju mobilnost. Mreže javnog prevoza se analizom velike količine podataka mogu optimizovati kako bi se na prometnim čvorovima povećala frekvencija vozila, dok bi se na onima koji nisu prometni mogla redukovati. Rešenja za pametne parkinge se mogu implementirati tako da pružaju građanima uvid u slobodna parking mesta i time znatno skrate vreme njihovog pronalaženja. Da bi se maksimalno iskoristilo svako vozilo, vozači mogu putem aplikacija ponuditi mesta u svojim automobilima i podeliti troškove sa ljudima koji takođe putuju na istu lokaciju što je izuzetno korisno za ljude koji stanuju u jednom gradu dok žive u drugom.
7. **Pametna proizvodnja** - Analizom potražnje na tržištu i trenutne opremljenosti skladišta u realnom vremenu može se optimizovati proizvodnja na nivou kompanije. Automatizacija mnogih repetitivnih poslova omogućava ljudima da se bave drugim, fizički manje zahtevnim i kreativnijim poslovima. Takođe se i sama logistika u lancu snabdevanja može pratiti i unaprediti analizom podataka.
8. **Otvoreni pristup podacima** - Koncept otvorenih podataka se bazira na pružanju uvida civilnom stanovništvu u sve relevantne informacije o projektima, nabavkama, tenderima, poslovima... na nivou opštine. Ovo omogućava običnim građanima da koristeći svoju struku sami daju ideje kako se neki projekti mogu poboljšati, od čega benefite, u slučaju realizacije preloženih rešenja, imaju i grad a i sami građani. Transparentnost se uvodi u tokove novca kroz gradske projekte što posledično dovodi do smanjenja mogućih pokušaja malverzacija.

4. STUDIJA SLUČAJA NOVI SAD

Konceptualni model prikazan na slici 1 prikazuje dva tipa korisnika koji će imati interakciju sa sistemom. Prvi tip su zaposleni u centrali javnog prevoza koji bi pratili stanje autobusa, gužvu na putevima, reagovali na kvarove i slično. Drugi tip su krajnji korisnici koji predstavljaju obične građane koji putem web stranice ili android aplikacije mogu proveriti poziciju autobusa na mapi Novog Sada u realnom vremenu. Klijentska web i android aplikacija su zamišljene tako da komuniciraju sa centralnom aplikacijom od koje dobijaju lokacije za liniju autobusa koju je korisnik odabrao. Centralna aplikacija u suštini prima podatke koji dolaze iz GPS uređaja autobusa, procesira ih i smešta u svoju bazu podataka. Ona takođe, na osnovu dobijenih podataka, kreira različite izveštaje u izabranim formatima (XML, CSV, HTML). Zaposleni u centrali javnog prevoza imaju pristup centralnoj aplikaciji i mogu da analizirajući izveštaje donesu zaključke o mogućim optimizacijama koje se mogu izvršiti u cilju boljeg funkcionisanja mreže javnog prevoza.



Slika 1. Konceptualni model

4.1. Implementacija centralne aplikacije

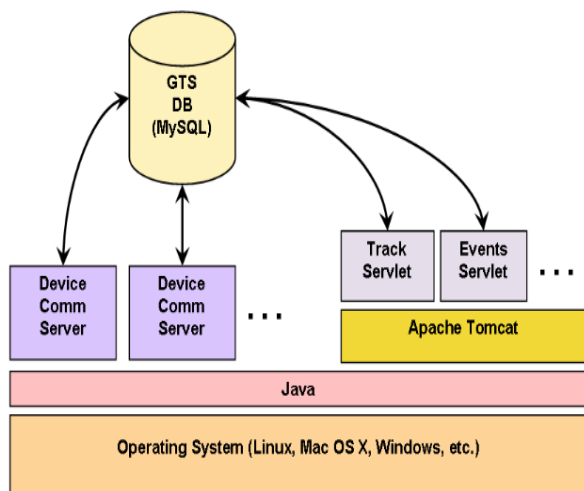
Za povezivanje samih GPS uređaja sa serverom centralne aplikacije predlaže se korišćenje generičkog open-source back-end web servisa za prikupljanje i prikazivanje GPS podataka pod nazivom OpenGTS (Open Source GPS Tracking System) koje je dizajnirano da radi nezavisno od specifičnog GPS uređaja ili protokola [5]. Sporedne usluge koje ovaj servis pruža su sledeće:

- Web bazirana autentifikacija – U okviru jednog akaunta može postojati više korisnika, od kojih svaki ima svoje korisničko ime i šifru kao i kontrolisani pristup različitim segmentima aplikacije.
- Nezavisnost od GPS uređaja – Uređaji za GPS praćenje različitih proizvođača se mogu pratiti simultano u realnom vremenu. Pored predefinisane liste uređaja za koju već postoji implementacija u OpenGTS-u, moguće je i ručno iskodirati implementaciju (uz u dokumentaciji pružen template) i tako integrisati novi uređaj.
- Promenljivi izgled web stranica – Web stranica se može prilagoditi tako da odgovara motivima korisnika.
- Različiti servisi za mapiranje – OpenGTS ima ugrađenu podršku za OpenLayers/OpenStreetMap, kao i za Google Maps, Microsoft Virtual Earth, and Mapstraction, a

drugi provajderi servisa za mapiranje se takođe na lak način mogu integrisati u servis.

- Različiti izveštaji - Interno se koristi XML bazirani engine za kreiranje izveštaja koji se mogu konfigurisati na različiti način tako da prikazuju željene podatke krajnjem korisniku.
- Nezavisnost od operativnog sistema – Servis je napisan u Javi, koristi Apache Tomcat kao web server i MySQL kao bazu podataka. Kao takav, može se pokrenuti na bilo kom sistemu koji podržava ove tehnologije (Linux, Mac OS X, FreeBSD, OpenBSD, Solaris, Windows XP, Windows Vista, Windows 20XX...).
- i18n kompatibilan - Podržava jednostavnu lokalizaciju na različite jezike poput Engleskog, Francuskog, Nemačkog, Grčkog, Bugarskog, Italijanskog, Portugalskog, Rumunskog, Ruskog, Slovačkog, Španskog i Turskog jezika. Ono što je jako bitno jeste da podržava i Srpski jezik.

Osnovna arhitektura OpenGTS sistema, prikazana na slici 2, sastoji se iz raličitih servera za komunikaciju sa uređajima (moduli koji primaju podatke od udaljenih GPS uređaja) koji se izvršavaju kao odvojeni proces. Track servlet (web interfejs), kao i drugi servleti (uključujući bilo koji server za komunikaciju sa uređajima baziran na http protokolu) se izvršavaju unutar servlet kontejnera, odnosno u našem slučaju Apache Tomcat-a. Ono što je jako bitno jeste da su projekat kao i sve različite konfiguracije koje omogućava detaljno dokumentovani [4]. OpenGTS je kompletno implementiran u Javi i radi na svakom sistemu koji podržava JRE (Java Runtime Environment).



Slika 2. Arhitektura OpenGTS servisa

4.2 Interfejs web aplikacije

Kako bi se aplikacija koristila, prvo sto je potrebno jeste da se korisnik uloguje sa svojim. U okviru jednog akaunta može postojati više korisnika. Korisnicima se mogu dodeliti i oduzimati prava pristupa za određene segmente stranice. Tipovi korisnika koje OpenGTS ima ugrađene su:

- Administrator akaunta
- Administrator geozone
- Administrator vozila
- Administrator grupe
- Vozač
- Običan korisnik

Nakon što se uloguje, korisnik (ukoliko ima prava pristupa naravno) može pristupiti stranici Mapping gde se vidi kretanje vozila u realnom vremenu. U padajućem meniju Vehicle Map može se izabrati uređaj/vozilo koje se prati. Klikom na pin možemo videti informacije o GPS lokaciji na kojoj se vozilo u tom trenutku nalazilo, vremenu kada je lokacija izmerena, prosečnoj brzini kojom se vozilo kretalo kao i neke dodatne informacije poput vozača datog vozila, njegovog telefona, procenta baterije GPS uređaja kao i distancu koju je vozilo prešlo... Takođe možemo i proveriti kretanje vozila za neki određeni datum u prošlosti tako što izaberemo u kalendaru vremenski opseg koji nas zanima. Za svaki period očitavanja GPS adrese, kreira se jedan pin na mapi. Pin će u zavisnosti od prosečne brzine vozila imati zelenu, žutu ili crvenu boju.

Ulogovani korisnik takođe ima opciju generisanja izveštaja po različitim kriterijumima na stranici Reports. Nakon što odabere kriterijume za generisanje izveštaja korisnik ima opciju nekoliko formata u kojima želi da vidi prikazane rezultate. Pored HTML, CSV i XML formata, korisnik ima i opciju slanja izveštaja na email adresu koju sam unosi.

4.3. Implementacija web i android aplikacije za krajnje korisnike

Uzimajući u obzir da je servis praćenja autobusa u gradskom prevozu namenjen najviše za krajnje korisnike, odnosno građane Novog Sada, dve implementacije aplikacije bi se mogle realizovati, jedna u obliku web aplikacije, a druga u vidu android aplikacije. Kako bi se olakšao razvoj predlog je implementirati obe aplikacije korišćenjem Java programskog jezika za kreiranje back-end servera, a zatim kreirati dve različite klijentske aplikacije (jednu za android telefone, a drugu za web stranicu) koje će se obraćati istom serveru. Predlog arhitekture i tehnologija koje se mogu koristiti je sledeći:

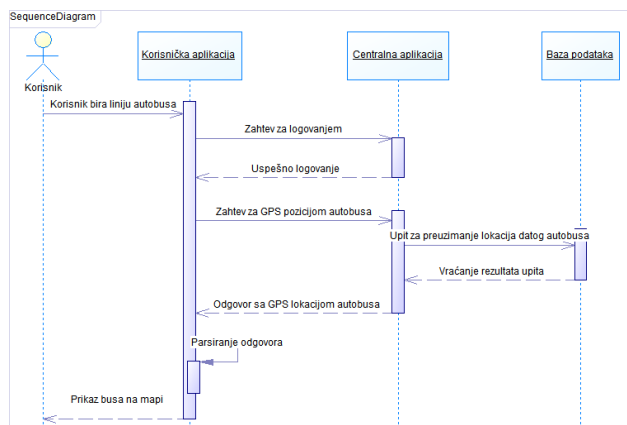
- Back-end server:
 - SpringBoot
 - MySQL
 - Maven
 - Apache Commons CSV library
- Prva klijentska aplikacija:
 - Android aplikacija
 - Google Maps API
- Druga klijentska aplikacija:
 - Angular 2+
 - Google Maps API

Dijagram sekvenci koji vidimo na slici 3 pokazuje jednu interakciju korisnika sa aplikacijom. Korisnik inicira radnju tako što izabere liniju autobusa za koju želi da zna pravac. Nakon toga korisnička aplikacija šalje zahtev za logovanje prema centralnoj aplikaciji sa predefinisanim kredencijalima.

Na centralnoj aplikaciji bi se kreirao jedan univerzalni korisnik koji bi imao prava pristupa samo izveštajima o pozicijama autobusa i ničemu više, i njegovi kredencijali bi se prosleđivali prilikom logovanja. Ovaj korak je neophodan kako bi se kasnije mogli slati HTTP zahtevi centralnoj aplikaciji. Nakon uspešnog logovanja šalje se GET zahtev za GPS pozicije autobusa na izabranoj liniji. Centralna aplikacija zatim formira upit za bazu podataka i

izvlači iz nje poslednjih 10 lokacija autobusa na datoj liniji.

Ukoliko su podaci uspešno izvučeni iz baze, formira se izveštaj u CSV formatu koji se prosledjuje korisničkoj aplikaciji u vidu HTTP odgovora. Back-end aplikacija zatim parsira CSV fajl korišćenjem Apache Commons CSV biblioteke i izvlači neophodne podatke koje prosledjuje klijentskoj aplikaciji. Klijentska aplikacija ima implementiran Google Maps API koji koristi da prikaže lokacije dobijene od strane back-end aplikacije.



Slika 3. Dijagram sekvenci

5. ZAKLJUČAK

Predmet ovog rada jeste bio istraživanje načina na koji bi se jedan od koncepata pametnog grada mogao realizovati na prostoru Novog Sada. Ono što je primećeno da nedostaje, a da se može realizovati, jeste mogućnost da korisnik prati GPS poziciju autobusa na datoj liniji i da ima prikaz o tome gde se autobus trenutno nalazi na mapi. Kako bi se ovakva ideja realizovala potrebno bi bilo pronaći isplativo i jednostavno rešenje.

Kao takvo pronađeno je OpenGTS rešenje, koje predstavlja projekat otvorenog koda, dostupan na internetu i besplatan za preuzimanje. Uz njega ide i obimna dokumentacija o konfigurisanju i pokretanju samog projekta kao i već gotova implementacija koja povezuje server sa GC-101/GX-101 GPS uređajima.

Projekat je uspešno realizovan u preko 100 zemalja i koristi se za praćenje više hiljada vozila na svih 7 kontinenata što govori o njegovoj pouzdanosti. U radu je izložen konceptualni model u kome se vidi kako bi se projekat mogao realizovati sa autobusima gradskog prevoza Novog Sada.

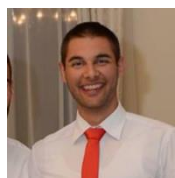
Pored konfigurisanja i pokretanja OpenGTS projekta i povezivanja GPS uređaja sa projektom bilo bi neophodno i realizovati dve klijentske front-end aplikacije i jednu back-end aplikaciju koje bi zapravo prikazivale informacije krajnjim korisnicima o poziciji autobusa na mapi.

Nakon predloženog modela izvršena je i analiza troškova gde se jasno vidi da implementacija ovakog rešenja ne bi predstavljala novčano prezahtevan projekat koji definitivno popeo Novi Sad na jednu stepenicu više u pogledu razvijenosti i uspešne primene IoT koncepata. Nakon predstavljanja modela i analize troškova takođe je pružen i uvid u moguća poboljšanja i unapređenja na kojima se može raditi nakon realizacije projekta.

6. LITERATURA

- [1] www.un.org (pristupljeno u septembru 2019.)
- [2] www.sr.wikipedia.org (pristupljeno u septembru 2019.)
- [3] www.mckinsey.com (pristupljeno u septembru 2019.)
- [4] GeoTelematic Solutions, Inc, „OpenGTS Installation and Configuration Manual”,
- [5] www.opengts.org/ (pristupljeno u septembru 2019.)

Kratka biografija:



Ivan Jančić rođen je u Smederevu 30.07.1995. god. Završio je Osnovnu školu „Dr Jovan Cvijić” u Smederevu. Nakon završene Osnovne škole upisuje srednju školu, Gimnaziju u Smederevu. 2014. godine upisuje „Fakultet tehničkih nauka” u Novom Sadu, smer „Računarstvo i automatika”. 2016/17. godine upisuje smer „Primenjene računarske nauke i informatika”. 2017/18. godine upisuje smer „Softversko inženjerstvo”. 2018. godine dobija zvanje Diplomirani inženjer elektrotehnike i računarstva. 2018/19. godine upisuje master akademске studije, smer „Primenjene računarske nauke i informatika – Elektronsko poslovanje”. kontakt: ivanjancic4@gmail.com

**ПРИМЕНА ДУБОКОГ УЧЕЊА УСЛОВЉАВАЊЕМ НА САМОВОЗЕЋИ
АУТОМОБИЛ У СИМУЛИРАНОМ ОКРУЖЕЊУ TORCS****USING DEEP REINFORCEMENT LEARNING TO TRAIN AN AUTONOMOUS
DRIVING AGENT IN A SIMULATED ENVIRONMENT TORCS**

Ђорђе Марјановић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду описан је систем у којем се дубоко учење условљавањем примењује на аутономно возилоу симулираном окружењу. Агент је трениран помоћу Deep deterministic policy gradient (DDPG) алгоритма, а окружење представља 3D тркачка видео игра TORCS. Након више од 200 епизода обучавања агент је успео да заврши цео круг без скретања ван стазе. DDPG алгоритам се показао веома успешно у окружењима са континуалним акцијама.

Кључне речи: дубоко учење условљавањем, DDPG, аутономна возила

Abstract – This paper presents an application of deep reinforcement learning for self-driving car in simulated environment. Agent is trained using Deep deterministic policy gradient (DDPG) algorithm and environment is 3D racing video game TORCS. After more than 200 episodes of training agent was able to finish a full lap without turning of the road. The results shot that the DDPG algorithm is very successfull in environments with continuous action space.

Keywords: deep reinforcement learning, DDPG, self-driving car

1. УВОД

Значајан напредак машинског учења и дубоких неуронских мрежа (енг. *Deep Neural Networks*) омогућио је развој система критичних за безбедност као што су аутономни аутомобили. Скорашњи резултати показују да су самовозећи аутомобили постали веома ефикасни у пракси и већ прешли милионе километара без икакве људске контроле [1]. Двадесет америчких држава укључујући Калифорнију, Тексас и Њу Јорк усвојиле су закон који омогућава тестирање и развој аутономних возила [2]. Аутономна возила обећавају велику корист за човечанство, као што је значајно смањење повреда и смртних случајева у саобраћајним несрећама и ефикасније коришћење саобраћаја у сврху смањења загађености ваздуха и смањењем трошкова. Међутим, у овом тренутку ова технологија је још увек у развоју.

НАПОМЕНА:

Овај рад је проистекао из мастер рада чији ментор је био др Александар Ковачевић, проф.

У овом раду описано је дубоко учење условљавањем (енг. *Deep Reinforcement Learning*) и његова примена на решавање проблема самовозећег аутомобила у симулираном окружењу. Као окружење користи се видео игра TORCS. Овај систем има клијент-сервер структуру, при чему серверски део обухвата окружење, а део система који помоћу алгоритма контролише кретање аутомобила представља клијента. Допринос овог рада је имплементација Deep Deterministic Policy Gradient алгоритма који у овом окружењу са континуалним акцијама одређује угао скретања аутомобила. Резултати су показали да обучавањем алгоритам успешно контролише кретање аутомобила не дозвољавајући скретање аутомобила изван ивица стазе.

Рад је подељен у шест секција. У наредној секцији дат је преглед постојећих решења. У трећој секцији дате су теоријске основе. Четврта секција опису детаље и начин имплементације система. У петој секцији приказани су резултати, а у шестој изложен је закључак овог рада.

2. СРОДНИ РАДОВИ

Једна од првих успешних демонстрација аутономног возила датира од 1986. године [3]. У то време, симулиране слике пута коришћене су за обуку трослојне неуронске мреже чија је излазна вредност правац који аутомобил треба да прати на путу. Од тада, истражује се широк спектар аспеката аутономне вожње укључујући перцепцију, локализацију, планирање и контролу. Људи су постепено почели да се фокусирају на реалније урбане средине са неизвесним саобраћајем, што је и даље важан циљ. У пројекту ALVINN [4] 1989. год. коришћена је потпуно повезана неуронска мрежа (један скривени слој са 29 неурона) за предвиђање управљачких команди за аутомобил. Пројекат DAVE [5] из 2004. је учио на основу људске вожње. Инспирисан DAVE пројектом, NVIDIA тим је тренирао велику конволутивну неуронску мрежу мапирањем слика добијених из вожње правог аутомобила на команде за управљање аутономног возила – *behavioral cloning* [6].

Учење условљавањем (енг. *Reinforcement learning*) такође налази примену у аутономној вожњи [7]. Учење условљавањем у почетку није било успешно у аутомобилским апликацијама, међутим скорашњи радови показују способност алгоритама учења условљавањем да стекну представу о окружењу у

коме оперирају. *Google DeepMind* је ово демонстрирао у пројектима где је агент научио да игра игре као што су *Atari* и *Go* [8].

Захваљујући најновијим достигнућима вештачке интелигенције, *Automotive Driver Assistance Systems (ADAS)* системи остварују брз напредак у протеклих неколико година. Тренутно, бројне истраживачке групе и компаније напорно раде на довођењу ове технологије на тржиште. Један од лидера је *Waymo*, Гуглов пројекат аутомобила без возача. Од настанка 2009. године прешао је преко 4 милиона километара на јавним путевима. *Uber* врши тестирања и планира да опреми 75.000 самовозећих аутомобила у 13 америчких градова до 2022. године. *Tesla* је лансирао свој аутопилот 2014. и до краја 2019. године очекују демонстрацију потпуно самосталне вожње.

3. ТЕОРИЈСКЕ ОСНОВЕ

Учење условљавање (енг. *Reinforcement learning*) је грана машинског учења која се бави секвенцијалним одлучивањем. Проблем учења условљавањем састоји се од агента (енг. *agent*) и окружења (енг. *environment*). Агент у окружењу извршава акције из дозвољеног скупа акција и добија одговор окружења у виду награде (енг. *reward*). Награда, позитивна или негативна вредност, је начин да се агенту допусти да зна колико је била добра акција у том тренутку. Задатак агента је да научи оптималну политику – мапирање из стања (енг. *state*) у акцију, које максимизује очекивану суму награде.

3.1. Q-Learning

Q-Learning алгоритам је један од најпопуларнијих техника учења условљавањем, заснива се на учењу вредности стања. Циљ овог алгоритма је да агент научи оптималну функцију $Q(s, a)$, при чему Q функција рачуна квалитет, односно Q вредност задатог пара (стање, акција). Ова функција представља итеративни поступак апроксимације, при чему се на сваком кораку ажурира тренутна процена Q вредности коришћењем процена оптималне вредности у будућности (*temporal difference*) [9].

3.2. Policy gradient методе

Метод градијента политике су приступ у учењу условљавањем јер директно оптимизују кумулативну награду и могу се лако користити са нелинеарним функцијама за апроксимацију као што су неуронске мреже [9].

3.3. Actor-Critic методе

Actor-Critic архитектура користи две структуре да оптимизује очекивани резултат. *Actor* и *Critic* раде заједно и посебно се обучавају. *Actor* дефинише политику и његова намена је да генерише акције у складу са тренутном политиком. Функција вредности (*Critic*) делује као критичар пошто процењује политику на основу грешке у временској разлици (*temporal difference error*). *Actor-Critic* методе пружају веома добра својства конвергенције и олакшавају израчунавање акције, посебно у простору са континуалним

акцијама. Као функције за апроксимацију веома често се користе неуронске мреже [10].

3.4. Deep deterministic policy gradient (DDPG)

Deep deterministic policy gradient (DDPG) [11] је алгоритам учења условљавањем који комбинује *Q-learning*, *policy gradient* и *actor-critic* меотде и веома погодан за проблеме са континуалним или великим бројем акција. Заснива се на *Deterministic Policy Gradient (DPG)* [12] методи и користи неуронске мреже као апроксимационе функције детерминистичке политике.

Q-Learning није ефикасан у окружењима са континуалним акцијама, као решење користи се *Actor-Critic* структура у облику неуронских мрежа. *Actor* функцију $\mu(s|\theta^\mu)$ која одређује политику која детерминистички за дато стање резултује конкретном акцијом која ће се извршити. *Actor* и *Critic* су међусобно повезани, тако да *Critic* користи тренутно стање и акцију добијену од *Actor*-а за то стање да одреди вредност акције $Q(s, a)$.

Показало се да директна примена неуронских мрежа у *DPG* алгоритму не доводи до стабилног обучавања. Као решење овог проблема користе се *target networks* које представљају копије *Actor* и *Critic* мрежа. Тежине ових мрежа се ажурирају веома споро тако да прате оригиналне мреже. Рачунају се по формули (1)

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta' \quad (1)$$

Мања вредност параметра $\tau \ll 1$ доводи до споријег праћења оригиналних тежина, али увелико побољшава стабилност обучавања. Као још један начин побољшања перформанси, *DDPG* алгоритам користи *experience replay*.

Циљ процеса обучавања *DDPG* алгоритма је да максимизује вредност функције $Q(s, a)$ коју користи астор за модификацију свог параметра θ . *Critic* се обучава користећи *Q-Learning* над узорком података из *replay-buffer*-а при чему се *TD-error* рачуна као сума непосредне награде и излазне вредности *target actor* и *target critic* мрежа.

Actor мрежа учи директно из посматраног простора користећи детерминистички *policy gradient* метод (*DPG*). Детерминистичка политика може се посматрати као посебан случај стохастичне политике у чијој расподели вероватноћа постоји једна екстремна не нула вредност за једну акцију. Заправо, доказано је да стохастична политика $\pi_{\mu, \sigma}$ је еквивалентна детерминистичкој политици μ_θ где је параметар $\sigma = 0$.

Применивши ову теорему у методи *policy gradient* долази се до закључка да је стохастични градијент политике еквивалентан детерминистичком градијенту политике. Пошто политика μ није зависна од акције, за обучавање *Actor* мреже довољан је градијент резултата *Critic* мреже помножен градијентом резултата *Actor* мреже над узорком података из *replay-buffer*-а [11][12].

Главни изазов учења у окружењима са континуалним акцијама представља експлорација. Предност *off-policy* алгоритама, као што је *DDPG*, је у томе што се проблем експлорације може третирали независно од учења алгорита. Овај проблем решава се додавањем шума (енг. *noise*) $\mathcal{N}$ на политику *actor*-а, тј. на саму акцију (2).

$$\mu'(s) = \mu(s_t|\theta^\mu) + \mathcal{N} \quad (2)$$

4. ИМПЛЕМЕНТАЦИЈА

Систем је структуриран као клијент-сервер апликација, при чему *TORCS*¹ видео игра представља серверски део, а за клијентски део користи се *SnakeOil*² библиотека помоћу које се контролише аутомобил. Ова библиотека омогућава добављање 21 параметра (сензора) аутомобила који описују стање агента, опсервацију, а то су:

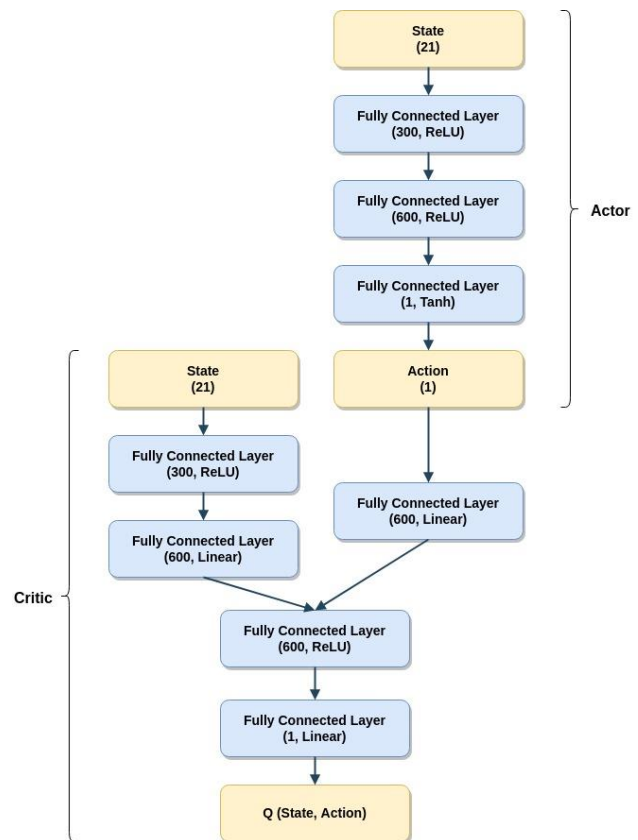
- **angle** $[-\pi, \pi](\text{rad})$ – угао између правца аутомобила и правца стазе
- **track** $[0, 200](\text{m})$ – низ од 19 сензора чије вредности представљају различита растојања аутомобила од ивица стазе
- **trackPos** $(-\infty, \infty)$ – растојање аутомобила и средишта стазе

За примену алгоритама учења условљавањем користи се *Gym-TORCS*³ емулатор који је потпуно компатибилан са *OpenAI Gym*⁴ пакетом алата. Убрзање и кочење аутомобила контролисаће се аутоматски од стране клијента а *DDPG* алгоритам одређује акцију агента – **угао скретања** у вредности $[-1, 1]$ при чему вредност -1 представља максимално скретање удесно, а 1 максимално скретање улево.

У решавању проблема учења условљавањем веома је битно добро дефинисати функцију награде како би учење било ефикасно. У овом пројекту циљ је да аутомобил научи да се креће на стази, да не стоји и да не скреће ван стазе. Према томе дефинише се функција награде која максимизује брзину аутомобила у правцу стазе, минимизује попречну брзину и додељује негативну вредност на већу удаљеност од средишта стазе. За ефикасније и брже обучавање додат је услов да при скретању ван стазе агент добија награду -200 и сигнал за прекид епизоде.

Actor и *Critic* конструисани су помоћу вишеслојних неуронских мрежа. Улазни податак у *actor* мрежу је стање агента у облику вектора са 21 елементом, а излаз је акција агента, тј угао скретања дефинисан активационом функцијом хиперболични тангенс која ограничава резултат на опсег $[-1, 1]$.

Critic неуронска мрежа као улаз прима два податка: стање агента и акцију агента за то стање које производи *Actor* мрежа. Архитектура *Actor* и *Critic* мрежа са активационим функцијама и бројем неурона приказана је на слици 4.1.



Слика 4.1. Actor-Critic архитектура

Структура *Actor* и *Critic* мрежа и избор хиперпараметара *DDPG* алгоритма формиран су по моделу описаном у раду [11] где је показано да алгоритам са истим овим параметрима примењен на различита окружења даје веома добре резултате. Комплетан код овог пројекта, документација и упутства налазе се на *GitHub* репозиторијуму⁵.

5. РЕЗУЛТАТИ

Циљ обучавања агента је да оствари што већу кумулативну награду, односно да се креће стазом у што дужем временском периоду. Скретање ван стазе сматра се лошом акцијом при чему агент добија негативну награду -200 и сигнал за завршетак епизоде.

Након више од 200 епизода обучавања аутомобил је успео да, без скретања ван стазе, константно вози више од два круга. У овој епизоди агент је извршио више од 6.000 акција и остварио највећу кумулативну награду до тог тренутка у вредности преко 130.000. Процес обучавања алгоритма, посматрајући кроз кумулативну награду по епизоди приказан је на графику 5.1.

¹ <http://torcs.sourceforge.net/>

² <http://xed.ch/project/snakeoil/>

³ https://github.com/ugo-nama-kun/gym_torcs

⁴ <https://gym.openai.com/>

⁵ <https://github.com/djo10/deep-rl-ddpg-self-driving-car>

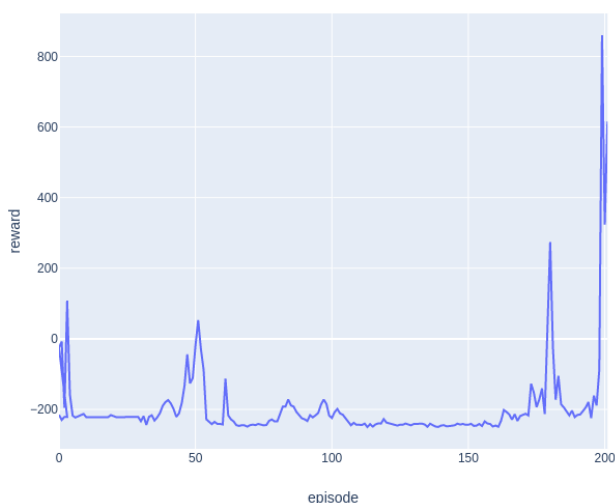


График 5.1. Кумулативна награда по епизоди

На основу графикона може се приметити да агент у раној фази обучавања извршава лоше акције које доводе до негативне кумулативне награде. Пошто алгоритам учења условљавањем учи на основу прикупљања искуства, после 200 епизода агент постаје способан да извршава исправне акције и максимизује кумулативну награду. Процес обучавања DDPG алгоритма у TORCS видео игри забележен је видео снимком⁶.

6. ЗАКЉУЧАК

У раду је представљен је самовозећи аутомобил у симулираном окружењу применом учења условљавањем. За управљање агента коришћен је *Deep Deterministic Policy Gradient (DDPG)* алгоритам, а тркачка видео игра TORCS је представљала окружење. Резултати су показали да је овај алгоритам веома ефикасан у окружењима са континуалним акцијама и његова примена у контроли кретања аутомобила показала се веома успешном. Предност алгоритама учења условљавањем је да константно могу да се усавршавају и тиме достигну, а чак и надмаше резултате постигнуте од стране човека.

Како су ове технологије и аутономни аутомобили још увек у развоју, овај пројекат има доста простора за унапређивање и усавршавање. Пре свега, може се модификовати алгоритам да поред скретања контролише брзину и кочење аутомобила. Већина савремених решења за самовозеће аутомобиле користе камере, јер камере представљају најбољу врсту сензора. Употреба конволутивних неуронских мрежа при обради стања агента (слике) представља корак ближе реалним системима аутономних возила. Такође у систем се може имплементирати посебан модул за препознавање препрека као што су пешаци, остали аутомобили у саобраћају, знакови, семафори итд. Заправо, главни циљ оваквих система је усавршавање окружења и алгоритма да обучавање у симулираном систему може успешно да се искористи у реалном свету аутономних возила.

7. LITERATURA

- [1] 2017. The Numbers Don't Lie: Self-Driving Cars Are Getting Good. <https://www.wired.com/2017/02/california-dmv-autonomouscardisengagement>. (2017).
- [2] 2017. Autonomous Vehicles Enacted Legislation. <http://www.ncsl.org/research/transportation/autonomous-vehiclesselfdriving-vehicles-enacted-legislation>. Aspx. (2017).
- [3] Takeo Kanade, Chuck Thorpe, and William Whittaker. Autonomous land vehicle project at cmu. In Proceedings of the 1986 ACM fourteenth annual conference on Computer science, pages 71–80. ACM, 1986. 1.1
- [4] Pomerleau, D. A. Alvin, an autonomous land vehicle in a neural network. Technical report, Carnegie Mellon University, Computer Science Department, 1989
- [5] Net-Scale Technologies. Autonomous off-road vehicle control using end-to-end learning. Technical report, 2004. Available at: <http://netscale.com/doc/net-scale-dave-report.pdf>. [Accessed 17 March 2017]
- [6] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J. et al. 2016. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316
- [7] A. El Sallab, M. Abdou, E. Perot, and S. Yogamani. Deep reinforcement learning framework for autonomous driving. Autonomous Vehicles and Machines, Electronic Imaging, 2017
- [8] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013
- [9] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018
- [10] V. R. Konda and J. N. Tsitsiklis, "On Actor-Critic Algorithms," SIAM Journal on Control and Optimization, vol. 42, pp. 1143–1166, Jan 2003.
- [11] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, pp. 1–14, 2015.
- [12] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 387–395, 2014

Кратка биографија:



Ђорђе Марјановић рођен је 16.3.1994. године у Ужицу. Основне студије уписао 2013. године на Факултету техничких наука, смер Софтверско инжењерство и информационе технологије. Основне студије завршио 2017. Након чега уписује мастер студије, смер Интелигентни системи.

⁶ <https://www.youtube.com/watch?v=XyvOfroVIwg>

SISTEM ZA LOKALIZACIJU OBJEKATA U ROBOTICI**SYSTEM FOR OBJECT LOCALIZATION IN ROBOTICS**Miloš Arbanas, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu analizirano je projektovanje sistema koji je u mogućnosti da detektuje željene objekte u okolini robota, a potom i odredi njihovu lokaciju. Analizirano je više pristupa rešavanju ovog problema, od korišćenja jednostavnih senzora za detekciju boje, do nešto naprednijih koji uključuju manipulaciju nad slikama i mašinsko učenje. Uređaj je praktično realizovan i prikazani su rezultati uspešnosti za nekoliko metoda, na osnovu čega su izvučeni zaključci. Na kraju su predloženi i dalji koraci koji bi mogli dodatno da poboljšaju rad uređaja.

Ključne reči: senzori, mašinska vizija, robotika, detektovanje objekata

Abstract – This paper analyzes the development of a system that is able to detect objects of interest, and also find their position. A few approaches and solutions were analyzed, from using simple color detection sensor, to somewhat complicated systems that include image manipulation and machine learning. The system was implemented and the results are shown for different methods, from which the conclusions were drawn. At the very end, some potential improvements are explained that could increase the performance of the system.

Keywords: sensors, machine vision, robotics, object detection

1. Uvod

Tema ovog završnog rada je projektovanje i izrada sistema koji je u mogućnosti da detektuje željene objekte u okolini robota, a potom odredi njihovu lokaciju.

Primer jedne situacije, gde se mogu uočiti objekti od interesa je prikazan na slici 1. Uređaj mora biti u mogućnosti da brzo detektuje željeni objekat i odredi njegovu poziciju u odnosu na poziciju robota.

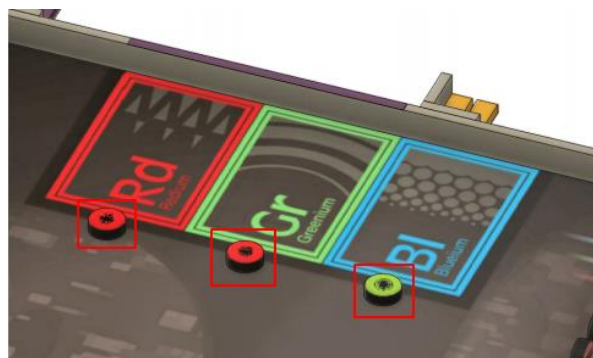
Jedan od najvećih izazova bio je napraviti sistem koji radi pod raznolikim uslovima, prvenstveno različitim intenzitetima i vrstama osvetljenja.

Analizirano je više pristupa rešavanju ovog problema, od jednostavnih senzora boje, do nešto kompleksnijih koji uključuju obradu slike sa kamere. Upoređeni su rezultati dobijeni klasičnim metodama obrade slike i nešto modernijim poput mašinskog učenja.

NAPOMENA:

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

Uređaj je praktično realizovan i izvršena su merenja uspešnosti detekcije željenih objekata. Sama ideja za izradu ovog uređaja nastala je usled potrebe da se detektuju objekti u okolini robota korišćenog na takmičenju Eurobot 2019. godine.



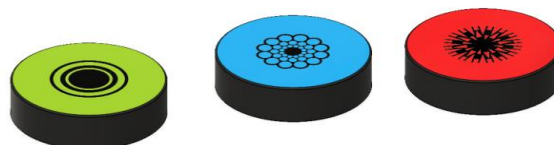
Slika 1. Primer terena sa obeleženim elementima

2. Analiza problema

Glavna primena projektovanog sistema je u okviru takmičenja u robotici, tako da se ovaj rad bazira na rešavanju konkretnog problema. Objekti od interesa, koje je potrebno detektovati su plavi, crveni i zeleni pakovi, sa određenim šarama na njima. Prikazani su na slici 2.

Kao logično rešenje se nameće rešenje sa kamerom, kojom mogu da se detektuju udaljeni objekti. Drugo moguće rešenje je detekcija senzorom boje. Ako su objekti od interesa statični, i unapred im je poznata lokacija, senzor boje je dovoljan. Međutim, ako ne znamo pozicije objekata unapred, kamera pruža mnogo više mogućnosti, iako je rešenje tog tipa komplikovanije, a samim tim potencijalno manje robusno.

Sistem sa redundantnim izvorima informacija je uglavnom sistem sa većom pouzdanošću. Tako da je potencijalno najbolje rešenje koristiti oba pristupa. Prvo detektovati objekte kamerom, prići, pokupiti ih aktuatorom u robota, a nakon toga proveriti senzorom boje da li je objekat zaista dobro detektovan.



Slika 2: objekti od interesa

2.1. Analiza pozicioniranja kamere

Kamera se može postaviti da bude van robota, statična, na mestu koje omogućava dobar pregled terena. Tada je površina koju kamera pokriva, na kojoj je moguće detektovati objekte, nezavisna od položaja samog robota. Problem koji postoji sa ovim rešenjem je to da je velika verovatnoća da nešto, robot na primer, zakloni deo terena, a time i neke od objekata.

Njih je tada nemoguće detektovati dok se objekat koji ih je zaklonio ne pomeri.

Sa druge strane, ako se postavi kamera na samog robota, dok se robot kreće, menja se i površina terena koju može da posmatra. Najviše ima smisla da se kamera stavi na prednju stranu robota, odnosno da pokriva površinu prema kojoj robot ide. Pozicija kamere zavisi i od mehanizama kojima se manipuliše nad detektovanim objektima.

Može biti da je potrebno i više kamera, neke isključivo da lokalizuju objekte, a druge da pomažu pri samoj manipulaciji objekata. Na primer, da se detektuje da li je mehanizam pokupio objekat, ili da li ga je ostavio na željeno mesto. U nekim slučajevima, kamera je preterano komplikovano rešenje i senzor boje može dati dovoljno informacija da se izbegne korišćenje kamere.

2.2. Analiza ključnih parametara pri izboru kamere

Kako polje mašinske vizije napreduje, tako se i širi izbor kamera koje se koriste u raznim slučajevima. Od same primene zavisi i na koje specifikacije treba obratiti pažnju. Postoji mnogo specijalizovanih, industrijskih kamera, koje su pravljenе isključivo za potrebe industrije. One pružaju mnoge pogodnosti, poput automatskog fokusa, prilagođavanja ekspozicije, pojačanja, balansa bele boje, otpornosti na teške uslove rada i slično... Takve kamere pružaju mnoge pogodnosti i olakšavaju obradu slike, a u industriji su poželjne i zbog robusnosti.

Pored profesionalnih industrijskih kamera, postoje kamere koje su pristupačnije, a pružaju dovoljno za zahteve postavljene u ovom radu. Neke od važnijih karakteristika su:

- Rezolucija

Parametar koji određuje koliko piksela ima dobijena slika sa kamere. Direktno utiče na kvalitet i oštrinu slike. Posebno je bitno ako su objekti udaljeni. Za detekciju objekata koji su blizu kamere (na primer, kamera se nalazi na robotu) nije potrebna velika rezolucija.

- Širina vidnog polja

Parametar koji je izražen u stepenima, a predstavlja ugao kojim kamera gleda. Od nje direktno zavisi na kolikoj površini je moguće detektovati objekat. Treba imati u vidu da veća širina vidnog polja znači i veću distorziju slike. Potrebno je uvesti dodatne metode za kompenzaciju, na primer kalibracija kamere.

- Način povezivanja kamere

S obzirom na to da je potrebno da sistem pokuša da lokalizuje objekte čim kamera uslika prostor ispred sebe, bitno je da kamera ima brzu komunikaciju sa sistemom za obradu. Web kamere sa USB interfejsom su pogodne, dok su GoPro kamere izuzetno teške za korišćenje zbog Wi-Fi ili micro HDMI interfejsa, iako su, najčešće, po svim ostalim specifikacijama superiornije.

- Brzina rada kamere

Važan parametar može broj odbiraka u sekundi (*Frames per second*). Ako se robot kreće, a potrebno je da u toku kretanja detektuje objekte ispred sebe, poželjno je da kamera ima dovoljno veliki FPS.

- Postojanje automatskog prilagođavanja i obrade slike

Veliki broj kamera u sebi sadrži dodatni procesor koji modifikuje izlaz pri različitim uslovima okoline. Automatski se prilagođava količini osvetljenja, filtrira se šum, pojačavaju se boje, uklanjaju se crvene zenice i slično. Treba biti svestan postojanja i paziti na koji način utiče na sliku koju je potrebno obraditi kasnije. Takođe je vrlo bitno da postoji i autofokus, kako bi se detektovali lepo i objekti koji nisu uvek na istoj udaljenosti u odnosu na kameru.

2.3. Ponašanje kamere pri različitim osvetljenjima

Jedan od najvećih problema je napraviti robusan sistem koji radi u različitim spoljašnjim uslovima. Jedan primer, koji opisuje ozbiljnost problema, je primer sa takmičenja. Dešava se u finalnim mečevima, kada se smanji broj terena na kojima se igra, da se svi reflektori prebace na jedan teren. Količina osvetljenja se drastično poveća i to utiče na rad kamere i na sliku koja se dobije. Slika koja je dobijena sa iste kamere, a pri različitim osvetljenjima, ima veoma različitu zasićenost boja ako sliku posmatramo u HSV formatu. To u mnogome otežava situaciju ako se koriste metode gde se na osnovu boje traže objekti.

2.4. Uređaj za obradu informacija

Sistem bi trebao biti mali, pogotovo ako ide na samog robota. Tako da je potreban i dovoljno mali procesor koji bi obradio signal sa senzora ili sliku sa kamere. Ako se koristi kamera, zahtevi su značajno veći. Potreban je mali kompjuter koji ima interfejs ka kameri i ima dovoljno računarske snage da obradi sliku dovoljno brzo.

Ako nije potrebno vršiti operacije nad velikim matricama, neki mali računari poput Raspberry Pi, UpBoard, BeagleBone dovoljni su za obradu.

U slučaju korišćenja neuronske mreže, zahtevi postaju izuzetno ozbiljni. Ne postoji velik izbor uređaja koji mogu da se koriste na optimalnom nivou. Najkorišćeniji su Nvidia Jetson sistemi, koji ubrzavaju izračunavanja slično grafičkim karticama. Druga mogućnost je korišćenje specijalizovanih procesora za mašinsko učenje poput Movidius Neural Compute Stick-a koji ubrzava rad neuronskih mreža. Još jedna opcija je kreiranje sopstvenih specijalizovanih procesora u FPGA tehnologiji, koji direktno odgovaraju potrebama konkretne izvedbe i lako mogu biti prilagođeni po potrebi.

3. Detaljan opis svih podsistema uređaja

U ovom poglavlju biće opisani podsistemi u konkretnoj praktičnoj izvedbi sistema. Odabrana su rešenja gde se koristi kamera pa se potom slike obrađuju različitim metodama. Jedan sistem se nalazi na robotu i gleda ispred robota. Elementi se detektuju kako se robot kreće i robot može da pokupi elemente ako je to u interesu.

Sistem se sastoji iz kamere, koja je fizički fiksirana na robota, i malog računara Raspberry Pi, koji obrađuje sliku sa kamere i preko mreže javlja robotu gde se objekat nalazi ako ga detektuje.

3.1. Kamera

Odabrana kamera za praktičnu izvedbu je Raspberry Pi Camera Module v2, sa Sony IMX2019 senzorom i rezolucijom od 8 Megapiksela. Dovoljno je malih dimenzija da se veoma lako montira na robota i ne zauzima mnogo mesta. Pored toga, interfejs je maksimalno pojednostavljen sa strane programera, tako da je laka i brza za korišćenje. Rezolucija je i više nego zadovoljavajuća s obzirom na to da kamera gleda u pod ispred robota, pa su svi objekti relativno blizu kameri.

3.2. Uređaj za obradu slike

Kao uređaj za obradu slike korišćen je Raspberry Pi 3 računar. S obzirom na njegovu popularnost, nameće se kao najbolji izbor u početku.

3.3. Algoritam rada

Konceptualno algoritam rada nije komplikovan i sastoji se iz 4 koraka:

- 1) Uslikati prostor ispred robota
- 2) Detektovati objekat
- 3) Odrediti lokaciju objekta
- 4) Poslati lokaciju upravljačkom sistemu robota

Koraci jedan i tri su trivijaleni i neće biti detaljno analizirani.

Korak dva je u stvari srž ovog rada i u narednom poglavlju će biti opisane metode za detekciju objekata od interesa.

3.3.1. Lokalizacija detektovanog objekta

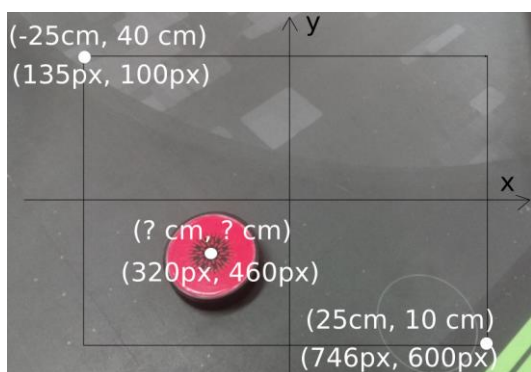
Lokaciju objekta nije teško naći ako je poznata koordinata objekta na slici. Pretpostavlja se da je objekat na podlozi po kojoj se kreće i robot. Algoritam neće raditi ako se objekat nalazi na nekoj platformi na primer. Pronalaženju objekta se može pristupiti na dva različita načina. Prvi je preračunavanje pozicije objekta ako se zna tačan ugao pod kojim kamera gleda teren, kao i tačna visina na kojoj se nalazi kamera. Drugi način je kalibracija bilinearnom interpolacijom. Odnosno definišu se dve tačke po svakoj koordinati i pronade se linearna funkcija koja preslikava koordinate u pikselima u udaljenost u centimetrima.

Na primeru na slici 3, x koordinata crvenog paka bi bila:

$$x = 0.08 * px - 36$$

$$x = 0.08 * 320 - 36 = -10.6 \text{ cm}$$

Ova formula važi dok se ne promeni pozicija kamere.



Slika 3: prikaz bilinearne interpolacije

4. Analiza metoda za detekciju objekata

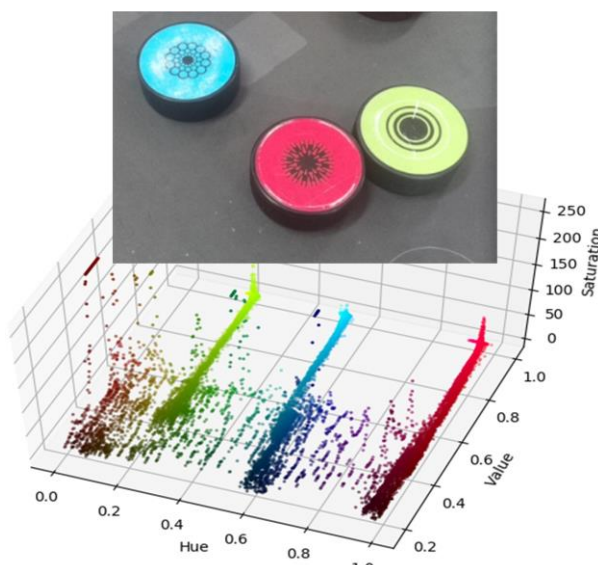
Dve metode su isprobane, u ovom poglavlju će biti ukratko opisane, a u narednom će biti predstavljeni rezultati.

4.1. Detekcija pomoću izolacije boje u HSV domenu

Prvi metod se sastoji u tome da se izoluju elementi određene boje. Objekti su plavi, zeleni i crveni, tako da se lako razlikuju. Nakon što se boja izoluje, nađu se sve konture, a potom se proveriti da li su konture krugovi.

Slika 4: prikaz HSV histograma elemenata od interesa

Izolacija boje se radi u HSV domenu. Ovaj domen je izuzetno pogodan jer *Hue* isključivo zavisi od boje, odnosno hromatskih osobina objekta i, za elemente koje



Slika 4: prikaz HSV histograma elemenata od interesa

je potrebno detektovati, vrlo širok prag tolerancije je moguće staviti.

Izolacija se vrši tako što se definiše opseg *Hue*-a, *Value*-a i *Saturation*-a koji isključivo izoluju elemente od interesa.

Na slici 4 se može videti slika i 3D prikaz histograma slike. Jasno se uočava da je potrebno definisati opseg po svakoj dimenziji da bi se izolovala određena boja.

Opsege je najlakše kalibrisati ručno, tako što se na velikom broju slika kalibrišu opsezi za svaku boju koji je dobro izoluju.

Nakon provlačenja slike kroz opsege, dobija se slika koja je crna svuda, osim na toj boji i pomoću funkcije ugrađanje u OpenCV koja se zove *findContours*, dobijaju se konture na slici. Potom je potrebno proveriti da li je kontura približno krug. Primitivno rešenje, koje se pokazalo kao dovoljno dobro, je da se definiše približna površina koju bi krug na tom rastojanju imao. I drugo, da se proveriti da li je pravougaonik koji opisuje krug približno kvadrat, sa nekim mogućim definisanim odstupanjem od idealnog.

Na taj način je moguće proveriti da li se kontura poklapa sa pretpostavljenom veličinom elementa koji bi mogao potencijalno da se nalazi na tom rastojanju.

Jasno je da je ovo rešenje ograničeno na primenu detektovanja elemenata na podlozi ispred samog robota.

4.2. Konvolucione neuronske mreže

Prethodno opisani metod u mnogome zavisi od velikog broja parametara. Svaka promena u sistemu zahteva ponovnu kalibraciju. Neki spoljašnji faktori mogu da dovedu do toga da metoda uopšte ne radi, kao što je odsjaj koji veći deo elementa može da prividno prikaže bez hromatskih osobina.

Primer jednog takvog slučaja je prikazan na slici 5. U ovom slučaju algoritam moguće da neće raditi jer površina konture potencijalno ne bi odgovarala površini kruga. Tolerancija se može povećati, ali se onda javlja problem da se možda detektuje kao krug nešto što nije krug.



Slika 5: prikaz odsjaja na elementu

Neuronske mreže, generalno, mogu da se posmatraju kao univerzalni aproksimatori funkcija. Konvolucione neuronske mreže su posebna vrsta neuronskih mreža koje su se izuzetno dobro pokazale u primenama mašinske vizije i obrade slike.

Glavna prednost u odnosu na prethodni metod je to što se mreža može istrenirati da detektuje objekte i pri nesavršenim uslovima. Potrebno je pri treningu ubaciti primere elemenata kao što je element na slici 5. Ideja je napraviti dovoljno velik i raznolik set za treniranje mreže da promene osvetljenja, odsjaji, promena kamere, pozicije kamere na robotu uopšte ne utiču na njen rad.

Pravljenje seta podataka za treniranje može biti dugotrajan i mukotrpan posao. Što veći set podataka, to će mreža bolje raditi. Kako bi se ubrzao proces pravljenja seta podataka za treniranje mreže, za obeležavanje elemenata na slikama za trening iskorišćeno je rešenje prethodnom metodom, a potom su ručno ispravljene greške, ako postoje. Postupak bi se sastojao iz toga da se uslika više različitih setova, sa velikim brojem slika, svaki sa karakterističnim uslovima (pozicija kamere, osvetljenje, sama kamera, ugao pod kojim se osvetljava itd.). Prvo se kalibrišu parametri koji dobro rade pod određenim uslovima, zatim se obeleže slike za set podataka za treniranje. Nakon toga se promene uslovi rada, ponovo se kalibrišu parametri, i ponovi se postupak. I tako više puta, da bi se pokrio što veći broj slučajeva. Treniranje neuronske mreže se vrši sa svim označenim slikama zajedno, na neki način su sve kalibrisane funkcije aproksimirane jednom velikom mrežom matrica.

Konkretna konvoluciona neuronska mreža koja se koristi u ovom rešenju je mreža Tiny-Yolo. To je mreža koja na svom izlazu daje koordinate detektovanih objekata od interesa. Jedan prolaz kroz mrežu je dovoljan da se detektuje veliki broj objekata [3]. Mreža je istrenirana da detektuje plave, crvene i zelene pakove sa takmičenja.

5. Merenje i poredenje rezultata

Uspešnost detekcije objekata se najčešće meri pomoću *Mean Average Precision* metrike [3]. Najvažnija karakteristika ove metode evaluacije je da uzima u obzir, pored uspešnih pogodaka (*True Positive*), i da li je detektovano nešto što ne treba (*False Positive*), da li je propušteno nešto što je trebalo biti detektovano (*False negative*).

Ukupno 200 slika je iskorišćeno za testiranje obe metode. Bitno je naglasiti da te slike nisu učestvovala u treniranju neuronske mreže. Rezultati su prikazani u tabeli 1.

	Crveni	Zeleni	Plavi	mAP
HSV metoda	59%	77%	71%	69%
Tiny-Yolo	87%	92%	90%	90%

Tabela 1: rezultati merenja

Ono što je najveći problem kod prvog metoda je odsjaj i postojanje objekata iste boje koji su sličnog oblika kao objekti od interesa. Na slici 6 se može uočiti iz kog razloga je detektovanje crvenih elemenata najteže za obe metode. Postoji slovo „d“ na terenu koje je gotovo istog oblika i veličine kao i elementi i uvek biva detektovano.



Slika 6: problematičan objekat na terenu

6. Zaključak

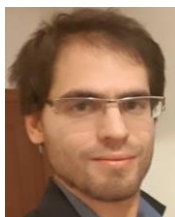
Uređaj je uspešno napravljen i zadovoljava potrebe takmičenja. Prva metoda za detekciju objekata, korišćenjem izolacije boje u HSV domenu i proveravanjem kontura, radi dobro za slučajeve kada nema odsjaja i kada nema objekata iste boje, koji su približno kružnog oblika. Neuronske mreže mogu biti obučene da se izbore i sa ovim slučajevima. Rezultati merenja pokazuju da obučena mreža radi značajno bolje kada se testira na slikama koje su uzete u raznim uslovima osvetljenja.

Najveći problem u vezi sa neuronskom mrežom je to što je množenje velikih matrica sporo. Dok prvi metod može da obradi sliku u par stotina milisekundi, da bi se obradila slika pomoću neuronske mreže potrebno je oko 3 sekunde (oba na Raspberry Pi 3 uređaju). Glavno unapređenje bi bilo korišćenje nekog procesora koji može da paralelizuje i ubrza računanje matrica poput Nvidia Jetson Nano modula.

7. Literatura

- [1] Richard Szeliski, „*Computer Vision: Algorithms and Applications*“, September 2010.
- [2] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, „*You Only Look Once: Unified, Real-Time Object Detection*“ University of Washington, 2016.
- [3] https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173 (pristupljeno u septembru 2019.)

Kratka biografija:



Miloš Arbanas rođen je u Sremskoj Mitrovici 1993. god. Završio je Mitrovačku gimnaziju u Sremskoj Mitrovici 2012. godine. Fakultet tehničkih nauka je upisao 2012. godine, a diplomirao na smeru Mikroročunarska elektronika 2016. godine. Iste godine upisuje master studije na smeru Embedded sistemi i računari.

PROJEKTOVANJE AKTIVNOG NISKOPROPUSNOG FILTRA U 0,18 μm CMOS TEHNOLOGIJI**DESIGN OF CMOS ACTIVE LOW-PASS FILTER IN 0,18 μm CMOS TECHNOLOGY**Kristina Nikolić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTOTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu je projektovan aktivni Batervortov niskopropusni filter šestog reda u 0,18 μm CMOS tehnologiji. Topologija se sastoji iz kaskadne veze tri filtera drugog reda. Filter drugog reda je projektovan korišćenjem Salen-Ki topologije koja se sastoji iz dvostepenog operacionog pojačavača, dva otpornika i dva kondenzatora. Projektovani filter ima maksimalno slabljenje u propusnom opsegu 9,5217 mdB i minimalno slabljenje u nepropusnom opsegu 50 dB. Granična učestanost je 52,3 kHz, a potrošnja kola 2,27 mW.

Ključne reči: CMOS tehnologija, integrisana kola, niskopropusni filter, operacioni pojačavač.

Abstract – The sixth-order active Butterworth low-pass filter has been designed in 0.18 μm CMOS technology in this paper. The topology consists of three second-order filters. The second-order filter is designed by using the Sallen-Key topology containing two stage operational amplifier, two resistors and two capacitors. The designed filter has a maximum passband attenuation of 9.5217 mdB and a minimum stopband attenuation of 50 dB. The cutoff frequency is 52.3 kHz and the power consumption is 2.27 mW.

Keywords: CMOS technology, integrated circuits, low-pass filter, operational amplifier.

1. UVOD

Aktivni filteri su električna kola koja se koriste za potiskivanje šuma ili neželjenih delova signala koji se dovodi na ulaz filtera [1]. Filteri takođe mogu da se koriste i za poravnanje signala na izlazu digitalno-analognih konvertora [1]. Spektralne karakteristike ulaznog signala se menjaju pri prolasku kroz filter u zavisnosti od uloge za koju je filter namenjen [2].

Osnovne aproksimacije prenosne karakteristike filtera su Batervortova aproksimacija, Čebiševljeva prvog reda, Čebiševljeva drugog reda, Eliptička i Beselova aproksimacija [3]. Najčešće korišćene topologije za realizaciju filtera (na frekvencijama reda MHz) u integrisanim tehnologijama su Gm-C topologija, biquad topologija, Salen-Ki topologija (engl. *Sallen Key topology*) i topologija sa višestrukim povratnim spregama (engl. *Multifeedback topology*).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Jelena Radić, docent

Cilj ovog rada jeste projektovanje filtera propusnika niskih učestanosti (NF) koji potiskuje komponente ulaznog signala na višim učestanostima, odnosno u nepropusnom opsegu. Korišćena je Batervortova aproksimacija prenosne funkcije za dobijanje maksimalno ravne karakteristike u propusnom opsegu. Za realizaciju filtera je odabrana Salen-Ki topologija, pre svega zbog njene jednostavnosti, ali i mogućnosti realizacije filtera višeg reda prostim kaskadnim vezivanjem.

2. PROJEKTOVANJE DVOSTEPENOG OPERACIONOG POJAČAVAČA

Na slici 1 je prikazana električna šema dvostepenog operacionog pojačavača koja je projektovana u programskom paketu *Cadence* korišćenjem 0,18 μm UMC CMOS tehnologije. Prvi stepen predstavlja diferencijalni pojačavač koji obezbeđuje veliko pojačanje, a drugi stepen je stepen sa zajedničkim sorsom koji obezbeđuje veći opseg napona na izlazu i dodatno pojačanje. Proračun početnih dimenzija tranzistora i vrednosti komponenti kola je urađen na osnovu postupaka prikazanih u [4].

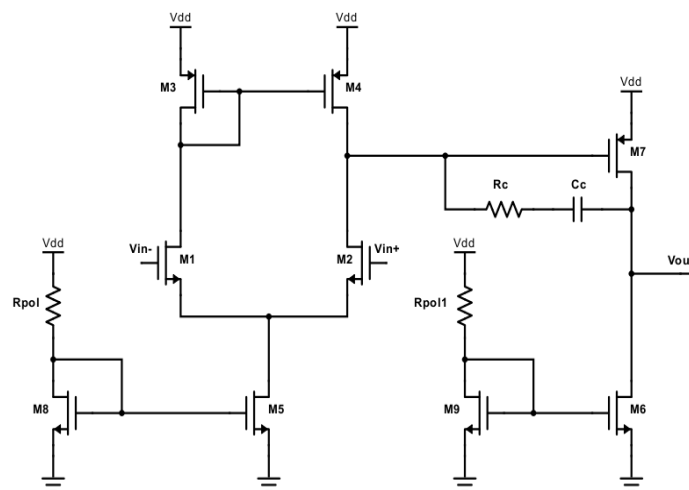
U toku simulacija se pokazalo da pojačavač ima bolje karakteristike ukoliko ova dva stepena imaju odvojene polarizacije zbog ograničenja korišćenih modela tranzistora koji za širinu tranzistora dozvoljavaju samo celobrojne umnoške vrednosti 5 μm , pa se ne može postići optimalna polarizacija ako se koristi isto strujno ogledalo.

2.1. Specifikacije parametara operacionog pojačavača

Zahtevi, odnosno granične vrednosti osnovnih parametara koje operacioni pojačavač treba da ispuni su sledeće:

- $A_{vdB} \geq 60 \text{ dB}$,
- $GBW > 50 \text{ MHz}$,
- $CMIR \sim 0.6 \text{ V} - 1.6 \text{ V}$,
- $SR \geq 10 \text{ V}/\mu\text{s}$,
- $P_{diss} \leq 1 \text{ mW}$,
- $PM \geq 70^\circ$,
- $V_{DD} = 1.8 \text{ V}$.

Parametar GBW predstavlja učestanost jediničnog pojačanja koja je jednaka proizvodu pojačanja i propusnog opsega (engl. *gain bandwidth* - GBW). Parametar CMIR je dozvoljeni opseg napona na ulazu (engl. *common mode input range* - CMIR). SR predstavlja brzinu promene signala na izlazu (engl. *slew rate* - SR), a PM faznu marginu (engl. *phase margin* - PM).



Slika 1. Električna šema operacionog pojačavača [4]

2.2. Rezultati simulacija operacionog pojačavača

Tokom analize naizmeničnih signala na ulaz operacionog pojačavača se dovodi naizmenični signal sa jednosmernom komponentom koja je neophodna za polarizaciju tranzistora na ulazu, a na izlazu je priključeno kapacitivno opterećenje koje ima vrednost 1 pF.

U DC analizi pojačavač se testira u zatvorenoj povratnoj sprezi i iz nje se mogu dobiti dozvoljeni opseg napona na ulazu kola i potrošnja.

Brzina promene signala na izlazu se dobija iz tranzijentne analize koja se takođe radi u zatvorenoj povratnoj sprezi pojačavača i umesto naizmeničnog signala, na ulaz kola se dovodi povorka pravougaonih impulsa.

Sve ostale karakteristike se dobijaju iz naizmenične (AC) analize.

Brojnim simulacijama i parametrizovanim analizama se dolazi do optimalnih dimenzija tranzistora i vrednosti elemenata kola koje su prikazane u tabeli 1. Korišćeni modeli tranzistora su ograničeni i imaju fiksnu dužinu kanala 0,18 μm , pa se svaki tranzistor u projektovanom kolu sastoji od redne veze dva tranzistora da bi dužina kanala bila 0,36 μm kako bi se izbegli neželjeni efekti kratkog kanala. Maksimalna širina korišćenih modela tranzistora je 105 μm , odnosno 21 prst (engl. *finger*) širine 5 μm . Neki tranzistori u projektovanom pojačavaču imaju

Tabela 1. Optimalne vrednosti elemenata operacionog pojačavača

Komponenta	Vrednost	Komponenta	Vrednost
$W_{1,2}$ [μm]	225,0	W_8 [μm]	105,0
W_3 [μm]	75,0	W_9 [μm]	70,0
W_4 [μm]	75,0	R_{pol} [k Ω]	11,0
W_5 [μm]	130,0	R_{pol1} [k Ω]	28,40
W_6 [μm]	135,0	R_c [k Ω]	1,0
W_7 [μm]	90,0	C_c [pF]	3,65

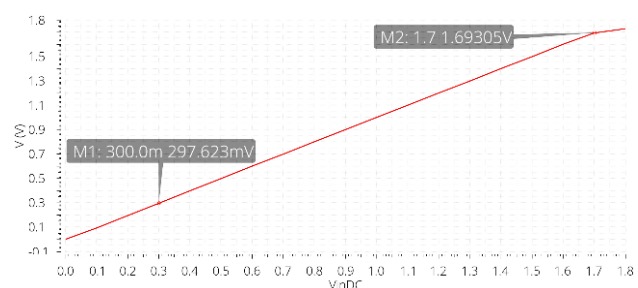
veću širinu, a to se postiže paralelnim vezivanjem tranzistora. Tako, na primer, tranzistori M1 i M2, koji

imaju širinu 225 μm se sastoje od paralelne veze tri tranzistora sa širinama od 75 μm .

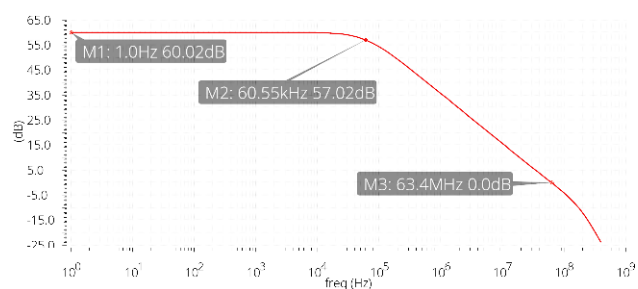
Na slici 2 je prikazana karakteristika zavisnosti izlaznog napona (V_{outDC}) od napona na ulazu u analizi jednosmernih signala. U linearnom režimu rada operacionog pojačavača, napon na izlazu treba da prati promene napona na ulazu i opseg napona koji zadovoljava taj uslov se naziva dozvoljeni opseg napona na ulazu, odnosno parametar $CMIR$. Vrednost parametra u ovom radu obuhvata napone od 0,3 V do 1,7 V.

Slika 3 prikazuje karakteristiku pojačanja (A_v) projektovanog dvostepenog operacionog pojačavača. Vrednost pojačanja pojačavača na nižim učestanostima iznosi 60,02 dB. Sa iste karakteristike se mogu očitati vrednosti učestanosti jediničnog pojačanja i 3 dB propusnog opsega pojačavača.

Propusni opseg projektovanog pojačavača se očitava na mestu gde je pojačanje opalo za 3 dB (57,02 dB) i iznosi 60,55 kHz.



Slika 2. Zavisnost izlaznog od ulaznog napona (DC)



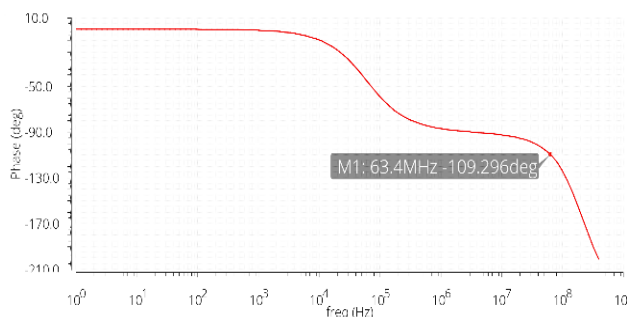
Slika 3. Pojačanje operacionog pojačavača

Jedinična učestanost (GBW) je učestanost na kojoj je pojačanje jednako jedinici, odnosno 0 dB u slučaju da je pojačanje izraženo u decibelima. Projektovani operacioni pojačavač ima vrednost parametra GBW od 63,4 MHz.

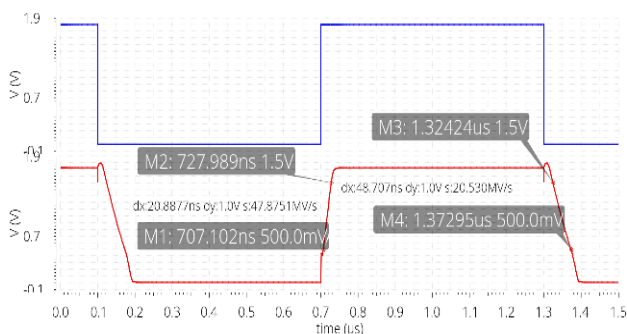
Na slici 4 je data fazna karakteristika projektovanog operacionog pojačavača. Fazna margina se dobija postavljanjem markera na dobijenoj jediničnoj učestanosti (63,4 MHz), očitavanjem faznog ugla i oduzimanjem očitane faznog ugla od 180°. Očitana vrednost faze je 109,295°, pa fazna margina iznosi 70,7°.

Na slici 5 su prikazani povorka pravougaonih impulsa koja je dovedena na ulaz operacionog pojačavača pomoću generatora implusnih signala i dobijeni signal na izlazu na osnovu kojih je određena vrednost parametra SR.

U slučaju projektovanog dvostepenog operacionog pojačavača brzina promene uzlazne ivice iznosi 47,88 V/μs, a brzina promene silazne ivice je 20,53 V/μs, što ispunjava definisane zahteve.



Slika 4. Fazna karakteristika operacionog pojačavača



Slika 5. Rezultati tranzijentne analize

3. PROJEKTOVANJE AKTIVNOG NISKOPROPUSNOG FILTRA

Prvi korak prilikom projektovanja filtra je određivanje reda filtra (n) i granične učestanosti filtra (f_n). Nakon toga se određuju koeficijenti prenosne funkcije filtra na osnovu zadatih karakteristika, dobijenog reda i granične učestanosti filtra. Ovi parametri se mogu dobiti korišćenjem programskog paketa *MATLAB*.

3.1. Specifikacije niskopropusnog filtra

Zahtevi koje filter treba da ispuni su:

- $W < 100$ kHz,
- $R_p < 1$ dB,
- $R_s < 50$ dB,
- *Maksimalno ravna karakteristika u propusnom opsegu.*

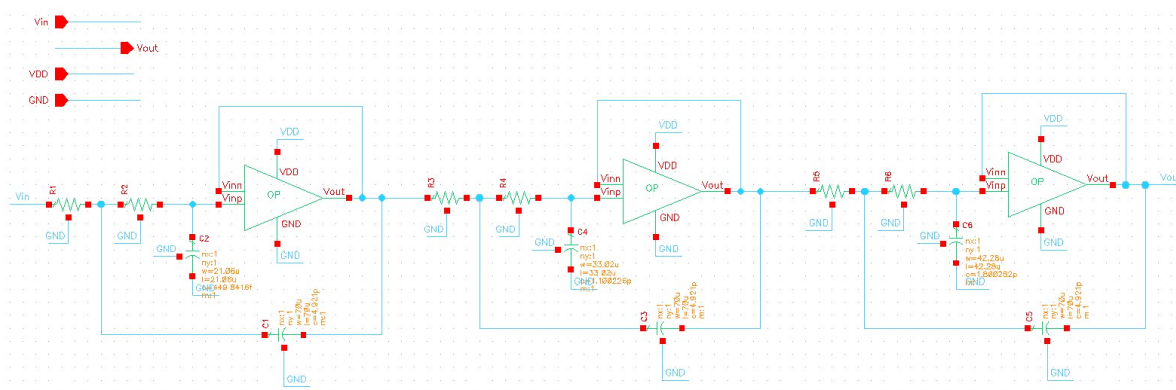
Širina prelazne oblasti W treba da bude manja od 100 kHz, slabljenje u propusnom opsegu (R_p) manje od 1 dB, a slabljenje u nepropusnom opsegu (R_s) manje od 50 dB. Maksimalno ravna karakteristika u propusnom opsegu se postiže korišćenjem Batervortove aproksimacije.

Red filtra se pronalazi funkcijom *buttord* i ima vrednost 6, a granična učestanost (f_n) ima vrednost 53,64 kHz, što je približno jednako 3 dB učestanosti projektovanog operacionog pojačavača (60,55 kHz).

S obzirom da je dobijen red filtra 6, on se sastoji od tri kaskadno vezana filtra drugog reda realizovana pomoću Salen-Ki topologije. Šema ovako realizovanog filtra je prikazana na slici 6 i projektovana je korišćenjem programskog paketa *Cadence*.

Da bi se dobili koeficijenti prenosne funkcije Batervortovog filtra koristi se funkcija *butter* iz programskog paketa *MATLAB*. Dobijena prenosna karakteristika ima polinom šestog reda u imeniocu, pa mora da se rastavi na tri prenosne funkcije sa polinomom drugog reda u imeniocu. Kada se polinom razloži, početne vrednosti otpornosti se biraju intuitivno, na osnovu iskustva, a početne vrednosti kondenzatora se računaju izjednačavanjem tri dobijene funkcije sa prenosnom funkcijom Salen-Ki topologije, koja ima oblik [3]:

$$A(s) = \frac{1}{R_1 R_2 C_1 C_2 s^2 + C_1 (R_1 + R_2) s + 1}, \quad (3.1)$$



Slika 6. Električna šema niskopropusnog filtra

gde indeksi otpornika i kondenzatora odgovaraju oznakama u prvom filtru u kaskadnom lancu prikazanom na slici 6.

3.2. Rezultati simulacija niskopropusnog filtra

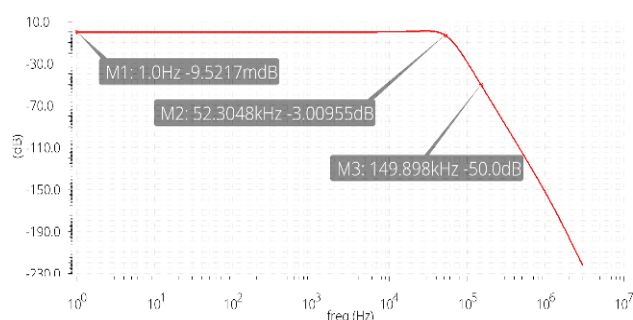
Na ulaz test kola projektovanog filtra se dovodi naizmenični signal, a na izlazu je priključeno kapacitivno opterećenje koje ima vrednost 1 pF, slično kao kod operacionog pojačavača.

Da bi se dobile optimalne vrednosti elemenata kola, koje su prikazane u tabeli 2 vrše se brojne simulacije i parametrizovane analize.

Na slici 7 je prikazana prenosna karakteristika projektovanog filtra. Markerom M1 je označeno slabljenje u propusnom opsegu i iznosi 9,4878 mdB. Marker M2 prikazuje frekvenciju na kojoj je signal oslabio za 3 dB, odnosno kraj propusnog opsega ili graničnu frekvenciju filtra koja iznosi 52,3 kHz (približno jednaka vrednosti 3 dB frekvencije operacionih pojačavača od kojih se filter sastoji).

Kraj prelazne oblasti i početak nepropusnog opsega predstavlja učestanost na kojoj slabljenje ima vrednost 50 dB (na osnovu postavljenih zahteva) i označena je markerom M3. Ova frekvencija ima vrednost 149,89 kHz, pa je širina prelazne oblasti 97,58 kHz.

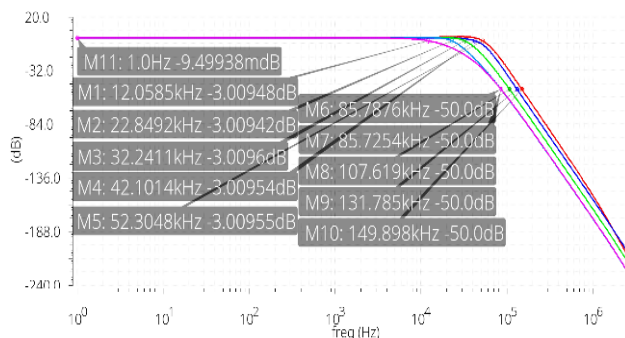
Simulacijama je utvrđeno da projektovani filter promene vrednosti odgovarajućih otpornosti i kapacitivnosti može imati različit propusni opseg, a da pri tome zadovoljava sve druge postavljene zahteve. Dakle, može se napraviti filter sa promenljivom širinom propusnog opsega korišćenjem mreže otpornika i kondenzatora čija vrednost se može digitalno upravljati, što nije zadatak rada, ali predstavlja podlogu za dalja istraživanja.



Slika 7. Prenosna karakteristika projektovanog filtra

Tabela 2. Optimalne vrednosti elemenata filtra

Komponenta	Vrednost	Komponenta	Vrednost
R_1 [MΩ]	1,2	C_1 [pF]	5,0
R_2 [MΩ]	1,2	C_2 [pF]	0,45
R_3 [MΩ]	1,0	C_3 [pF]	5,0
R_4 [MΩ]	1,0	C_4 [pF]	1,1
R_5 [kΩ]	800,0	C_5 [pF]	5,0
R_6 [kΩ]	800,0	C_6 [pF]	1,8



Slika 8. Prenosne karakteristike filtra

Na slici 8 su prikazane prenosne karakteristike filtra za širine propusnog opsega od 12 kHz do 52 kHz sa korakom 10 kHz, dobijene za različite vrednosti elemenata kola.

Sa slike 8 se vidi da je pojačanje u propusnom opsegu uvek manje od 10 mdB i da su prenosne karakteristike maksimalno ravne. Širine svih prelaznih oblasti su manje od 100 kHz.

4. ZAKLJUČAK

U ovam radu je opisan postupak projektovanja dvostepenog operacionog pojačavača, a zatim pomoću njega realizacija Batervortovog aktivnog niskopropusnog filtra korišćenjem Salen-Ki topologije. Projektovani pojačavač i filter moraju da zadovolje određene zahteve, na osnovu koji se na početku vrši proračun vrednosti elemenata kola, a zatim se radi optimizacija istih dok se ne dođe do željenih karakteristika. Projektovanje predstavlja dugotrajan proces sa ponekad neželjenim efektima koji sam proces simuliranja mogu vratiti i po nekoliko koraka unazad.

Namena rada je bila detaljna analiza postupka projektovanja topologije niskopropusnog filtra u UMC 180 nm CMOS tehnologiji, sa ciljem dobijanja što boljih karakteristika. Dat je predlog za unapređenje filtra pružanjem mogućnosti podešavanja njegovih karakteristika projektovanjem topologije sa promenljivim propusnim opsegom korišćenjem digitalno upravljivih otporničkih i kondenzatorskih mreža, pri čemu se zadržava vrednost pojačanja u propusnom opsegu i širina prelazne oblasti.

5. LITERATURA

- [1] B. Razavi, *RF Microelectronics*, Second Edition, Castleton, New York, USA, 2011.
- [2] H. Uhrmann, R. Kolm, H. Zimmermann, *Analog Filters in Nanometer CMOS*, Springer, Verlag Berlin Heidelberg, 2014.
- [3] P.V. Ananda Mohan, *VLSI Analog Filters*, New York, USA, 2013.
- [4] P. E. Allen, D. R. Holberg, *CMOS Analog Circuit Design*, Oxford University Press

Kratka biografija:



Kristina Nikolić rođena je u Pirotu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetika, elektronika i telekomunikacije uradila je 2019.god. kontakt: kristinanikolic@uns.ac.rs

AUTOMATSKO ODREĐIVANJE TEMA KNJIGA POMOĆU TEHNIKA ZA PROCESIRANJE PRIRODNOG JEZIKA**AUTOMATIC BOOK TOPIC DETECTION USING NATURAL LANGUAGE PROCESSING**

Vlada Đurđević, *Fakultet tehničkih nauka, Novi Sad*

Oblast - RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj - Ovaj rad bavi se analizom performansi LDA modela kreiranog sa ciljem određivanja tema koje se pojavljuju u nekom korpusu knjiga. Opisan je skup podataka sa kojim se radi kao i svi problemi koji se javljaju prilikom implementacije ovakvog modela. Detaljno su analizirana četiri glavna koraka kreiranja modela, preprocesiranje podataka, NER metoda, određivanje optimalnog broja tema i izbor konkretnog algoritma za implementaciju. Za svaki od koraka su demonstrirani različiti pristupi rešavanju problema koji se javljaju. Izvršena je evaluacija rezultata za svaki od ovih pristupa nakon čega je odabran optimalan pristup sa ciljem da čini sastavni deo krajnjeg modela.

Ključne reči: Latent Dirichlet Allocation, Named Entity Recognition

Abstract - This paper presents a performance analysis of an LDA model created for determining topics from a book corpus. A detailed analysis of four crucial steps regarding the implementation of the model is presented, data preprocessing, NER method, determining the optimal number of topics and choosing the best implementation algorithm. For each of the steps, a number of different methods for overcoming the problems that arise are demonstrated. The obtained results for each of the different methods are presented and discussed in detail. Finally, the optimal method is chosen to be a part of the resulting model.

Keywords: Latent Dirichlet Allocation, Named Entity Recognition

1. UVOD

Procesiranje prirodnog jezika (*Natural language processing* - NLP) predstavlja granu veštačke inteligencije koja se bavi interakcijom između ljudi i računara uz pomoć prirodnog jezika. Krajnji cilj NLP-a je mogućnost čitanja, razumevanja i upotrebe različitih ljudskih jezika.

Jedna od metoda koja se koristi prilikom procesiranja ljudskog jezika je LDA (*Latent Dirichlet Allocation*), sa ciljem modelovanja kolekcija dokumenata i određivanja tema koje su u njima javljaju. Svaka pojedinačna tema predstavlja distribuciju verovatnoće nekih reči, tj. pruža nam verovatnoću da će se u njoj pojaviti neki određen skup reči.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kovačević, vanr.prof.

Ideja na kojoj se temelji LDA jeste da se svaki dokument sastoji od skupa različitih tema a da se svaka od tih tema sastoji od skupa različitih reči. Ovaj master rad opisuje upotrebu i analizira rezultate dobijene uz pomoć LDA metode u okviru sistema za predikciju popularnosti knjiga [1]. Cilj ovakvog sistema je da pronade i analizira sve atribute od interesa koji mogu uticati na popularnost neke knjige.

Pod popularnošću se smatra rejting knjige, tj. ocena koju je knjiga dobila na sajtu *Goodreads* [2]. LDA metod je korišćen prilikom analize naslova i sažetka knjige radi određivanja tema koje se pojavljuju u samoj knjizi. Takođe je upotrebljena NER (*Named Entity Recognition*) metoda radi uklanjanja ličnih imena iz tekstova koji čine sažetke knjiga. Isprobano je nekoliko različitih implementacija LDA modela pa je nakon toga odabran model koji daje najbolje rezultate. Ovaj model je potom primenjen na analizu sažetaka kao i na analizu naslova same knjige pošto se iz rezultata moglo zaključiti da nema potrebe za kreiranjem dva različita modela.

2. SRODNI RADOVI

Postoji velik broj radova koji se bave određivanjem skupa tema koje se pojavljuju u nekoj kolekciji dokumenata. U ovoj sekciji su predstavljeni radovi koji su najviše uticali na pristupe rešavanju problema koji se javljaju pri svakom od koraka kreiranja modela.

Rad [3] najbolje ilustruje osnovni cilj jednog LDA modela i predstavlja osnovu na kojoj se temelji ovaj master rad. U njemu je implementirano klasterovanje i vizualizacija dokumenata uz pomoć LDA metode i samoorganizujućih mapa (*Self-Organizing Maps* - SOM). Autori ovog rada su implementirali LDA-SOM model i primenili ga na dva različita skupa podataka da bi evaluirali njegovu efektivnost. Korišćeni skupovi podataka su *20 Newsgroups*, koji se sastoji od 11,269 dokumenata i *Neural Information Processing Systems* (NIPS), koji se sastoji od 1500 dokumenata. Implementirani su samo osnovni koraci preprocesiranja podataka a evaluacija modela je izvršena upotrebom metrika za evaluaciju SOM.

Naredni rad [4] se bavi isključivo veličinom skupa podataka nad kojim će se obučavati LDA model, tačnije brojem potrebnih dokumenata za kreiranje dobrog LDA modela. Intuitivno, deluje da će veći obučavajući skup uvek rezultovati boljim krajnjim modelom ali to ne mora uvek da bude slučaj. U okviru ovog rada autori su došli do zaključka da je obučavajući skup koji se sastoji od oko 40.000 dokumenata i 100.000 različitih termina dovoljan za kreiranje kvalitetnog LDA modela.

U okviru rada [5] fokus je bio na pretprocesiranju podataka. Cilj rada je bio pronalaženje tema koje se javljaju u grupi tvitova. Autori ovog rada su implementirali 7 različitih koraka pretprocesiranja, neophodnih za funkcionisanje krajnjeg LDA modela, nakon čega se izgubila skoro trećina obuhvaćajućeg skupa.

Prilikom kreiranja LDA modela broj tema koje model treba da odredi u okviru datog korpusa mora biti unapred zadat. U navedenim radovima broj tema je određen ručno, na osnovu samog označenog skupa podataka ili nasumično. Nasuprot tome, rad [6] daje pregled metoda za automatsko određivanje optimalnog broja tema koje se pojavljuju u bilo kakvom skupu podataka. Jedna od najpopularnijih metoda koje se koriste sa ovim ciljem je računanje *perplexity*¹ vrednosti za niz modela sa različitim brojem tema a zatim odabir modela sa najnižom *perplexity* vrednošću.

Poslednji od radova koji će biti navedeni u ovom poglavlju se odnosi na vizualizaciju rezultata nekog LDA modela. Bez obzira na kvalitet samog LDA modela krajnji rezultati su samo skup brojeva koji predstavljaju verovatnoće tema. Takve rezultate je prilično teško analizirati a još teže prikazati korisniku na jednostavan način. U radu [7] autori predstavljaju potencijalno rešenje ovog problema, opisuju jedan od alata za vizualizaciju i poređenje rezultata modela.

3. METODOLOGIJA I ALATI

U ovoj sekciji se nalazi pregled svih metoda, algoritama, alata, okruženja i biblioteka korišćenih u okviru ovog rada.

3.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) predstavlja generativni probabilistički model neke kolekcije diskretnih podataka, poput tekstualnog korpusa. Sačinjava ga troslojni Bajesovski hijerarhijski model kod koga se svaki element neke kolekcije modeluje kao konačna kombinacija postojećih tema. Opšti cilj LDA modela je pronalaženje kratkog opisa za svaki element neke kolekcije, radi efikasnog procesiranja ogromnih kolekcija podataka, dok se u isto vreme očuvava esencijala statistička zavisnost među njima. LDA model podrazumeva da se svaki dokument W unutar korpusa D može dobiti prateći generativni proces definisan na sledeći način.

1. Odabere se $N \sim$ na osnovu Poasonove distribucije
2. Odabere se $\theta \sim$ na osnovu Dirihleove distribucije
3. Za svaku od N reči w_n :

3.1 Odabere se tema $z_n \sim$ na osnovu Multinomialne distribucije

3.2 Odabere se reč w_n iz multinomialne verovatnoće $p(w_n|z_n, \beta)$

Kao posledica ovako definisanog generativnog procesa, dobija se sledeća verovatnoća za ceo korpus:

$$p(D|\alpha, \beta) = \prod_{d=1}^M \int p(\theta_d) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d$$

Parametri α i β su jedinstveni na nivou korpusa i pretpostavlja se da se sempluju samo jednom prilikom generisanja korpusa. Varijable θ_d su jedinstvene na nivou pojedinačnog dokumenta i sempluju se jednom za svaki

dokument, a varijable z_{dn} i w_{dn} su jedinstvene na nivou pojedinačnih reči i sempluju se jednom za svaku reč u svakom dokumentu.

3.2 Named Entity Recognition

Named Entity Recognition (NER) se koristi kao naziv za podkategoriju zadataka koje obavlja *Information Extraction* (IE), a koja se fokusira na prepoznavanje informacionih jedinica unutar nestruktuiranog teksta, poput imena ljudi, organizacija, lokacija, kao i datuma, vremena, i novčanih iznosa. Ključan problem koji NER sistemi treba da reše je identifikovanje unapred nepoznatih entiteta. Postoji nekoliko pristupa rešavanju ovog problema. U okviru ovog rada korišćena je alat koji implementira *Linear Chain Conditional Random Fields* model, koji spada u metode nadgledanog učenja.

3.3 NLTK

NLTK (*Natural Language ToolKit*) čini platformu za izgradnju *Python* programa koji se bave obradom običnog teksta. Pruža interfejse ka velikom broju različitih korpusa i leksičkih resursa, poput *WordNet*-a, kao i velik broj biblioteka za procesiranje teksta. Prilikom implementacije sistema, NLTK platforma je najviše korišćena za pretprocesiranje teksta, uz pomoć metoda tokenizacije, stemovanja, lematizacije i POS tagovanja.

3.4 MALLET

MALLET (*MAchine Learning for Language Toolkit*) je *Java*-baziran paket za procesiranje prirodnog jezika, klasifikaciju dokumenata, određivanje tema koje se pojavljuju u dokumentima, ekstrakciju informacija a sadrži i velik broj drugih metoda mašinskog učenja za obradu teksta.

3.5 Stanford NER

Stanford NER predstavlja *Java*-baziranu implementaciju *Named Entity Recognition* sistema, koja u svom centru sadrži *Linear-chain Conditional Random Fields* model. U sklopu ove implementacije postoje tri različita klasifikaciona modela: 7-class, 4-class i 3-class model. U okviru ovog rada korišćen je 3-class model jer obuhvata entitete neophodne za rešavanje problema predstavljenog u radu a u isto vreme pruža značajno bolje rezultate od druga dva modela.

3.6 Multidimensional scaling

Multidimensional scaling (MDS) predstavlja tehniku za nelinearnu redukciju dimenzionalnosti. Podatke koji se nalaze u visoko dimenzionom prostoru, predstavlja preko matrice udaljenosti. Zatim pokušava da očuva taj kriterijum nakon projekcije podataka u prostor niže dimenzije. Svodi se na optimizacioni problem minimizacije funkcije koja predstavlja gubitak informacija.

3.7 Jensen-Shannon divergencija

Jensen-Shannon divergencija (JSD) predstavlja metod za merenje sličnosti između dve distribucije verovatnoće. Ova divergencija je simetrična pa se u opštem slučaju može koristiti kao mera udaljenosti, a takođe je i konačna što je čini pogodnom za računanje sličnosti između distribucija verovatnoće.

3.6 pyLDAvis

PyLDAvis predstavlja *Python* biblioteku kreiranu sa ciljem da pomogne korisnicima prilikom interpretacije

¹ *Perplexity* predstavlja meru koja određuje koliko dobro neki statistički model opisuje određen skup podataka, gde manja vrednost ukazuje na bolji statistički model.

tema koje se pojavljuju kao rezultati nekog modela za određivanje tema. Uzima potrebne informacije iz obučenog LDA modela i predstavlja ih u vidu interaktivne web aplikacije koja omogućava globalan pogled na sve teme, kao i detaljnu analizu svake pojedinačne teme i svih termina koji se u njoj pojavljuju. Teme se predstavljaju u 2D prostoru upotrebom MDS tehnike i Jensen-Shannon divergencije.

4. EKSPERIMENTALNA POSTAVKA

U ovoj sekciji će biti predstavljeni problemi rešavani u okviru ovog rada. Fokus rada je prvenstveno bio na kreiranju optimalnog LDA modela, i posebna pažnja je posvećena problemima koji treba da se reše prilikom kreiranja modela.

4.1 Problemi

Kreiranje modela je predstavljeno preko četiri glavna koraka: pretprocesiranje podataka, NER metoda, određivanje optimalnog broja tema i izbor konkretnog algoritma za implementaciju.

Prvi problem koji treba da se prevaziđe prilikom kreiranja LDA modela predstavlja pretprocesiranje podataka. U osnovi podaci predstavljaju nestruktuirani tekst i treba da prođu dosta koraka pretprocesiranja pre nego što model može da ih upotrebi na željeni način. U okviru rada su upoređeni rezultati dobijeni različitim vidovima pretprocesiranja. Prvo je kreiran model bez pretprocesiranja podataka, zatim model koji implementira metod stemovanja a za kraj model koji implementira metod lematizacije i POS tagovanja. Takođe postoje neki koraci koji su zajednički za sve metode, poput tokenizacije reči, prebacivanje velikih slova u mala, izbacivanja znakova interpunkcije i izbacivanja reči koje se izrazito često pojavljuju.

Naredni korak predstavlja NER metoda, koja takođe u nekoj meri predstavlja vid pretprocesiranja podataka. U okviru ovog rada se koristila *Stanford NER 3-class* implementacija ove metode, koja pronalazi i identifikuje imena ljudi, organizacija i lokacija u nestruktuiranom tekstu. Razlog korišćenja NER metode proističe iz činjenice da postoji velik broj knjiga koje imaju više nastavaka a glavni likovi ostaju isti. Zbog ovoga, imena tih likova mogu da predstavljaju reči od velikog značaja za temu koja je dominantna u tim knjigama. Posledica ove činjenice je da ukoliko se pojavi neka knjiga sa istim imenima likova postoji veća verovatnoća da će dobiti istu temu, iako ime lika ne bi trebalo da utiče na temu knjige. Najkompleksniji korak prilikom kreiranja bilo kog LDA modela predstavlja određivanje optimalnog broja tema koji taj model treba da odredi. U okviru rada je isproban broj tema određen na osnovu sličnih radova, broj tema određen na osnovu žanrova knjige i broj tema određen na osnovu koherentnosti² krajnjeg modela. Poslednji korak kreiranja modela predstavlja izbor algoritma za implementaciju. Isprobane su dve različite implementacije, Gensim i MALLET implementacija i pokazalo se da MALLET implementacija pruža bolje rezultate..

4.2 Rezultati

Za problem pretprocesiranja podataka najbolji rezultati su postignuti upotrebom metode lematizacije i POS tagovanja, ali treba imati u vidu da je ovaj pristup značajno vremenski zahtevniji od metode stemovanja. NER metoda je takođe rezultovala poboljšanim modelom ali se javlja isti problem vremenske zahtevnosti. Isto tako treba primetiti da je problem koji se rešava ovom metodom specifičan za domen problema definisanog u ovom radu, dok se u velikom broju drugih slučajeva on uopšte ne javlja. Najkompleksniji korak kreiranja modela je predstavljalo određivanje optimalnog broja tema. U ovom radu je cilj bio da se ostvari jedna od osnovnih ideja LDA modela, predstavljanje velike količine podataka uz pomoć relativno kratkog opisa. Radi ostvarivanja ovog cilja je pronađen najmanji mogući skup tema koje mogu da opišu dokumente iz korpusa na osnovu vrednosti za koherentnost modela, ali u opštem slučaju čak i dosta veći broj tema se ne bi mogao klasifikovati kao pogrešan. Konačno, za samu implementaciju modela je odabrana MALLET biblioteka na osnovu rezultata koherentnosti modela, iako je Gensim implementacija značajno brža. Rezultujuća koherentnost za MALLET implementaciju bila je 0.426 dok je za Gensim implementaciju bila samo 0.311. Teme određene krajnjim, optimalnim modelom su prikazane na slici 1.



Slika 1: Teme optimalnog modela

5. ZAKLJUČAK

Problem rešavan u okviru ovog rada je implementacija i analiza performansi LDA modela kreiranog sa ciljem određivanja tema koje se pojavljuju u nekom korpusu knjiga. Model je implementiran uz pomoć programskog jezika Python uz korišćenje NLTK, Gensim i MALLET programskih biblioteka kao i Stanford NER i pyLDAvis alata.

Model je kreiran sa ciljem da pronađe što manji skup tema koji je dovoljan da na ispravan način predstavi sve knjige iz obučavajućeg korpusa. Teme određene uz pomoć ovog modela su predstavljene kao vektor vrednosti verovatnoća i zatim korišćene u okviru sistema za predikciju popularnosti knjiga, gde se analizom rezultata algoritma koji određuje važnost atributa moglo primetiti da ovako određene teme u velikoj meri utiču na razultujuću popularnost neke knjige.

Detaljno su analizirana četiri glavna koraka kreiranja modela, pretprocesiranje podataka, NER metoda, određivanje optimalnog broja tema i izbor konkretnog algoritma za implementaciju. Pretprocesiranje je obavljeno upotrebom Gensim i NLTK biblioteka i predstavlja osnovni korak prilikom implementacije modela ali takođe

² Koherentnost predstavlja meru za evaluaciju kvaliteta tema koje odredi neki model, a ustvari određuje koliko je neka tema pogodna za interpretaciju od strane čoveka. U okviru ovog rada korišćena je Gensim implementacija mere koherentnosti koja predstavlja implementaciju četvorodelnog procesa definisanog u [8].

i vremenski najzahtevniji. NER metoda je implementirana uz pomoć Stanford NER alata i fokus je bio isključivo na pronalaženju i uklanjanju ličnih imena iz korpusa dokumenata. Demonstrirani su različiti pristupi određivanju optimalnog broja tema, a za potrebe ovog rada je odabran pristup koji uključuje računanje vrednosti koherentnosti modela. Kao poslednji korak kreiranja modela su predstavljena dva različita načina implementacije modela, upotrebom Gensim i MALLET biblioteka, kao i razlika među njima.

Dalji pravci istraživanja uključuju analizu uticaja još različitih metoda pretprocesiranja, poput predstavljanja karakterističnih fraza u obliku bigrama ili trigrama, modifikaciju NER metoda sa ciljem uklanjanja imena lokacija i organizacija, izmenu samih algoritama za implementaciju sa ciljem optimizacije za konkretan problem definisan u ovom radu.

6. LITERATURA

- [1] O. Hrnjaković, V. Đurđević, D. Bujiša, *Predikcija popularnosti knjiga*, Fakultet tehničkih nauka, Novi Sad, 2019
- [2] Goodreads. (2018). [online] Dostupno na: <https://www.goodreads.com/>
- [3] J. Millar, G. Peterson, M. Mendenhall, *Document Clustering and Visualization with Latent Dirichlet Allocation and Self-Organizing Maps*, Air Force Institute of Technology, 2009
- [4] S. Crossley, M. Dascalau, D. McNamara, *How Important Is Size? An Investigation of Corpus Size and Meaning in both Latent Semantic Analysis and Latent Dirichlet Allocation*
- [5] D. Alvarez-Melis, M. Saveski, *Topic Modeling in Twitter: Aggregating Tweets by Conversations*, Massachusetts Institute of Technology, Cambridge, MA, USA, 2016
- [6] W. Zhao, J. Chen, R. Perkins, Z. Liu, W. Ge, Y. Ding, W. Zou, *A heuristic approach to determine an appropriate number of topics in topic modeling*, 2015
- [7] J. Murdock, C. Allen, *Visualization Techniques for Topic Model Checking*, Program in Cognitive Science, Indiana University, USA
- [8] M. Roder, A. Both, A. Hinneburg, *Exploring the Space of Topic Coherence Measures*, Leipzig University, R&D, Unister GmbH, Martin-Luther University, Germany

Kratka biografija:

Vlada Đurđević je rođen 24.04.1995. godine u Novom Sadu, Republika Srbija. Osnovnu školu „23. Oktobar” u Sremskim Karlovcima završio je 2010. godine. Nakon toga upisuje opšti smer u gimnaziji „Svetozar Marković” u Novom Sadu. Srednju školu završava 2014 godine. Iste godine se upisuje na Fakultet tehničkih nauka u Novom Sadu, odsek Elektrotehnika i računarstvo, smer Računarstvo i automatika. Školske 2016/17 se opredeljuje za usmerenje Primenjene računarske nauke i informatika, a potom školske 2017/18 za usmerenje Inteligentni sistemi. Zvanje diplomirani inženjer elektrotehnike i računarstva stekao je 2018. godine, sa prosečnom ocenom 8.72. Iste godine upisao je master akademske studije na smeru Računarstvo i automatika na Fakulteta tehničkih nauka u Novom Sadu. Uža specijalnost na master studijama bila je inteligentni sistemi. Položio je sve ispite predviđene planom i programom.

**GENERISANJE FORMI ZA POSLOVNE PROCESSE KORIŠĆENJEM MAGRITTE
RAZVOJNOG OKVIRA****FORMS GENERATION FOR BUSINESS PROCESSES USING MAGRITTE
FRAMEWORK**Nina Miladinović, *Fakultet tehničkih nauka, Novi Sad***Oblast – SOFTVERSKO INŽENJERSTVO**

Kratak sadržaj – U radu su opisani poslovni procesi, karakteristike formi za unos podataka i programski jezik Smalltalk i njegov dijalekt Pharo. Opisani su razvojni okviri neophodni za implementaciju rešenja. Predstavljena su postojeća rešenja uz akcenat na delovima koji se tiču generisanja formi za unos podataka. Predstavljeno je implementirano rešenje, prikazane su klase koje su nosioci rada aplikacije i prikazano je dobijeno rešenje.

Ključne reči: Poslovni procesi, generisanje formi, Pharo programski jezik, Magritte razvojni okvir

Abstract – The thesis describes business processes, characteristics of data entry forms and the programming language Smalltalk and its dialect Pharo. Frameworks needed to be implemented into the solution are described. Existing solutions are presented, with an emphasis on parts concerning the generation of data entry forms. The implemented solution is presented, the classes that are the carriers of the application work are shown and screenshots of developed application are shown.

Keywords: process engine, generating forms, smalltalk – Pharo, Magritte framework

1. UVOD

Poslovni proces predstavlja niz aktivnosti koje su potrebne da se izvrše kako bi se postigao jasno definisani cilj. Ove aktivnosti mogu biti izvršene na razne načine, od strane različitih učesnika. Ljudi kao učesnici i dalje su neophodni u izvršavanju određenih zadataka i to je deo koji još uvek ne može da bude automatizovan. Ovo dovodi do zahteva za korisničkim interfejsom koji će omogućiti korisnicima da izvršavaju potrebne zadatke. Zadatak ovog rada jeste projektovanje web aplikacije za upravljanje zadacima korisnika u poslovnim procesima. Ovo zahteva i logovanje korisnika, pregled i odabir zadataka i generisanje formi za popunjavanje podacima. Rešenje je potrebno implementirati u Pharo¹ programskom jeziku uz upotrebu Magritte² razvojnog okvira. Ono treba da predstavi proširenje namenjeno NewWave³ softveru, treba da predstavi proširenje namenjeno NewWave softveru,

NAPOMENA:

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

koji u Pharo programskom jeziku implementira funkcije potrebne za praćenje i upravljanje poslovnim procesima.

2. TEORIJSKE OSNOVE

U ovom poglavlju se nalazi opis teorijskih osnova na kojima je zasnovan ovaj rad.

2.1 Poslovni procesi

Svaki proizvod ili usluga koju kompanije nude zapravo je proizvod niza aktivnosti – poslovni proces. Kako se teži da se svaki problem prebaci u domen informatike, tako se i za poslovne procese našlo rešenje u oblasti pod nazivom Business Process Management (skraćeno BPM). Svaki proces predstavlja niz aktivnosti koje imaju tačno jedan ulaz, ali mogu imati više izlaza. Na izlazu se proizvodi nova vrednost za krajnjeg korisnika. Redosled kojim će se izvršavati aktivnosti unutar procesa jeste bitan.

Životni ciklus poslovnog procesa predstavlja sve faze kroz koje on prolazi od prvog trenutka pa dalje kroz razvoj i primenu. Prva faza u životnom ciklusu jesu dizajn i analiza poslovnog procesa. Druga faza je konfigurisanje i ona se dešava samo u slučaju kada implementacija rešenja zahteva korišćenje softverskih sistema za upravljanje poslovnim procesima. Izvršavanje je treća faza životnog ciklusa koja predstavlja realno izvršavanje poslovnog procesa pokretanjem instanci, u svrhu ostvarivanja poslovnih ciljeva kompanije. Četvrta faza u životnom ciklusu poslovnog procesa je evaluacija, odnosno ocena funkcionalnosti poslovnog procesa.

Iz ugla IT sektora, sistem za upravljanje poslovnim procesima je softverski sistem koji definiše, kreira i upravlja izvršavanjem radnih procesa (workflow) korišćenjem odgovarajućeg softvera, koji je u sposoban da interpretira definiciju procesa, obezbedi interakciju sa učesnicima radnog procesa i obezbedi (gde je to potrebno) korišćenje IT alata i aplikacija [1]. U okruženju i procesima u kome su promene česte, fiksni redosled izvršavanja operacija kod aplikacija je neželjena karakteristika. Zato upotreba sistema za upravljanje poslovnim procesima omogućava laku izmenu sistema u skladu sa novim zahtevima i neophodnim promenama. Problem koji ovakvi sistemi donose jeste pitanje raspodele zadataka, jer se kod korisnika stvara osećaj da ga softversko odlučivanje preopterećuje. Takođe, ekspertskim korisnicima se mora pružiti fleksibilnost da se ne bi osećali sputano, ali im se ipak ne sme ostaviti prostor za narušavanje opšteg toka procesa.

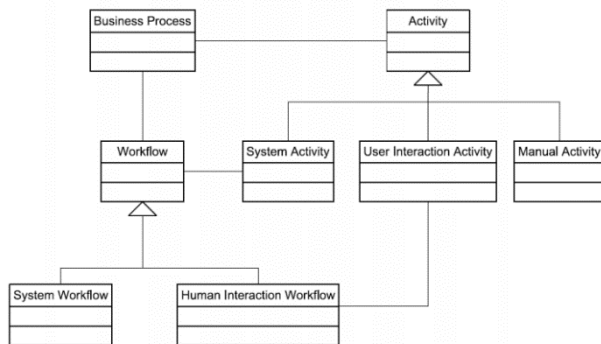
¹ <https://pharo.org/>

² <https://github.com/magritte-metamodel/magritte>

³ <https://github.com/skaplar/NewWave>

2.1.3 Meta-model poslovnih procesa

Meta-model poslovnih procesa je dat na slici 1. Da bi se postigao željeni cilj, u poslovnim procesima se moraju koordinisano izvršavati aktivnosti koje mogu biti sistemske, interaktivne (korisničke) i manuelne.



Slika 1. Meta-model poslovnog procesa [1]

Meta-model procesa definiše osnovne elemente modela (elementi modela su instance elemenata meta-modela). Njega čine model procesa, čvorovi i grane.

2.2 Forme za unos podataka

Prilikom definisanja formi za unos podataka, potrebno je voditi računa o pronalaženju minimalnog skupa funkcija potrebnih za izvršenje zadatka. Ovakav način definisanja formi ne zahteva pamćenje veće količine informacija i ubrzava proces učenja. Komponentu koja će se upotrebiti za unos podataka za određeno polje je potrebno pažljivo odabrati.

Na primer, u slučaju potrebe za unosom većeg teksta treba odabrati drugačiji element u odnosu na potrebu unosa samo nečijeg imena. Elementi koji se u praksi koriste su sledeći: *textField*, *passwordField*, *textArea*, *checkBox*, *radioButton*, *selectionList*, *comboBox* i *label*.

3. SMALLTALK – PROGRAMSKI JEZIK

Smalltalk je objektno orijentisani programski jezik pomoću kog se može napisati bilo koja desktop ili web aplikacija. On je osmišljen ranih sedamdesetih godina u Xerox-ovom centru za istraživanje *Palo Alto Research Center* (PARC).

1980. godine je *Smalltalk-80* verzije 1 postala prva verzija ovog jezika koja je data u javnu upotrebu, ali samo određenom broju kompanija poput *Apple*, *Hewlett-Packard* i *UC Berkley*.

Tri godine kasnije, *Smalltalk-80* verzije 2 je dat na korišćenje široj programerskoj javnosti i ovo je verzija na koju se generalno misli pri pomenu *Smalltalk* jezika. *American National Standards Institute* je 1998. godine ratifikovao zvaničnu verziju *Smalltalk*-a na kojoj se zasnivaju savremene implementacije. Dok u drugoj polovini devedesetih godina kompanije IBM i *ObjectShare* radile na komercijalnom razvoju *Smalltalk*-a, nekoliko *open-source* dijalekata *Smalltalk*-a (između ostalih i *Pharo*), objavljeni su i dobili su značajan udeo na tržištu. Ipak, tokom 2000-ih, rast *Smalltalk*-a je zaustavljen, ali danas uživa u preporodu zahvaljujući uspehu razvojnih okvira poput *Seaside* i *AIDA/web*-a [2].

3.1 Pharo programski jezik

Pharo je objektno orijentisani programski jezik i moćno okruženje koje je fokus stavilo na jednostavnost i neposredni odgovor na svaku liniju koda. On podržava programiranje „uživo“: neposrednu manipulaciju objektima i programskim kodom u okviru pokrenutog programa. *Pharo* ima 2 karakteristična principa: „sve je objekat, pa čak i klasa i poruka“ i „sve se dešava slanjem poruka objektu“ [3]. Poruka predstavlja nameru, ono što treba da se desi. Metode implementiraju ono kako treba da se desi. Novi objekat može biti kreiran slanjem poruke jednog objekta ka drugom.

3.2 Seaside razvojni okvir

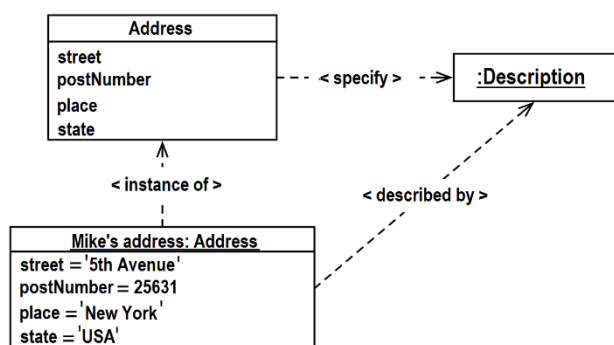
Seaside je razvojni okvir koji omogućava lako razvijanje naprednih i dinamičnih web aplikacija uz korišćenje *Smalltalk*-a. Klase unutar *Seaside*-a se mogu nasledivati i tako prilagođavati individualnim potrebama. *Seaside* se zasniva na mogućnost stvaranja nezavisnih komponenti koje se iznova mogu koristiti i uklapati za prikaz u web pregledaču. Svaka komponenta je odgovorna za svoj prikaz, stanje i tokove kontrole. *Seaside* nudi i mehanizam povratnog poziva (*callback*) koji klikom na odgovarajući deo komponente poziva izvršavanje određenog dela koda. Velika prednost *Seaside*-a je debugovanje aplikacije uz pomoć veoma moćnog dinamičkog alata, kao i mogućnost izmene koda dok je server još aktivan – takozvano „on the fly“ [4]. Komponenta u *Seaside*-u može da ima interakciju sa drugim komponentama na dva načina. Prvi je da može da ugradi sadržaj i funkcionalnost drugih komponenti u sopstvenu web stranicu. Drugi način interakcije je da može da pozove druge komponente i time im omogući da preuzmu web stranicu dok ne vrate rezultat glavnoj komponenti.

3.3 Magritte razvojni okvir

Magritte je razvojni okvir za opis meta-podataka. Pomoću *Magritte*-a se objektima dodeljuju opisi, a mogu se dobiti *Seaside* komponente koje imaju uključenu validaciju. Štaviše, *Magritte* opisuje sam sebe, omogućavajući automatsko generisanje *meta-editora* koji se može prilagoditi i iskoristiti za izgradnju web aplikacije. Ključna ideja koja stoji iza *Magritte*-a je sledeća: potrebno je da postoji jedan objekat sa skupom atributa i opisom tih atributa, na osnovu kojih će se automatski kreirati alati koji obrađuju opise i eventualno kreiraju i *Seaside* komponente. Program će interpretirati opise date u klasi i to na jedan od više načina: kao upit za bazu podataka, kao generički interfejs ili kao *Seaside* komponentu [4]. Primer klase koja predstavlja model i koja sadrži potreban *Magritte* opis dat je na slici 2.

Unutar klase, dodaju se metode čiji nazivi po konvenciji počinju sa „description“. Ove metode se koriste za definisanje opisa različitih entiteta ili atributa. U listingu 1 metoda klase *Address*-a kreira objekat *Magritte* opisa koji ima labelu „Street“, prioritet i dva pristupa atributu *Street*.

Opisi su centar *Magritte*-a i sadrže mehanizam za pristup vrednostima (*accessors*) i ograničenja vrednosti (*conditions*). Jednom kada je objekat opisan, *Seaside* komponenta se može dobiti tako što se objektu pošalje poruka „Object >> asComponent“.



Slika 2. Objekat koji sadrži opis nad klasom [4]

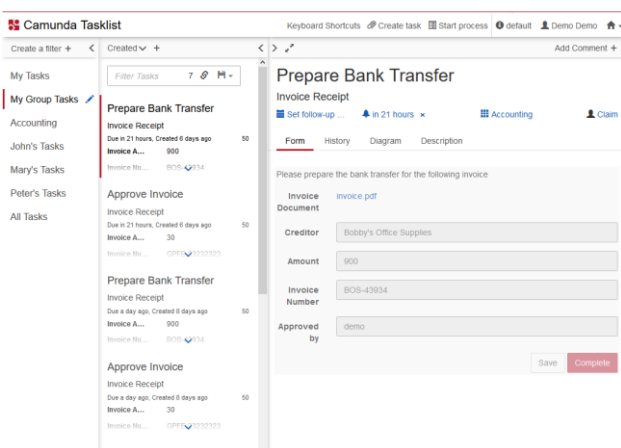
```

Address >> descriptionStreet
^ MStringDescription new
accessor: #street;
label: 'Street';
priority: 100;
yourself
  
```

Listing 1. Metoda *descriptionStreet* klase *Address-a*

4. POSTOJEĆA REŠENJA

Kako bi se bolje razumeli zahtevi postavljeni zadatkom generisanja formi za poslovne procese, poželjno je imati uvid u neka već postojeća rešenja.



Slika 3. Camunda Tasklist A aplikacija

4.1 Camunda

Camunda⁴ je razvojni okvir zasnovan na Java-i koji podržava BPMN notaciju za poslovne procese i automatizaciju procesa. On se koristi kroz Java biblioteku koja je odgovorna za izvršavanje BPMN 2.0 procesa. Ima nezahtevno POJO (Plain Old Java Object) jezgro, a koristi se i relacionom bazom podataka. ORM (Object-relational mapping) je podržan uz pomoć MyBatis razvojnog okvira [7]. Camunda se sastoji od nekoliko delova za upravljanje poslovnim procesima, među kojima je Camunda Tasklist A (web aplikacija za upravljanje korisničkim zadacima) najrelevantnija za ovaj rad. Postoje dve različite vrste formi koje krajnji korisnici mogu primeniti za rešavanje korisničkog zadatka ukoliko se forma generiše od strane Camunda web aplikacije: generisane forme zadataka i ugrađene forme zadataka

(HTML fajl/stranica ugrađena u projekat) [5]. Na slici 3 je prikazana jedna izgenerisana forma u okviru Camunda Tasklist A aplikacije.

4.2 Activiti

Activiti je razvojni okvir za upravljanje poslovnim procesima pisan u Java programskom jeziku i može izvoditi poslovne procese opisane pomoću BPMN 2.0 notacije. Tehnički zahtevi koje ima Activiti su Java runtime i Apache Tomcat⁵, iako se zapravo može koristiti bilo koji drugi web container jer je oslonac na servletima. Activiti pruža pogodan i fleksibilan način za dodavanje formi za unos podataka u poslovnim procesima. Podržava dve strategije za rad sa formama: prikaz ugrađene forme i prikaz eksterne forme.

4.3 jBPM

jBPM je alat za izgradnju poslovnih aplikacija koje pomažu u automatizaciji poslovnih procesa i odluka. Poslovna aplikacija može se definisati kao automatizovano rešenje, izgrađeno u odabranim okvirima i sa mogućnostima koji implementiraju poslovne funkcije ili poslovne probleme. Napisan je potpuno u Java sistemu i radi na bilo kojoj JVM-i (Java Virtual Machine) i dostupan je i u Maven Central⁶ repozitorijumu.

5. IMPLEMENTACIJA REŠENJA

NewWaveFieldUserTask je proširenje (plug-in) za NewWave Process Engine⁷. NewWave se razvija na Katedri za informatiku Fakulteta tehničkih nauka u Novom Sadu kao softversko rešenje otvorenog koda. Poput Camunda Tasklist A ili Activiti Explorer-a, ima namenu da omogućiti krajnjim korisnicima pristup odabiru i rešavanju zadataka, a rešavanje podrazumeva prikaz forme za unos podataka.

Programski jezik korišćen za implementaciju aplikacije je Pharo, a razvojno okruženje Pharo Virtual Machine. Rešenje je implementirano korišćenjem Magritte i Seaside razvojnih okvira. Za stilizovanje izgleda u web prenteru, korišćen je Bootstrap wrapper za Seaside. Komunikacija sa NewWave okruženjem se odigrava preko REST servisa, a uz pomoć razvojnog okvira Zinc HTTP Components⁸. NeoJSON⁹ razvojni okvir se koristi za mapiranje JSON opisa na odgovarajuće klase.

Na slici 4 prikazan je dijagram klase. U okviru projekta korišćene su Seaside klase „WASession“ i „WAComponent“. „WASession“ nasleđena je od strane „FUSession“ klase i pomoću nje se prati sesija. Druga klasa „WAComponent“ predstavlja Seaside komponentu koja služi za prikaz u web pregledaču. Nju nasleđuju klase „FUApplicationRootComponent“, „FUHeader“ i „FUScreenComponent“. „FUDOCClassDescription“ predstavlja klasu za koju je potrebno izgenerisati formu i ona je zapravo zadatak koji je prosleđen korisniku. Sadrži

⁵ <http://tomcat.apache.org/>

⁶ <https://search.maven.org/>

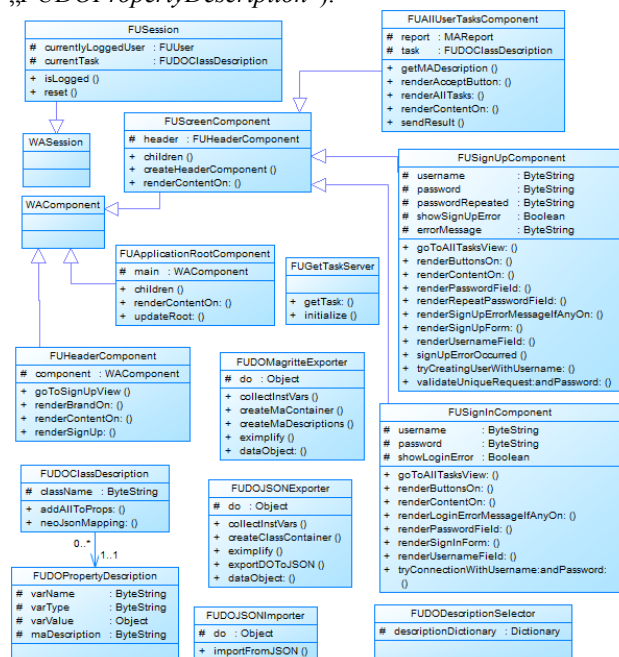
⁷ <https://github.com/skaplar/NewWave>

⁸ <https://ci.inria.fr/pharo-contribution/job/EnterprisePharoBook/lastSuccessfulBuild/artifact/book-result/Zinc-HTTP-Client/Zinc-HTTP-Client.html>

⁹ <https://ci.inria.fr/pharo-contribution/job/EnterprisePharoBook/lastSuccessfulBuild/artifact/book-result/NeoJSON/NeoJSON.html>

⁴ <https://camunda.com/>

atribute „className“ (naziv klase) i „properties“ (niz polja odnosno atributa klase koji su tipa „FUDOPROPERTYDESCRIPTION“).



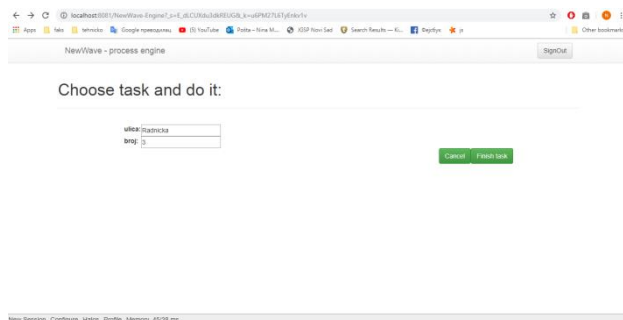
Slika 4. Dijagram klasa NewWaveFieldUserTask proširenja

Sadržaj klase „FUDOCClassDescription“ se dostavlja u vidu JSON opisa kao odgovor na REST zahtev upućen serveru u okviru NewWave-a (prikazan na listingu 26). Ovaj opis se prosleđuje klasi „FUDOJSONImporter“ slanjem JSON opisa kako parametra u poruci „importFromJSON“. Povratna vrednost ove metode je „FUDOCClassDescription“.

Klasa „FUDOJSONExporter“ definiše metode koje se koriste za konstruisanje JSON opisa. Klasa navedene u dijagramu kao „FUDOMagritteExporter“ sadrži metode za generisanje Magritte opisa („createMaDescription“) i kontejnera („createMaContainer“). Metoda „createMaContainer“ kao povratnu vrednost ima objekat (prikazan na slici 16) koji nasleđuje Seaside klasu „WAComponent“. Klasa „FUGetTaskServer“ sadrži metodu čijim se pozivanjem formira i upućuje REST zahtev ka NewWave serveru. Prikaz izgenerisane forme za prikupljanje podataka dat je na slici 5.

6. ZAKLJUČAK

U radu su opisani poslovni procesi, date su smernice i podele koje se tiču formi za unos podataka, opisan je programski jezik Smalltalk i njegovi najznačajniji razvojni okviri. Predstavljena su postojeća rešenja, kao i njihov pristup problemu generisanja formi. Prezentovano je rešenje, objašnjene najznačajnije klase i metode i dat je primer korišćenja aplikacije. Kreirano rešenje može biti korisno za NewWave Process Engine napisan u Pharo programskom jeziku, ali da bi ovo rešenje našlo primenu u praksi, ono mora biti značajno prošireno. Takođe, korišćenje Magritte razvojnog okvira, donelo je velike prednosti i olakšanja u generisanju polja u formi.



Slika 5. Prikaz forme za popunjavanje

Rad u ovoj oblasti ukazao je na veliki broj mesta i prilika za proširenje. Generisanje formi može biti odrađeno na više načina, ne samo upotrebom Magritte opisa. Ipak, Magritte opisi pružaju veliku pomoć pri generisanju upita za bazu podataka. Takođe, aplikacija za generisanje formi je napravljena isključivo za korisnike, ali se može proširiti tako da se njom mogu koristiti i operatori procesa ili administratori.

7. LITERATURA

- [1] dr Miroslav Zarić, Slajdovi sa predavanja iz predmeta „Upravljanje poslovnim procesima“, Katedra za informatiku, FTN, Novi Sad, 2019. godine
- [2] Jon Penland, *Smalltalk: The Original Object-Oriented Programming Language?*, <https://www.whoishostingthis.com/resources/smalltalk/>, 2019. godine
- [3] Pharo MOOC, <http://mooc.pharo.org/#week1>, 2019. godine
- [4] Stéphane Ducasse, Lukas Renggli, C. David Shaffer, Rick Zaccane with Michael Davies, *Dynamic Web Development with Seaside*, 2019. godine
- [5] Felix Müller, Mike Winters, *Introducing Embedded Forms Generator: A Camunda Modeler Plugin for Easier Embedded Form Building*, <https://blog.camunda.com/post/2018/08/modeler-plugin-embedded-form-generator/>, 2018. godine

Kratka biografija:



Nina Miladinović rođena je 1995. godine u Novom Sadu. Završila je Osnovnu školu „Đorđe Natošević“ i gimnaziju „Jovan Jovanović Zmaj“ u Novom Sadu kao nosilac diploma „Vuk Karadžić“. 2014. godine, upisuje se na studije na Fakultetu tehničkih nauka, odsek Računarstvo i automatika. Za dalje usmerenje u okviru pomenutog odseka, odlučuje se za smer Primenjene računarske nauke i informatika. Diplomirala je 4. septembra 2018. godine i zatim se upisuje na master studije na istom odseku i smeru, a za podsmer je odabrala Elektronsko poslovanje.

STEGOSPLOIT U SISTEMIMA ZA PRENOS I PRIKAZ DIGITALNIH SLIKA

STEGOSPLOIT IN SYSTEMS FOR TRANSFERING AND DISPLAYING IMAGES

Aleksandra Mišić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – Primenjeno softversko inženjerstvo

Kratak sadržaj – U ovom radu predstavljena je vrsta napada pod nazivom stegosploit. Opisan je postupak kreiranja malicioznog fajla, načini za isporuku fajla žrtvi, kao i načini odbrane od ove klase kiber pretnji.

Ključne reči: stegosploit, slike, sigurnost, steganografija, JPEG, PNG, polyglot

Abstract – This paper presents the class of attacks called stegosploit. We describe how the malicious files are created and delivered, as well as possible security controls for mitigating this class of cyber threats.

Keywords: stegosploit, images, security, steganography, JPEG, PNG, polyglot

1. UVOD

Stegosploit je tehnika sakrivanja potencijalno malicioznog JavaScript koda u digitalne slike. Osmislio ju je istraživač sigurnosti u oblasti računarskih sistema, Saumil Shah i predstavio javnosti 2015. godine [1]. Naziv tehnike se zasniva na pojmu steganografija, koji označava skup tehnika za prikrivanje poruka unutar drugih poruka, slika, videa.

Stegosploit čine sledeći koraci:

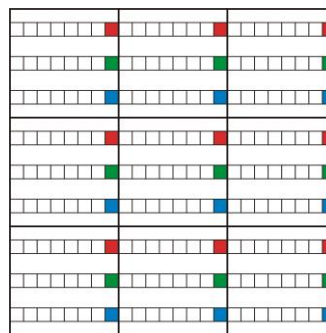
1. Upisivanje koda u piksele slike uz pomoć odabranog steganografskog algoritma. U ovom radu, biće prikazan postupak za slike formata – *Joint Photographic Experts Group* (JPEG) i *Portable Network Graphics* (PNG);
2. Kreiranje poliglotskog fajla (eng. *polyglot*) koji u isto vreme predstavlja validan *Hypertext Markup Language* (HTML) dokument kao i validnu sliku;
3. Isporuka stegosploita žrtvi.

2. UPISIVANJE KODA U PIKSELE SLIKE

Uprkos postojanju sofisticiranog steganografskog algoritma, pod nazivom F5, koji čuva integritet kodirane poruke prilikom osnovnih transformacija slike, stegosploit koristi jednu od jednostavnijih tehnika (ili njenu varijaciju) – sakrivanje u najmanje značajne bitove (eng. *Blind Hide in Least Significant Bit* (LSB)) [2]. Razlog je potreba za konciznim i jednostavnim algoritmom, koji je u isto vreme sasvim dovoljno efikasan.

Digitalna slika se može predstaviti kao niz piksela. Svaki piksel sadrži najmanje tri kanala – crveni, plavi i zeleni (R, G, B). Neki formati slika (npr. PNG) podržavaju i

četvrti – alfa kanal (A), koji se koristi za nivo transparentnosti piksela [3]. Ukoliko se za primer uzme slika čiji je svaki kanal predstavljen uz pomoć osam bitova, slika se može razložiti u osam bit-slojeva. Svaki bit-sloj je kreiran tako što se zadrže vrednosti bitova na odgovarajućoj poziciji u nizu, za svaki kanal, dok se ostali bitovi anuliraju. Na primer, za nulti nivo zadržaću se svi bitovi najmanjeg značaja (eng. *least significant bit*, LSB) (slika 1), dok će se za sedmi nivo zadržati svi bitovi najvišeg značaja (eng. *most significant bit*, MSB).



Slika 1. Nulti bit-sloj

Algoritam *Blind Hide in Least Significant Bit* podrazumeva da se poruka pretvori u niz bitova i upiše u nulti bit-sloj, u jedan proizvoljni kanal ili u sva tri. Nakon upisivanja poruke, potrebno je pretvoriti niz piksela u JPEG ili PNG format, gde dolazi do problema kompresije podataka.

2.1. JPEG i PNG kompresije

Kompresija podataka (eng. *data compression*) je čest postupak u računarstvu koji podrazumeva transformisanje digitalnih podataka u oblik koji sadrži manje bitova od polaznog, najčešće u svrhu smanjenja zauzeća memorije i komunikacionih kanala, kao i trajanja transmisije fajlova [4]. Iako postoji veliki broj različitih formata i algoritama za kompresiju podataka, može se napraviti opšta podela na algoritme bez gubitaka podataka (eng. *lossless compression*) i algoritme sa izvesnim gubicima (eng. *lossy compression*). Algoritmi bez gubitaka podataka komprimuju podatke na način koji ima inverzan postupak za tačnu rekonstrukciju originalnih podataka. Sa druge strane, prilikom kompresije uz pomoć algoritama sa gubicima, dolazi do nepovratnog gubitka dela informacija, što implicira da nema garancije da će dekomprimovan fajl biti identičan polaznom.

U kontekstu digitalnih slika, tipičan primer kompresije sa gubicima jeste JPEG, dok PNG implementira algoritam kompresije bez gubitaka. JPEG definiše niz postupaka za kompresiju sirovih piksela, međutim najpoznatiji i najviše

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor prof. dr Imre Lendak.

korišćen metod jeste onaj koji definiše *JPEG File Interchange Format* (JFIF). Mreža piksela se deli na podmreže dimenzija 8x8 piksela, a zatim se nad svakom mrežom izvršava diskretna kosinusna transformacija (eng. *Discrete Cosine Transform* (DCT)). Ovaj postupak prima parametar koji definiše željeni nivo kvaliteta kompresije. Što je veća vrednost ovog parametra, aproksimacija piksela je blaža, a gubitak podataka manji.

2.2. Algoritam zapisivanja koda u JPEG sliku

Da bi se izbegao problem pogrešnog dekodiranja koda iz JPEG slike prouzrokovan kompresijom sa gubicima, postupak enkodiranja i dekodiranja, kompresije i dekompresije se iterativno ponavljaju. Ukoliko se ovom procesu proslede odgovarajući parametri, greška se smanjuje nakon svake iteracije. Drugim rečima, broj različitih piksela između slika na početku i kraju iteracije, u pravilnim okolnostima, teži nuli.

Uvode se sledeće oznake:

S – originalna JPEG slika

K – kod za enkodovanje u sliku S

C – sirovi pikseli

ENCODE – steganografska funkcija za enkodovanje

DECODE – steganografska funkcija za dekodiranje

JPG – funkcija koja transformiše sirove piksele u JPEG sliku (JPEG kompresija)

INV_JPG – funkcija koja transformiše JPEG sliku u sirove piksele

b – bit-sloj (0-7)

c – kanal (R, G, B, svi)

g – mreža kodiranja

q – kvalitet JPEG kompresije [0, 1]

Pre početka prve iteracije, slika S se učitava u HTML element, *canvas*, i dobijaju se sirovi pikseli C (listing 1). U prvoj iteraciji se K upisuje u C uz pomoć ENCODE funkcije, a zatim se, uz pomoć JPG funkcije, postojeći pikseli kompresuju u JPEG sliku – S'. U nastavku prve iteracije, slika S' se učitava u *canvas* – C'' i dekodira se kod iz C'' uz pomoć DECODE funkcije.

```
C = INV_JPG(S)
C' = ENCODE(K, C, b, c, g)
S' = JPG(C', q)
C'' = INV_JPG(S')
K' = DECODE(C'', b, c, g)
```

Listing 1. Deo iteracije upisivanja koda u JPEG sliku

Dalje se proverava da li postoji razlika između K i K'. Ukoliko ne postoji, postupak enkodiranja je završen i konačan rezultat je S'. Ukoliko postoji, potrebno je zabeležiti trenutni broj različitih piksela između C' i C'', da bi se u sledećoj iteraciji proverilo da li se broj smanjuje, tj. da li teži nuli. Ukoliko ne teži, potrebno je izmeniti neke od parametara b, g i q.

Ukoliko je potrebna sledeća iteracija, postupak se ponavlja tako što se u C'' ponovo enkodira K.

2.3. Parametri funkcija enkodiranja i dekodiranja

Funkcije ENCODE i DECODE treba da budu svesne bit-nivoa (b), kanala (c) i mreže kodiranja (g) koji se koriste i

ovi parametri moraju imati jednake vrednosti za obe funkcije.

Broj iteracija neophodan za uspešno enkodiranje zavisi od same biblioteke koja se koristi za JPEG kompresiju i dekompresiju (biblioteke su ugrađene u internet pretraživač). Mogu se sumirati faktori koji utiču na brzinu i verovatnoću konvergencije:

1. JPEG enkoderi primaju parametar kvaliteta kompresije (q) koji može imati vrednosti između 0 i 1. Vrednost 1 obezbeđuje najveći kvalitet krajnje slike i minimizaciju aproksimizacije, što ne garantuje kompresiju bez gubitaka, ali utiče na smanjenje broja iteracija i povećava verovatnoću konvergencije.

2. Ukoliko se bitovi koda ne upisuju u uzastopne bitove bit-sloja, već se preskače određen broj bitova u obe dimenzije (kvadratna mreža kodiranja - g), smanjuje se broj potrebnih iteracija. Pokazalo se da korišćenje mreža veličina 3x3 ili 4x4 donosi znatno bolje rezultate, dok dalje povećanje ovog broja ne pravi značajnu razliku.

3. Upisivanje koda u nulti ili prvi bit-sloj (b) najčešće ne dovodi do konvergencije. Slojevi 2 i 3 pokazuju bolje rezultate, dok je za više slojeve rezultat primetan za ljudsko oko.

2.4. Algoritam zapisivanja koda u PNG sliku

PNG slike koriste kompresiju bez gubitaka za skladištenje piksela. Iz tog razloga ne postoji problem gubitka steganografski enkodiranih poruka i nema potrebe za iterativnim postupkom opisanim u prethodnim poglavljima. I u ovom slučaju koristi se *canvas* i ugrađene funkcije PNG kompresije za kreiranje konačnog fajla.

Enkodovanje poruka u PNG slike ima određene prednosti u odnosu na enkodovanje poruka u JPEG slike:

1. Za proces enkodovanja, potrebna je jedna iteracija.

3. Moguće je enkodovanje u nulti bit-sloj, što ima najbolje vizuelne rezultate.

3. Enkodiranje i dekodiranje funkcioniše ispravno kada se koriste različiti pretraživači (eng. *cross browser encoding and decoding*).

3. KREIRANJE POLIGLOTSKOG FAJLA

U kontekstu računarskih sistema, poliglotski fajl predstavlja dokument koji zadovoljava strukturu dva ili više formata. Na primer, moguće je napraviti *Portable Document Format* (PDF) + ZIP poliglotski fajl koji se pravilno otvara u PDF čitaču kada mu je ekstenzija .pdf, odnosno uspešno eksportuje ako se ekstenzija promeni u .zip.

Stegosploit koristi HTML + JPEG/PNG poliglotske fajlove za isporuku napada žrtvi. Ukoliko se fajl otvori sa ekstenzijom .png ili .jpg, prikazaće se pravilna slika. Sa druge strane, ako se ekstenzija preimenuje u .html, u internet pretraživaču će se otvoriti validan HTML dokument.

3.1. JPEG/JFIF format

Kako bi se razjasnio postupak kreiranja HTML + JPEG poliglotskih fajlova, neophodno je upoznati se sa strukturom JPEG fajla, odnosno JPEG/JFIF formata. JFIF fajl se sastoji iz niza markera i marker segmenata. Svaki

marker se sastoji iz dva bajta od kojih prvi uvek ima heksadecimalnu vrednost FF, dok je drugi različit od FF i 00 i određuje tip markera [4]. Neki markeri su samostalni, dok većina označava početak marker segmenta koji ima strukturu kao na slici 2.

Marker	Segment	
FF xx	s1 s2	[Data bytes]

Slika 2. Struktura JFIF marker segmenta

Bajtovi s1 i s2 zajedno formiraju 16-bitni ceo broj koji označava dužinu *data bytes* dela, uključujući i dva bajta koje zauzima sam broj. Broj bajtova u nastavku iznosi: $265 * s1 * s2 - 2$. U zavisnosti od tipa markera, *data bytes* sadrži različite specifične podatke.

Svaka JFIF slika obavezno počinje SOI (*Start of Image*) markerom, nakon kog sledi JFIF APP0 marker segment koji sadrži različite parametre slike (slika 3).

SOI	FF D8						
JFIF-APP0	FF E0	Dužina	JFIF\0				
		Verzija	U	Xres	Yres	W	H
DQT	FF DB						
...	...	ostali segmenti					

Slika 3. Početak JFIF slike

3.2. Kreiranje HTML + JPEG poliglotskog fajla

Da bi se napravio HTML + JPEG poliglotski fajl, neophodno je ubaciti HTML deo fajla na odgovarajuće mesto u JFIF strukturu sa slike 3. Kako APP0 marker segment ne utiče na renderovanje slike, već služi aplikacijama kao izvor metapodataka, najbolje mesto za upis jeste u ovaj segment nakon poslednjeg bajta. Iako nije neophodno, poželjno je ponovo podesiti vrednost polja za dužinu APP0 segmenta.

Na slici 4. se vidi sadržaj JPEG slike nakon upisivanja HTML koda. Plavom bojom obeležena su izmenjena polja.

SOI	FF D8						
JFIF-APP0	FF E0	Dužina	JFIF\0				
		Verzija	U	Xres	Yres	W	H
		<html>					
		<head> ...deko... </head>					
		<body> ...img... </body>					
		</html>					
DQT	FF DB						
...	...	ostali segmenti					

Slika 4. Poliglotski HTML + JPEG fajl

3.3. PNG format

Svaka PNG slika počinje osmobajtnim potpisom: 89 50 4E 47 0D 0A 1A 0A (heksadecimalno). Nakon toga sledi niz *Four Character Code* (FourCC) bajtova (eng. *chunk*) gde svaki ima sledeću strukturu:

1. Dužina: 4 bajta, pozitivan ceo broj, predstavlja dužinu podataka.
2. Tip: 4 bajta, primeri su IHDR, IDAT, IEND...
3. Podaci: varijabilna dužina.
4. *Cyclic Redundancy Check* (CRC): 4 bajta, predstavlja vrednost koja se generiše na osnovu polja – tip i podaci [5].

Svaki PNG fajl mora da sadrži tačno jedan IHDR segment i tačno jedan IEND segment. Između njih može biti proizvoljan broj IDAT segmenata koji sadrže podatke o pikselima (slika 5).

PNG Zaglavlje	89	50	4E	47	0D	0A	1A	0A
IHDR	Dužina	IHDR		podaci segmenta				CRC
IDAT	Dužina	IDAT		piksel podaci				CRC
IDAT	Dužina	IDAT		piksel podaci				CRC
IDAT	Dužina	IDAT		piksel podaci				CRC
IEND	0	IEND		CRC				

Slika 5. Struktura PNG slike

3.4. Kreiranje HTML + PNG poliglotskog fajla

PNG fajlove je lakše proširiti nego JFIF fajlove, jer je dovoljno samo dodati još jedan segment u kom će se nalaziti HTML kod. Poseban tip segmenta je tEXt segment koji se može koristiti za skladištenje metapodataka o slici. HTML kod će biti upisan u jedan ovakav segment odmah ispod IHDR segmenta.

Na slici 6. se može videti PNG slika nakon što je u nju upisan HTML kod.

PNG Zaglavlje	89	50	4E	47	0D	0A	1A	0A
IHDR	Dužina		IHDR		podaci segmenta			CRC
tEXt	Dužina		tEXt		<html>			
	<head> ...deko... </head>							
	<body> ...img... </body>							
	</html>						CRC	
IDAT	Dužina		IDAT		piksel podaci			CRC
IDAT	Dužina		IDAT		piksel podaci			CRC
IDAT	Dužina		IDAT		piksel podaci			CRC
IEND	0		IEND		CRC			

Slika 6. Poliglotski HTML + PNG fajl

3.5. Dekodiranje i izvršavanje koda iz slike

Poliglotski fajlovi HTML + JPEG/PNG, dobijeni u prethodnom poglavlju, sadrže nekoliko ključnih komponenti:

1. sliku (JPEG ili PNG) sa steganografski enkodiranim JavaScript kodom;
2. JavaScript deko... koji dekodira kod iz slike i izvršava ga;
3. HTML strukturu koja prikazuje samo sliku i pokreće deko... po završetku učitavanja dokumenta.

Telo (eng. *body*) HTML dokumenta sadrži samo *img* element unutar *div* elementa:

```
<div>
    
</div>
```

Listing 2. HTML deo poliglotskog fajla

Treba primetiti da je vrednost *src* atributa „#“, što znači da će vrednost izvora slike biti *Unified Resource Locator* (URL) trenutnog dokumenta. Drugim rečima, slika koja sadrži enkodiran kod, koja je deo dokumenta, biće učitana u *img* element.

Po učitavanju dokumenta, pokreće se funkcija deko... koja preuzima sliku iz *img* elementa i kreira *canvas*. U njega učitava piksele slike, a zatim briše originalni *img* element i na njegovo mesto postavlja pomenuti *canvas*. Kod se dekodira iz piksela *canvas*-a, inverznim

steganografskim postupkom, uz korišćenje odgovarajućih parametara koji se poklapaju sa vrednostima parametara enkodiranja (kanal, bit-sloj, veličina mreže enkodiranja). Od niza dekodiranih bitova, kreira se JavaScript string.

Najjednostavniji način da se dobijeni kod izvrši je uz pomoć `eval()` funkcije, međutim, ova funkcija se često obeležava kao potencijalno opasan kod i privlači neželjenu pažnju. Drugi način da se izvrši JavaScript string jeste da se kreira anoniman Function objekat i da se njegovom konstruktoru prosledi string kao argument (listing 3).

```
function execute(string) {
    var a = setTimeout(new Function(string), 100);
}
```

Listing 3. Funkcija za izvršenje malicioznog koda

Ukoliko se ovako konstruisan poliglotski fajl prikaže u pretraživaču kao .html fajl, pri vrhu prikaza će biti vidljiv tekstualni višak koji potiče od zaglavlja slike (slika 7).



Slika 7. Višak teksta na vrhu prikaza

Da bi se rešio ovaj problem, u fajl je dodat *Cascading Style Sheets* (CSS) deo kojim se višak teksta sakriva. Kako bi se maksimalno povećala verodostojnost poliglotskog fajla kada se otvara kao .html dokument, kreiran je dodatan CSS kod koji obezbeđuje da prikazi izgledaju identično kada se poliglotski fajl otvori u pretraživaču sa ekstenzijom .html i ekstenzijom .jpg ili .png (listing 4).

```
<style>
  body {
    background: #0e0e0e;
    color: transparent;
  }
  canvas {
    margin: auto;
    display: block;
    position: absolute;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
  }
  img {
    visibility: hidden;
  }
</style>
```

Listing 4. CSS dodatak

CSS kod u ovom primeru je prilagođen često korišćenim internet pretraživačima iz listinga 5.

```
Google Chrome Version 76.0.3809.132
Firefox Quantum Version 68.0.2
Opera Version 62.0.3331.116
```

Listing 5. Internet pretraživači

3. DOSTAVLJANJE NAPADA ŽRTVI

Konačan poliglotski fajl moguće je dostaviti žrtvi na više načina. Jedna opcija je da napadač postavi fajl na server nad kojim ima kontrolu, a druga da postavi fajl na neki od sajtova koji služe za skladištenje fajlova. Primer ovakvog sajta je Filebin koji ne zahteva registraciju, dozvoljava *upload* HTML fajlova i pruža mogućnost kopiranja linka do dokumenta koji se otvara direktno u pretraživaču. Nakon što je postavio dokument na željeni sajt, napadač može da pošalje direktan link žrtvi uz pomoć nekog vida socijalnog inženjeringa. Napad se izvršava čim žrtva klikne na link.

4. NAČINI ODBRANE OD STEGOSPLOITA

Načini odbrane se mogu posmatrati iz ugla korisnika internet pretraživača i iz ugla programera koji kreiraju *web* aplikacije za *upload*-ovanje fajlova proizvoljnih formata. Korisnici treba da imaju svest da ne pristupaju sadržaju koji ne potiče od izvora od poverenja, što nažalost, često nije slučaj. Programeri koji kreiraju pomenute aplikacije, treba da budu pažljivi pri odabiru načina prikaza pojedinih tipova fajlova u internet pretraživačima. HTML fajlove nipošto ne treba prikazivati na način koji omogućava izvršavanje potencijalnih skripti unutar fajlova, već ih treba prikazati kao običan tekst.

5. ZAKLJUČAK

Stegosploit je vrlo efektna tehnika prikrivanja programskog koda unutar digitalnih slika formata JPEG i PNG. U ovom radu je prikazana tehnika ubacivanja neželjenog programskog koda u poliglotske JPEG i PNG fajlove. Na kraju rada su diskutovane mere bezbednosti koje omogućavaju zaštitu od potencijalnih napada kroz stegosploit.

6. LITERATURA

- [1] Stegosploit, <https://www.youtube.com/watch?v=np0mPy-EHII> (datum poslednjeg pristupa 14.09. 2019)
- [2] Dr. M. Umamaheswari, Prof. S. Sivasubramanian, S. Pandiarajan "Applied Optimal Control", 2010.
- [3] <https://stegosploit.info/> (datum poslednjeg pristupa 14.09. 2019)
- [4] Data compression, https://en.wikipedia.org/wiki/Data_compression (datum poslednjeg pristupa 14.09. 2019)
- [5] JPEG format podataka, https://en.wikipedia.org/wiki/JPEG_File_Interchange_Format (datum poslednjeg pristupa 14.09. 2019)
- [6] PNG format podataka, <https://www.w3.org/TR/2003/REC-PNG-20031110/> (datum poslednjeg pristupa 14.09. 2019)

Kratka biografija:



Aleksandra Mišić rođena je 1994. godine u Novom Sadu. Završila je Gimnaziju "Jovan Jovanović Zmaj" u Novom Sadu, 2013. godine. 2015. godine postala je stipendista u firmi "Schneider Electric DMS NS", a osnovne akademske studije završila je 2018. godine na Fakultetu tehničkih nauka u Novom Sadu.

**IMPLEMENTACIJA VIZUALIZATORA STRUKTURA PODATAKA KORIŠĆENJEM
WEBASSEMBLY PLATFORME****DATA STRUCTURE VISUALIZATION IMPLEMENTATION USING WEBASSEMBLY
PLATFORM**Rade Radišić, Srđan Popov, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Izučavanje struktura podataka može da bude apstraktno za studente koji su početnici u programiranju. Njihovo razumevanje memorijskog modela računara nije na dovoljno visokom nivou. Strukture podataka temelje se na teoriji grafova, koji se na lak način mogu vizualizovati i intuitivno razumeti. Tema ovog rada je softverska implementacija vizualizatora za strukture podataka.

Ključne reči: Vizualizacija struktura podataka, Rust, JavaScript, WebAssembly

Abstract – Studying of data structures can be abstract for beginner student programmers. Their understanding how computer memory works is vague. Data structures are based on graph theory, which could be easily visualized and understood intuitively. The subject of this paper is a software implementation of a data structure visualization.

Keywords: Data structure visualization, Rust, JavaScript, WebAssembly

1. UVOD

Strukture podataka su, zajedno sa algoritmima, fundamentalni principi računarskih nauka. Budući softverski inženjeri bave se njihovim izučavanjem na početku svog školovanja. Da bi se efikasno izučavali algoritmi i strukture podataka, neophodna su određena znanja iz diskretne matematike kao što su poznavanje formalne logike, skupovi, relacije, funkcije i teorija grafova [1]. Matematičke definicije i zapisi pretenduju da budu apstraktni za nekog kome je bitno da se sa pragmatičnog stanovišta upozna sa upotrebom algoritama i struktura podataka. Taj problem moguće je izbeći korišćenjem vizualizacija, kao pomoćnim sredstvom. Graf G je uređen par $(V(G), E(G))$ gde su $V(G)$ skup njegovih čvorova i $E(G)$, disjunktan od $V(G)$, skup njegovih ivica zajedno sa funkcijom učestalosti ψ_G , koja formira svaku ivicu grafa G kao (ne)uređen par čvorova grafa G [2].

U zavisnosti od tipa relacije, grafovi mogu biti usmereni i neusmereni. Jednostavnije objašnjenje grafova bi bilo da su čvorovi tačke, a svaka ivica je prikazana kao linija koja spaja dva čvora. Vizualni prikaz grafova može biti upotrebljen kao sredstvo za vizualizaciju struktura podataka, jer se one definišu kao usmereni grafovi.

Prikazivanje podataka je bitna problematika kojim se računarstvo bavi. U tu svrhu razvijana su mnogobrojna softverska rešenja. Njihova rasprostranjenost i upotrebljenost zavisila je od faktora kao što su multiplatformska podrška i otvorenost koda.

Moderno računarstvo fokusira se na izradu web aplikacija, jer se one koriste i prikazuju u internet pretraživačima, čiji je osnovni nefunkcionalni zahtev da budu multiplatformske, odnosno, da se izvršavaju na identičan način na različitim operativnim sistemima i uređajima.

Postoji više pristupa koji se primenjuju prilikom izučavanja programiranja i programskih jezika. Pristup „odozdo prema dole” bio bi da student uči programiranje u programskom jeziku visokog nivoa, kao što su *Python*, *Java* ili *JavaScript*. Ove i njima slične jezike odlikuje osobina postojanja velikog broja apstrakcija, koje uprošćavaju sam jezik i time ga čine lakšim za učenje i dalje korišćenje. Cena tih apstrakcija je da su autori jezika napravili izbor kako se rukuje primitivnim operacijama. Primer ovakve apstrakcije je postojanje mehanizma automatskog rukovanja memorijom, *Garbage Collection*. Pristup „odozdo prema gore” podrazumeva učenje jezika niskog nivoa¹ poput programskog jezika *C*. Prilikom rada sa ovakvim jezicima, programer mora da vodi računa o dinamički alociranoj memoriji, odnosno da je oslobodi kada ona više nije neophodna.

Prethodno opisani pristupi direktno utiču na studentovo razumevanje struktura podataka. U prvom slučaju, on će imati osnovno razumevanje rada sa strukturama podataka, dok će u drugom dobiti i tačnu sliku stanja u memoriji prilikom rada. Problem kod drugog slučaja nalazi se u tome što je potrebno više vremena pre nego što student ovlada principima koji stoje iza struktura podataka. Trenutno stanje je da nijedan od dva navedena pristupa nije dovoljno dobar, te su, u svrhu smanjenja jaza između pristupa učenju, autori razvili softverski alat pomoću kog se mogu vizualizovati strukture podataka pomoću vizualizacije usmerenih grafova.

2. PREGLED KORIŠĆENIH TEHNOLOGIJA**2.1. WebAssembly**

WebAssembly je mašinski format instrukcija za virtuelnu mašinu baziranu na stek arhitekturi [3]. Ona omogućava drugim programskim jezicima koji nisu *JavaScript* da se

NAPOMENA:

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

¹ Pod jezicima niskog nivoa uglavnom se misli na assembler i mašinski jezik računara. Pod ovom konotacijom, podrazumevaju se jezici u kojim se, pre svega, manuelno upravlja memorijom koju koriste, ali su za razliku od asemblerske grupe jezika, platformski nezavisni.

izvršavaju u okviru internet pretraživača [4]. *WebAssembly* podržavaju svi najpopularniji internet pretraživači, ali implementacija same virtuelne mašine postoji i van njih, te ju je moguće koristiti i za samostalne aplikacije. Koncept virtuelne mašine je veoma sličan bajtkodu *Java* virtuelne mašine (*JVM*), kao i *IL Assembler*-u *.NET Framework*-a [3].

Prednosti ovakve implementacije u odnosu na *JavaScript* sastoji se u tome što se, pre svega, radi o mašinskom formatu, dok je *JavaScript* kod tekstualan i interpretiran. To uključuje dodatne korake prilikom izvršavanja *JavaScript* koji uključuju leksar i parser jezika, dok je binarni format *WebAssembly*-ja odmah moguće izvršiti od strane virtuelne mašine.

WebAssembly koristi linearni model memorije. U pitanju je kontinualni blok bajtova, koji može biti deklarisan unutar *WebAssembly* modula, izveden iz modula ili uvezen iz platforme na kojoj se *WebAssembly* izvršava [3]. Na ovaj način *WebAssembly* i *JavaScript* međusobno komuniciraju. Važno je napomenuti da *WebAssembly* još uvek ne može direktno da manipuliše *Document Object Model (DOM)* stablom, te se kroz linearnu memoriju mogu proslediti podaci pomoću kojih će *JavaScript* kod ažurirati *DOM* stablo.

Virtuelna mašina *WebAssembly*-ja je veoma jednostavna, tako da ne podržava *heap* memoriju, odnosno nema dinamičku alokaciju objekata pomoću *new* operatora. Umesto toga, postoje četiri primitivna tipa koja se koriste [3]:

Tabela 1. Tipovi podataka u *WebAssembly*-ju

i32	32-Bit Integer
i64	64-Bit Integer
f32	32-Bit Floating-Point Number
f64	64-Bit Floating-Point Number

WebAssembly poseduje tekstualnu sintaksu koju je moguće pisati i prevoditi u binarni format. Ona umnogome podseća na *LISP* programski jezik. Uglavnom nije slučaj da programer direktno piše takav kod, već se jezici visokog nivoa prevode, a ciljani jezik je binarni format koji virtuelna mašina može da izvršava. Trenutna situacija je da jezici čije je upravljanje memorijom automatsko putem korišćenja mehanizma *Garbage Collection* nisu podržani, ili nisu podržani u potpunosti. Neki od takvih jezika koji su podržani su programski jezik *Go*, *C#* i *AssemblyScript*, koji je jezik dijalekat *TypeScript*-a. Nasuprot jezicima sa automatskim upravljanjem memorijom, jezici koji nemaju *Garbage Collection* kao što su programski jezici *C/C++* i *Rust* su podržani i najčešće se koriste prilikom rada sa *WebAssembly*-jem.

2.2. Rust programski jezik

Rust [4] programski jezik je moderan programski jezik. Pojavio se kao sistemski jezik, ali je ubrzo našao primenu i u drugim oblastima kao što je razvoj web aplikacija. Glavna osobina po čemu se razlikuje od drugih jezika je tzv. *ownership* sistem, koji predstavlja način manuelnog upravljanja memorijom, ali na način da su pravila implementirana u samom kompajleru, odnosno, ukoliko je neko od pravila za upravljanje memorijom prekršeno, program se neće kompajlirati.

Ownership sistem je alternativa automatskim upravljanjem memorijom kod drugih modernih programskih jezika, koji omogućava visoke performanse programa, dok sa druge strane, model upravljanja memorijom je urađen tako da uslovljava programera da piše memorijski bezbedan kod. Mana *ownership* sistema je što čini da kriva učenja/usvajanja jezika bude strmija, odnosno, potrebno je više vremena kako bi se shvatili razlozi principa koji iza njega stoje.

Podrška za rad sa *WebAssembly*-jem je raznovrsna za programski jezik *Rust*. *wasm-bindgen* je biblioteka i komandni alat u programskom jeziku *Rust* koja služi za interakciju između *WebAssembly* modula i *JavaScript*-a [5]. *web-sys* i *js-sys* su biblioteke pomoću kojih se iz *Rust* koda mogu pozivati funkcije u *JavaScript*-u za manipulaciju *DOM* stablom i funkcije standardne *Javascript* biblioteke, respektivno.

2.3. JavaScript biblioteka D3.js

Za vizualizaciju grafova korišćena je *D3.js JavaScript* biblioteka, kao jedno od najboljih i dokazanih rešenja za vizualizaciju podataka u *JavaScript*-u.

D3.js podržava prikazivanje podataka iz *JavaScript Object Notation (JSON)* objekata, kao i manipulaciju vektorskim formatom, odnosno *Scalable Vector Graphics (SVG)*. Velika pogodnost ovog vektorskog formata je što je podržan u svim internet pretraživačima.

D3.js funkcioniše po principu vezivanja podataka za *DOM* stablo i primenom podacima vođenim transformacijama nad njime [6]. Nije monolitan radni okvir koji pokušava da odgovori svakom zahtevu, već je fleksibilno rešenje, koje daje osnovu za manipulaciju *DOM* stablom, a na onome ko je koristi je da upotrebi različite web standarde zarad uspešne vizualizacije.

3. DATA STRUCTURE VISUALIZER TOOL

3.1. Pregled osnovnih funkcionalnosti

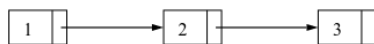
Alat je razvijen u svrhu lakšeg savladavanja rada sa strukturama podataka, na osnovu problematike objašnjene u uvodnom delu. Sastoji se od dva interfejsa:

- “Interfejs studenta”, grafički, interaktivni interfejs
- “Interfejs nastavnika”, tekstualni jezik specifičan za domen (*Domain Specific Language - DSL*), koji se koristi za definiciju strukture podataka

Interfejs studenta je interaktivna, grafička aplikacija u kojoj je moguće manipulirati strukturama podataka. Student može da izabere rad sa više različitih struktura podataka.

Deo za rad sa konkretnom strukturom podataka sastoji se iz dva dela: grafičkog prikaza trenutnog stanja strukture podataka i panela koji prikazuje moguće operacije. Na primeru spregnute realizacije steka sa slike, vidi se da postoje operacije *push* i *pop*.

Unosom konkretne vrednosti i pritiskom “Push” dugmeta, dodaće se vrednost na kraj steka, dok pritiskom na dugme “Pop” biće uklonjena vrednost sa kraja steka.



Slika 1. Primer spregnute realizacije steka

Interfejs nastavnika koristi se za definiciju struktura podataka koje će student koristiti. Nastavnik ima na raspolaganju sve neophodne operacije, a na njemu je da izabere koje operacije će i gde biti korišćene. Kao dodatni nivo fleksibilnosti, nastavnik može posebno nazvati izabrane operacije za datu strukturu podataka, kao i samu strukturu podataka.

Ova opcija dodata je iz razloga što različite literature iste strukture podataka i njihove operacije nazivaju različitim imenima, tako da se alat može prilagoditi terminologiji koja se koristi u nastavi. Konkretni primer koda jezika specifičnog za domen koji bi nastavnik koristio za definiciju strukture podataka je sledeći:

```
datastructure! {
  name          "Stack"
  type          "linear"
  implementation "linked"
  operation      "Push",   push_back
  operation      "Pop",    pop_back
  operation      "Clear",   clear
}
```

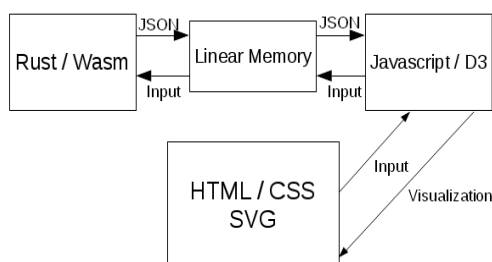
Primer 1. Primer definicije spregnute realizacije steka

Alat je upotrebljiv iz internet pretraživača, bez ikakve dodatne instalacije.

3.2. Arhitektura softverskog rešenja

Graf je modelovan pomoću struktura u programskom jeziku *Rust* tako da se sastoji od dva vektora: jedan za čvorove, drugi za ivice grafa. Svaka operacija nad grafom biće sračunata na *Rust* strani, odnosno, unutar *WebAssembly* virtualne mašine.

Nakon dobijanja novog stanja grafa, ono će biti konvertovano u *JSON*, što će dalje biti preneseno kroz linearnu memoriju do *JavaScript* koda, gde će *D3.js* funkcije biti zadužene za iscrtavanje grafa na osnovu dobijenih podataka. Funkcije za definisanje interfejsa su takođe na *Rust* strani, ali aktivno koriste funkcije iz *web-sys* biblioteke, kako bi se manipulisalo *DOM* stablom.



Slika 2. Arhitektura vizualizatora struktura podataka

Za komunikaciju između panela sa komandama i koda napisanog u *Rust*-u koristi se *requestAnimationFrame* metoda, čija je svrha da bude *event loop* i čeka događaje koji se generišu pritiskom na dugmad. Onog momenta kada se pojavi odgovarajući događaj, on će biti pročitao od strane *event loop-a*, na osnovu čega će biti izvršena odgovarajuća akcija, koja će rezultirati promenom stanja grafa, gde će se generisati novi *JSON* objekat, koji će biti poslat na iscrtavanje. Tako će biti promenjen prikaz grafa. Kako je prethodno pomenuto, tekstualni jezik specifičan za domen koristi se kako bi se definisala konkretna struktura podataka.

Primer 1. daje izgled jedne takve definicije. U pitanju je implementacija pomoću *Rust* makroa, gde će se makro transformisati u kod koji koristi *Builder* softverski obrazac [7]. Definicija strukture iz primera 1. prevešće se u sledeći kod u programskom jeziku *Rust*:

```
LinearStructureBuilder::new(
  "Stack".to_string()
).push_back("Push".to_string()).unwrap()
.pop_back("Pop".to_string()).unwrap()
.build();
```

Primer 2. Prevođenje koda definicije strukture podataka iz *Rust* makroa

Rezultat *wasm_bindgen* build-a je *WebAssembly* modul zapakovan kao *Node Package Manager (NPM)* paket. U tom obliku, moguće ga je uključiti u bilo koju modernu *JavaScript* frontend aplikaciju koja koristi *NPM* za upravljanje zavisnostima.

4. ZAKLJUČAK

Vizualizacija je moćno sredstvo koje pomaže pri usvajanju novih koncepata. Strukture podataka su usmereni grafovi, te se njihove vizualizacije mogu koristiti kao sredstvo za objašnjavanje struktura podataka, koje su, inače vrlo apstraktan pojam. Alat koji je predstavljen je još u ranom razvoju.

Dalji planovi razvoja sastoje se u implementaciji svih struktura podataka, kao i omogućavanje većeg nivoa interaktivnosti, što bi dalje rezultovalo podrškom za vizualizaciju algoritama, odnosno, njihovo izvršavanje „korak po korak”. Dobar primer kakvom alatu autori teže razvijen je na univerzitetu u San Francisku. To vizualno rešenje razvija se već dugi niz godina i promenilo je i nekoliko tehnologija, gde je trenutno aktuelno rešenje u *JavaScript*-u.

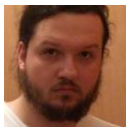
Ono što je mana tog rešenje jeste nemogućnost prilagođavanja terminologije, za razliku od rešenja prikazanog u ovom radu.

Mogućnosti koje pruža *WebAssembly* su velike, jer omogućava ostalim jezicima opšte namene da se izvršavaju unutar internet pretraživača. Apetiti tima koji radi na *wasm* platformi su veći, te ga oni vide kao ciljnu platformu za izvršavanje softvera odgovornog za rad *cloud* sistema. Sa druge strane, mnoge biblioteke i softveri su na ovaj način našle put do izvršavanja u internet pretraživačima, samim tim je težnja da se kod piše za jednu, uniformnu platformu dostigla novi nivo.

5. LITERATURA

- [1] Prof. Tom Leighton, Dr. Marten van Dijk
“Mathematics for Computer Science”, MIT course
- [2] J.A. Bondy, U.S.R. Murty “Graph Theory”, 3rd
edition, 2008
- [3] Kevin Hoffman “Programming WebAssembly with
Rust”
- [4] Steve Klabnik, Carol Nichols “The Rust Programming
Language”
- [5] “wasm-bindgen guide”, official wasm-bindgen
documentation
- [6] “D3.js - Data-Driven Documents”, official D3.js
documentation
- [7] E. Gamma, R. Helm, R. Johnson, J. Vlissides “Design
Patterns Elements of Reusable Object-Oriented
Software”

Kratka biografija:



Rade Radišić rođen je u Zrenjaninu 1992. godine, gde je završio osnovnu i srednju elektrotehničku školu. 2011. godine upisuje Fakultet tehničkih nauka, smer Računarstvo i automatika. 2015. godine brani diplomski rad i upisuje master studije. Položio je sve ispite predviđene planom i programom.

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	

