



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXXIX

Број: 11/2024

Нови Сад

Едиција: „Техничке науке – Зборници“

Година: XXXIX Свеска: 11

Издавач: Факултет техничких наука Нови Сад

Главни и одговорни уредник: проф. др Срђан Колаковић, декан Факултета техничких наука у Новом Саду

Уредништво:

Проф. др Срђан Колаковић
Проф. др Александар Купусинац
Проф. др Борис Думнић
Проф. др Дарко Стефановић
Проф. др Себастиан Балаш
Проф. др Дејан Лукић
Проф. др Јован Дорић
Проф. др Мирослав Кљајић
Проф. др Немања Тасић
Проф. др Дејан Убавин

Проф. др Милан Видаковић
Проф. др Мирјана Дамњановић
Проф. др Јелена Атанацковић Јеличић
Проф. др Игор Пешко
Проф. др Драган Јовановић
Проф. др Небојша Ралевић
Доц. др Сања Ожват
Проф. др Немања Кашиковић
Проф. др Теодор Атанацковић

Редакција:

Проф. др Дарко Стефановић, главни уредник
Проф. др Жељен Трповски, технички
уредник

Проф. др Драгољуб Новаковић
Проф. 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 Стеван Станковски, 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
Срђан Колаковић. – 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 cm. –(Едиција: Техничке науке – зборници)

Месечно

ISSN 0350-428X

COBISS.SR-ID 58627591

ПРЕДГОВОР

Поштовани читаоци,

Пред вама је једанаеста 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 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 21.05.2024. до 16.09.2024. год., а који се промовишу 27.01.2025. год. То су оригинални прилози студената са главним резултатима њихових мастер радова.

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

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

У овој свесци, са редним бројем 11, објављени су радови из области:

- машинства и
- електротехнике и рачунарства.

У свесци са редним бројем 12. објављени су радови из области:

- грађевинарства,
- саобраћаја,
- графичког инжењерства и дизајна,
- архитектуре,
- инжењерства заштите на раду и заштите животне средине,
- мехатронике,
- математике у техници и
- геодезије и геоматике.

У свесци са редним бројем 1. из 2025. године објављени су радови из области:

- инжењерског менаџмента,
- инжењерства информационих система,
- биомедицинског инжењерства,
- анимација у инжењерству,
- чистих енергетских технологија и
- сценске архитектуре и технике.

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. Александар Бардак,
ПРИМЈЕНА ФАЗНО ПРОМЈЕНЉИВИХ МАТЕРИЈАЛА – СИМУЛАЦИЈА УТИЦАЈА НА
ТЕМПЕРАТУРУ И ЕНЕРГЕТСКУ ПОТРОШЊУ ОБЈЕКТА 1553-1556
2. Andreja Živkov,
PRORAČUN PROTOČNE KARAKTERISTIKE REGULACIONOG VENTILA PRIMENOM
RAČUNARSKE DINAMIKE FLUIDA 1557-1560
3. Марко Вукић,
НУМЕРИЧКИ ПРОРАЧУН ПРОТОЧНЕ КАРАКТЕРИСТИКЕ РЕГУЛАЦИОНОГ ВЕНТИЛА 1561-1564
4. Radovan Ratković,
PRORAČUN DVOFAZNOG SEPARATORA U OKVIRU SABIRNO-OTPREMNE NAFTNE
STANICE “ЇДОШ” 1565-1568
5. Aleksandra Kosanović,
ANALIZA TAČNOSTI I KVALITETA OBRADE PRI SEČENJU LASEROM, 1569-1572

Radovi iz oblasti: Elektrotehnika i računarstvo

1. Ana Atanacković,
GPT ARHITEKTURA I PRIMENA U SOFTVERSKOJ INDUSTRIJI 1573-1576
2. Владан Живановић, Дејан Јеркан,
НУМЕРИЧКИ МОДЕЛ ЗА ДИНАМИЧКЕ ПРОЦЕСЕ ТРОФАЗНЕ КАВЕЗНЕ АСИНХРОНЕ
МАШИНЕ 1577-1580
3. Milena Laketić,
SISTEM ZA PRAĆENJE I ANALIZU ŠAHOVSKIH PARTIJA SA PLATFORME LIČES U
REALNOM VREMENU 1581-1585

4. Nikola Kolović, PROJEKTOVANJE INFRASTRUKTURE ZA POTREBE KLIJENT-SERVER APLIKACIJE I EVALUACIJA PROJEKTOVANE INFRASTRUKTURE KROZ PRIZMU DOBROG ARHITEKTONSKOG OKVIRA	1586-1589
5. Угљеша Поповић, Ивана Тодоровић, Дарко Марчетић, РАЗВОЈ И ПРОГРАМИРАЊЕ ЕДУКАТИВНЕ СТАНИЦЕ ЗА ПРОУЧАВАЊЕ КОМУНИКАЦИОНИХ ПРОТОКОЛА У ИНДУСТРИЈСКОМ ОКРУЖЕЊУ	1590-1593
6. Dragan Borkovac, UPOTREBLJIVOST GRAPHQL BIBLIOTEKE ZA RAZVOJ NLP INTERFEJSA	1594-1597
7. Đorđe Novaković, PRIMENA OPERATIVNOG SISTEMA ZA RAD U REALNOM VREMENU U INTERNETU STVARI	1598-1602
8. Nikola Sarkić, UTICAJ OBNOVLJIVIH IZVORA ENERGIJE NA ELEKTRODISTRIBUTIVNU MREŽU	1603-1606
9. Aleksa Ivanić, POREĐENJE POKRIVENOSTI AKADEMSKIH BAZA	1607-1610
10. Милош Гравара, MiniOptimizer: ПОЈЕДНОСТАВЉЕНИ РЕЛАЦИОНИ ОПТИМИЗАТОР УПИТА	1611-1614
11. Душан Миљковић, ФОРЕНЗИКА macOS-a	1615-1618
12. Зорица Чулић, ОПТИМИЗАЦИЈА ИЗБОРА ТРАНСФОРМАТОРА И НАПОЈНИХ КАБЛОВА ЗА НАПАЈАЊЕ НОВЕ ГЛАВНЕ ЦРПНЕ СТАНИЦЕ 1 У НОВОМ САДУ	1619-1622
13. Miloš Marinković, SEMANTIČKA SEGMENTACIJA POKRIVENOSTI ZEMLJIŠTA U GORNJEM PODUNAVLJU NA SLIKAMA SENTINEL-2 SATELITA	1623-1626
14. Cvijetin Mladenović, NAPREDNE TEHNIKE ZA SENTIMENT ANALIZU: STUDIJA KLASIFIKACIONIH I GENERATIVNIH MODELA NA ONLINE KOMENTARIMA	1627-1630
15. Igor Jorgovanović, ESTIMACIJA KRVNOG PRITISKA NA OSNOVU FOTOPLETIZMOGRAMA KORISTEĆI VEŠTAČKE NEURONSKE MREŽE	1631-1634
16. Сандра Бркић, НАМЕНСКИ ЈЕЗИК И ОКРУЖЕЊЕ ЗАСНОВАНО НА ИНЖЕЊЕРСТВУ ВОЂЕНОМ МОДЕЛИМА ЗА ОПИСИВАЊЕ И ГЕНЕРИСАЊЕ БАСНИ	1635-1638
17. Елена Акић, НАМЕНСКИ ЈЕЗИК И ОКРУЖЕЊЕ ЗАСНОВАНО НА ИНЖЕЊЕРСТВУ ВОЂЕНОМ МОДЕЛИМА ЗА ПРИСТУП РАЗЛИЧИТИМ ВЕКТОРСКИМ БАЗАМА ПОДАТАКА	1639-1642
18. Nebojša Koledin, Vladimir Popović, UPRAVLJANJE SINHRONIM MOTOROM SA PERMANENTNIM MAGNETIMA NA ROTORU U ELEKTROMOTORNIM POGONIMA BEZ DAVAČA NA VRATILU	1643-1646
19. Никола Поповић, ОДРЕЂИВАЊЕ ПАРАМЕТАРА ЕКВИВАЛЕНТНЕ ШЕМЕ АСИНХРОНОГ МОТОРА УПОТРЕБОМ СОФТВЕРСКИХ АЛАТА	1647-1650
20. Андрија Станишић, ПРИМЕНА ФЕДЕРАТИВНОГ УЧЕЊА У РЕШАВАЊУ ПРОБЛЕМА БИНАРНЕ КЛАСИФИКАЦИЈЕ	1651-1654
21. Milan Gvozdenović, Stevan Cvetičanin, ANALIZA PRELAZNIH REŽIMA NA MODELU 20[kV] MREŽE PRILIKOM AUTOMATSKOG PONOVOG UKLJUČENJA	1655-1658

22. Kristina Vajda, PREDIKCIJA HRONIČNE BOLESTI BUBREGA NA OSNOVU SCINTIGRAFSKIH SNIMAKA UPOTREBOM METODA MAŠINSKOG UČENJA.....	1659-1662
23. Vladimir Lunić, KLASIFIKACIJA TIPA RADARSKOG OBJEKTA KORISTEĆI LSTM NEURONSKE MREŽE SA KONVOLUCIJAMA NA SKUPOVIMA TAČAKA	1663-1667
24. Миле Кајтез, РАЗВОЈ ВЕБ АПЛИКАЦИЈЕ ЗА ПРЕЗЕНТОВАЊЕ ПОДАТАКА О ФАКУЛТЕТИМА И ДАВАЊЕ ПРЕДЛОГА УПОТРЕБОМ "CHAT GPT AI" МОДЕЛА	1668-1671
25. Vladimir Jovanović, SOFTVERSKO REŠENJE ZA AUTOMATIZOVANU RAZMENU ZDRAVSTVENIH KARTONA ZASNOVANO NA TEHNOLOGIJI BLOKČEJN	1672-1675
26. Biljana Milošević, ANALIZA DELOVANJA TERMIČKE ZAŠTITE KOD OPASNIH POGONSKIH STANJA ASINHRONIH MOTORA	1676-1679
27. Aleksandar Džever, Marko Vekić, PRIMARNA I SEKUNDARNA KONTROLA ZA KOMPENZACIJU NESIMETRIJE NAPONA ZAJEDNIČKE SABIRNICE SADEJSTVOM PRETVARAČA U MIKROMREŽI	1680-1683
28. Nebojša Gordić, ANALIZA MEHANIZAMA MREŽNE BEZBEDNOSTI NA AWS	1684-1687
29. Nataša Rajtarov, JEZIK ZA SPECIFIKACIJU PRODAJNIH AKCIJA	1688-1691
30. Aleksa Simić, FANTAZIJSKE IGRE BAZIRANE NA VELIKIM JEZIČKIM MODELIMA	1692-1695
31. Nevena Radešić, ANALIZA ALGORITAMA ROJA ČESTICA ZA PROBLEM RUTIRANJA ELEKTRIČNIH VOZILA	1696-1699
32. Dragana Filipović, Nikola Luburić, BEZBJEDNOSNA ANALIZA CAN PROTOKOLA KORIŠĆENOG U AUTOMOBILIMA	1700-1703
33. Danilo Bulatović, ZAGONETNA AVANTURA U VIRTUELNOJ REALNOSTI	1704-1707
34. Jovan Svorcan, IZAZOVI U KREIRANJU 2D HEKANDSLEŠ VIDEO IGRE	1708-1711
35. Vukašin Bukvić, Ivan Todorović, TOK SNAGE IZMEĐU KANALA IZOLOVANOG VIŠEKANALNOG PRETVARAČA	1712-1715

ПРИМЈЕНА ФАЗНО ПРОМЈЕНЉИВИХ МАТЕРИЈАЛА – СИМУЛАЦИЈА УТИЦАЈА НА ТЕМПЕРАТУРУ И ЕНЕРГЕТСКУ ПОТРОШЊУ ОБЈЕКТА

APPLICATION OF PHASE CHANGE MATERIALS – SIMULATION OF THE IMPACT ON BUILDING TEMPERATURE AND ENERGY CONSUMPTION

Александар Бардак, Факултет техничких наука, Нови Сад

Област– МАШИНСТВО

Кратак садржај – Технологија са великим потенцијалом, која може одиграти кључну улогу у човјековој зеленој будућности је технологија фазно промјенљивих материјала. У овом раду приказан је механизам фазно промјенљивих материјала и њихов утицај на енергетску ефикасност објекта. Кроз теоријски преглед и спроведене симулације у програму EnergyPlus представљени су ефекти које имплементација ових савремених материјала има на температуру у објекту, потрошњу енергије, потенцијалне уштеде и смањење емисије штетних гасова.

Кључне речи: Фазно промјенљиви материјали, хлађење, латентна топлота, симулација, EnergyPlus

Abstract – A technology with great potential, which can play a key role in human's green future, is the technology of phase change materials. This paper presents the mechanism of phase change materials and their influence on the energy efficiency of the building. Through a theoretical review and conducted simulations in the EnergyPlus program, the effects of implementing these modern materials on the temperature in the building, energy consumption, potential savings, and reduction of harmful gas emissions are presented.

Keywords: Phase change materials, cooling, latent heat, simulation, EnergyPlus

1. УВОД

У погледу глобалног загријавања и све топлијих временских услова, потреба за хлађењем објеката постаје све већа, те представља изазов који захтијева иновативна рјешења. Данас око 35% домаћинстава има бар један клима уређај. Енергија која се утроши на хлађење објеката чини значајан дио укупне утрошене енергије у свијету и због тога примјена технологија које омогућавају смањење утрошка енергије за хлађење постаје кључна [1].

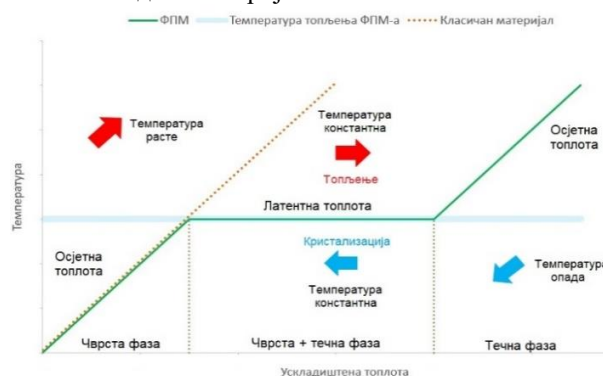
Фазно промјенљиви материјали (ФПМ) управо представљају такво иновативно рјешење, односно кључну иновацију у домену термалне регулације у грађевинским објектима, које нуди ефикасно рјешење за контролу температуре унутар објекта.

Напомена:

Овај рад проистекао је из мастер рада чији је ментор био др Мирослав Кљајић, ванр. проф.

2. КАРАКТЕРИСТИКЕ ФПМ-а

Ефекат складиштења, односно похране топлоте грађевинских елемената могуће је значајно повећати примјеном ФПМ-а. ФПМ омогућавају велико складиштење топлоте при промјени фазе, а специфични топлотни капацитет тих материјала је знатно већи од конвенционалних материјала. Прелаз ФПМ-а из чврстог у течно агрегатно стање, односно промјена фазе, одвија се при својственој тачно дефинисаној температури топљења за дати материјал.



Слика 1: Фаза ФПМ-а у зависности од температуре

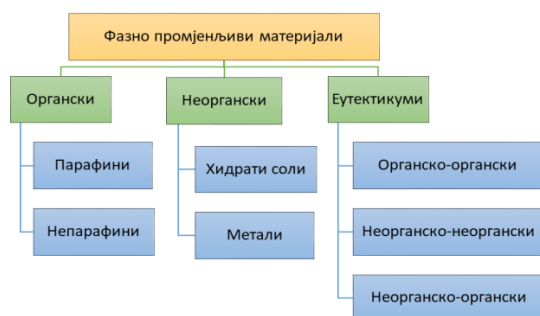
На тај начин, долази до значајног прихватања расположиве количине топлотне енергије из околине, при чему не долази до повећања температуре самог материјала, него само до промјене фазе, односно агрегатног стања. Тако акумулирану топлоту називамо латентном топлотом.

Када се током дана одвија топљење, тада се топлота доводи материјалу, док се ноћу јавља процес очвршћавања, при чему се топлота ослобађа из материјала, као што се види на слици 1. ФПМ омогућава одржавање унутрашње температуре просторије у жељеним границама, јер се топлота не преноси на унутрашњост него се троши на промјену фазе материјала.

Долазимо до закључка да количина ускладиштене топлоте код конвенционалних материјала једнака је прихваћеној осјетној топлоти, док код ФПМ-а укупна прихваћена количина топлоте једнака је збиру похрањене осјетне и латентне топлоте.

Фазно промјенљиви материјали су хемијски разноврсна група материјала, чију класификацију видимо на слици 2. Широко је температурни опсег у коме су доступни ови савремени материјали. Свака група

материјала има свој специфичан опсег температуре топљења и опсег енталпије топљења.



Слика 2. Класификација ФПМ-а [2]

3. ИМПЛЕМЕНТАЦИЈА ФПМ-а

Имплементација ФПМ-а може бити реализована како у оквиру пасивних, тако и у оквиру активних система. За пасивне системе није потребна механичка опрема и није потребан утршак додатне енергије. У таквим системима температура ваздуха опада и расте у односу на тачку топљења, при чему ФПМ мијења фазу, а природна вентилација омогућава циркулацију ваздуха. Са друге стране, активни системи за хлађење захтијевају механичку опрему како би ФПМ могао да обавља своју функцију, односно прихвата и отпушта топлотну енергију.

Високо ефикасни омотачи зграда су најбољи начин за смањење топлотних потреба зграда, а да се при томе топлотни комфор унутрашњости објекта одржи на жељеном нивоу. Овакав приступ омогућава ФПМ-у да постане дио грађевинске конструкције или компоненте зграде, готово непримјетно, а да при томе ефикасно обавља своју функцију. Имплементацијом савремених ФПМ-а са погодним карактеристикама за термичку регулацију, добијају се и боље вриједности индекса инерције који се користи при процјени способности материјала да задржи стабилну унутрашњу температуру објекта. Материјали су у том смислу занимљиви за проучавање, јер се помоћу њих може доћи до веће ефикасности омотача, који је кључан фактор у одрживој градњи и може значајно допринијети смањењу потрошње енергије за хлађење.

Коришћење ФПМ-а у омотачима зграда можемо да подијелимо на: интеграцију у грађевинском материјалу или као посебну компоненту.

4. Симулације у програму EnergyPlus

Одабир програма EnergyPlus првенствено је базиран на његовој способности у раду са својствима материјала [3]. Односно, могућност рада са материјалима са промјеном фазе. У овом раду описане су симулације, на основу којих је анализирана примјена ФПМ-а, са циљем оптимизације термичких перформанси објекта, анализан је и утицај на потрошњу енергије и утицај различитих конфигурација ФПМ-а.

4.1 Анализирани објекат

Како би се на што једноставнији начин приказао утицај коришћења ФПМ-а, за потребе симулације коришћен је једноставан једнозонски објекат чија је

свака површина у додиру са околином. Слојевита структура подова и плафона је коришћена у симулацији, како би се постигао довољан коефицијент пролаза топлоте, ког захтијева програм KnaufTerm2. На овај начин је смањен утицај на промјену температуре објекта од стране пода и плафона, у чијим слојевима нису имплементирани фазно промјенљиви материјали. Замисао такве употребе ФПМ-а је да се прикаже предност кориштења ових материјала само у спољашњим зидовима, као основа за даље анализе и могућности употребе ФПМ-а у сложенијим објектима.

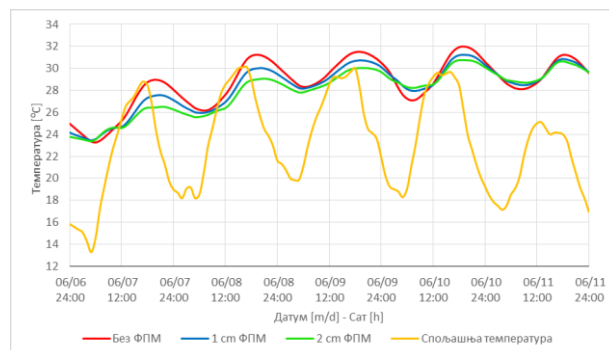
4.2 Спроведене симулације

Документ времена је датотека у којој су садржани метеоролошки подаци, изабраног географског мјеста, потребних за симулацију у програму EnergyPlus. Ова датотека се користи за потребе симулације како би се симулирали стварни услови околине. За потребе ове симулације коришћена је датотека времена за Београд.

У циљу истраживања употребе фазно промјенљивих материјала, спроведено је неколико симулација у софтверу EnergyPlus. Симулације су урађене са истим објектом, али у неколико различитих конфигурација. Према томе, симулирана је промјена температуре објекта, са и без ФПМ-а. Такође, симулирана је и потрошња енергије за одржавање температуре објекта у случајевима са ФПМ-ом у спољашњим зидовима објекта, као и без њега. Симулације су спроведене у временском периоду од пет мјесеци, од маја до септембра. Разлог изабраног опсега су управо више температуре него у остатку године, а самим тим и највећа потреба за хлађењем је током ових мјесеци.

4.3 Утицај ФПМ-а на унутрашњу температуру

У анализираном периоду од пет мјесеци, остварене су знатно ниже температуре у конфигурацији објекта са имплементираним ФПМ-ом у слоју спољашњих зидова. Примјери флукуација температура у све три конфигурације, као и спољашња температура представљени су на дијаграму 1. Током симулираног периода добијене су ниже унутрашње дневне температуре, као и ниже максималне температуре унутар објекта.



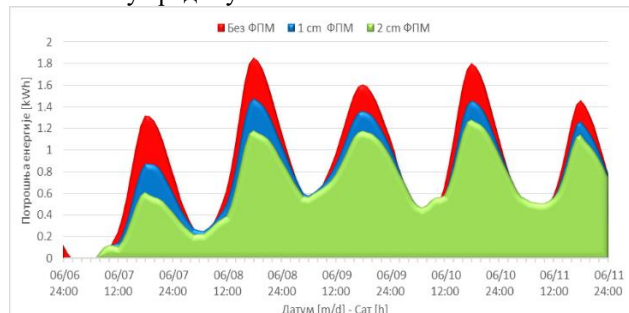
Дијаграм 1: Температуре појединих дана у јуну

Према томе, ФПМ успјешно обавља своју функцију термалних акумулатора у објекту са ФПМ-ом и остварене температуре су осјетно ниже него у објекту без слоја ФПМ-а. Ово је потврда да употреба оваквих материјала доприноси приближавању унутрашње

температуре ка нивоу комфора. Поред тога у мјесецу мају остварен је занимљив ефекат, гдје су се у објектима са имплементираним ФПМ-ом током појединих дана температуре у објекту налазиле у границама комфора, при чему је потпуно елиминисана потреба за хлађењем. У јулу и августу током појединих дана упркос високим дневним и релативно високим ноћним температурама, ФПМ је ефикасно регулисао температуру.

4.4 Утицај ФПМ-а на потрошњу енергије

У симулацијама је подешено да температура просторије увијек буде испод или на вриједности од 24°C. Потрошња енергије за потребе хлађења објекта значајно је смањена у објектима са имплементираним ФПМ-ом. На дијаграму 2 представљена је потрошња енергије у сва три објекта у мјесецу јуну, гдје се јасно може видјети остварена разлика коју остварују објекти са имплементираним ФПМ-ом. Као што је и очекивано очигледне су и разлике у потрошеној енергији у две конфигурације са ФПМ-ом. Црвена површина представља енергију коју је употребом ФПМ-а могуће смањити, те тако позитивно утицати на животну средину.



Дијаграм 2: Потребна енергија за хлађење - јун

4.5 Укупна уштеда енергије

Подаци су комплетирани на мјесечном нивоу у табели 1 и приказују потрошњу енергије у свим конфигурацијама објекта. Ова анализа пружа увид у динамику потрошње енергије током различитих мјесеци, те омогућава идентификацију ефикасности примјењеног слоја ФПМ-а.

Табела 1: Утрошена енергија по мјесецима

Утрошена енергија за хлађење [kWh]						
	Мај	Јун	Јул	Авг.	Сеп.	Ук.
Без ФПМ	210	363	560	497	141	1770
1cm ФПМ	181	317	512	449	113	1572
2cm ФПМ	168	290	484	421	100	1463

На основу података из табела 2 и 3 лако се може закључити да су смањења потрошње енергије употребом ФПМ-а остварена током свих мјесеци. Иако током одређених периода смањење температуре у објектима са ФПМ-ом није било велико, симулацијама је утврђено да је уштеда у потрошњи енергије за хлађење значајна.

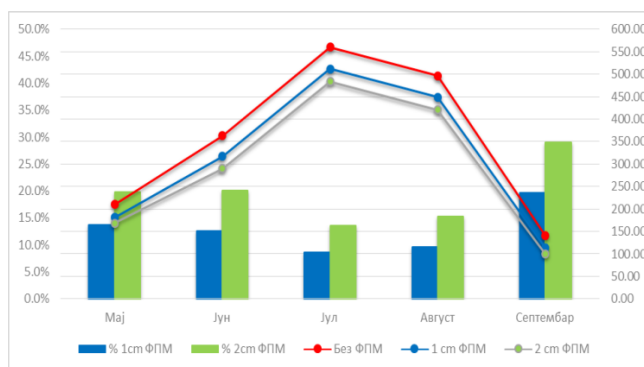
Са дијаграма 3 и видимо да је одговарајућа температура топлења материјала позитивно је утицала на енергетску ефикасност објекта, те на тај начин оствареним цикличним радом ФПМ-а добијена су смањења потрошње енергије.

Табела 2: Остварена уштеда у kWh

Уштеђена енергија [kWh]						
ФПМ	Мај	Јун	Јул	Авг.	Сеп.	Ук.
1cm	28.8	45.9	48.2	47.6	27.6	198.0
2cm	41.5	73.0	76.3	75.5	40.7	307.2

Табела 3: Уштеда у различитим конфигурацијама [%]

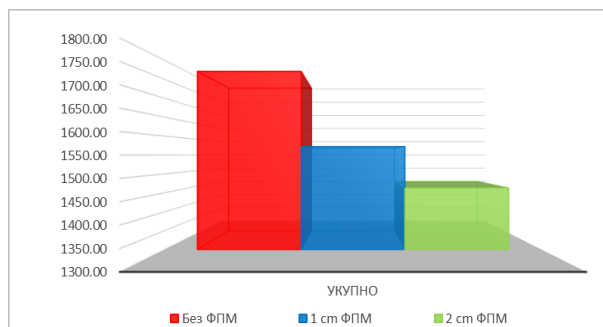
Уштеђена енергија [%]						
ФПМ	Мај	Јун	Јул	Авг.	Сеп.	Ук.
1cm	13.7	12.6	8.6	9.6	19.6	11.2
2cm	19.8	20.1	13.6	15.2	29.0	17.4



Дијаграм 3: Потрошња и остварена уштеда током симулираних мјесеци

Највеће смањење утрошене енергије јавило се током најтоплијих мјесеци. Као што је очекивано највећа потреба за енергијом за хлађење јавила се у току јула и августа. Иако процентуално нешто ниже вриједности уштеде, у овим мјесецима су већ у конфигурацији са 1 cm ФПМ-а остварене уштеде од преко 47 kWh, док у опцији са 2 cm ФПМ-а и за преко 75 kWh.

Према томе, употребљени револуционарни ФПМ су ефикасно извршили своју способност похрањивања и отпуштања топлотне енергије, те омогућили стабилизацију температуре унутар анализираних објекта. У овим мјесецима је и највећи утицај хлађења на електроенергетску мрежу. Стога, оствареним ефектима ФПМ-а се смањује ризик од преоптерећења и нестабилности мреже.



Дијаграм 4: Укупно утрошена енергија за хлађење

Према добијеним вриједностима потрошње енергије за потребе хлађења све три конфигурације објекта, израчуната су и новчана средства које је потребно издвојити у Републици Србији, како би се оствариле циљане температуре у објекту, приказана у табели 4.

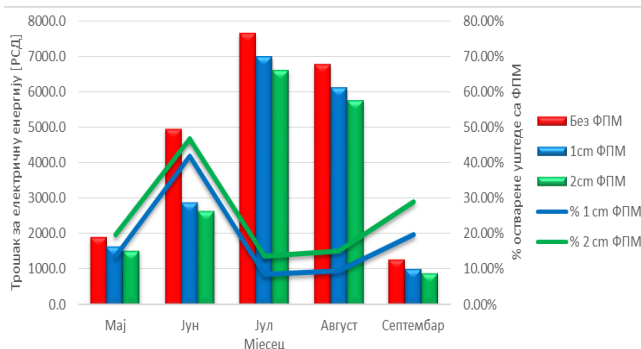
Табела 4: Трошак за хлађење објекта у РСД

	Без ФПМ	1 cm ФПМ	2 cm ФПМ
Мај	1913.4	1650.8	1534.9
Јун	4957.8	2887.4	2639.8
Јул	7651.2	6993	6608.6
Август	6787.7	6137.7	5755.7
Септембар	1280.3	1029.3	909.1
Укупно	22590.3	18698	17448

Цијена електричне енергије која је анализирана је узета за двотарифно мјерење [4], односно за вишу дневну тарифу. Разлог узимања више тарифе је чињеница да се хлађење објекта врши у периоду када је електрична енергија скупља, односно током дана, када су и спољашње температуре високе.

Табела 5: Процентуално остварене уштеде у потрошњи електричне енергије

	Новчана уштеда [%]					
ФПМ	Мај	Јун	Јул	Авг.	Сеп.	Ук.
1cm	13.7	41.8	8.6	9.6	19.6	17.2
2cm	19.8	46.7	13.6	15.2	29.0	22.8



Дијаграм 5: Трошак за електричну енергију и остварена уштеда у %

ФПМ материјали пружају конкретне економске предности у хлађењу објекта, видљиве на дијаграму 5. Ако погледамо остварену уштеду у симулацијама, можемо рећи да употреба ФПМ-а директно утиче на смањење рачуна за електричну енергију, пружајући корисницима значајне уштеде на дугорочном нивоу. Још једна занимљивост представља могућност да се употребом оваквих материјала корисник смањењем потрошње енергије нађе у нижој зони потрошње, гдје је цијена електричне енергије знатно повољнија, што је остварено у мјесецу јуну. Уштеђена средства могу се преусмјерити у неки од додатних начина повећања енергетске ефикасности објекта.

4.6 Емисија штетних гасова

Како употреба ФПМ-а смањује потребу за радом расхладних уређаја, тако се смањује и потреба за производњом исте количине електричне енергије. ФПМ позитивно утиче на електродистрибутивни систем, те умањује потребу за радом електрана које троше фосилна горива, што резултира смањеним емисијама штетних гасова. Како би се одредила емисија штетних материја која је настала при производњи електричне енергије утрошене за потребе хлађења симулираног објекта, кориштени су коефицијенти емисије. У Србији се 70% електричне енергије добија у термоелетранама [5], те су

вриједности емисионих фактора узете за лигнит, које се виде у табели 6.

Табела 6. Емисиони фактори за лигнит [6]

CO ₂	NO _x	SO _x	PM ₁₀	PM _{2.5}
0.36360	0.00062	0.00302	0.00042	0.00039

Табела 7. Количина емитованих честица у kg

	CO ₂	NO _x	SO _x	PM ₁₀	PM _{2.5}
Без ФПМ	450.5	0.772	3.747	0.522	0.482
1cm ФПМ	400.1	0.685	3.328	0.464	0.428
2cm ФПМ	372.4	0.638	3.097	0.431	0.398

Вриједности емисије штетних гасова за овако мали објекат су значајне. Експанзија употребе ФПМ-а би довела до осјетних разлика у потрошњи енергије за потребе хлађења, а посљедишно и до великих побољшања у еколошком смислу.

5. ЗАКЉУЧАК

Симулације су показале да је максимална температура употребом ових савремених материјала смањена за неколико степени, што осјетно доприноси угодности корисника у објекту. Један од најважнијих ефеката који се остварује имплементацијом савремених ФПМ-а јесте значајно смањење потрошње енергије за потребе хлађења самог објекта, које у две симулиране конфигурације износи 11,2% и 17,4%.

Технологија ФПМ-а повећањем ефикасности објекта доводи до низа позитивних ефеката у смислу мањег утицаја на електроенергетску мрежу, новчаних уштеда, смањене емисије штетних гасова, и као таква технологија постаје кључни елемент у постизању жељене енергетске ефикасности и одрживог развоја.

6. ЛИТЕРАТУРА

- [1] International Energy Agency, "World Energy Outlook", 2022 [S.l.], v. 39, p. 4, p. 33-43, jan. 2017
- [2] Souayfane F., Fardoun F., Biwole P.-H. "Phase change materials (PCM) for cooling applications in buildings: A review", 2016
- [3] <https://energyplus.net/> (pristupljeno u oktobru 2023.)
- [4] Skupština EPS AD Beograd, "Odluka o reguliranoj ceni električne energije za garantovano snabdevanje", Beograd, 2023
- [5] <https://www.eps.rs/lat/Poslovanje-EE/Stranice/Proizvodnja-elen.aspx> (pristupljeno u aprilu 2024.)
- [6] D.R. Gómez, J.D. Watterson, B.B. Americano, C. Ha, G. Marland, E. Matsika, L. N. Namayanga, B. O.-Elasha, J.D.K. Saka and K. Treanton (IEA), "IPCC Guidelines for National Greenhouse Gas Inventories", Chapter 2 Stationary Combustion, 2006

Кратка биографија:



Александар Бардак рођен у Добоју 1998. год. Основне академске студије завршио је на Факултету техничких наука у Новом Саду 2021. год. Мастер рад на Факултету техничких наука из области Машинство – Енергетика и процесна техника одбранио је 2024. год.
Контакт: aco.bardak@gmail.com

**PRORAČUN PROTOČNE KARAKTERISTIKE REGULACIONOG VENTILA
PRIMENOM RAČUNARSKE DINAMIKE FLUIDA**

**CALCULATION OF CONTROL VALVE FLOW CHARACTERISTICS USING
COMPUTATIONAL FLUID DYNAMICS**

Andreja Živkov, *Fakultet tehničkih nauka, Novi Sad*

**Mašinstvo – ENERGETIKA I PROCESNA
TEHNIKA**

Kratak sadržaj – U radu je izvršen teorijski i numerički proračun protočne karakteristike regulacionog ventila. Teorijski proračun protočne karakteristike ventila izvršen je na osnovu formula iz literature, dok je za numerički proračun korišćen program „SimFlow”. Cilj ovog rada bio je dokazivanje protočne karakteristike ventila dobijene od strane proizvođača numeričkim proračunom.

Ključne reči: regulacioni ventil, koeficijent protoka, numerički proračun

Abstract – The paper provides theoretical and numerical calculations of a control valve flow characteristic. The theoretical calculation is based on well-known analytical expressions, while for numerical calculation the SimFlow software has been used. The aim for this paper is to prove flow characteristics given by the valve manufacturer using numerical calculation.

Keywords: control valve, flow coefficient, numerical calculations

1. UVOD

Tema ovog rada je numerički proračun protočne karakteristike regulacionog ventila VENR DN100 PN16 – EQ proizvođača „Termovent SC” iz Temerina. Oznaka EQ ukazuje da je reč o regulacionom ventilu sa ravnoprocenatnom karakteristikom, što je trebalo proveriti proračunom u okviru ovog rada.

Regulacioni ventili su važne komponente različitih industrijskih postrojenja gde često imaju odgovorne uloge u kontroli procesa, posebno kada je reč o regulisanju pritiska ili protoka. Regulacioni ventili po pravilu imaju striktno definisanu protočnu (ili regulacionu) karakteristiku koja predstavlja zavisnost koeficijenta protoka K_v (odnosno koeficijenta lokalnog otpora) od otvorenosti ventila. Cilj ovog rada je da se do protočne karakteristike ventila dođe numeričkim proračunom.

U radu su dati osnovni pojmovi o regulacionim ventilima, ukazano je na njihovu funkciju u sistemima i date su najčešće karakteristike ventila. Izvršen je teorijski i numerički proračun koeficijenta protoka K_v kako bi se

dokazala protočna karakteristika ventila data od strane proizvođača. Teorijski proračun protočne karakteristike datog regulacionog ventila izvršen je na osnovu poznatih formula iz literature, dok se numerički proračun zasniva na metodi konačnih zapremina. U okviru numeričkog proračuna korišćen je program „SimFlow”, koji se zasniva na metodi konačnih zapremina. Teorijski i numerički proračuni ventila bazirani su na određivanju koeficijenta protoka K_v pri promeni njegove otvorenosti h .

2. METOD KONAČNIH ZAPREMINA

Metod konačnih zapremina je prvenstveno razvijen na bazi metode konačnih razlika. Algoritam rešavanja matematičkog modela koji opisuje neki strujni problem na bazi metode konačnih zapremina sastoji se od sledećih koraka:

- integracija osnovnih transportnih jednačina po kontrolnoj zapremini,
- diskretizacija, koja u stvari pretvara sistem integralnih jednačina u sistem algebarskih jednačina i
- rešavanje dobijenih algebarskih jednačina iterativnim postupkom [1].

Rezultati dobijeni integracijom čuvaju tačna fizička svojstva za svaku ćeliju konačne veličine. Glavni razlog primene metoda konačnih zapremina je upravo postojanje jasnih odnosa između numeričkih algoritama i osnovnog principa očuvanja fizičkog svojstva. Očuvanje fizičkog svojstva ϕ u okviru kontrolne zapremine definiše se kao ravnoteža između različitih procesa koji teže da je povećaju ili smanje. Za rešavanje problema dinamike fluida metodom konačnih zapremina, koriste se:

1. osnovne jednačine održanja,
2. jednačina stanja i
3. model viskoznih napona (npr. Njutnov).

Bez primene prethodno navedenih jednačina i pretpostavki ne može se dobiti matematički zatvoren sistem, samim tim nije moguće rešiti. Osnovna transportna jednačina nema ograničenja sa aspekta viskoznosti.

Opšti oblik transportne jednačine za stacionarna strujanja u integralnom obliku glasi:

$$\int_A (\rho \cdot \phi \cdot \vec{u}) d\vec{A} = \int_A (\Gamma \cdot \text{grad} \phi) d\vec{A} + \int_{KZ} S dV \quad (1)$$

Za rešavanje problema metodom konačnih zapremina pored definisanja svih potrebnih transportnih jednačina, neophodno je definisati granične i početne uslove.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Slobodan Tašin, vanr. prof.

3. REGULACIONI VENTILI

Regulacioni ventili su uređaji, koji se koriste za direktno regulisanje parametara radnog fluida kao što su pritisak i protok. Pored toga oni mogu i da vrše indirektnu regulaciju parametara kao što su temperatura, gustina i nivo tečnosti. Regulacija parametara ostvaruje se na osnovu promene protočnog preseka ventila, tačnije njegove otvorenosti. Upravljanje regulacionim ventilom može biti ručno i automatsko.

Koeficijent protoka je jedan od najvažnijih parametara ventila. Koeficijent protoka ventila K_V predstavlja protok radnog fluida izražen u metrima kubnim na čas kroz ventil (izazivač otpora) pri padu pritiska od 1 bar.

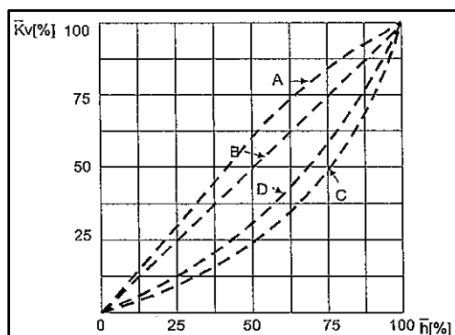
$$K_V = 36000 \cdot A_v \cdot \sqrt{\frac{2}{\zeta}} \quad (2)$$

3.1. Karakteristike ventila

Regulacioni ventili se međusobno razlikuju na osnovu protočne karakteristike ventila. Ventil može imati neku od sledećih protočnih karakteristika:

- A. brzootvarajuću karakteristiku,
- B. linearnu karakteristiku,
- C. ravnoprocenatnu karakteristiku i
- D. paraboličnu karakteristiku [2].

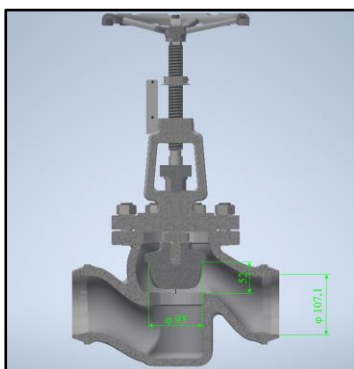
Na slici 1 prikazane su sve navedene karakteristike ventila.



Slika 1. Karakteristike ventila [3]

4. PRORAČUN

U okviru rada urađen je proračun protočne karakteristike ventila, odnosno izvršen je proračun koeficijenta protoka pri različitim otvorenostima ventila.



Slika 2. Regulacioni ventil

Proračun protočne karakteristike izvršen je za regulacioni ventil „Termovent SC”, model VENR DN 100 PN 16 – EQ. Koeficijent protoka pri punoj otvorenosti ventila je

$K_{VS} = 160 \text{ m}^3/\text{h}$. Od proizvođača je dobijen 3D model sa naznačenim svim osnovnim dimenzijama (slika 2).

4.1. Teorijski proračun

Prema proizvođaču, razmatrani regulacioni ventil treba da ima ravnoprocenatnu protočnu karakteristiku koja se može odrediti iz sledeće formule:

$$K_V(h_{rp}) = K_{VS} \cdot R^{-1} \cdot e^{h_{rp} \cdot \ln R} \quad [\text{m}^3/\text{h}] \quad (3)$$

Regulacioni odnos R u izrazu (3) je parametar regulacionih ventila koji se određuje eksperimentalnim putem u laboratorijskim postrojenjima i predstavlja odnos maksimalnog protoka (pri maksimalnoj otvorenosti ventila) i minimalnog protoka kroz ventil pri kojem ventil još uvek zadržava svoju regulacionu karakteristiku. Regulacioni odnos zavisi od tipa same protočne karakteristike ventila i za ventile sa ravnoprocenatnom karakteristikom najčešće iznosi 50 [4, 5].

U tabeli 1 prikazana je promena koeficijenta protoka posmatranog ventila pri različitoj otvorenosti ventila sračunata na osnovu (3).

Tabela 1. Koeficijent protoka pri promeni otvorenosti posmatranog regulacionog ventila

$h_{rp} [-]$	$K_V [\text{m}^3/\text{h}]$
0	3,2
0,1	4,73
0,2	7,00
0,3	10,35
0,4	15,30
0,5	22,63
0,6	33,46
0,7	49,48
0,8	73,17
0,9	108,20
1	160

4.2. Numerički proračun

Numerički proračun ventila izvršen je u okviru programskog paketa „SimFlow”, koji je slobodno dostupan na internet adresi: www.sim-flow.com. Ovaj program primenjuje metod konačnih zapremina za određivanje nepoznatih parametara strujanja. Kako bi program mogao da uradi numerički proračun karakteristike protoka ventila, neophodno je uneti geometriju ventila i ravne cevne deonice neposredno ispred i iza ventila. Ravna cevna deonica je dužine 1000 mm i ona je ključna za stabilizaciju strujnih parametara neposredno ispred i iza ventila. Ilustracije radi, na slici 3. prikazan je osnovni model sistema.



Slika 3. Model ventila i ravne cevne deonice

Nakon što se unese potrebna geometrija, neophodno je generisati mrežu. Izgled generisane mreže na postojećem modelu dat je na slici 4.



Slika 4. Model sa generisanom mrežom

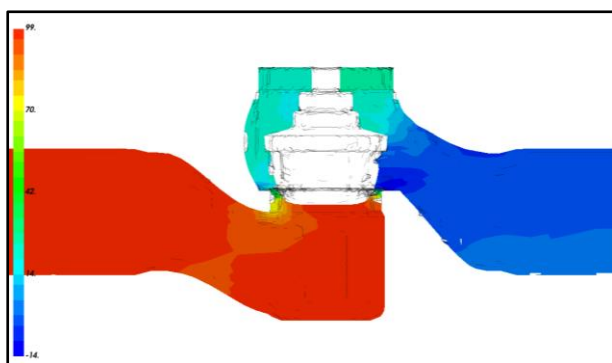
Kako bi se odredio koeficijent protoka pri različitoj otvorenosti ventila neophodno je podesiti još neke parametre programa. Pre svega, potrebno je odabrati da je transportovani fluid voda, kao i koji algoritam se koristi za rešavanje postavljene simulacije. Takođe, neophodno je podesiti i odgovarajuće granične uslove odnosno podešava se pritisak na izlazu i na ulazu.

Poznavajući definiciju koeficijenta protoka, bilo je neophodno podesiti da je pad pritiska kroz ventil 1 bar, što znači da je na ulazu u ventil pritisak npr. 1 bar a na izlazu iz ventila 0 bar. U tabeli 2. prikazana je sračunata promena karakteristike protoka pri različitoj otvorenosti ventila.

Tabela 2. Sračunate vrednosti koeficijenta protoka pri različitoj otvorenosti

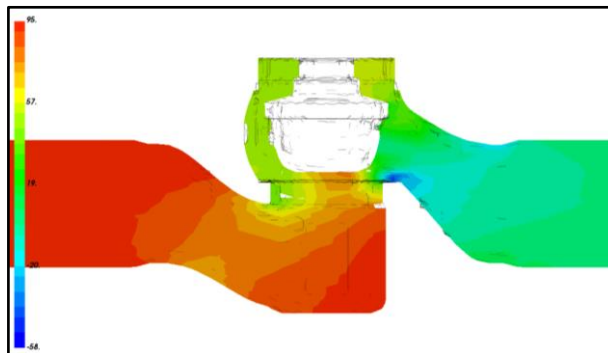
h_{rp} [-]	Q [m ³ /s]	K_V [m ³ /h]
0	0	0
0,1	0,00038	1,37
0,2	0,00094	3,38
0,3	0,00331	11,9
0,4	0,00608	21,89
0,5	0,01093	39,34
0,6	0,01738	62,58
0,7	0,02545	91,62
0,8	0,03115	112,15
0,9	0,03307	119,04
1	0,03382	121,76

Kao ilustracija rezultata koji su dobijeni numeričkim proračunima pod navedenim uslovima na slici 5 je prikazano polje pritiska u ventilu pri otvorenosti ventila od 60%.



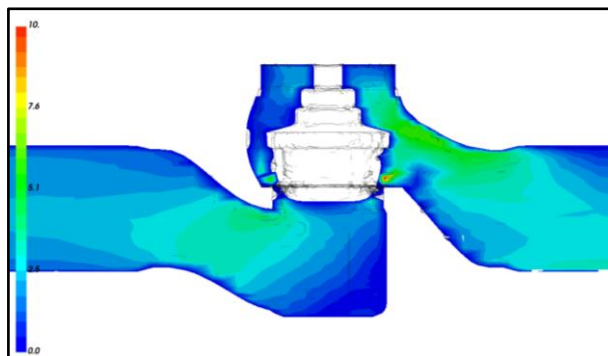
Slika 5. Polje pritiska u sistemu ventila i cevovoda pri otvorenosti ventila od 60%

Na slici 6 prikazano je polje pritiska u ventilu pri otvorenosti ventila od 100%.

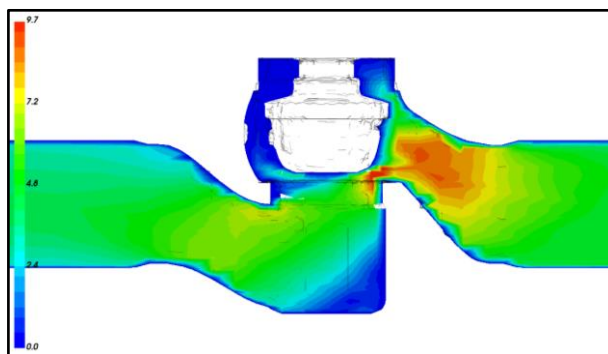


Slika 6. Polje pritiska u sistemu ventila i cevovoda pri otvorenosti ventila od 100%

Na slici 7. prikazano je polje brzina u ventilu pri otvorenosti ventila od 60%, dok je na slici 8. prikazano polje brzina u ventilu pri otvorenosti ventila od 100%.



Slika 7. Polje brzina u sistemu ventila i cevovoda pri otvorenosti ventila od 60%



Slika 8. Polje brzina u sistemu ventila i cevovoda pri otvorenosti ventila od 100%

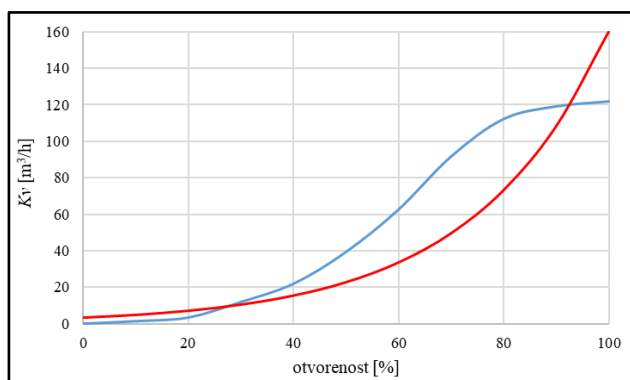
Na osnovu prethodnih slika (slike 5-8) može se zaključiti da prilikom numeričkog rešavanja protoka fluida sam program ima određena ograničenja. Bolji rezultati dobijaju se pri većoj otvorenosti ventila, iz razloga što program tada lakše prepozna same granice modela.

Ova simulacija ukazala je na nedostatak programa u pogledu nemogućnosti kreiranja sitnije mreže u oblasti sedišta i regulišućeg tela ventila. Sitnija mreža u ovoj oblasti bi omogućila preciznije podatke o tome kako se radni fluid ponaša u okviru oblasti oko regulišućeg tela koja utiče na karakteristiku protoka.

4.3. Analiza rezultata

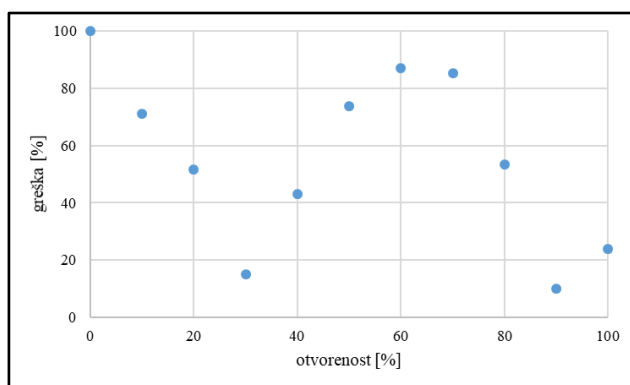
Vrednosti koeficijenta protoka dobijene teorijskim i numeričkim proračunom se znatno razlikuju, što se može zaključiti na osnovu analize podataka datih u tabeli 1 i tabeli 2 kao i na slici 9. Teorijska pretpostavka karakteristike ventila dobijena od strane proizvođača poklapa se samo delimično sa rezultatima koji su dobijeni numeričkim proračunom.

Na slici 9. prikazane su protočne karakteristike ventila koje su dobijene teorijskim i numeričkim proračunom. Crvenom bojom predstavljena je protočna karakteristika ventila dobijena teorijskim proračunom, dok je plavom bojom predstavljena protočna karakteristika ventila dobijena numeričkim proračunom.



Slika 9. Protočne karakteristike ventila dobijene teorijskom i numeričkom metodom

Kao što je već primećeno, postoje znatna odstupanja protočne karakteristike dobijene teorijskim i numeričkim proračunom. Najveća odstupanja javljaju se pri zatvorenom ventilu, jer pri teorijskim proračunu smatra se da postoji određeni protok, dok u realnom slučaju to nije moguće. Na slici 10. može se videti procentualna razlika koeficijenta protoka dobijenog teorijskim i numeričkim proračunom. Takođe, sa slike 10. može se videti da su znatna odstupanja i pri otvorenosti od 50% do 70%.



Slika 10. Procentualna greška koeficijenta protoka pri različitoj otvorenosti

5. ZAKLJUČAK

U okviru ovog rada izvršen je proračun protočne karakteristike jednog ventila. Ventil je marke „Termovent SC”, model VENR DN100 PN16 - EQ. Oznake EQ ukazuje da je reč o ventilu ravnoprocentne karakteristike. Na osnovu poznatih parametara ventila, datih od strane proizvođača,

izvršen je teorijski i numerički proračun. Teorijski, kao i numerički proračun ventila se zasniva na određivanju koeficijenta protoka pri promeni njegove otvorenosti.

Numerički proračun ventila je izvršen u programu „SimFlow”. Za potrebe numeričkog proračuna bilo je neophodno prvenstveno dodati pravu cevnu deonicu neposredno ispred i iza ventila, što je urađeno u programu „AutoCAD”, kako bi se ustalio protok fluida kroz cev. U okviru numeričkog proračuna pretpostavljeno je da je fluid koji struji voda.

Na osnovu teorijskog i numeričkog proračuna, dobijena je protočna karakteristika ventila odnosno zavisnost koeficijenta protoka od otvorenosti ventila. Upoređivanjem ove dve krive zaključeno je da se teorijska pretpostavka protočne karakteristike ventila u određenoj meri poklapa sa protočnom karakteristikom ventila dobijenom numeričkim proračunom. Kako se pri određenoj otvorenosti koeficijent protoka dobijen teorijskom i numeričkom metodom znatno razlikuje, može se zaključiti da regulacionom ventilu dobijenom od strane proizvođača ne odgovara u potpunosti ravnoprocentna karakteristika. Predlaže se proizvođaču da dodatno ispita protočnu karakteristiku predmetnog ventila eksperimentalnim putem i ukoliko je potrebno izvrši neke korekcije po pitanju samog dizajna ventila.

6. LITERATURA

- [1] H. Versteeg, W. Malalasekera, “An introduction to Computational Fluid Dynamics – The Finite Volume Method”, Longman Group Ltd, Harlow, 1995.
- [2] J. Ferziger, M. Perić, “Computational Methodes for Fluid Dynamics”, Springer-Verlag Berlin Heidelberg, Germany, 1996.
- [3] D. Uzelac, “Pumpne i kompresorske stanice-autorizovana predavanja”, Fakultet tehničkih nauka, Novi Sad, 2014.
- [4] <https://www.flocontrol.ltd.uk/what-is-rangeability-and-turndown-ratio-and-why-is-it-important/> (pristupljeno u junu 2024.)
- [5] <https://www.cedengineering.com/userfiles/Control%20Valves%20Basics%20-%20Sizing%20&%20Selection.pdf> (pristupljeno u junu 2024.)

Kratka biografija:



Andreja Živkov rođena je u Kikindi 2000. god. Master rad na Fakultetu tehničkih nauka iz oblasti Energetike i procesne tehnike-Računarska dinamika fluida odbranila je 2024.god.
kontakt: andreja.zivkov@gmail.com

НУМЕРИЧКИ ПРОРАЧУН ПРОТОЧНЕ КАРАКТЕРИСТИКЕ РЕГУЛАЦИОНОГ ВЕНТИЛА**NUMERICAL CALCULATION OF A CONTROL VALVE FLOW CHARACTERISTIC**

Марко Вукић, Факултет техничких наука, Нови Сад

Област – МАШИНСТВО

Кратак садржај – У овом раду извршено је поређење проточне карактеристике вентила, добијене применом рачунарске динамике флуида, у програму „SimFlow“, са карактеристикама вентила добијених применом познатих аналитичких израза из литературе.

Кључне речи: Регулациони вентил, рачунарска динамика флуида, нумерички прорачун, карактеристика вентила, параметар протока

Abstract – In this paper, a comparison was made of the valve flow characteristic obtained using computational fluid dynamics, in the "SimFlow" software, with the results obtained using known analytical formulae taken from literature.

Keywords: Regulating valve, Computational fluid dynamics, Numerical calculation, Valve characteristic, Flow coefficient

1. УВОД

Рачунарска динамика флуида је моћан инжењерски алат који се користи за симулацију и анализу феномена везаних за динамику флуида кроз нумеричко решавање тзв. основних једначина. Коришћењем математичких метода и рачунарских алгоритама, рачунарска динамика флуида омогућава инжењерима и истраживачима да стекну увид у сложена понашања флуида, предвиде перформансе и оптимизују конструкције различитих машина и апарата у широком спектру апликација.

Регулација протока и притиска је кључан аспект у бројним индустријама, од нафте и гаса, до третмана воде и шире. У области регулације система концепти параметра протока и проточне карактеристике вентила су од фундаменталног значаја. Параметар протока вентила представља запремински проток флуида који пролази кроз вентил под одређеним условима, обично при паду притиска од 1 bar. Функција која представља промену параметра протока у зависности од отворености вентила назива се проточна карактеристика вентила.

У овом раду урађено је поређење нумерички израчунате проточне карактеристике вентила, у функцији

отворености вентила, са карактеристикама израчунатих помоћу познатих аналитичких израза преузетих из литературе. Анализиран је регулациони вентил компаније „Термовент СЦ“, називног пречника DN 100, називног притиска PN 16, параметра протока при максималној отворености од 160 m³/h и хода од 52 mm. Регулационо тело овог вентила је обликовано (према произвођачу) тако да вентил има линеарну карактеристику, међутим, карактеристика овог вентила никада заправо није испитана. Параметар протока нумерички је израчунат применом рачунарске динамике флуида, методом коначних запремина у програму „SimFlow“.

2. МАТЕРИЈАЛ И МЕТОД

Параметар протока, који зависи од отворености вентила $\bar{h}$ [%], једнак је протоку кроз вентил (Q [m³/h]) при паду притиска на вентилу од 1 bar, односно

$$K_v(\bar{h}) = \frac{Q}{\sqrt{\Delta p}}, \quad (1)$$

при чему су:

Q [m³/h] – проток кроз вентил, и

Δp [bar] – пад притиска кроз вентил.

Произвођачи вентила често у својим каталозима дају дијаграме који приказују зависност параметра протока од отворености вентила. Ова зависност може да буде: линеарна, равнопроцентна (истопроцентна), брзоотварајућа, итд. Карактеристике вентила у функцији отворености вентила могу се израчунати аналитички, тако да за линеарну, равнопроцентну, параболичну и брзоотварајућу карактеристику важе следеће формуле, респективно, [1, 2]:

$$K_v(\bar{h}) = (R^{-1} + (1 - R^{-1}) \cdot \bar{h}_{lin}) K_{vs}, \quad (2)$$

$$K_v(\bar{h}) = (R^{-1} e^{\bar{h}_{rp} \cdot \ln R}) K_{vs}, \quad (3)$$

$$K_v(\bar{h}) = (R^{-1} + (1 - R^{-1}) \cdot \bar{h}_{par}^2) K_{vs} \quad \text{и} \quad (4)$$

$$K_v(\bar{h}) = \sqrt{\bar{h}_{bo}} K_{vs}, \quad (5)$$

при чему су:

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Слободан Ташин, ванр. проф.

K_{vs} [m^3/h] – параметар протока при отворености вентила од 100% (потпуно отворен вентил), и

R [-] – регулациони однос вентила, тј. однос максималног и минималног протока кроз вентил.

Карактеристика вентила зависи, као што је речено, од отворености вентила, док је тип зависности условљен обликом самог регулационог тела. Различити облици регулационог тела приказани су на слици 1.

За манипулацију 3Д модела вентила коришћен је програм „Inventor Professional“ компаније „Autodesk“, док је за нумеричку симулацију коришћен је програм „SimFlow“ који је развила компанија „SIMFLOW Technologies“.

У програму „Inventor Professional“ направљено је 10 верзија регулационог вентила, са цевном деоницом дужине 1 m испред и иза вентила, које се разликују само по положају регулационог тела. Прва верзија направљена је тако да регулационо тело буде на 10% отворености, док је свака наредна верзија имала већу отвореност, са кораком по 10%. Самим тим, у последњој верзији отвореност вентила 100%. Свака верзија модела сачувана је посебно у .stp формату, како би касније могла да се отвори у програму „SimFlow“.

У програму „SimFlow“ учитан је један од претходно креираних 3Д модела. Применом генератора мреже који је уграђен у „SimFlow“, генерисана је мрежа која представља запремину флуида унутар регулационог вентила. За тип домена изабран је квадар (енг. Box). Димензије овог квадра су такве да је дуж x -осе маргинално краћи од крајева цеви, како би их препознао као почетак, односно крај домена, док дуж осталих оса таквих димензија да обухвата читаву геометрију модела.

Број подела домена је 18 дуж x -осе, између 17 и 18 дуж y -осе, и између 15 и 16 дуж z -осе, у зависности од варијанте отворености вентила. Границе су адекватно дефинисане, како би програм препознао улаз, односно излаз из домена, а материјална тачка је постављена унутар геометрије како би програм препознао да се ради о унутрашњем струјању. Због ограничења

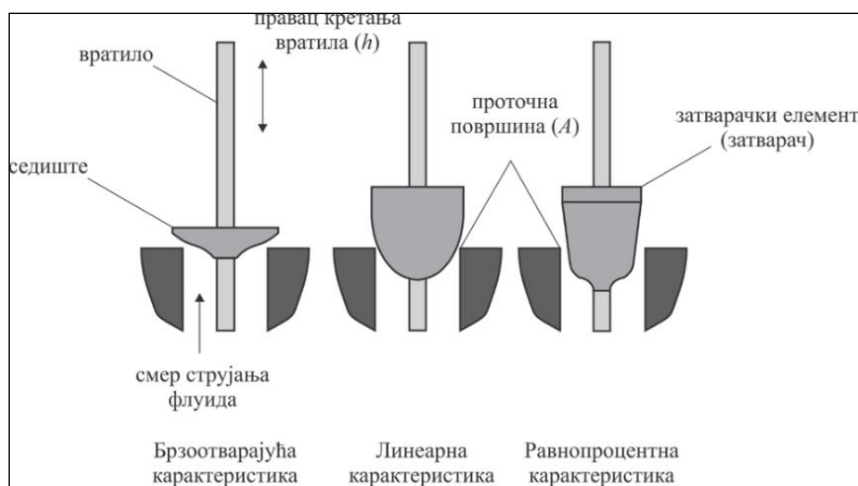
лиценце, број ћелија који је генерисан износи између 100 и 109 хиљада, у зависности од варијанте отворености вентила. Да би мрежа имала адекватан квалитет (у смислу облика ћелија), ћелије граничног слоја нису генерисане, те су све расположиве ћелије „потрошене“ на основну мрежу. Програм „SimFlow“ садржи 50 различитих решавача. За проблем струјања воде кроз регулациони вентил, као решавач одабран је SIMPLE.

Решавач SIMPLE погодан је управо за случајеве стационарног, некомп्रेसибилног струјања једнофазног флуида. У програму „SimFlow“ постоји могућност одабира групе модела турбуленције, односно изостављања истог. За овај случај изабран је RANS приступ моделирању турбуленције, односно у оквиру њега одабран је модел турбуленције $k-\omega$ SST (Shear Stress Transport – *срп. пренос тангенцијалног напона*). С обзиром да се ради о стационарном струјању, дискретизација по времену није потребна. Због своје стабилности при раду са мрежама лошијег квалитета, за шему дискретизације брзине, кинетичке енергије турбуленције, и брзине дисипације кинетичке енергије изабрана је Upwind шема дискретизације. 3

а градијенте је примењена линеарна Грин-Гаусова метода дискретизације, и изабран је метод линеарне интерполације за рачунање вредности кроз цео домен. У зависности од изабраног решавача, програм даје могућност промене само оних термофизичких величина које су релевантне за изабрани решавач и изабрани модел флуида (у овом случају изабран је Њутновски модел флуида, пошто се ради о води).

За посматрани проблем, једина релевантна величина јесте кинематска вискозност. Програм „SimFlow“ садржи библиотеку, односно каталог најчешће коришћених материјала (под материјалом се подразумевају флуиди) са њиховим термофизичким величинама, тако да корисник не мора да их памти, или тражи из литературе.

За тражену кинематску вискозност узета је кинематска вискозност воде, која је у програму преддефинисана као $10^{-6} \text{ m}^2/\text{s}$.



Слика 1. Различити облици регулационог тела [1]

На улазу у цевовод (енг. *Inlet*) гранични услов дефинисан је као „Pressure Inlet“ (срп. *притисак на улазу*), и унето је да тотални притисак износи $100 \text{ m}^2/\text{s}^2$, из разлога што је притисак за нестишљиве флуиде у програму „SimFlow“ увек изражен као кинематски притисак, односно притисак подељен са густином. Пошто је густина воде 1000 kg/m^3 , то значи да је притисак на почетку цевовода заправо дефинисан као 10^5 Pa , односно 1 bar . На излазу из цевовода (енг. *Outlet*) дефинисан је гранични услов „Pressure Outlet“ (срп. *притисак на излазу*) и притисак је унет као $0 \text{ m}^2/\text{s}^2$. Самим тим, пошто разлика између притисака на почетку и крају цевовода износи 1 bar , проток који се добије биће заправо једнак параметру протока K_v . Брзина, турбулентна кинетичка енергија, као и специфична брзина дисипације аутоматски су дефинисане у програму и те вредности нису мењане. Пошто програм „SimFlow“ нема могућност праћења протока у току итерација, дефинисано је да се прати брзина на улазу у цевовод. Урађено је по 500 итерација за свих 10 варијанти отворености вентила.

Пошто је познат унутрашњи пречник цевовода ($107,1 \text{ mm}$), може се израчунати проток на улазу

$$Q_{num} = \frac{4 \cdot u_{num}}{d^2 \pi} \quad (6)$$

при чему су:

u [m/s] – брзина струјања, и

d [m] – унутрашњи пречник цевовода.

Иста вредност, јасно је, добила би се и коришћењем брзине на излазу, пошто због једначине континуитета проток мора бити једнак и на улазу и на излазу у цевовод. Као што је објашњено, пошто је у питању струјање воде при константном паду притиска од 1 bar

$$K_{vnum} = Q_{num} \quad (7)$$

Проверена је релативна грешка добијене нумеричке вредности у односу на аналитички израчунату вредност за линеарну карактеристику (која је израчуната према једначини (2)), као и за равнопроцентну, параболичну и брзоотварајућу карактеристику (према једначинама (3), (4), и (5), респективно):

$$\varepsilon_{lin} = \frac{K_{vnum} - K_{vlin}}{K_{vlin}} \quad (8)$$

$$\varepsilon_{rp} = \frac{K_{vnum} - K_{vrp}}{K_{vrp}} \quad (9)$$

$$\varepsilon_{par} = \frac{K_{vnum} - K_{vpar}}{K_{vpar}} \quad (10)$$

$$\varepsilon_{bo} = \frac{K_{vnum} - K_{vbo}}{K_{vbo}} \quad (11)$$

Регулациони однос вентила који фигурише у аналитичким једначинама за параметар протока усвојен је према препорукама из литературе [3]. Он представља однос максималног и минималног протока при којем вентил задржава регулациону карактеристику. Конкретно, за линеарну карактеристику усвојен је регулациони однос 33, за брзоотварајућу 20, док је за равнопроцентну и параболичну карактеристику усвојен регулациони однос 50.

3. РЕЗУЛТАТИ И ДИСКУСИЈА

Резултати урађених нумеричких симулација, аналитички резултати, као и релативне грешке, дати су у табели 1. На слици 2. су графички приказани аналитички и нумерички добијени параметри протока.

Као што се може видети, карактеристика вентила добијена нумеричком симулацијом не изгледа у потпуности као аналитички добијена линеарна карактеристика. Види се да не изгледа ни као аналитички добијена равнопроцентна, параболична или брзоотварајућа карактеристика.

Међутим, као што је речено, карактеристика овог вентила никада није проверена од стране произвођача, тако да се не може искључити могућност да су резултати симулације исправни, а да карактеристика вентила није заправо линеарна као што произвођач очекује да буде.

Оно што се такође да приметити јесте да нумерички добијен параметар протока при максималној отворености вентила K_{vs} не износи очекиваних $160 \text{ m}^3/\text{h}$, већ $128,77 \text{ m}^3/\text{h}$, што значи да релативна грешка износи $19,52\%$, што није уопште занемарљиво. Гледајући резултате уопштено, може се приметити да највеће одступање у односу на линеарну карактеристику настаје при отворености мањој од 30% , и већој од 80% .

Један од могућих разлога добијеног одступања нумеричких вредности од аналитичких јесте што мали број ћелија који је генерисан (услед ограничења лиценце) потенцијално не даје довољно прецизне резултате. Због релативно малог броја генерисаних ћелија, исте су у одређеним деловима мреже веома неправилних облика, што свакако неповољно утиче на поузданост симулације. Такође, приликом генерисања мреже нису генерисане ћелије које представљају гранични слој (такође услед ограничења лиценце), што такође има утицај на резултате, с обзиром да је утицај граничног слоја битан, поготово када се ради о веома малим удаљеностима између зидова (као што је нпр. удаљеност између регулационог тела и зида вентила, посебно при нижим вредностима отворености вентила).

Битно је такође нагласити да је ова анализа ограничена релативно уским опсегом параметара: симулације су урађене само у једном програму, разматран је само један вентил, један модел турбуленције, као и само једна од могућих варијанти граничних услова.

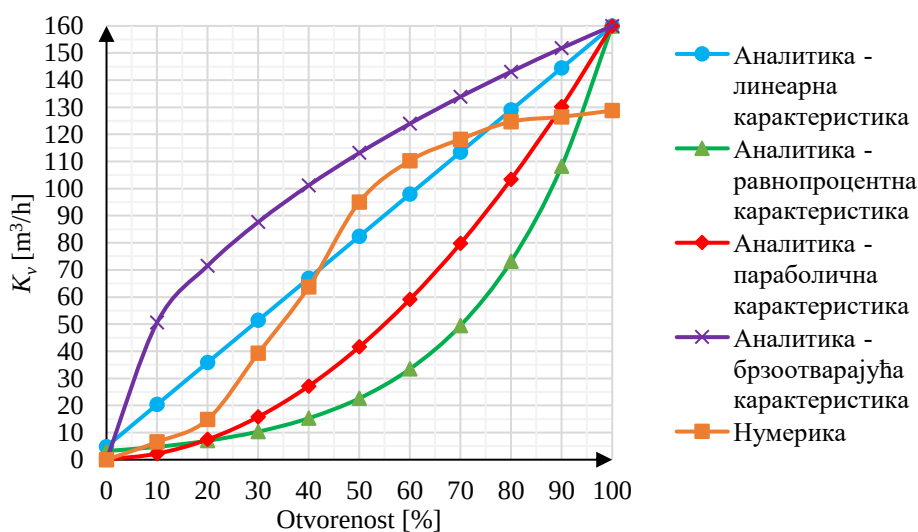
4. ЗАКЉУЧАК

Све наведено указује на потребу даљег развоја овакве нумеричке симулације (у неком другом програму за РДФ, уз повећање броја ћелија, за различите величине вентила, уз различите моделе турбуленције, различите граничне услове, итд.). Пре свега, требало би заправо

испитати карактеристику вентила, али свакако би требало и поновити ову нумеричку симулацију за параметре наведене у овом раду али уз генерисање далеко већег броја ћелија, након чега треба размотрити добијене резултате, како би се могло одлучити о даљем поступку.

Табела 1. Резултати нумеричке симулације и поређење са аналитичким резултатима

$\bar{h}$ [%]	u_{num} [m/s]	Q_{num} [m ³ /s]	K_{vnum} [m ³ /h]	K_{vlin} [m ³ /h]	K_{vvp} [m ³ /h]	K_{vpar} [m ³ /h]	K_{vbo} [m ³ /h]	ε_{lin} [%]	ε_{vp} [%]	ε_{par} [%]	ε_{bo} [%]
0	0	0	0	0	0	0	0	—	—	—	—
10	0,201	0,00181	6,52	20,36	4,73	2.23	50.6	67,98	-37,84	-192.38	87.11
20	0,457	0,00412	14,83	35,88	7	7.46	71.55	58,67	-111,86	-98.79	79.27
30	1,21	0,0109	39,24	51,39	10,35	15.78	87.64	23,64	-279,13	-148.67	55.23
40	1,96	0,01766	63,74	66,91	15,3	27.16	101.19	4,74	-316,6	-134.68	37.01
50	2,93	0,0264	95,04	82,42	22,63	41.62	113.14	-15,31	-319,97	-128.35	16
60	3,4	0,03063	110,27	97,94	33,46	59.15	123.94	-12,59	-229,56	-86.42	11.03
70	3,65	0,03288	118,21	113,45	49,48	79.75	133.87	-4,2	-138,9	-48.23	11.7
80	3,85	0,03468	124,69	128,97	73,17	103.43	143.11	3,32	-70,41	-20.55	12.87
90	3,9	0,03513	126,47	144,48	108,2	130.18	151.79	12,47	-16,89	2.85	16.68
100	3,97	0,03577	128,77	160	160	160	160	19,52	19,52	19.52	19.52

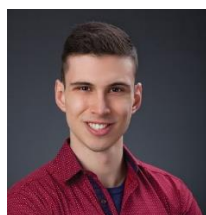


Слика 2. Поређење аналитичких карактеристика вентила са нумеричком

5. ЛИТЕРАТУРА

- [1] https://www.spiraxsarco.com/learn-about-steam/control-hardware-electric-pneumatic-actuation/control-valve-characteristics?sc_lang=en-GB (приступљено у марту 2024.)
- [2] Д. Узелац, Пумпне и компресорске станице, ФТН скрипта, 2014
- [3] <https://www.flocontrol.ltd.uk/what-is-rangeability-and-turn-down-ratio-and-why-is-it-important/> (приступљено у марту 2024.)

Кратка биографија:



Марко Вукић рођен је у Новом Саду 2000. год. Дипломски рад на Факултету техничких наука из области машинства – енергетике и процесне технике одбранио је у јулу 2023. У новембру 2023. год. изабран је у звање сарадника у настави за ужу научну област механика флуида, хидропнеуматска, гасна и нафтна техника, на Катедри за механику флуида и хидропнеуматске системе, у оквиру Департамента за енергетику и процесну технику.

контакт: marko.vukic@uns.ac.rs

**PRORAČUN DVOFAZNOG SEPARATORA U OKVIRU SABIRNO-OTPREMNE
NAFTNE STANICE "IDOŠ"****CALCULATION OF THE TWO-PHASE SEPARATOR WITHIN THE CRUDE OIL
GATHERING AND DISPATCHING STATION "IDOŠ"***Radovan Ratković, Fakultet tehničkih nauka, Novi Sad***Oblast – MAŠINSTVO**

Kratak sadržaj – Predmet rada je proračun dvofaznog sabirnika-separatora (nafta/ voda) u okviru sabirne naftne stanice „Idoš“.

U radu je dat kratak opis sabirne stanice „Idoš“, navedena je najvažnija oprema stanice i izvršeno je dimenzionisanje dvofaznog separatora. U radu je ispraćena zakonska regulativa Republike Srbije iz oblasti rudarstva i prerade i transporta ugljovodonika.

Ključne reči: separatori, nafta, sabirna stanica.

Abstract–The subject of the work is the calculation of the two-phase collector-separator (oil/water) within the crude oil gathering and dispatching station "Idoš". The paper gives a brief description of the gathering and dispatching station "Idoš", the critical equipment of the station is listed and the sizing of the two-phase separator is carried out. The legal regulations of the Republic of Serbia in the field of mining and processing and transportation of hydrocarbons are presented in the paper.

Key words: separators, crude oil, gathering station

1. UVOD

Sabirno otpremna stanica „Idoš“ je centralno postrojenje za prihvatanje i primarnu preradu nafte koje pokriva eksploataciono polje „Idoš“. Eksploataciono polje „Idoš“ je do maja 2018. g. imalo osam pozitivnih bušotina, a trenutno se očekuje priključenje nekoliko novih bušotina. Sirova nafta se od bušotina do sabirno-otpremnice stanice transportuje cjevovodima. Sama sabirno-otpremnica stanica opremljena je kompletnom opremom za primarnu pripremu nafte za transport.

Tehnologija prerade je prilagođena konkretnom tipu sirove nafte. Prerađena nafta se privremeno skladišti u rezervoarima, koji su sastavni dio postrojenja. Iz rezervoara se nafta transportuje autocisternama do eksploatacionog polja „Turija“ odakle se dalje transportuje do naftnog skladišta u Novom Sadu, a potom naftovodom do rafinerije Pančevo na finalnu preradu. Sabirna naftna stanica Idoš nalazi se na prostoru sjevernog Banata i u vlasništvu je „NIS a.d.“

NAPOMENA:

Ovaj rad je proistekao iz master rada čiji je mentor bio prof. dr Slobodan Tašin.

U okviru rada data je tehnološka šema sabirno-otpremnice stanice „Idoš“ i opisani su tokovi fluida od bušotina do rezervoara za naftu. Date su teorijske osnove za proračun dvofaznog vertikalnog separatora kao glavne procesne opreme za pripremu nafte za transport. Prikazane su teorijske osnove proračuna dvofaznog separatora kao i sam proračun dvofaznog vertikalnog separatora. Ostala oprema u sklopu sabirno-otpremnice stanice je detaljno opisana i date su njene osnovne tehničke karakteristike.

2. TEORIJSKE OSNOVE

Sabirno-transportni sistem čine podzemne instalacije, te objekti i uređaji od glave bušotine do otpremne stanice odnosno rafinerije.

Osnovne funkcije nekog sabirno-transportnog sistema za naftu su:

- prikupljanje nafte,
- priprema nafte za transport,
- transport nafte do otpremne stanice odnosno rafinerije.

2.1 Prikupljanje nafte

Prikupljanje, kao jedna od osnovnih funkcija sabirno-transportnog sistema, podrazumijeva transport bušotinskog fluida iz pojedinačnih bušotina do zajedničke lokacije gdje se obavlja priprema za transport. Bušotinski fluid se transportuje ili pojedinačnim naftovodima ili kolektorskim sistemima cjevovoda, što zavisi od mnogo faktora kao što su veličina ležišta, morfologija terena, raspored i broj bušotina te pritisak na glavama bušotina. Smješa nafte i gasa dobijena iz bušotina usmjerava se u mjerne ili sabirne stanice na kojima se vrši odvajanje gasne i tečne faze, mjerenje dobijenih količina nafte i naftnog gasa i skladištenje nafte ako se radi o sabirnoj stanici. [1]

Postoje tri sistema prikupljanja smjese nafte i gasa:

- pojedinačni sistem,
- sistem odvojenih cjevovoda sa više mjernih ili sabirnih stanica i
- zbirni sistem.

Karakteristika pojedinačnog sistema je separacija i mjerenje dobijenih količina nafte i gasa na svakoj bušotini. Sistem se primjenjuje samo u slučaju velikog kapaciteta (davanja) bušotina što opravdava ekonomske troškove. Drugi sistem uključuje više sabirnih ili mjernih stanica pri čemu je svaka bušotina vezana za odgovarajuću sabirnu ili mjernu stanicu

zasebnim cjevovodom. Ovaj sistem naziva se i sistem odvojenih cjevovoda.

Treći sistem prikupljanja podrazumijeva dopremanje smjese nafte i naftnog gasa iz svake bušotine kratkim cjevovodom do zajedničkog cjevovoda (kolektorski cjevovod) kojim se ukupno dobijena količina iz svih bušotina otprema do centralnog sabirnog mjesta. Ovakav sistem pogodan je ukoliko su ostvareni visoki pritisci na glavama bušotina te kada ne postoji ekonomsko opravdanje za korišćenje sistema koji spada u drugu grupu sistema sabiranja.

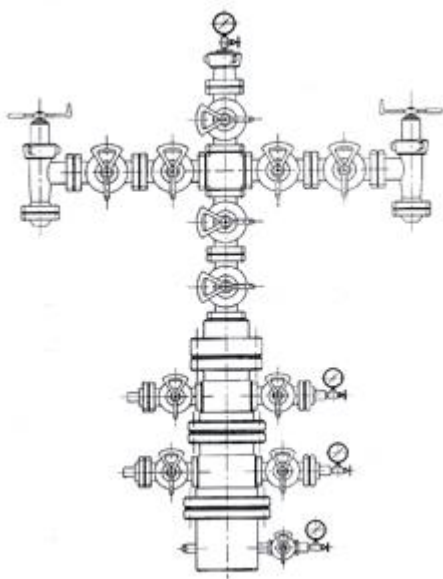
2.2 Transport nafte

Transport nafte podijeljen je na unutrašnji transport, koji obuhvata fazu prikupljanja i pripreme nafte, te magistralni transport, utvrđen kao transport nafte od otpremnih stanica do centralne otpremne stanice. Završnu fazu transporta predstavlja transport iz otpremnih stanica do rafinerija.

3. DIJELOVI TEHNOLOGIJE ZA PROIZVODNJU NAFTE

3.1 Erupcioni uređaj

Erupcioni uređaj je dio opreme na ustima bušotine pomoću kog se vrši eksploatacija bušotinskog fluida i koji omogućavaju održavanje pritiska u bušotini. Zahvaljujući erupcionom uređaju omogućena je direktna i obratna cirkulacija fluida u bušotini.



Slika 1. Erupcioni uređaj

Konstrukcija erupcionog uređaja bazira se na istim principima pa ne postoji neka bitna razlika između uređaja različitih proizvođača. Kod izbora erupcionog uređaja mora se voditi računa o očekivanom pritisku na datoj bušotini, isto tako treba imati u vidu karakter erupcije i količinu pijeska koji dolazi sa fluidom.

3.2 Separatori

Separatori su čelične posude pod pritiskom. Mogu biti smješteni na bušotini, mjernoj ili sabirnoj stanici gdje se vrši prihvati i obrada smjese s obližnjih proizvodnih bušotina. Njihov je glavni zadatak razdvajanje fluida na gasnu i tečnu fazu. Tip separatora koji će se koristiti zavisi najprije o svojstvima i količini fluida, te radnom pritisku. U postupku

odabira separatora treba uzeti u obzir i buduće promjene svojstava proizvedenog fluida, da bi se u kasnijoj fazi proizvodnje izbjegli problemi u radu separatora i mogući dodatni troškovi. Zapjenjenost i korozivnost smjese koja ulazi u separator, tip proticanja i sadržaj pijeska takođe su parametri koji u određenoj mjeri određuju tip separatora ili elemente koji se u njega ugrađuju. [1]

Bez obzira na izgled, tip ili konstrukciju, svi bi separatori trebali omogućavati sljedeće:

- zadržavanje tečnosti radi postizanja fazne ravnoteže (od 3 do 30 minuta);
- visoki stepen iskorišćenja separacije;
- neprekidnost i pouzdanost procesa;
- dovoljnu propusnu moć, tako da na postrojenju ne dolazi do stvaranja „uskog grla“;
- jednostavnu implementaciju automatske regulacije procesa;
- jednostavno održavanje i mogućnost naknadne ugradnje ili izmjene elemenata unutar separatora.

4. OPIS SABIRNO-OTPREMNE STANICE „IDOŠ“

Povezivanje bušotina sa sabirno-otpremnom stanicom Idoš (SOS Idoš) će se izvršiti čeličnim podzemnim cjevovodima DN 65. Na glavi svake bušotine predviđeno je postavljanje stabilne dizne (konstantnog protočnog presjeka) za regulaciju protoka pri eruptivnoj proizvodnji. Bušotinski cjevovodi će u krugu SOS Idoš nakon izlaska iz zemlje biti priključeni na ulazne priključke automatskog mjernog uređaja (AMU) u kojem će se vršiti mjerenje proizvodnje gasa, nafte i slojne vode pojedinačnih bušotina. Predviđena je ugradnja AMU-a sa 14 ulaznih priključaka. Radni pritisak AMU-a kretać se u rasponu od 2 bar do 5 bar. Pritisak na početku bušotinskog cjevovoda određen je pritiskom u separatorskom sistemu SOS Idoš i padom pritiska pri sabiranju fluida koji je uslovljen karakteristikama sabirnog sistema, karakteristikama i protokom faza bušotinskog fluida.

Imajući u vidu geološka svojstva nafte, posebno visoku temperaturu tečenja (prosječna vrednost 27 °C), kao i niske brzine proticanja, protočnost bušotinskih cjevovoda biće obezbijedena:

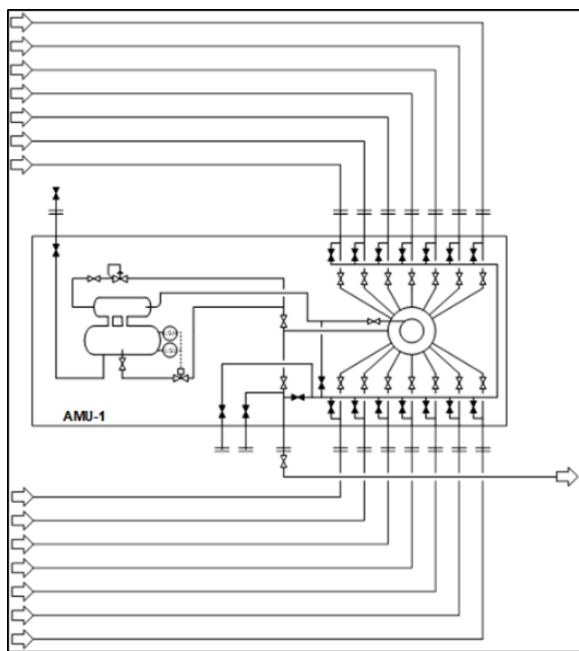
- ugradnjom elektrootpornog grijača nominalne snage 7,5 kW na nadzemnom dijelu svih cjevovoda na lokaciji bušotina,
- toplovodnim grijanjem bušotinskih cjevovoda,
- čišćenjem cjevovoda mehaničkim čistačima,
- termoizolacijom podzemnih i nadzemnih dijelova cjevovoda (nadzemni dijelovi bušotinskog cjevovoda izolovani su slojem mineralne vune debljine 40 mm a podzemni dijelovi cjevovoda izolovani su slojem poliuretanske pjene debljine 60 mm),
- ugradnjom priključaka za ispiranje cjevovoda u slučaju nepredviđenog prestanka proizvodnje.

Parametri transporta fluida od erupcionog uređaja do tačke priključenja na AMU pratiće se preko:

- lokalnog manometra i termoindikatora između erupcionog zasuna i regulacione dizne,
- transmitera pritiska ispred otpremne čistačke slavinne,
- lokalnog manometra između otpremne čistačke slavinne i elektrootpornog grijača,

- lokalnog termoindikatora posle elektrootpornog zagrijača i
- lokalnog manometra i termoindikatora ispred tačke priključenja na AMU.

AMU-1 predstavlja automatski mjerni uređaj protočnog tipa koji funkcionalno zamjenjuje kolektorski i mjerni separatorski sistem. Uređaj je predviđen za sabiranje, mjerenje i daljinsko praćenje proizvodnje bušotinskog fluida. Jedan uređaj čine blokovi tehnologije i automatike, smješteni u zasebne kontejnere. Šema automatskog mjernog uređaja prikazana je na slici 3.



Slika 2. Tipska tehnološka šema AMU-a

5. PRORAČUN SEPARATORA

5.1 Teorijske osnove separacije nafte i gasa separatora i dimenzionisanje separatora

Struja fluida po ulasku u separator se grubo razdvaja na tečnu i gasnu fazu pri čemu, uslijed razlika u gustini, tečna faza ide u donji dio separatora, a gas u gornji dio. U gornjem dijelu separatora – gasnom dijelu – nalazi se gas u kome su zaostale čestice tečne faze koje podliježu gravitacionom izdvajanju, taloženju.

Na tečnu kapljicu sfernog oblika unutar gasne faze (u gornjem dijelu separatora) djeluje sila težine $F_g = m \cdot g$, uslijed čega se kapljica kreće na dole i sila otpora gasa kretanju kapljice, F_D . Ova sila ima suprotan smjer od sile težine. Sila otpora, F_D , zavisi od brzine kretanja kapljice, poprečne površine kapljice (površine upravne na smjer kretanja) i od gustine gasa.

$$F_D = \frac{C_D \cdot A \cdot \rho_g \cdot v_c^2}{2} \quad (1)$$

gdje su:

- F_D - sila otpora
- C_D - koeficijent otpora
- ρ_g - gustina gasne faze kg/m^3
- v_c - brzina taloženja, m/s

A - površina poprečnog presjeka kapljice, m^2 .

U momentu kada se sila težine i sila otpora izjednače, kapljica dobija konstantnu brzinu kretanja koju nazivamo brzina taloženja.

Razlikujemo taloženje sitnih i krupnih čestica. Taloženje sitnih čestica podleže Stoksovom zakonu. To su čestice veličine 3 do $100 \mu\text{m}$. U uslovima kada imamo ovu veličinu tečnih kapljica i kada je strujanje gasa oko kapljice laminarno, $Re_g < 1$ onda je koeficijent otpora $C_D = 24/Re_g$ pa sila otpora ima oblik [1]:

$$F_D = \frac{24}{Re_g} \cdot A \cdot \rho_g \cdot v_c^2 = 3 \cdot \pi \cdot \mu_g \cdot v_c \cdot D \quad (2)$$

Pošto je $v_c = v_g$

Ako silu otpora F_D , izjednačimo sa silom teže F_g , pri čemu je masa sferične kapljice $m = V \rho_c$, a zapremina $V = D^3 \pi / 6$ i uz uzimanje u obzir Arhimedovog zakona jer je tečna kapljica gustine ρ_l u gasu gustine ρ_g , onda je

$$F_D = (D^3 \pi / 6) (\rho_l - \rho_g) g. \quad (3)$$

Izjednačavanjem jednakosti (2) i (3) dobija se izraz za za brzinu taloženja po Stoksovom zakonu:

$$v_c = \frac{D^2 \cdot g (\rho_l - \rho_g)}{18 \cdot \mu_g} \quad (4)$$

gdje su:

- μ_g - viskoznost gasa, Pas
- D - prečnik kapljice, m
- g - gravitaciono ubrzanje $g = 9,81 \text{ m/s}^2$
- ρ_l - gustina tečne faze, kg/m^3
- ρ_g - gustina gasa, kg/m^3
- v_c - brzina taloženja, m/s .

U uslovima kada je veličina kapljice veća od $800 \mu\text{m}$, u automodelnoj oblasti $1000 < Re < 200.000$, koeficijent otpora iznosi $C_D = 0,44$ a brzina taloženja podliježe Njutnovom zakonu [2]

$$v_c = 1,74 \sqrt{\frac{D \cdot g (\rho_l - \rho_g)}{\rho_g}}. \quad (5)$$

Kada se krupnije čestice kreću laminarno, onda je koeficijent otpora $C_D = 18,5/Re_g^{0,6}$.

Ukoliko se taloženje obavlja po Stoksovom zakonu, ali pod dejstvom centrifugalne sile, onda je brzina taloženja jednaka:

$$v_c = \frac{D^2 \cdot a (\rho_l - \rho_g)}{18 \cdot \mu_g} \quad (6)$$

gdje je „a“ ubrzanje izazvano centrifugalnom silom.

Kapljice koje treba da se izdvoje u separatoru iz gasne struje podliježu Stoksovom zakonu. Iz ove jednačine dobijamo brzinu taloženja tečnih kapljica u gasnoj struji, pri

čemu brzina gasa treba da bude nešto malo ispod brzine taloženja, mada se u proračunima uzima da su jednake. U [3] daje se kao osnova za projektovanje modifikacija kod koje koeficijent otpora, C_D , ima sledeću vrijednost:

$$C_D = \frac{24}{Re} + \frac{3}{\sqrt{Re}} + 0,34 \quad (7)$$

pa jednačina za brzinu taloženja kapljica dobija oblik:

$$v_c = 0,0036271 \left[\left(\frac{(\rho_l - \rho_g)}{\rho_g} \right) \frac{d_m}{C_D} \right]^{\frac{1}{2}} \quad (8)$$

gdje su:

d_m - prečnik kapljice

C_D - koeficijent otpora

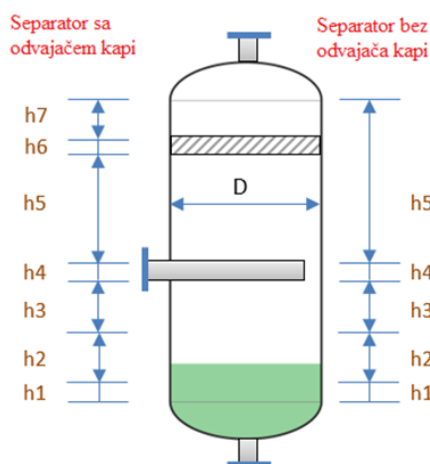
ρ_g - gustina gasa, kg/m³

ρ_l - gustina tečne faze, kg/m³.

5.2 Dimenzionisanje vertikalnog trofaznog separatora na sabirnoj otpremnoj stanici „Idoš“

U nastavku na slici 3 su prikazani tabelarni rezultati proračuna vertikalnog dvofaznog separatora za sabirno-otpremnu stanicu „Idoš“, kao i pozicije pojedinih dimenzija.

ULAZNI PODACI/INPUT			
Protok pare	415,29 Kg/h	Radni pritisak	2 bar
Gustina pare	2,7 Kg/m ³	Kriterijum K vrijednosti	Without Mesl Pad
Protok tečnosti	11254,8 Kg/h	Izračunata K vrijednost	0,053 m/s
Gustina tečnosti	914,8 Kg/m ³	Izračunati prečnik	1100 mm
Vrijeme zadržavanja tečnosti	3 min	Pretpostavljeni prečnik	1200 mm
Kapacitet	0,9 m ³		
L/D maksimum	3		
KONAČNE DIMENZIJE SEPARATORA/SEPARATOR SIZING SUMMARY			
Tip	Bez odvajanja kapi		
D	Prečnik separatora		1200 mm
L	Dužina separatora		3000 mm
h1	Nizak nivo tečnosti do dna T/L		150 mm
h2	Raspon nivoa		950 mm
h3	Visoki nivo tečnosti do ulazne mlaznice		768 mm
h4	Prečnik ulazne mlaznice		114 mm
h5	Od ulazne mlaznice do vrha separatora T/L		1075 mm
L/D			2,5



Slika 3. Dimenzije vertikalnog dvofaznog separatora

6. ZAKLJUČAK

U okviru ovog rada dat je kratak pregled osnovnih dijelova sabirnih stanica za naftu, kao i teorijske osnove proračuna dimenzija dvofaznih separatora za naftu.

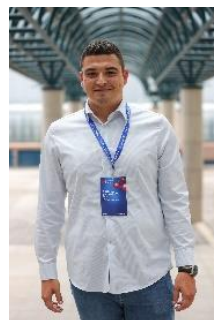
Iz bušotina na naftnom polju bušotinski fluid se transportuje prvo do izmjenjivača toplote, gdje mu se povećava temperatura iznad temperature tečenja. Potom se tako zagrejan fluid transportuje do separatora gdje se vrši separacija na gasnu i tečnu fazu (naftu). Gasna faza jednim dijelom ide do kotlarnice sa gasnim kotlom. Izdvojena tečna faza iz separatora se transportuje do rezervoara, gdje se skladišti do utkanja u autocisternu.

Konstrukcija i dimenzije separatora zavise od vrste i protoka bušotinskog fluida, pritiska, temperature i brzine taloženja kapljica. U radu su date teorijske osnove proračuna i prikazan je proračun osnovnih dimenzija dvofaznog vertikalnog separatora za konkretan fluid koji se eksploatiše sa naftnog polja „Idoš“.

7. LITERATURA

1. Prstojević, B. (2012). Priprema nafte, gasa i ležišnih voda. Rudarsko-geološki fakultet Univerziteta u Beogradu.
2. Zelić, M. (1987). Tehnologija sabiranja i pripreme nafte i plina za transport. INA- Naftaplin, Zagreb.
3. Nedeljković, V. (1965) Eksploatacija naftnih i gasnih ležišta, Deo I - Tehnologija ležišta i proizvodnje, Naftagas, Novi Sad.
4. Nedeljković, V. (1965) Eksploatacija naftnih i gasnih ležišta, Deo II - Metode eksploatacije, Naftagas, Novi Sad.

Kratka biografija:



Radovan Ratković rođen je u Trebinju 1995. god. Osnovne studije iz oblasti Mašinstva završio je na Fakultetu tehničkih nauka u Novom Sadu 2020. god. Master rad iz oblasti mašinstva odbranio je na istom fakultetu 2024. god.

ANALIZA TAČNOSTI I KVALITETA OBRADE PRI SEČENJU LASEROM**ANALYSIS OF ACCURACY AND QUALITY DURING LASER CUTTING**Aleksandra Kosanović, *Fakultet tehničkih nauka, Novi Sad***Oblast – PROIZVODNO MAŠINSTVO**

Kratak sadržaj – Lasersko sečenje je napredni proces čije performanse zavise od mnogobrojnih faktora. Zbog nedostatka pouzdanih praktičnih podataka o ovim faktorima, tehnologija laserskog sečenja često nije dovoljno iskorišćena. U ovom radu je korišćena Taguchi metoda za identifikaciju optimalnih vrednosti ključnih parametara procesa, kao što su brzina sečenja, snaga lasera i pritisak pomoćnog gasa, kako bi se smanjili ugao nagiba reza i srednja aritmetička hrapavost. Predstavljena metodologija analize i optimizacije omogućava preciznije i kvalitetnije upravljanje procesom laserskog sečenja.

Ključne reči: Lasersko sečenje, optimizacija parametara obrade, Taguchi metoda

Abstract – Laser cutting is an advanced process whose performance depends on numerous factors. Due to a lack of reliable practical data on these factors, the laser cutting technology is often underutilized. In this paper, the Taguchi method was used to identify the optimal values of key process parameters, such as cutting speed, laser power, and assist gas pressure, in order to minimize the kerf angle and arithmetic mean roughness. The presented analysis and optimization methodology enables more precise and higher-quality control of the laser cutting process.

Keywords: Laser cutting, machining parameters optimization, Taguchi method

1. UVOD

Iako je lasersko sečenje napredna tehnologija, na njenu efikasnost utiče mnoštvo faktora. Laser se široko koristi u industriji za precizno sečenje, zavarivanje, graviranje i bušenje različitih materijala. Njegova primena obuhvata automobilske, medicinske i elektronske sektore, kao i proizvodnju, gde omogućava visoku tačnost i kvalitet obrade. Ipak, nedostatak pouzdanih praktičnih podataka o ovim faktorima često dovodi do suboptimalne upotrebe ove tehnologije, ograničavajući njen puni potencijal [1].

Brojni autori posvetili su se istraživanju uticaja ulaznih parametara na izlazne performanse obrade. Jedno od takvih detaljnih istraživanja, koje je obuhvatilo proučavanje uticaja parametara na hrapavost površine pri laserskoj obradi nerđajućeg čelika 304, sprovedla je grupa istraživača predvođena Vikasom [2].

Njihovo istraživanje, koje kombinuje eksperimentalne analize sa metodom odzivne površine, omogućilo je du-

binsko ispitivanje odnosa između parametara obrade. Rezultati su pokazali da frekvencija impulsa, snaga lasera, brzina sečenja, kao i interakcije između određenih parametara, značajno utiču na hrapavost površine. Primenom metoda višekriterijumske optimizacije zasnovane na relacionoj analizi u radu istraživani su parametri sečenja vlaknastim laserom čelika nerđajućeg tipa AISI 304L debljine 20 mm [3]. Korišćenjem Taguchi tehnike pojedinačno su optimizovane brzina sečenja, fokalna pozicija, frekvencija i radni ciklus, uz istovremenu optimizaciju hrapavosti površine i širine sečenja. Ova istraživanja doprinose razumevanju procesa laserskog sečenja debelih nerđajućih čeličnih materijala i može imati značajnu primenu u industriji.

U ovom radu [4], autori su proučili uticaj različitih parametara laserskog sečenja na kvalitet površine i širinu reza metala. Poseban fokus bio je na ključnim faktorima ovog tehnološkog procesa, istražujući kako promene u parametrima mogu uticati na konačna svojstva obrađenih materijala. Detaljno su razmotrene karakteristike laserskog sečenja, uključujući brzinu sečenja, snagu lasera i osobine materijala, sa ciljem postizanja optimalne ravnoteže između preciznosti sečenja, kvaliteta površine i širine reza. Rad takođe pruža uvid u moguće izazove i ograničenja koja se mogu javiti tokom procesa laserskog sečenja metala.

Pregled literature pokazuje da su u većini istraživanja analizirani uticaji dva, tri ili četiri faktora procesa (najčešće: snaga lasera, brzina sečenja i pritisak pomoćnog gasa) na jednu, dve ili tri karakteristike procesa, pri čemu su najčešće razmatrani kvalitet reza, širina reza, hrapavost i zona uticaja toplote (HAZ). Eksperimentalni rezultati pokazuju da različiti faktori imaju različite efekte na performanse procesa u različitim intervalima variranja. Dosadašnja istraživanja su se pretežno fokusirala na sečenje metala, polimernih i kompozitnih materijala, kao i keramike. Pregled literature takođe ukazuje da je istraživanje laserske obrade i dalje aktuelno, s obzirom na stalni razvoj tehnologije i potrebu za unapređenjem procesa kako bi se postigle bolje performanse i kvalitet.

U ovom istraživanju detaljno je opisan postupak laserskog sečenja, sa posebnim naglaskom na analizu tačnosti i kvaliteta obrade. U okviru eksperimentalnog istraživanja, na mašini je izrađeno 9 uzoraka od niskougljeničnog čelika, poznatog kao crni lim sa oznakom SRPS EN 10130:2011. Cilj eksperimenta bio je isecanje 9 uzoraka koristeći različite parametre na mašini, uključujući snagu lasera, brzinu sečenja i pritisak pomoćnog gasa. Za analizu i optimizaciju parametara primenjena je Taguchi metoda, koja je korišćena za optimizaciju izlaznih

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Rodić.

performansi, srednje aritmetičke hrapavosti obrađene površine i ugla nagiba reza. Eksperimenti su sprovedeni koristeći Taguči ortogonalni niz $L_9 (3^4)$ za lasersko sečenje crnog lima. Posmatran je uticaj promena brzine, snage lasera i pritiska pomoćnog gasa na kvalitet reza. Takođe, izvršena je analiza rezultata pomoću ANOVA metode.

2. MATERIJAL I METODE

Sečenje materijala je izvedeno na laserskoj mašini "Bystar 3015" poznatoj u industriji zbog svoje preciznosti i efikasnosti. Osnovne karakteristike mašine su: veličina radnog prostora: 3000 x 1500 mm, maksimalna masa obratka: 890 kg, maksimalna debljina obratka: 20 mm, snaga lasera: 3500W.

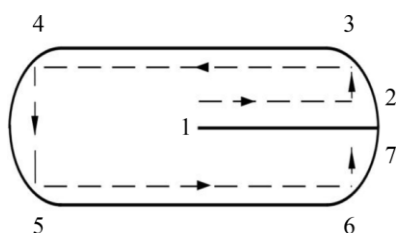
Eksperimenti su izvršeni na toplovaljanom limu od niskougljeničnog čelika, u praksi poznatijem kao crni lim, oznake S235JR (1.0330) prema SRPS EN 10130:2011. Osnovne mehaničke karakteristike ovog čelika su: zatezna čvrstoća $360 \div 510$ MPa, granica razvlačenja 235 MPa, tvrdoća $104 \div 154$ HB, modul elastičnosti 210 GPa. Osnovni hemijski sastav je: Fe ($\leq 98,5\%$), C ($0,17 \div 0,20\%$), Mn ($\leq 1,4\%$), Si ($\leq 0,035\%$), P ($\leq 0,045\%$), S ($\leq 0,045\%$).

Proces je izveden na 9 uzoraka s varijacijama u parametrima: snazi lasera, brzini sečenja i pritisku pomoćnog gasa. Zbog složenosti laserskog sečenja, različiti faktori mogu značajno uticati na kvalitet reza. Na osnovu literature [5] i iskustava operatera, ključni parametri su: snaga lasera, brzina sečenja i pritisak pomoćnog gasa. Pri odabiru vrednosti za ove parametre uzete su u obzir preporuke proizvođača, ograničenja mašine i karakteristike materijala. S obzirom na kompleksnost odnosa između ulaznih parametara i karakteristika reza, primenjena su tri nivoa varijacije za svaki faktor kako bi se postigao potpuni rez materijala debljine 3 mm.

Tabela 1. Vrednosti ulaznih faktora procesa laserskog sečenja

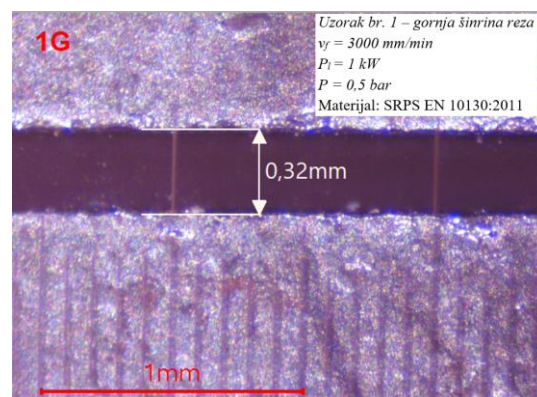
Faktori procesa	Jedinica	Nivo		
		1	2	3
Brzina sečenja, v_f	(m/min)	3000	3400	3800
Snaga lasera, P_l	(kW)	1	1,5	2
Pritisak pomoćnog gasa, P	(MPa)	0,5	0,7	1

Proces laserskog sečenja počinje bušenjem otvora, pozicija 1, zatim kretanjem rezne glave po zadatoj konturi, slika 1.



Slika 1. Prikaz putanje sečenja uzorka

Kao indikator tačnosti procesa laserskog sečenja korišćen je ugao nagiba reza koji se dobija računskim putem merenjem gornje i donje širine reza. Merenje je obavljeno pomoću mikroskopa Euromex Nexius Zoom EVO sa digitalnom kamerom CMEX-5 Pro, koja omogućava visoku preciznost merenja od 0,01 mm sa skaliranjem 1 DIV = 0,01 mm, slika 2. Ova visoka rezolucija omogućava detaljnu analizu površine sečenja, što je važno za postizanje visokog kvaliteta proizvoda.



Slika 2. Prikaz merenja širine reza

Hrapavost površine reza je ocenjena koristeći srednje aritmetičko odstupanje profila R_a . Merenja su izvršena pomoću instrumenta Mitutoyo surfest SJ-210, koji radi po principu povlačenja mernog pipka po površini reza i detektovanja neravnina, slika 3. Ovi rezultati se mogu lako obraditi i kategorizovati putem povezivanja uređaja sa računarnom. Mašina se sastoji od stega za fiksiranje materijala, osnovne strukture mašine, mernog stuba i uređaja s mernim pipkom.



Slika 3. Prikaz merenja površinske hrapavosti uzorka na instrumentu Mitutoyo surfest SJ-210

U radu je korišćena Taguči metoda za jednokriterijumsku optimizaciju ulaznih parametara laserskog sečenja, sa ciljem poboljšanja kvaliteta i smanjenja troškova. Taguči metoda se fokusira na sužavanje parametara oko nominalnih vrednosti, pružajući sistematski pristup optimizaciji bez potrebe za matematičkim modelima. Primena Taguči ortogonalnog niza $L_9 (3^4)$ omogućila je analizu uticaja varijacija u parametrima, kao što su snaga lasera, brzina sečenja i pritisak pomoćnog gasa, na kvalitet reza crnog lima debljine 3 mm.

Rezultati analize prikazali su uticaj ulaznih faktora sa intervalom poverenja od 95%.

Tabela 2. Taguči ortogonalni niz L_9 (3^4) plan eksperimenta i rezultati merenja

R. br.	Faktor			Merene veličine			
	v_f mm/min	P_l kW	P bar	w_g mm	w_d mm	α °	R_a μm
1.	3000	1	0,5	0,35	0,45	0,96	0,68
2.	3000	1,5	0,7	0,47	0,33	1,34	1,1
3.	3000	2	1	0,46	0,33	1,24	1,43
4.	3400	1	0,7	0,3	0,37	0,67	0,96
5.	3400	1,5	1	0,44	0,4	0,38	1,45
6.	3400	2	0,5	0,48	0,43	0,48	1,48
7.	3800	1	1	0,31	0,35	0,38	0,89
8.	3800	1,5	0,5	0,42	0,34	0,76	1,31
9.	3800	2	0,7	0,47	0,37	0,96	3,05

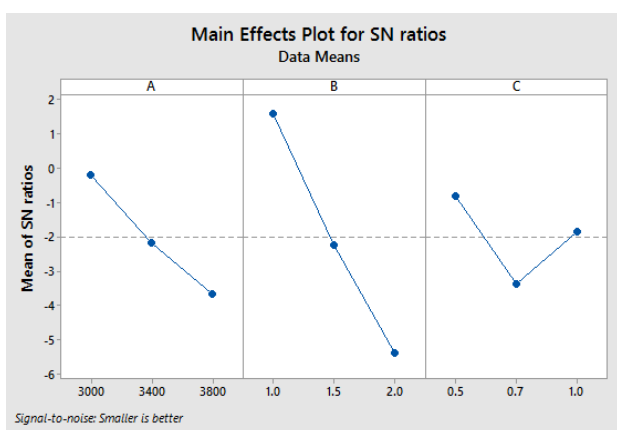
3. ANALIZA REZULTATA

U skladu sa tabelom 2., na osnovu kriterijuma “manje je bolje“, izračunate su S/N vrednosti za svaki nivo ulaznih parametara procesa laserskog sečenja. Rezultati su predstavljeni u tabeli 3.

Tabela 3. Vrednosti S/N odnosa za parametar hrapavosti R_a

Redni broj	Faktori	S/N odnos			max - min	Rang
		Nivoi				
		1	2	3		
1.	A - v_f	-0,20	-2,17	-3,67	3,47	2
2.	B - P_l	1,58	-2,23	-5,39	6,97	1
3.	C - P	-0,81	-3,38	-1,85	2,57	3

Prema analizi tabele 3. i grafika prikazanog na slici 4., izvršena je procena uticaja ulaznih parametara na srednju aritmetičku hrapavost obrađene površine. Rezultati ukazuju da snaga lasera ima najveći uticaj na hrapavost reza, dok brzina sečenja i pritisak pomoćnog gasa pokazuju manji uticaj.



Slika 4. Grafik glavnih uticaja S/N vrednosti na parametar hrapavosti R_a

Zaključeno je da optimalna kombinacija za postizanje minimalne hrapavosti obuhvata brzinu sečenja od 3000 mm/min, snagu lasera od 1 kW i pritisak gasa od 0,5 bara. Prema Tagučijevoj metodi, ova kombinacija daje S/N odnos od 4,59 i predviđenu hrapavost od 0,32

mikrometra, što garantuje visok kvalitet reza. Optimalni parametri iz tabele 3. i slike 4. su: A (nivo 1), B (nivo 1), C (nivo 1), dok je optimalno podešavanje za hrapavost R_a prikazano u tabeli 4.

Tabela 4. Optimalno podešavanje ulaznih faktora za parametar hrapavosti R_a

Ulazni parametri	Nivo	Podešavanje parametara	R_a dobijeno primenom Tagučij metode
A - v_f	1	3000	$S/N=4,59$ $R_a=0,32$
B - P_l	1	1	
C - P	1	0,5	

U tabeli 5. prikazana je početna ANOVA analiza srednje aritmetičke hrapavosti reza pri sečenju crnog lima debljine 3 mm. Rezultati pokazuju da snaga lasera ima najveći uticaj na hrapavost, sa učešćem od 51,97%. Brzina sečenja je na drugom mestu, sa uticajem od 18,63%, dok pritisak gasa ima najmanji uticaj, od 13,12%. S obzirom na to da su u analizu uključena samo tri faktora (četvrta kolona iz Tagučij plana L_9 (3^4) je isključena iz analize) što je dovelo do procentualnog učešća greške od 16,28%. Analiza se smatra uspešnom kada je procenat greške manji od 50.

Tabela 5. ANOVA analiza za parametar hrapavosti R_a

Faktor	Stepen	Suma kvadrata	Varijacija	F test	Procenat %
A	2	0,71	0,36	1,16	18,63
B	2	1,98	0,99	3,23	51,97
C	2	0,50	0,25	0,81	13,12
Greška	2	0,61	0,31		16,28
Ukupno	8	3,81			

Prikazana ANOVA analiza potvrđuje rezultate dobijene primenom Tagučij metode, ističući da su snaga lasera i brzina sečenja ključni faktori koji utiču na hrapavost reza, dok pritisak pomoćnog gasa igra manju, ali i dalje relevantnu ulogu u procesu.

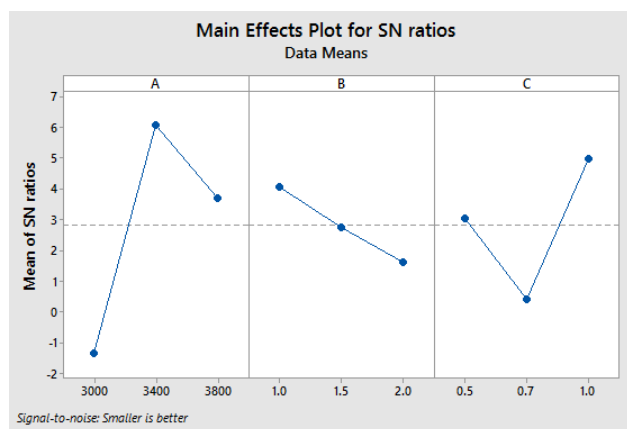
Analizom podataka iz tabele 2. i vrednosti S/N odnosa za svaki nivo ulaznih parametara procesa laserskog sečenja, procenjen je uticaj pojedinačnih faktora na ugao nagiba reza. Rezultati, prikazani u tabeli 6. i grafiku sa slike 5., pokazuju da brzina sečenja ima najveći uticaj, dok su pritisak pomoćnog gasa i snaga lasera manje značajni.

Tabela 6. S/N vrednosti faktora za ugao nagiba reza α

Redni broj	Faktori	S/N odnos			max - min	Rang
		Nivoi				
		1	2	3		
1.	A - v_f	-1,35	6,09	3,71	7,44	1
2.	B - P_l	4,08	2,75	1,62	2,46	3
3.	C - P	3,04	0,43	4,98	4,55	2

Na osnovu grafičkih prikaza sa slike 5., vrednosti S/N odnosa i analize ugla nagiba krive, preporučuje se optimalna kombinacija parametara: brzina sečenja 3400 mm/min, snaga lasera 1 kW i pritisak gasa 1 bar. Ova kombinacija, prema Tagučijevoj metodi, daje S/N odnos

od 9,51 i predviđeni ugao nagiba reza od 0,25 stepeni. Takva konfiguracija ne samo da omogućava precizno i efikasno sečenje, već i minimizuje nepravilnosti na površini reza, što je ključno za postizanje visokog kvaliteta finalnog proizvoda.



Slika 5. Grafik glavnih uticaja S/N vrednosti na ugao nagiba reza α

Optimalna kombinacija nivoa faktora koja minimizuje ugao nagiba reza, prema tabeli 7., iznosi A (nivo 2), B (nivo 1) i C (nivo 3), što predstavlja najpouzdaniji set za postizanje maksimalnih performansi u procesu laserskog sečenja.

Tabela 7. Optimalno podešavanje ulaznih faktora za ugao nagiba reza α

Ulazni parametri	Nivo	Podešavanje parametara	α dobijeno primenom Taguchi metode
A - v_f	2	3400	S/N=9,51 $\alpha=0,25$
B - P_l	1	1	
C - P	3	1	

ANOVA analiza iz tabele 8. potvrđuje da brzina sečenja dominira sa 70,2% uticaja na ugao nagiba reza, dok pritisak pomoćnog gasa doprinosi sa 16,83%, a snaga lasera sa 7,92%. Ovi podaci jasno ukazuju na ključnu ulogu brzine sečenja u optimizaciji procesa za postizanje minimalnog ugla nagiba reza.

Tabela 8. ANOVA analiza za ugao nagiba reza

Faktor	Stepen	Suma kvadrata	Varijacija	F test	Procenat %
A	2	0,71	0,36	17,51	70,2
B	2	0,08	0,04	1,93	7,92
C	2	0,17	0,09	4,28	16,83
Greška	2	0,04	0,02		
Ukupno	8	1,01			

Snaga lasera najviše utiče na hrapavost reza jer direktno određuje količinu prenete energije. Optimalno podešena snaga obezbeđuje bolje topljenje materijala, smanjujući nepravilnosti i rezultira bolji kvalitet obrade, dok prevelika snaga može izazvati pregrevanje i oštećenja. Brzina sečenja smanjuje vreme interakcije lasera i

materijala – brže sečenje može dovesti do manje kontrole i veće hrapavosti reza. Pritisak pomoćnog gasa uklanja rastopljeni materijal iz zone sečenja. Viši pritisak poboljšava uklanjanje materijala, ali prevelik pritisak destabilizuje proces i povećava hrapavost.

Brzina sečenja je ključna za ugao nagiba reza jer prebrzo sečenje smanjuje ravnomernost prenosa energije, povećavajući nagib. Niža brzina omogućava stabilnije sečenje i manji nagib. Pritisak pomoćnog gasa takođe utiče na ugao nagiba reza. Optimalan pritisak daje ravnomerno uklanjanje materijala, dok prevelik ili premali pritisak povećava nagib. Snaga lasera najmanje utiče, ali veća snaga omogućava dublje sečenje, dok preniska snaga može povećati nagib reza.

4. ZAKLJUČAK

Na osnovu rezultata, optimalna kombinacija faktora za smanjenje srednje aritmetičke hrapavosti reza je brzina sečenja (A) na nivou 1, snaga lasera (B) na nivou 1 i pritisak pomoćnog gasa (C) na nivou 1, što odgovara snazi od 1 kW, brzini od 3000 mm/min i pritisku od 0,5 MPa. Za minimizaciju ugla nagiba reza, optimalni parametri su brzina sečenja (A) na nivou 2, snaga lasera (B) na nivou 1 i pritisak gasa (C) na nivou 3, što odgovara brzini od 3400 mm/min, snazi od 1 kW i pritisku od 1 bar. ANOVA analiza pokazuje da snaga lasera ima najveći uticaj na hrapavost reza sa 51,97%, dok brzina sečenja i pritisak gasa utiču sa 18,63% i 13,12%. Za ugao nagiba reza, brzina sečenja je dominantna sa 70,2%, dok pritisak gasa i snaga lasera imaju manji uticaj, sa 16,83% i 7,92%. Zaključak je da laserska obrada, uz optimizaciju parametara, može značajno poboljšati preciznost i kvalitet, zadovoljavajući visoke standarde industrije.

5. LITERATURA

- [1] M. Gostiurović, "Nekonvencionalni postupci obrade", Fakultet tehničkih nauka, Novi Sad, 2012.
- [2] Magdum, V. B., V. M. Nandedkar, and A. D. Vibhute, "Surface roughness optimization in laser machining of stainless steel 304 using response surface methodology," *Materials Today: Proceedings*, Vol. 59, pp. 540-546, 2022.
- [3] D. Milovanović, A. Milosavljević, Ž. Radovanović, and R. Jović, "Laser treatments of Nimonic 263 nickel-based superalloy," **Zastita Materijala**, Vol. 60, No. 1, pp. 26-43, 2019.
- [4] F. A. Al-Sulaiman, B. S. Yilbas, and M. Ahsan, "CO2 laser cutting of a carbon/carbon multi-lamelled plain-weave structure," **Journal of Materials Processing Technology**, Vol. 173, No. 3, pp. 345-351, 2006.
- [5] M. Madić, "Matematičko modeliranje i optimizacija procesa laserskog sečenja primenom metoda veštačke inteligencije," Ph.D. dissertation, 2013.

Kratka biografija:

Aleksandra Kosanović rođena je u Zvorniku 2000. godine. Diplomirala je na Fakultetu tehničkih nauka u Novom Sadu, na smeru Proizvodno mašinstvo, 2023. godine. Master rad iz oblasti Proizvodnog mašinstva – „Analiza tačnosti i kvaliteta obrade pri sečenju laserom“, odbranila je 2024. godine.

Kontakt: kosanovic00@gmail.com

GPT ARHITEKTURA I PRIMENA U SOFTVERSKOJ INDUSTRIJI**GPT ARCHITECTURE AND APPLICATION IN THE SOFTWARE INDUSTRY**

Ana Atanacković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – GPT modeli su modeli za predviđanje jezika opšte namene. To su kompjuterski programi koji mogu da analiziraju, izdvajaju, sumiraju i na drugi način koriste informacije za generisanje sadržaja. Oni kreiraju tekst sličan čoveku, a da nisu eksplicitno programirani da to urade. Kao rezultat, mogu se podesiti za niz zadataka obrade prirodnog jezika, uključujući odgovaranje na pitanja, prevod jezika i rezimiranje teksta.

Ključne reči: GPT, Transformer, veštačka inteligencija

Abstract – GPT models are general purpose language prediction models. These are computer programs that can analyze, extract, summarize and otherwise use information to generate content. They create human-like text without being explicitly programmed to do so. As a result, they can be tuned for a range of natural language processing tasks, including answering questions, translating languages and summarizing text.

Keywords: GPT, Transformer, AI

1. UVOD

GPT jeste skraćenica za *Generative Pre-trained Transformers*. Generativno označava tehnologiju sposobnu da proizvodi sadržaj, kao što su tekst i slike. Prethodno obučeno označava sačuvane mreže koje su već naučene da reše problem ili da ostvare određeni zadatak koristeći veliki skup podataka. Transformer označava bazu GPT modela. Modela za predviđanje jezika zasnovanog na neuronskim mrežama koji je izgrađen na arhitekturi Transformer [1]. Ovakvi modeli analiziraju upite prirodnog jezika i predviđaju najbolji mogući odgovor na osnovu njihovog razumevanja jezika. Oni su prethodno obučeni za ogromne količine podataka, kao što su knjige i veb stranice, da bi se generisao kontekstualno relevantan i semantički koherentan jezik. To su računarski programi koji mogu da kreiraju tekst sličan čoveku, a da nisu eksplicitno programirani da to urade. Kao rezultat, mogu se podesiti za niz zadataka obrade prirodnog jezika, uključujući odgovaranje na pitanja, prevod jezika i rezimiranje teksta.

U okviru ovog teksta provešćemo detaljnu istragu o tome kako izgleda arhitektura GPT-a i kako funkcioniše tok obrade nekog unosa, kako bi se dobio valjani ishod.

NAPOMENA:

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

Zatim ćemo se fokusirati na softverski aspekt GPT-a, gde ćemo pokazati konkretne primere korišćenja.

2. MODEL GPT-A

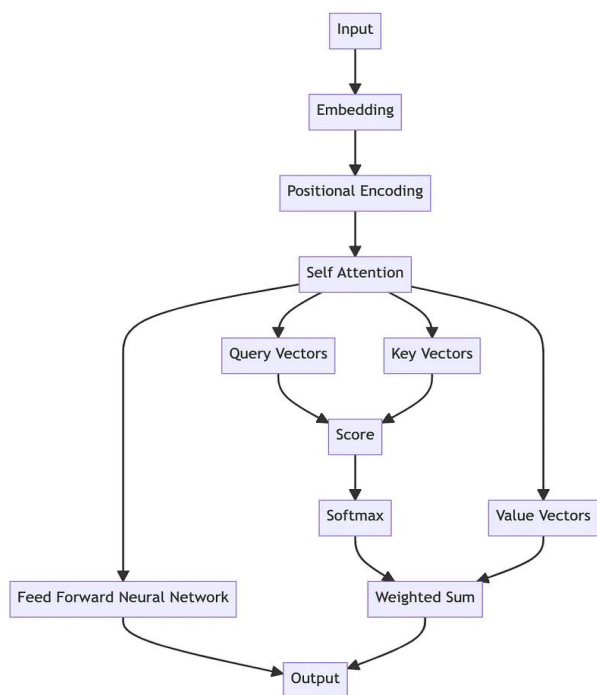
GPT arhitektura predstavlja značajan korak napred u obradi prirodnog jezika, omogućavajući mašinama da generišu tekst sličan čoveku i obavljaju bezbroj zadataka vezanih za jezik. Temelj ovog modela bazira se na arhitekturi Transformer, koja je objavljena još 2017-te godine [2]. Sledeća poglavlja otkriće ključne komponente modela, kao i njegove osnovne mehanizme.

Ovaj model prolazi kroz dva koraka pre njegovog korišćenja, obuku i usavršavanje. Tokom obuke, model je izložen ogromnoj količini raznovrsnih tekstualnih podataka, koje mu omogućavaju da shvati kompleksnost jezika i njegovih sastavnih veza. Ključnost ove faze leži u učenju bez nadzora, tako da model sam razvije široko razumevanje jezičkih termina i njihovih mogućih značenja. Za namene usavršavanja, potrebno je prvo izabrati odgovarajuće zadatke, koji si planirani da se izvrše pomoću ovog modela. Onda kada imamo krajnji cilj, model će se prilagoditi specifičnom zadatku za koji je adaptiran. Ova prilagodljivost čini GPT svestranim alatom za različite aplikacije za obradu prirodnog jezika.

2.1. Transformer

Transformatori su postali baza mnogih najsavremenijih modela obrade prirodnog jezika zbog svoje sposobnosti da efikasno pronađu zavisnosti i veze između podataka. Model Transformer sastoji se od enkodera i dekodera, od kojih je svaki sastavljen od niza identičnih slojeva, koji se sastoje od dva podsloja (videti izgled na slici 1.):

1. Višeglavi samosvesni mehanizam – omogućava modelu da odmeri i kodira različite reči u ulaznoj sekvenci na osnovu njihove relevantnosti za trenutnu reč koja se obrađuje, i koristeći više glava detektuje različite tipove odnosa između reči. Time se podržava paralelna obrada (model se istovremeno fokusira na različite delove ulazne sekvence) i poboljšano prikazivanje učenja. Ovaj mehanizam pomaže da se nauči o kontekstualnom odnosu između reči.
2. Poziciono povezana mreža za napredovanje – standardna neuronska mreža koja se primenjuje nezavisno za svaku poziciju, kako bi se promenio izlaz samosvesnog sloja. Ova mreža pomaže da se nauči o samim rečima.



Slika 1. Model transformera

Struktura po slojevima omogućava modelu da uhvati hijerarhijske karakteristike i složene paterne u podacima. Ovim se olakšava i skalabilnost, davajući prostora za proširenje modela u neke kompleksnije modele.

2.2. Samosvestan mehanizam

Sam temelj arhitekture Transformera jeste mehanizam pažnje/svesnosti, koji hvata zavisnosti i odnose između bilo koje dve tačke u nizu, bez obzira na njihovu udaljenost. Sa ovim, moguće je iskoristiti i znanje iz prošlosti, kako bi se uticalo na buduća predviđanja. Ovaj mehanizam izračunava i težinu koja predstavlja važnost ili relevantnost tačke u nizu. Tako se model fokusira na najrelevantnije delove ulaznih podataka za predviđanje.

Razumejući ovo, vidimo da on hvata zavisnosti i odnose unutar ulaznih sekvenci, i omogućava modelu da identifikuje i odmeri važnost različitih delova ulazne sekvence bivajući samosvestan.

Višeglavi samosvestan mehanizam podržava paralelizam. Ovo ga čini mnogo bržim i efikasnijim kada se radi o velikim količinama ulaznih parametara. Ipak, on ima i kvadratnu računsku složenost u odnosu na dužinu sekvence, što može učiniti mehanizam neefikasnim za veoma duge sekvence.

Samosvesni slojevi mogu pružiti oblik interpretabilnosti, pošto težine pažnje mogu dati uvid u to koje parametre model smatra važnim za svako predviđanje. Oni se koriste u obradi prirodnog jezika, za akcije prevodenja, sumiranja i analize osećanja.

2.3. Poziciono kodiranje

Tehnika koja se koristi da se modelu daju informacije o relevantnosti položaja reči u rečenicama - poziciono kodiranje. Transformatori su zasnovani na samosvesnim mehanizmima, koji sami po sebi ne uzimaju u obzir redosled reči u rečenicama, te se poziciono kodiranje

dodaje ulaznim ugrađenim elementima (rečima), pre nego što se proslede u samosvesni sloj. Ovim model može da uhvati sekvencijalnu prirodu podataka, koja je ključna za zadatke koji zahtevaju razumevanje reda reči i konteksta.

2.4. Računanje težine

Već prethodno napomenuta, težina ili značaj ulaznog parametra, može se matematički sračunati. Kada gledamo ceo ulazni niz, važnost svake reči za trenutnu reč određuje se na osnovu njihovog semantičkog i sintaksnog odnosa. Ova mera se računa u koraku obuke modela, i može da se menja u zavisnosti od konteksta u kome se reč koristi. Drugi naziv za nju jeste i ocena pažnje.

Ako uzmemo za primer rečenicu „Rad je završen u toku prethodnog dana.“, kada se kodira reč „rad“, samosvestan mehanizam može da oceni reč „završen“ sa visokom ocenom, misleći da je za razumevanje konteksta reči „rad“ ova reč bitna.

Unos u sloj se transformiše u vektore upita, ključa i vrednosti kroz množenje sa tri različite matrice težine. Vektori upita i ključa se koriste za izračunavanje težine, a vektori vrednosti za kreiranje kombinacije težine na osnovu ovih rezultata. Kombinacije težine su ujedno i izlazi ovog sloja.

2.5. Ugrađivanje reči

Postoji više načina na koje se reč može numerički predstaviti, jedan od njih je ugrađivanjem reči. Ovaj pristup koristi Transformer. Svaka reč u rečniku se preslikava u vektor realnih brojeva, nasumično izabranih, koji kreira njenu ugradnju reči. Tokom treninga, ugradnja se menja u odnosu na kontekst u kom je reč korišćena. Slične ugradnje reči bi na primer bile „diplomski rad“ i „master rad“, jer su pronađene da se često koriste u rečenicama o radovima. Ovakav vid obrade zove se kontekstualno učenje.

Model pravi predviđanja na osnovu trenutnih ugradnji, a zatim se izračunava greška između ovih predviđanja i stvarnih vrednosti. Ova greška se zatim koristi za prilagođavanje ugrađivanja kako bi se predviđanja približila stvarnim vrednostima. Ovim se ostvaruje proces nazadnog širenja znanja modela. Ovakav model je fleksibilan i jednostavan, kako za ovakvu akciju nije potrebno neko predznanje. On uključuje rad unazad od izlaza do ulaza, kako bi se otkrilo kako smanjiti broj grešaka i učiniti model pouzdanijim. Ovakvi algoritmi primenljivi su za mnoge scenarije, što ih čini idealnim metodom za poboljšanje performansi neuronskih mreža.

2.6. Višeglava svesnost

Višeglava svesnost je ključna komponenta arhitekture modela Transformatora, koja se koristi da omogući modelu da se fokusira na različite vrste informacija. Mehanizam svesnosti se primenjuje više puta paralelno (različite glave su aktivne), i time se dobijaju drugačiji tipovi veza između podataka.

Svaka glava pažnje ima sopstvene linearne transformacije koje primenjuje na ugrađen unos kako bi dobila svoje upite, ključeve i vrednosti. Ovim glave bivaju nezavisne i

mogu učiti odvojene stvari, te se jedna grana može fokusirati na sintaksne veze a druga na semantičke. Svi rezultati (izlazi glava), spajaju se i linearno se transformišu za konačan rezultat.

Svaka glava će biti nezavisna, ne namerno već na osnovu nasumične (stohastičke) prirode procesa obuke, kao i činjenice da svaka glava počinje se drugačijim slučajno izabranim parametrima. Sa ovim, potencijalno moguće je da se ostvari nejedinstvenost grana, ali šanse za to su male.

3. PRIMENE U SOFTVERU

GPT pomaže programerima da poboljšaju svoju produktivnost, kreativnost i kvalitet svog rada. On može automatizovati zadatke kao što su pisanje koda, otklanjanje greški, testirati i dokumentovati, edukovati i odgovarati na pitanja [3]. Isto tako, on nudi sugestije i ocenjuje već postojeće rezultate [4].

3.1. Generisanje koda

Sposobnost generisanja koda na osnovu opisa prirodnog jezika jedna je od boljih funkcija GPT-a. Programeri mogu da obezbede model sa opisom željene funkcionalnosti, a on će generisati odgovarajući kod u ciljnom programskom jeziku. Programeri mogu da opišu svoje zahteve ili da traže specifične šablone koda, a GPT može da generiše šablone koda ili da pruži primere implementacije kako bi pomogao u procesu razvoja. Ova sposobnost ubrzava proces razvoja, smanjuje ljudske greške i omogućava programerima da se fokusiraju na složenije zadatke. Sa pružanjem jasnih i sažetih opisa mogu se dobiti bolji rezultati upita. Ovo će pomoći modelu veštačke inteligencije da generiše tačan i efikasan kod uz minimiziranje potencijalnih nesporazuma.

3.2. Optimizacija koda

GPT može da koristi tehnologije obrade prirodnog jezika da brzo analizira i identifikuje oblasti koda koje treba poboljšati. Pruža se lako razumljiv korisnički interfejs za programere da identifikuju funkcionalne specifikacije, nefunkcionalne zahteve, slučajeve upotrebe, arhitekturu softvera, obrasce dizajna, implementaciju i testiranje. Dobijaju se preporuke za lako kreiranje i testiranje različitih verzija koda, dok se ne ispune željeni nivoi performansi. Na taj način se štedi vreme, novac i energija, kreiranjem optimizovanog koda u manjem broju iteracija nego bez pomoći modela.

3.3. Automatski pregledi koda i asistencija

Pregledi koda su suštinski deo procesa razvoja softvera, obezbeđujući da kod bude visokog kvaliteta i da se održava u skladu sa standardima kodiranja. Ovaj model se može koristiti za automatizaciju pregleda koda otkrivanjem potencijalnih problema, kao što su sintaksne greške, nesavršenosti u stilu, neefikasni algoritmi i potencijalni bezbednosni propusti. Ova automatizacija smanjuje vreme i trud koje programer mora da uloži na ručne preglede koda, omogućavajući mu da se fokusira na vrednije zadatke.

GPT može da analizira kod za potencijalne ranjivosti kao što su *SQL injection* i *XSS*. Takođe, može da otkrije potencijalne logičke greške i mrtav kod, smanjujući vreme potrebno za otklanjanje grešaka kod testa. Ove funkcije olakšavaju programerima da identifikuju i poprave probleme u kodu pre nego što uđu u produkciju.

3.4. Testiranje

GPT modeli, sa svojim sofisticiranim razumevanjem jezika i konteksta, sada su uključeni u različite scenarije testiranja. U obrazovanju, oni pomažu u kreiranju i ocenjivanju testova, nudeći personalizovane povratne informacije učenicima. U razvoju softvera, on pomaže u generisanju i izvršavanju test slučajeva, povećavajući efikasnost i pokrivenost. Sposobnosti modela da oponašaju ljudske reakcije takođe ih čini idealnim za testiranje scenarija korisničke službe i čet-bota.

3.5. Otklanjanje grešaka na bazi AI-a

Programeri mogu da unesu delove svog koda u GPT, a model može da analizira ove isečke da bi identifikovao greške ili predložio poboljšanja. Ovaj proces se ne odnosi samo na pronalaženje sintaksnih grešaka, on takođe može da predloži bolje prakse kodiranja ili efikasnije načine za obavljanje zadataka.

GPT može pomoći u identifikaciji uobičajenih i složenih grešaka u kodu. Što je još važnije, on objašnjava ove greške na način koji je razumljiv i početnicima i iskusnim programerima. Ovaj obrazovni aspekt pomaže programerima ne samo da poprave svoje trenutne probleme već i da uče iz svojih grešaka. Kada se problem identifikuje, GPT predlaže više rešenja ili zaobilazna rešenja. Ova mogućnost je posebno korisna u scenarijima u kojima može postojati više od jednog načina za rešavanje problema, omogućavajući programerima da izaberu onaj koji najbolje odgovara njihovom specifičnom kontekstu.

3.6. Dokumentovanje i deljenje znanja

GPT je u mogućnosti da pojednostavi proces dokumentovanje generisanjem tačne i precizne dokumentacije analizom generisanog koda. Tehnologija razume zahteve korisnika i kreira dokumentaciju sa minimalnim znanjem. Isto tako, sposoban je da napravi tutorijale i da odgovori na često pitana pitanja, čime programer uspeva da brže shvati nove koncepte i tehnologije.

3.7. Komunikacija

GPT, čiji zadatak jeste da bude inteligentni posrednik komunikacionog toka, može automatski rezimirati duže diskusije, odgovoriti na pitanja koja su kontekstualno povezana sa diskusijama i predložiti potencijalna rešenja za postojeće probleme u razgovoru.

Jezičke barijere mogu ometati efikasnu komunikaciju i saradnju. GPT ima potencijal da premosti ove praznine pružanjem mogućnosti prevodenja u realnom vremenu [5].

3.8. Personalizovana obuka

Analizom skupa veština svakog programera i praćenjem preferenci učenja GPT može da generiše personalizovane planove učenja koji se fokusiraju samo na relevantne teme i resurse. Uz ovakvo učenje, programeri uspevaju da nauče nove koncepte brže i efikasnije, povećavajući njihovu produktivnost i doprinos timu.

3.9. Kontrola verzije

Krucijalna komponenta za razvoj većih projekata, gde je uključen veći broj programera, jeste kontrola koda, tj. njegov izgled u svakom momentu. Za te potrebe vrši se kontrola verzije koda. GPT takođe nudi lako skladištenje, upravljanje i deljenja koda u okviru svojih projekata razvoja softvera, praćenje napretka pojedinaca i identifikovanje konflikata u kodu.

Moguće je i vratiti se na prethodne verzije koda, kao i vršiti automatske izgradnje, testiranja i puštanja koda. Kada se pristupi nekoj verziji koda, možemo dobiti uvid u detaljne informacije verzije, za potrebe analize i deljenja.

3.10. Kontinuirana integracija

Kontinuirana integracija je proces koji spaja sve promene od programera u jednu glavnu granu u bazi koda. GPT olakšava programerima da identifikuju greške, prate napredak i testiraju promene pre stizanja koda do glavne grane, automatizacijom zadataka kao što su pokretanje testova i pravljenje slika.

3.11. Puštanje koda

GPT pomaže pri puštanju koda, te on upravlja paketima i integracijom, pruža razne prilagođene usluge razvoja softvera kao što su primena u *Cloud*-u i servisi podrške. To znači da kompanije mogu biti sigurne da se njihova prilagođena rešenja primenjuju ispravno i na vreme. Isto, on ima snažan sistem praćenja koji omogućava programerima da prate performanse svojih rešenja u proizvodnji.

3.12. Analiza podataka

GPT model može pomoći analitičarima da efikasno sakupi velike količine podataka. Jezički modeli pronalaze tražene podatke i izračunavaju i prikazuju rezultate u tabeli sa podacima. Neke aplikacije mogu i da iscrtaju rezultate na grafikonu ili da kreiraju sveobuhvatne izveštaje [6].

3.13. Organizacija projekta

Da bi razvoj softvera bio završen efikasno i na vreme, kao i da bi se upravljalo resursima i budžetima, potrebni su menadžeri projekta. GPT može da obezbedi moćno rešenje za upravljanje projektima za razvoj softvera. Automatizacijom radnji kao što su zakazivanje sastanaka, praćenje resursa i rokova, on omogućava menadžerima projekata da pojednostave procese, davajući im obaveštenja i upozorenja.

4. ZAKLJUČAK

GPT je prilično mlado otkriće koje se razvija ubrzanim tempom, ali nema sumnje, da će usled popularnosti i uspeha, ova oblast biti sve više izučavana i unapređivana. Očekujemo sve veće pogodnosti ovog modela, u vidu aplikacija i alata. Što se tiče softvera, već sada vidimo ogromnu količinu slučajeva korišćenja u realnom sistemu. Isto tako postoje i rizici na koje moramo ozbiljno posvetiti pažnju. Moramo obezbediti bezbednost naših podataka i shvatiti moguću nepouzdanost nekih alata GPT-a. Adekvatna protekcija i pametno korišćenje GPT-a, proizvešće odlične rezultate za bilo koju oblast.

5. LITERATURA

- [1] https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- [2] https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- [3] <https://arxiv.org/pdf/2311.07361.pdf>
- [4] <https://arxiv.org/pdf/2302.14520.pdf>
- [5] <https://arxiv.org/pdf/2304.01852.pdf>
- [6] <https://aws.amazon.com/what-is/gpt/>

Kratka biografija:



Ana Atanacković rođena je u Novom Sadu 1998. god. Studijski program Računarstvo i automatika, upisala je 2017. godine, a završila 2021. godine. Nakon toga upisuje master akademske studije Primenjene računarske nauke i informatika, iz oblasti Elektrotehnike i računarstva.

НУМЕРИЧКИ МОДЕЛ ЗА ДИНАМИЧКЕ ПРОЦЕСЕ ТРОФАЗНЕ КАВЕЗНЕ АСИНХРОНЕ МАШИНЕ**NUMERICAL MODEL FOR DYNAMIC PROCESSES OF THREE-PHASE SQUIRREL CAGE INDUCTION MACHINE**Владан Живановић, Дејан Јеркан, *Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

Кратак садржај – Тема рада јесте динамички модел трофазне кавезне асинхроне машине са уважањем ефектима којима доприноси специфичности конструкције, као што је утицај жлебова на статору и ротору разматране машине. Извршене су симулације противструјног кочења и кочења једносмерном струјом на моделу који је развијен у MATLAB Simulinku. Такође, тако добијени резултати су упоређени са теоријским разматрањима.

Кључне речи: Кавезна асинхрона машина, метода коначних елемената, динамички модел, противструјно кочење,

Abstract – In this paper a dynamic model of a three-phase squirrel cage induction machine is developed while considering the effects which are contributed by the specificity of the construction, such as the influence of the stator's and rotor's slots of the considered machine. On the model developed in MATLAB Simulink, simulations of plugging and DC injection breaking were performed. Also, the results thus obtained were compared with theoretical considerations.

Keywords: Squirrel cage induction machine, Finite Element Analysis, dynamic model, DC injection breaking.

1. УВОД

Асинхрони мотори су најчешће коришћена врста електричних машина, нарочито у савременој индустрији, где представљају њену погонску снагу. У циљу побољшања рада, асинхроне машине су на путу до савремене изведбе претрпеле низ измена. Имају широку примену у индустрији, поготово у погонима који захтевају изузетно брзо и прецизно управљање. Такође, њихова употреба је распрострањена и у домаћинствима, то јест различитим врстама кућних апарата. Главни недостатак асинхроних машина и њихова препрека за ширу примену је био мали опсег промене брзине обртања. Развојем и применом уређаја енергетске електронике, постало је могуће у ширем опсегу управљати брзином обртања асинхроне машине. Самим тим предност погона са машинама једносмерне струје, пренесена је на погоне са асинхроним машинама.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Дејан Јеркан.

Познавање динамичког модела машине је потребно за решавање проблема управљања генераторима и моторима, за пројектовање заштите и надзора, за одређивање структуре и параметара регулације у роботизи, као и за решавање проблема у аутоматизацији производње, електричним возилима и другим сличним применама [1]. Математичка репрезентација система који се посматра, односно математички модел треба да буде једноставан, прегледан и употребљив. Дакле, потребно је да модел задржи битне физичке карактеристике моделоване машине и истовремено суштински осликава природу појава у њој. Са друге стране, потребно је да довољно детаљан модел буде погодан и за имплементацију у рачунару (рачунарске симулације). Значај математичког моделовања се огледа у анализирању различитих радних режима машине, развијања сложених алгоритама управљања истих и проучавања добијених карактеристика.

2. МАТЕМАТИЧКИ МОДЕЛ ТРОФАЗНЕ КАВЕЗНЕ АСИНХРОНЕ МАШИНЕ

С обзиром на то да се у овом раду говори о математичком моделу трофазног кавезног асинхроног мотора, при чему је модел динамички, јер омогућава проучавање транзијентних феномена у електричним машинама, неопходно је одредити улазне параметре модела применом методе коначних елемената. Сам модел се ослања на скуп релација које су познате под именом вишеструко спрегнута електрична кола (енгл. *Multiple Coupled Circuit Approach*) [2]. Математички модел је у оригиналном, нетрансформисаном моделу и као такав је повољан и довољно детаљан за проучавање свих појава и процеса везаних за изучавање трофазне кавезне асинхроне машине. Модел је математичка репрезентација система и као такав описује понашање истог, међутим да би се добила његова упрошћена представа потребно је занемарити небитне утицаје, то јест неопходно је уважити одређене апроксимације [3]: појаве у електричној машини се могу описати довољно детаљно помоћу сконцентрисаних параметара, појаве капацитивног карактера су, као и снага губитака у гвожђу занемарљиве, док се феромагнетски материјал у магнетском колу сматра линеарним. Омски отпори статора и ротора су константни, као и моменат инерције ротирајућих маса. Из разматрања се изузимају и струје цурења које се затварају ван штапних проводника. Нема закошења штапних проводника.

Једна од чињеница, свакако јесте да ротор (роторски кавез) нема електричних приступа, те је проблем на први поглед, пронаћи адекватан начин за дефинисање свих појединачних електричних кола кавеза. Проблем се решава представљањем електричног кола ротора помоћу роторских петљи. Једну роторски петљу сачињава затворен проводни пут који се састоји од две суседна роторска штапа и два наспрамна сегмента краткоспојних прстенова [4].

Математички модел се састоји од електричног и механичког подсистема. Реч је о електричним колима са временски променљивим струјама, те се диференцијална једначина напонске равнотеже у матричној форми може записати помоћу следеће релације:

$$\mathbf{u} = \mathbf{R} \cdot \mathbf{i} + \frac{d\boldsymbol{\Psi}}{dt} \quad (1)$$

$\mathbf{u}$ – вектор напона свих електричних кола којима је разматрана електрична машина моделована,

$\mathbf{i}$ – вектор струје статорских намотаја и свих роторских петљи, укључујући и струје краткоспојног прстена,

$\boldsymbol{\Psi}$ – вектор свих флуksних обухвата статора и ротора,

$\mathbf{R}$ – матрица свих статорских и роторских отпорности

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_s & \\ & \mathbf{R}_r \end{bmatrix} \quad (2)$$

$\mathbf{R}_s$ – дијагонална субматрица статорских отпорности димензија 3×3 и

$\mathbf{R}_r$ – субматрица роторских отпорности димензија $(N_B+1) \times (N_B+1)$.

Ако се са N_B означи број роторских петљи (роторских штапних проводника), потребно је израчунати N_B+1 струја у роторском колу, уважавајући и краткоспојне прстенове. Субматрица статорских отпорности $\mathbf{R}_s$ је дијагонална матрица која је проста и генерише се тривијално. Генерисање субматрице роторских отпорности $\mathbf{R}_r$ није тривијално, зато што матрица није дијагонална и она описује топологију роторског кавеза, тј. начин успостављања напонске равнотеже у роторским петљама.

Саставни део електричног подсистема математичког модела је и једначина флуksног обухвата, која даје везу између струја и флуksева преко нестационарне матрице индуктивности:

$$\boldsymbol{\Psi} = \mathbf{L} \cdot \mathbf{i} \quad (3)$$

Матрицу индуктивности $\mathbf{L}$ трофазне асинхроне машине чине четири субматрице, представљена је у следећем облику:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{ss} & \mathbf{L}_{sr} \\ \mathbf{L}_{rs} & \mathbf{L}_{rr} \end{bmatrix} \quad (4)$$

$\mathbf{L}_{ss}$ – субматрица свих сопствених и међусобних индуктивности намотаја статора,

$\mathbf{L}_{sr}$ – субматрица статорско-роторских индуктивности,

$\mathbf{L}_{rr}$ – субматрица свих сопствених и међусобних роторских индуктивности и

$\mathbf{L}_{rs}$ – субматрица роторско-статорских индуктивности. Субматрица статорских индуктивности $\mathbf{L}_{ss}$ има димензије 3×3 и на њеној главној дијагонали се налазе елементи сопствених индуктивности намотаја статора, док на позицијама вандијагоналних елемената стоје коефицијенти међусобних индуктивности. Уважавање утицаја жлебова доводи до постојања

варијација у магнетској пермеабилности по ободу ваздушног зазора. Самим тим елементи матрице $\mathbf{L}_{ss}$ нису непроменљиви, већ су сложенопериодичне функције положаја ротора. Субматрица роторске индуктивности $\mathbf{L}_{rr}$ је знатно сложеније форме (увек сложенопериодична функција положаја ротора). Димензија ове матрице су $(N_B+1) \times (N_B+1)$, због изузетно великог броја међусобних спрега унутар роторског кавеза. Суштински значај субматрице статорско-роторских $\mathbf{L}_{sr}$ и роторско-статорских $\mathbf{L}_{rs}$ индуктивности је заправо та могућност да се помоћу њих описује доминантна компонента оствареног електромагнетског момента. Субматрице међусобних индуктивности, димензија $3 \times (N_B+1)$ су симетричне, па важи да је $\mathbf{L}_{sr} = \mathbf{L}_{rs}^T$. Таласни облици елемената субматрице међуиндуктивности су идентични, са одговарајућим фазним помаком између елемената врста матрице $\mathbf{L}_{sr}$ који износи $2\pi/N_B$, док су елементи колоне међусобно фазно померени за угао $2\pi/3$. Међутим, то не важи за елементе последње колоне у матрици $\mathbf{L}_{sr}$, нулти елементи указују на одсуство магнетске спреге између статорских намотаја и петље краткоспојног прстена.

Након електричног подсистема, неопходно је дефинисати и механички подсистем да би математички модел трофазне кавезне асинхроне машине био употпуњен. Израз за електромагнетни моменат, односно моменат конверзије, представљен је следећом релацијом:

$$M_{em} = \frac{1}{2} \cdot p \cdot \begin{bmatrix} \mathbf{i}_s \\ \mathbf{i}_r \end{bmatrix}^T \cdot \frac{d}{d\theta} \left(\begin{bmatrix} \mathbf{L}_{ss} & \mathbf{L}_{sr} \\ \mathbf{L}_{rs} & \mathbf{L}_{rr} \end{bmatrix} \right) \cdot \begin{bmatrix} \mathbf{i}_s \\ \mathbf{i}_r \end{bmatrix} \quad (5)$$

M_{em} – електромагнетни моменат машине,

p – број пари полова трофазне асинхроне машине и

θ – електрични угао који описује тренутни положај ротора у односу на статор.

Математичком моделу, тј. његовом механичком подсистему се мора придодати и Њутнова диференцијална једначина за случај ротационог кретања, која је прилагођена моторском режиму рада.

$$M_{em} - M_m = J \cdot \frac{d\omega_m}{dt} + K_m \cdot \omega_m \quad (6)$$

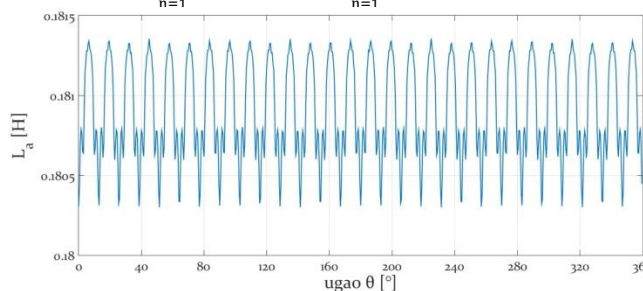
3. ОДРЕЂИВАЊЕ ЕЛЕКТРИЧНИХ ПАРАМЕТАРА КАВЕЗНЕ АСИНХРОНЕ МАШИНЕ

Познавање елемената свих матрица индуктивности и отпорности је неопходно ради извршавања рачунарска симулација над наведеним развијеним математичким моделом. Отпорности намотаја се добијају експериментално или адекватним прорачунима. За одређивање непознатих индуктивности машине користи се метода коначних елемената која подразумева спорвођење низа магнетостатичких симулација. То се најчешће реализује помоћу програмског пакета FEMM 4.2 [5]. Извођење магнетостатичких симулација подразумева побуђивање одређених намотаја или роторских петљи једносмерном струјом. Периодична зависност свих елемената матрице индуктивности од положаја ротора, потиче од уважавања постојања жлебова и промене релативног међусобног положаја намотаја статора и роторских петљи. Пошто су елементи матрице индуктивности сложенопериодичне функције положаја ротора, вредности индуктивности добијене магнетостатичким симулацијама се могу представити преко коефицијената Фуријеовог реда. Наиме, развој у Фуријеов ред је дефинисан искључиво за

континуалне функције, те се добијени дискретни одбирци могу добити као континуалне функција применом линеарне интерполације.

Израз за развој сопствене индуктивности статорског намотаја фазе а у Фуријеов ред дат је релацијом (7), а пример израчунавања ове индуктивности дат за конкретну машину је приказан на слици 1. Уколико се израчуна сопствена индуктивност статорске фазе а, сопствене индуктивности преостале две фазе се добијају фазним померајем за $-2\pi/3$, односно $+2\pi/3$,

$$L_a(\theta) = L_{ss0} + \sum_{n=1}^{\infty} L_{ssan} \cdot \cos(n\theta) + \sum_{n=1}^{\infty} L_{ssbn} \cdot \sin(n\theta) \quad (7)$$



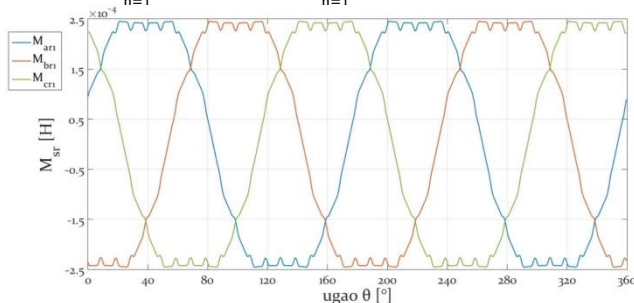
Слика 1. Сопствена индуктивност статорске фазе а

Израз за развој међусобне индуктивности статорских намотаја M_{ab} у Фуријеов ред дат је релацијом (8), док су релације за развој индуктивности M_{ac} и M_{bc} исте, али фазно померене за $-2\pi/3$, односно $+2\pi/3$, респективно.

$$M_{ab}(\theta) = M_{ss0} + \sum_{n=1}^{\infty} M_{ssan} \cdot \cos(n\theta) + M_{ssbn} \cdot \sin(n\theta) \quad (8)$$

Слика 2 приказује таласне облике међусобних индуктивности статорских намотаја фаза а, б и с и прве у низу роторских петљи унутар краткоспојеног кавезног ротора разматране машине (роторска петља са индексом 1), а израз за развој индуктивности M_{ar1} у Фуријеов ред дат је релацијом (9). С обзиром на то да су елементи врста субматрице статорско-роторских индуктивности L_{sr} међусобно фазно померени за $2\pi/3$, таласни облици представљених индуктивности су такође померени за исти угао.

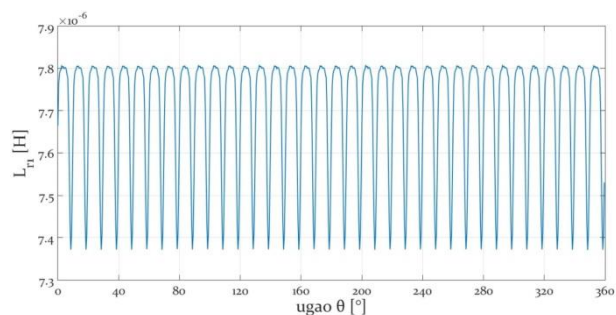
$$M_{ar1}(\theta) = \sum_{n=1}^{\infty} M_{sran} \cdot \cos(n\theta) + \sum_{n=1}^{\infty} M_{srbn} \cdot \sin(n\theta) \quad (9)$$



Слика 2. Статорско-роторске индуктивности

Израз који представља развој у Фуријеов ред сопствене индуктивности прве у низу роторских петљи унутар кавезног ротора машине дат је релацијом (10), док је пример израчунавања ове индуктивности приказан на слици 3.

$$L_{r1}(\theta) = L_{rr0} + \sum_{n=1}^{\infty} L_{rran} \cdot \cos(n\theta) + \sum_{n=1}^{\infty} L_{rrbn} \cdot \sin(n\theta) \quad (10)$$



Слика 3. Сопствена индуктивност прве роторске петље

Све преостале сопствене индуктивности роторских петљи имају идентичан таласни облик, уз одговарајући фазни помак између суседних роторских петљи за угао $2\pi/N_B$. Израз за развој међусобних роторско-роторских индуктивности у Фуријеов ред, дат је следећом релацијом:

$$M_{rjk}(\theta) = M_{ij-k|0} + \sum_{n=1}^{\infty} M_{ij-k|an} \cdot \cos(n\theta) + M_{ij-k|bn} \cdot \sin(n\theta) \quad (11)$$

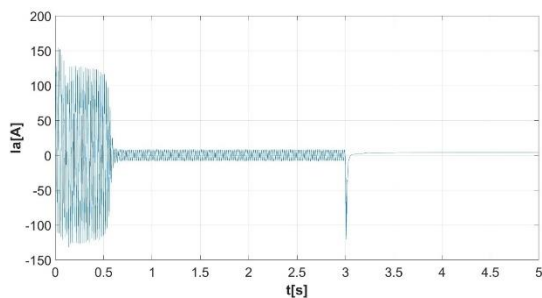
4. КОЧЕЊЕ У ПОГОНУ СА ТРОФАЗНИМ АСИНХРОНИМ МОТОРОМ

Погони са асинхроним моторима се у појединим индустријским системима заустављају природним путем, што се постиже искључењем напајања мотора. Међутим, наведени начин кочења није прихватљив, због честе потребе за скраћењем времена кочења. У условима електричног кочења мотор развија момент који делује тако да погон успорава **Error! Reference source not found.** Стога се електрично кочење намеће као једноставно и ефикасно решење. Са становишта практичне примене, електрично кочење у погонима са асинхроним моторима се традиционално реализује као противструјно кочење, генераторско кочење (рекуперативно и отпорничко) и кочење једносмерном струјом. Када је реч о кавезним асинхроним моторима, противструјно кочење (енгл. *counter-current braking* или *plugging*) се постиже променом смера обртања обртног магнетног поља укрштеним превезивањем двеју фаза мотора. Отпорничко кочење (енгл. *rheostatic braking*), као врста динамичког кочења, примењује се када се асинхрона машина напаја преко погонског претвараача чији улазни степен не може да врати генерисану електричну енергију у мрежу. Алтернативу отпорничком кочењу, као врста динамичког кочења, представља кочење једносмерном струјом (енгл. *DC injection braking*). Изводи се тако што се статорски намотаји одвоје од извора наизменичног напона напајања и повежу на извор једносмерног напона. У роторским проводницима се индукује електромоторна сила, услед стојеће магнетопобудне силе коју ствара једносмерна струја која се успоставља у намотајима статора.

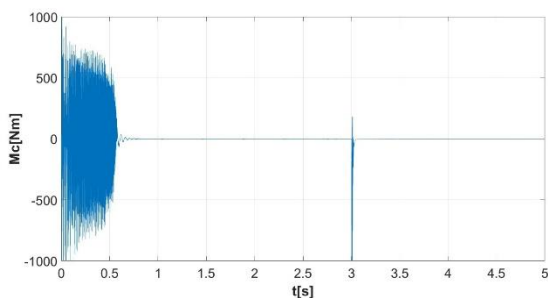
5. МАТЛАВ СИМУЛАЦИЈЕ НА РАЗВИЈЕНОМ МОДЕЛУ МАШИНЕ

За реализацију динамичког математичког модела трофазне кавезне асинхроне машине, користи се програмски пакет MATLAB/Simulink који омогућава истраживање свих релевантних појава и феномена који се одвијају унутар машине као средства електромеханичке конверзије путем најразличитијих рачунарских симулација. На примеру логичке замене реалног система у форми математичког модела, рачунарске симулације служе за прибављање

података и информација о различитим појавама које се одвијају унутар електричне машине без стварних догађаја.



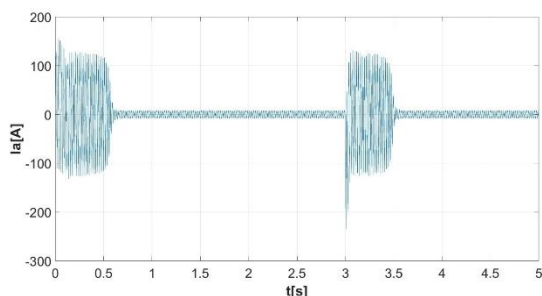
Слика 4. Фазна струја мотора током кочења једносмерном струјом



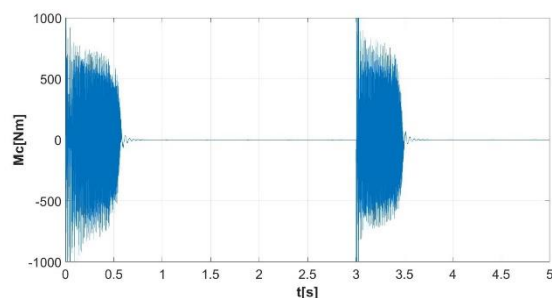
Слика 5. Момент конверзије током кочења једносмерном струјом

Дијаграм струје кроз један прикључни вод мотора приказан је на слици 4. Вредност струје је нагло порасла након краће безнапонске паузе и довођења напона на прикључне крајеве мотора, што је последица електромоторне силе у намотајима статора због постојања струје у намотајима ротора које нису у потпуности ишчезле у тренутку довођења једносмерног напона. На слици 5 приказана је динамичка моментна карактеристика мотора који се кочи једносмерном струјом. Велике осцилације у оствареном моменту су директна последица сложенопериодичних одзива струја и флуксева.

Дијаграм фазне струје (фаза а) приликом противструјног кочења приказан је на слици 6. На датој слици, може да се уочи скоковита промена струје мотора у тренутку након укрштеног превезивања фаза, што је последица промене електромоторне силе услед промене смера ротације. На слици 7 приказана је динамичка моментна карактеристика мотора који се кочи противструјно. Након промене смера ротације, мотор генерише негативан моменат, што указује на обраћање мотора у супротном смеру услед ефекта противструјног кочења.



Слика 6. Фазна струја мотора током противструјног кочења



Слика 7. Момент конверзије током противструјног кочења

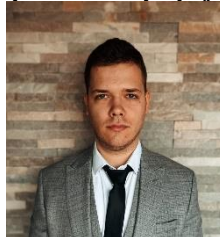
6. ЗАКЉУЧАК

У овом раду је развијен математички модел трофазне кавезне асинхроне машине. Модел је динамички, јер омогућава проучавање транзијентних појава у електричним машинама и ослања се на скуп релација које су познате као вишеструко спрегнута електрична кола. Улазни параметри модела се добијају применом методе коначних елемената. Приказани су одзиви фазних струја мотора и момента конверзије добијени симулацијом противструјног кочења и кочења једносмерном струјом неоптерећене машине са номиналним моментом инерције. Могућа су и бројна друга истраживања и изучавања најразличитијих феномена и процеса који се манифестују током рада трофазне кавезне асинхроне машине.

7. ЛИТЕРАТУРА

- [1] Слободан Н. Вукосавић, „Електричне машине“, Универзитет у Београду, Електротехнички факултет, „Академска мисао“, 2010.
- [2] Дејан Рељић, „Откирвање кvara ротора кавезног асинхроног мотора применом техника анализе терминалних величина“, докторска дисертација, Факултет техничких наука, Нови Сад, 2017.
- [3] Владан Вучковић, „Општа теорија електричних машина“, ИП „Наука“, 2009.
- [4] Дејан Јеркан, „Динамички модел трофазне кавезне асинхроне машине заснован на методи коначних елемената“, докторска дисертација, Факултет техничких наука, Нови Сад, 2016.
- [5] <http://www.femm.info/wiki/HomePage>
- [6] Дарко Марчетић, Петар Матић, Дигитално регулисани електромоторни погони, Академска мисао, Београд, 2020.

Кратка биографија:



Владан Живановић рођен је у Ужицу 1998. год. Дипломирао је на Факултету техничких наука у Новом Саду 2022. године. Мастер студије из области Електротехника и рачунарство – Енергетска електроника и електричне машине је уписао школске 2022/23. године.

SISTEM ZA PRAĆENJE I ANALIZU ŠAHOVSKIH PARTIJA SA PLATFORME LIČES U REALNOM VREMENU**SYSTEM FOR REAL-TIME TRACKING AND ANALYSIS OF CHESS GAMES FROM THE LICHESS PLATFORM**

Milena Laketić, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj - U ovom radu predstavljen je sistem za obradu i analizu podataka o šahovskim partijama prikupljenih sa platforme Ličes [1]. Glavni cilj sistema je omogućiti uvid u kvalitet partije tokom njenog odigravanja na platformi Ličes, kao i identifikacija potencijalno sumnjivih aktivnosti igrača. Kvalitet poteza analizira se koristeći šahovski alat Stokfiš [2], a igrači se na osnovu metričkih vrednosti aktivnosti mogu klasifikovati kao sumnjivi. Sistem je projektovan da bude skalabilan kako bi omogućio prikupljanje i obradu podataka u realnom vremenu.

Cljučne reči: Šah, Ličes, Stokfiš analiza, Detekcija prevara, Kafka Strims, Obrada podataka u realnom vremenu, Vizualizacija rezultata obrade.

Abstract - This paper introduces a system for processing and analyzing data on chess games collected from the Lichess platform. The main goal of the system is to provide insights into the quality of the game during its play on the Lichess platform, as well as to identify potentially suspicious activities of players. The quality of moves is analyzed using the chess tool Stockfish, and players can be classified as suspicious based on the metric values of their activities. The system is designed to be scalable to enable real-time data collection and processing.

Keywords: Chess, Lichess, Stockfish analysis, Fraud detection, Kafka Streams, Real-time data processing, Visualization of processing results.

1. UVOD

Postoji širok spektar softvera dostupnih za igru i analizu šaha, uključujući desktop programe, mobilne aplikacije i internet platforme. Funkcionalnosti koje nude obuhvataju: šahovske baze podataka, mogućnost definisanja treninga i kurseva, proučavanje odigranih partija i igranje protiv drugih korisnika. Obično su korisnicima dostupne analize samo sopstvenih partija, dok je analiza ograničena na već završene igre. Motivacija za razvijanje sistema za analizu šahovskih partija u realnom vremenu proizilazi iz potrebe za naprednim alatom koji obezbeđuje ne samo praćenje partija u trenutku njihovog održavanja, već i mogućnosti stručne analize i detekcije nepravilnosti ili potencijalnih prevara tokom igre.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Dimitrieski, vanr. prof.

Sistem opisan u ovom radu usmeren je ka analizi brzih, nekoliko-minutnih partija, omogućavajući detaljne uvide u dinamiku i strategije specifične za ovu kategoriju igara. Ličes, kao jedna od vodećih šahovskih platformi, pruža bogat izvor podataka o igračima i partijama koje se odvijaju na platformi. Cilj ovog rada je pružiti korisnicima, uključujući igrače, trenere i organizatore turnira na Ličesu, mogućnost da prate igru i analiziraju poteze uz visok stepen pouzdanosti upotrebom šahovskog alata Stokfiš.

Sistemi za obradu podataka u realnom vremenu suočeni su sa raznim izazovima. Sistem mora obrađivati podatke sa minimalnim kašnjenjem kako bi obezbedio pravovremene uvide. Visoko kašnjenje može učiniti podatke zastarelim, naročito u aplikacijama gde odluke moraju biti donete trenutno. Sistem mora biti u mogućnosti da horizontalno skalira kako bi se nosio sa sve većim obimom podataka. Sa ovim u vidu, obezbeđivanje konzistentnosti podataka u distribuiranim sistemima je izazovno, naročito kada se radi o podacima koji pristižu velikom brzinom. Na kraju, sistemi u realnom vremenu moraju biti otporni na greške. Svaki zastoj može dovesti do gubitka kritičnih podataka i uticati na pouzdanost sistema, pa je otpornost na otkaze veoma važna karakteristika [3].

Cljučne komponente sistema za detekciju anomalija uključuju monitoring ponašanja korisnika kao što su vreme prijave, aktivnosti na sistemu i sl. U kontekstu projektovanog sistema to su aktivne šahovke partije, odigrani potezi i slično. Zatim, definisanje tipičnog ponašanja korisnika kao i identifikacija i računanje relevantnih karakteristika koje mogu ukazati na nedozvoljeno ponašanje.

2. PREGLED TRENUTNOG STANJA U OBLASTI

U ovom poglavlju predstavljeni su neki od najkorišćenijih softvera za analizu šaha u trenutku pisanja rada,

ChessBase predstavlja jedan od popularnijih šahovskih softvera i postoji u vidu desktop i mobilne aplikacije. To je plaćeni softver koji omogućava korisnicima skladištenje i analizu šahovskih partija. Za analizu koristi alate kao što su Stokfiš, Komodo i *Fritz* [4]. Namenjen je profesionalnim igračima i trenerima.

Internet platforme za šah dostupne su direktno preko web pregledača, što korisnicima pruža veću pristupačnost i fleksibilnost u korišćenju. Po broju korisnika izdvajaju se platforme Ličes i Chess.com. Ličes je besplatna onlajn platforma koja nudi igranje šaha protiv drugih igrača, kao i funkcionalnosti poput obuka i analiza odigranih partija.

Korisnici mogu učestvovati u različitim organizovanim turnirima, gde se igre mogu pratiti uživo preko *Lichess TV*. Partije su kategorizovane po vremenu dostupnom igračima. Jedna od kategorija ima naziv Metak i predstavlja partije u kojima je svakom od igrača dostupan jedan minut za igru.

Projektovani sistem nadovezuje se na internet platformu Ličes uvodeći analizu igre u realnom vremenu. Omogućava gledaocima da prate partije drugih igrača uz stručne komentare pomoću alata Stokfiš, nudeći uvid u kvalitet odigranih poteza. Omogućava način praćenja aktivnosti igrača kroz metrike kao što su prosečna preciznost igrača, odnos dobrih ka lošim potezima i drugo. Značajnije metrike predstavljene su u [poglavlju 3](#). Dodatno, ovaj sistem pruža mehanizme za detekciju potencijalnih varalica bez potrebe za kreiranjem i treniranjem složenih neuronskih mreža, čime se poboljšava integritet i fer-plej na onlajn platformi.

3. SPECIFIKACIJA ZAHTEVA

Projektovani sistem može biti od velike koristi za različite korisnike platforme Ličes. **Menadžer platforme** ima ključnu ulogu u nadzoru popularnosti i performansi platforme, fokusirajući se na praćenje zdravlja platforme i aktivnosti korisnika, kao i na praćenje funkcionalnosti Ličes TV, što uključuje broj učesnika, partija i poteza. Takođe otkriva i reaguje na bilo kakve anomalije, kao što je odsustvo aktivnih igara i slično.

Organizator turnira upravlja i organizuje šahovske turnire, pri čemu su mu potrebni uvidi u realnom vremenu o ponašanju i performansama igrača. Ova uloga uključuje praćenje specifičnih metrika za otkrivanje sumnjivih aktivnosti ili varanja u realnom vremenu.

Analitičar šaha bavi se analizom istorijskih podataka prikupljenih iz šahovskih igara, sa ciljem generisanja novih saznanja o šahu, naročito u kontekstu brzih partija. Ovo uključuje identifikovanje trendova i obrazaca u podacima o igri.

Igrači ili gledaoci turnira, zainteresovani za detaljne metrike performansi, koriste podatke za analizu performansi i ponašanja protivnika, pregled ključnih indikatora performansi i za razvoj novih strategija i tehnika igre.

Projektovani sistem omogućava predstavljenim profilima uvid u igre preko funkcionalnih zahteva predstavljenih u narednom poglavlju.

3.1 Funkcionalni zahtevi

Kako bi podržao specifične zahteve korisnika, projektovani sistem prikuplja relevantne podatke o šahovskim partijama iz kategorije Metak sa servera Ličes. Ova kategorija odlikuje se kratkim trajanjem od samo jednog minuta po igraču. Ove partije generišu znatno više događaja zbog brzine igre, što direktno utiče na količinu podataka koje sistem mora efikasno obraditi u realnom vremenu.

Profilisanje igrača ključno je za omogućavanje gledaocima i igračima da procene kvalitet i taktiku tokom igre, kao i organizatorima turnira i menadžerima

platforme da identifikuju anomalije ili atipična ponašanja igrača. To podrazumeva praćenje svakog odigranog poteza i ocenjivanje njegovog kvaliteta pomoću alata Stokfiš, koji poteze kategorizuje u različite klase, od briljantnih do veoma pogrešnih, u zavisnosti od njihove evaluacije. Aktivnost igrača predstavljena je kroz ključne indikatore performansi (KPI). Metrike obuhvataju broj poteza po kategoriji, kao i izvedene metrike kao što su preciznost igrača tokom partije. Metrike koje se računaju za svakog igrača i ažuriraju izračunatim vrednostima za pojedinačnu partiju, prikazane su u [tabeli 3.1](#).

Standardna devijacija igračeve preciznosti posebno je korisna za procenu varijabilnosti u performansama igrača, omogućavajući detekciju potencijalne upotrebe nedozvoljenih alata tokom igre. Igrači sa visokim varijacijama u preciznosti između partija mogu biti označeni za dalju analizu i monitoring.

Naziv metrike	Definicija
Odnos pobeda ka porazima	Broj pobeda / Broj poraza
Odnos ispravnih ka neispravnim potezima	Broj ispravnih / Broj neispravnih poteza
Srednja vrednost igračeve preciznosti	Broj ispravnih / Ukupan broj poteza
Makro preciznost igrača	Prosečna vrednost igračeve preciznosti u svim partijama
Medijan igračeve preciznosti	Pedeseti percentil svih zabeleženih preciznosti
Standardna devijacija igračeve preciznosti	Rasipanje igračeve preciznosti u odnosu na srednju vrednost

Tabela 3.1. Ključni indikatori performansi igrača

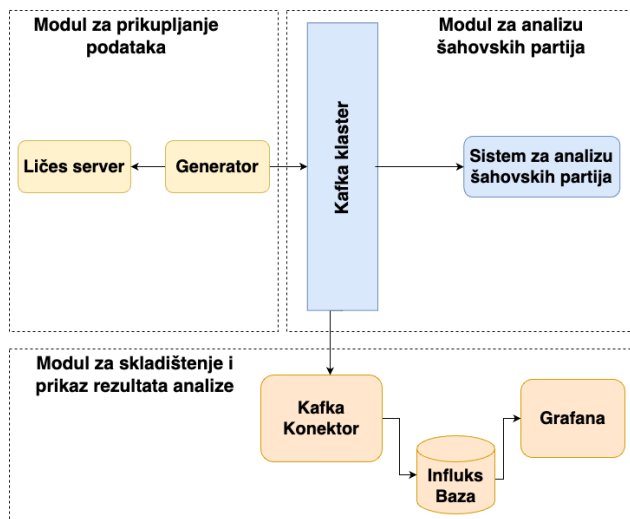
4. ARHITEKTURA SISTEMA

Projektovani sistem sastoji se iz tri modula: modul za prikupljanje podataka sa platforme Ličes, modul za analizu šahovskih partija pomoću alata Stokfiš i modul za skladištenje i prikaz rezultata analize.

Ključna komponenta u ovom sistemu jeste Kafka, sistem za asinhronu razmenu poruka koji omogućava odvajanje i nezavisnost pomenutih komponentata. Kafka služi kao posrednik u komunikaciji između generisanja i obrade tokova podataka. Na [slici 4.1](#) prikazana je arhitektura sistema, a u narednim poglavljima dat je opis komponenti.

4.1 Modul za prikupljanje podataka sa platforme Ličes

Prikupljanje relevantnih podataka sa servera Ličes postiže se aktiviranjem generator aplikacije.



Slika 4.1. Arhitektura sistema

Ova komponenta inicira periodičnu komunikaciju sa serverom, transformiše dobavljene podatke u odgovarajuće strukture, vrši Avro serijalizaciju, i šalje poruke na teme Kafke. Komponente aplikacije generator su:

1. **Generator partija:** Prikuplja podatke o trenutno aktivnim partijama i generiše poruke koje sadrže podatke kao što su identifikator partije, detalji o igračima, odigrani potezi, i ostale relevantne podatke.
2. **Generator poteza:** Dobavlja poteze iz određene partije u realnom vremenu. Poruke sadrže odigrani potez, redni broj poteza, trenutnu poziciju na tabli u FEN notaciji (objašnjena u nastavku poglavlja) i drugo.
3. **Generator kraja partije:** Emituje događaje o završetku partije.
4. **Generator igrača:** Sakuplja podatke o korisnicima koji učestvuju u trenutno aktivnim igrama, kao što su rang (ELO), broj odigranih partija, i drugo.

Forsajt-Edvardsova notacija (FEN) predstavlja standardizovani metod za tekstualni opis pozicija na šahovskoj tabli. U jednom redu teksta predstavljen je raspored figura, sledeći igrač na potezu, mogućnosti rokade, mogućnost *en passant*, broj polupoteza od poslednjeg značajnog poteza, i ukupan broj poteza u partiji. FEN je efikasan za dokumentovanje i analizu šahovskih igara, omogućavajući lako deljenje i pregled stanja igre u bilo kom trenutku. Predstavlja standardan način reprezentovanja šahovske table u najkorišćenijim šahovskim alatima kao što je Stokfiš [5].

4.2 Modul za analizu šahovskih partija u realnom vremenu

Obrada podataka u realnom vremenu uključuje kontinuiranu obradu neograničenih skupova podataka koji se nazivaju tokovi. Karakteristike toka uključuju uređenost i nepromenljivost događaja u njima [3]. U poređenju sa tradicionalnim pristupima obrade podataka, kao što su obrada po zahtevu ili paketna obrada, obrada toka podataka nudi znatno nižu latenciju i sposobnost da se podaci ažuriraju i analiziraju u realnom vremenu. Ovakav

pristup pruža korisnicima mogućnost brzog odgovora na promene i događaje kako se oni dešavaju.

Komponenta za obradu šahovskih partija u realnom vremenu rešava sledeće zadatke:

1. Praćenje i analiza aktivnosti igrača: Aplikacija efikasno prati aktivnosti igrača na platformi Ličes, koristeći metrike koje su definisane u prethodnom poglavlju.
2. Integracija sa alatom Stokfiš: Alat pruža analizu stanja na šahovskoj tabli uz evaluaciju odigranog poteza. Komunikacija sa alatom realizuje se prema UCI (engl. *Universal Chess Interface*) protokolu [5]. Korišćena metrika je centi-pion koja predstavlja stoti deo jednog piona.
3. Deduplikacija događaja: Važan deo obrade poteza uključuje filtriranje i odbacivanje duplikata poruka, što pomaže u očuvanju integriteta podataka. Ovo je ključno za održavanje performansi sistema na visokom nivou i osiguranje da su svi analitički uvidi zasnovani na tačnim i ažuriranim podacima.
4. Detekcija potencijalnih varalica: Postiže se analizom odstupanja u igračevom ponašanju u odnosu na njegov uobičajeni nivo preciznosti u igrama.

Ova komponenta razvijena je uz upotrebu biblioteke Kafka Strims. Kafka Strims omogućava obradu podataka s veoma niskom latencijom. Pored toga, aplikacije mogu lako skalirati horizontalno, što znači da se kapacitet obrade može povećati dodavanjem više instanci aplikacije bez prekida postojećeg rada. Podržava replikaciju podataka i mehanizme oporavka koji minimiziraju uticaj potencijalnih grešaka ili kvarova u sistemu. Podržava širok spektar operacija obrade podataka, od jednostavnih transformacija do složenih agregacija, operacija spajanja (engl. *join*) i grupisanja [6].

Biblioteka Kafka Strims omogućava i definisanje različitih tipova prozora za grupisanje događaja po vremenskim intervalima. Ovo je posebno korisno za izračunavanje agregata ili analizu trendova unutar definisanog vremenskog perioda. U projektovanom sistemu korišćen je sesijski (engl. *session*) tip prozora prilikom računanja broja ispravnih odnosno neispravnih poteza u trenutno aktivnoj partiji.

Aplikacije sa bibliotekom Kafka Strims podržavaju obradu sa čuvanjem stanja (engl. *stateful*), što znači da čuvaju informacije iz prethodnih događaja kako bi izvršile trenutnu obradu [6]. Ovo je korisno za operacije koje zahtevaju kontekst. U projektovanom sistemu korišćen je ovaj tip obrade prilikom deduplikacije događaja, određivanja prednosti igrača i računanja ključnih indikatora performansi.

4.3. Modul za skladištenje i prikaz rezultata analize

Modul za skladištenje i prikaz rezultata analize koristi tehnologiju Kafka Konekt za efikasno upravljanje podacima između tema Kafke i trajnih skladišta podataka. Kafka Konekt predstavlja robusno rešenje za prenos podataka, omogućavajući konfiguraciju konektora koji automatski upisuju podatke iz tema Kafke u izabranu bazu. Konektori mogu biti konfigurisani kao samostalni

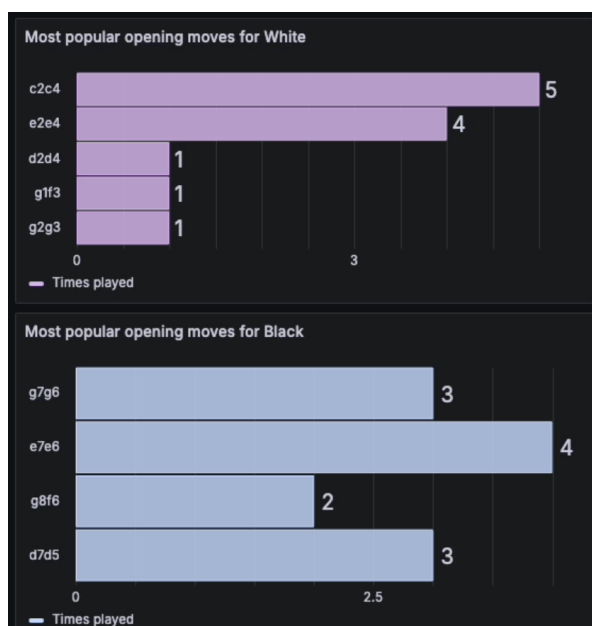
proces ili kao distribuirani servis. Korišćena baza, InfluxDB, specijalizovana je za rad sa vremenskim serijama, što je čini dobrim izborom za skladištenje podataka koji su vremenski osetljivi.

Nakon što su podaci smešteni u InfluxDB, oni se zatim vizualizuju kroz alat Grafana. Otvorenog je koda i omogućava korisnicima da kreiraju kontrolne table za vizuelni prikaz podataka, pružajući jasne uvide kroz grafove i metrike. Na ovaj način omogućeno je korisnicima da na efektivan način vizualno prate i analiziraju šahovske partije. Grafana nudi fleksibilne mogućnosti za prilagođavanje kontrolnih tabla, podržavajući sve od jednostavnih grafikona sa prebrojavanjima poteza do složenijih analiza performansi igrača kroz vreme. Kontrolne table koje koriste Grafana su interaktivne i mogu se prilagoditi za prikaz različitih metrika, što korisnicima omogućava da uvidu u podatke prilagode svojim specifičnim potrebama i interesovanjima.

5. PRIMER UPOTREBE

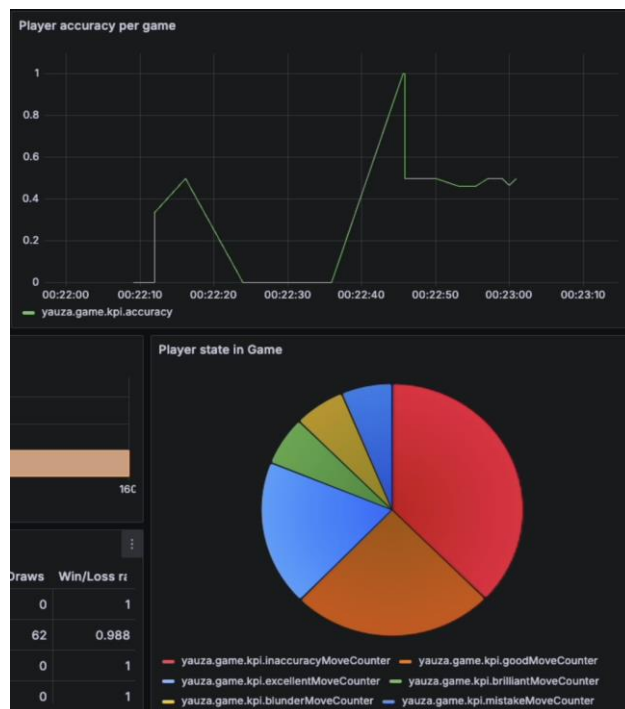
U ovom poglavlju dat je primer kontrolne table kreirane u alatu Grafana. Prikazana kontrolna tabla na slici 5.1 omogućava korisnicima da analiziraju najčešće poteze u partijama kategorije Metak, identifikujući popularne strategije i taktike. Prikazani paneli pod nazivom *Most popular opening moves for White* i *Most popular opening moves for Black* daju uvid u najčešće igrane prve poteze iz ugla belog i crnog igrača. Može se primetiti da beli igrači partiju najčešće započinju pomeranjem piona sa e2 na e4 ili sa d2 na d4, dok za crnog najčešći odgovor predstavlja skakač sa g8 na f6 odnosno pion sa g7 na g6.

Kontrolna tabla može uključivati panel koji prikazuje liste trenutno aktivnih partija na Ličes TV, a koje su praćene kroz projektovani sistem. Korisnicima može biti omogućeno da unose identifikatore za partije i igrače. Nakon unosa, prikazuju se dva panela na kontrolnoj tabli: Promena igračeve preciznosti tokom vremena pod nazivom *Player accuracy per game* i stanje igrača tokom igre - *Player state in game*. Na slici 5.2 predstavljeni su opisani paneli.



Slika 5.1. Najčešće igrani prvi potezi za belog/crnog igrača

Igračeva preciznost definisana je kao odnos ispravnih ka ukupnom broju poteza koje je igrač odigrao. Na dijagramu prikazana je promena ove vrednosti za vreme trajanja partije. Stanje igrača prikazano je korišćenjem pita dijagrama gde se može uočiti: broj ispravnih (dobri, odlični i briljantni potezi) i broj neispravnih poteza (kategorije nesigurni, pogrešni i veoma pogrešni potezi). Na slici 5.2 ove vrednosti su predstavljene pod sledećim nazivima *goodMoveCounter*, *excellentMoveCounter*, *brilliantMoveCounter*, *inaccuracyMoveCounter*, *mistakeMoveCounter*, *blunderMoveCounter*, redom.



Slika 5.2. Preciznost i stanje igrača tokom partije

Grafana pruža mogućnost kreiranja novih panela kojima korisnici mogu pratiti i analizirati podatke u skladu sa svojim interesovanjima.

6. ZAKLJUČAK

U ovom radu predstavljen je sistem za praćenje i analizu šahovskih partija sa platforme Ličes u realnom vremenu. Razvijeni sistem pruža mogućnost detekcije potencijalnih prevara i analize igara tokom njihovog trajanja, što predstavlja značajan napredak u odnosu na postojeće sisteme koji nude samo istorijske analize već završenih partija.

Kroz upotrebu tehnologija kao što su Kafka i Kafka Strims, obezbeđeno je efikasno prikupljanje i obrada podataka o igračima, odigranim potezima, partijama koje se održavaju i drugo, u realnom vremenu, što omogućava pravovremene uvide i reakcije. Korišćenje alata Stokfiš za evaluaciju poteza omogućilo je procenu kvaliteta odigranih poteza tokom igre i evaluaciju stanja oba igrača. Na osnovu definisanih metrika igrači se profilšu i njihove aktivnosti se prate što omogućava detektovanje anomalija u igri. Identifikacija ovih karakteristika i njihovo praćenje ključno je za identifikaciju potencijalnih varalica. Korišćenjem Grafane za vizuelizaciju rezultata omogućeno je praćenje metrika i aktivnosti igrača u

realnom vremenu, što pruža dodatnu vrednost potencijalnim korisnicima platforme - organizatorima ili gledaocima turnira, analitičarima šaha i drugim profilima krajnjih korisnika. Projektovani sistem je skalabilan i otporan na greške, što omogućava njegovu primenu u okruženjima sa velikim obimom podataka i visokom potrebom za pouzdanošću.

Postoje ograničenja od strane platforme Ličes kao što su limiti na broj zahteva ka njihovom serveru sa jedne internet adrese, što može umanjiti efikasnost prikupljanja podataka. Ovi problemi mogu se potencijalno rešiti upotrebom više adresa ili proksi servera.

Dalji napredak mogao bi biti ostvaren poboljšanjem algoritma za detekciju prevara. Trenutni pristup koji se zasniva na praćenju preciznosti igrača kroz igre mogao bi se poboljšati ukoliko bi se analiza samih poteza proširila, odnosno povećala dubina pretrage u alatu Stokfiš. Ovo bi omogućilo tačniju evaluaciju stanja igrača u partiji nakon odigranog poteza.

Osim poboljšanja trenutnog pristupa za detekciju anomalija, upotrebom algoritama mašinskog učenja za profilisanje igrača mogla bi se poboljšati tačnost detekcije prevara. Ovo uključuje upotrebu dubokih neuronskih mreža, slično pristupu koji sama platforma Ličes implementira.

Treći vid unapređenja projektovanog sistema uključuje integraciju dodatnih izvora podataka u sistem. Uvođenje dodatnih izvora, kao što su istorijski podaci o korisnicima i odigranim partijama, moglo bi da obezbedi bolji uvid u ponašanje igrača i njihovo profilisanje.

7. LITERATURA

- [1] Lichess. (2024). *Lichess API Documentation*. [Onlajn]. Dostupno na: <https://lichess.org/api>
- [2] Stockfish Chess. (2024). *Stockfish Chess - Open Source Chess Engine*. [Onlajn]. Dostupno na: <https://stockfishchess.org/>

- [3] T. Akidau, S. Chernyak, and R. Lax, *Streaming Systems: The What, Where, When, and How of Large-Scale Data Processing*, 1st edition. Sebastopol, California: O'Reilly Media, 2018.

- [4] ChessBase. (2022). *ChessBase 17 Manual* [Onlajn]. Dostupno na: <https://help.chessbase.com/cb17-eng.pdf>

- [5] Shredder Chess. (2024). UCI - Universal Chess Interface. [Onlajn].

- Dostupno na: <https://www.shredderchess.com/chess-features/uci-universal-chess-interface.html>

- [6] Apache Kafka. (2024). Developer Guide - Apache Kafka Streams. [Onlajn].

- Dostupno na: <https://kafka.apache.org/24/documentation/streams/developer-guide/>

Kratka biografija:



Milena Laketić rođena je 12. aprila 1998. godine u Rumi, Vojvodina. Poхаđala je gimnaziju "Jovan Jovanović Zmaj" u Novom Sadu, smer obdareni učenici u matematičkoj gimnaziji. Fakultet tehničkih nauka upisala je 2017. godine na smeru Softversko inženjerstvo i informacione tehnologije. Master studije na studijskom programu Softversko inženjerstvo i informacione tehnologije usmerenje Inteligentni sistemi upisuje 2021. U toku master studija izabrana je na poziciju Saradnika u nastavi na fakultetu tehničkih nauka i tokom jedne godine izvršava predviđene obaveze na predmetu Programski prevodioci.

PROJEKTOVANJE INFRASTRUKTURE ZA POTREBE KLIJENT-SERVER APLIKACIJE I EVALUACIJA PROJEKTOVANE INFRASTRUKTURE KROZ PRIZMU DOBROG ARHITEKTONSKOG OKVIRA

INFRASTRUCTURE DESIGN FOR CLIENT-SERVER APPLICATIONS AND EVALUATION OF DESIGNED INFRASTRUCTURE THROUGH THE PRISM OF A WELL ARCHITECTURAL FRAMEWORK

Nikola Kolović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – U sklopu ovog rada je projektovana i implementirana infrastruktura za potrebe klijent-server aplikacije na AWS platformi. Opisani su korišćeni servisi na kojima se infrastruktura zasnika, i prikazani su detalji implementacije servisa. Dodatno, izvršena je analiza i evaluacija projektovane infrastrukture kroz prizmu dobrog arhitektonskog okvira koji AWS platforma obezbeđuje prema dva osnovna stuba: bezbednost i operativna efikasnost. Dobijeni rezultat rada predstavlja sigurnu i skalabilnu infrastrukturu za potrebe klijent server aplikacija, koja je lako proširiva i jednostavna za nadogradnju koje su predložene rezultatima evaluacije.

Ključne reči: *Amazon Web Services, Računarstvo u oblaku, Terraform*

Abstract – As part of this work, the infrastructure for the client-server application on the AWS platform was designed and implemented. The services used on which the infrastructure is based are described, and the details of the service implementation are presented. Additionally, an analysis and evaluation of the designed infrastructure was performed through the prism of a well architected framework provided by the AWS platform according to two pillars: security and operational efficiency. The resulting work represents a secure and scalable infrastructure for the needs of client-server applications, which is easily expandable and simple to upgrade, as suggested by the evaluation results.

Keywords: *Amazon Web Services, Cloud computing, Terraform*

1. UVOD

U savremenom svetu, svedoci smo ubrzanog rasta informacionih tehnologija, i sve veće potrebe za kompleksnijim aplikacijama koje treba da opsluže milione korisnika. Tradicionalne aplikacije, koje su se bazirale na monolitnom pristupu, i lokalnim infrastrukturama postaju irelevantne u odnosu na brzo-rastuća rešenja u oblaku.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio docent Željko Vuković, red. prof.

Računarstvo u oblaku donosi novi pristup kreiranja infrastrukture koji omogućavaju jednostavnije kreiranje i održavanje infrastrukture, a krajnjim korisnicima bolje korisničko iskustvo. Pored jednostavnijeg kreiranje infrastrukture, postoje mnogi benefiti računarstva u oblaku kao što su povećana bezbednost, mogućnost skaliranja resursa, visoka dostupnost servisa, robusnost sistema i druge.

Ovaj rad baviće se projektovanjem infrastrukture u oblaku na AWS (Amazon Web Services) [1] platformi, i implementacija korišćenjem IaaS (Infrastructure as a Service) prakse za kreiranje resursa u oblaku. Implementirana infrastruktura će biti namenjena klijent-server aplikacijama. Analiziraćemo projektovanu infrastrukturu kroz prizmu AWS Well-Architected Framework-a [2], gde ćemo se fokusirati na dva stuba ovog okvira: bezbednost i operativnu efikasnost. Čilj je da se projektuje i implementira infrastruktura koja je dovoljno fleksibilna za buduće izazove, a istovremeno bezbedna, skalabilna i pouzdana.

U drugom poglavlju predstavljeni su AWS servisi i tehnologije koje su korišćene u radu. Opis i detalji implementacije projektovane infrastrukture prikazani su u trećem poglavlju. Četvrto poglavlje sadrži evaluaciju projektovane infrastrukture kroz prizmu dobrog arhitektonskog okvira. U petom poglavlju je iznet zaključak gde su opisane prednosti i mane projektovane infrastrukture, i izneti predlozi za dalji razvoj.

2. KORIŠĆENE TEHNOLOGIJE I AWS SERVISI

Za implementaciju infrastrukture koristili smo alat Terraform [3] koji je trenutno industrijski standard za obezbeđivanje infrastrukture pisanjem programskog koda.

AWS servisi koje smo koristili i koji integrisani zajedno zadovoljavaju potrebe veličine klijent-server aplikacija:

1. AWS Identity and Access Management - koristi se za kreiranje identiteta i upravljanjem dozvolama na platformi
2. AWS Organization - omogućava centralnu organizaciju naloga u okruženjima sa više naloga
3. AWS Identity Center - omogućava centralno upravljanje korisnicima i dozvolama u okruženjima sa više naloga

4. AWS Virtual Private Cloud - koristi se za kreiranje virtuelne privatne mreže radi izolacije resursa, kao i za konfiguraciju dodatnih sigurnosnih postavki
5. AWS Elastic Compute Cloud - omogućava kreiranje virtuelnih mašina u oblaku
6. AWS Elastic Load Balancer - koristi se za kreiranje i konfiguraciju balansa opterećenja
7. AWS Elastic Container Service - omogućava pokretanje i orkestraciju kontejnerizovanih aplikacija u oblaku
8. AWS Relational Database Service - koristi se za kreiranje i upravljanje relacionim bazama podataka
9. AWS Simple Storage Service - predstavlja visokodostupno, skalabilno skladište podataka
10. AWS Cloudfront - omogućava prenos statičkog i dinamičkog saobraćaja ultra brzim mrežama za prenos sadržaja
11. AWS Elastic Container Registry - omogućava čuvanje i upravljanje Docker slikama
12. AWS Secret Manager - koristi se za čuvanje, kao i za rotiranje tajni

3. OPIS PROJEKTOVANE INFRASTRUKTURE

Dizajn rešenja koje je implementirano na AWS platformi je bio poseban izazov zbog potrebe za visokim nivoom skalabilnosti, dostupnosti i sigurnosti. Cilj je bio da se kreira rešenje dovoljno fleksibilno da podrži rastuće zahteve aplikacije, a istovremeno dovoljno robusno da prihvati različite vrste opterećenja.

3.1 Dizajn infrastrukture

Dijagram na slici 1. [4, 5] prikazuje servise koji su korišćeni i njihov raspored unutar virtuelne privatne mreže, čime smo želeli da obezbedimo sigurno okruženje sa sto manje manjkavosti i sigurnosnih propusta. Za implementaciju virtuelne privatne mreže koristili smo AWS VPC, kojim smo kreirati tri grupe pod mreža: javne, privatne, i izolovane. U pod mreže su smešteni resursi zavisno od potrebe za internet konekcijom. Važno je napomenuti da je svaki tip pod mreže kreiran u dve zone dostupnosti, čime obezbeđujemo visoku dostupnost aplikacije.

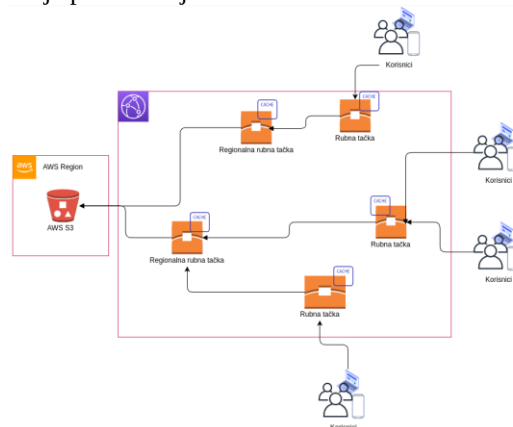
Bezbednosne grupe su postavljene tako da bazi podataka mogu pristupiti samo servisi iz privatne pod mreže, i dodatno adresa bastion host-a čime obezbeđujemo pristup bazi podataka razvojnom timu. Servisima u privatnoj pod mreži mogu pristupiti servisi iz javne pod mreže, dok smo za servise u javnoj pod mreži prilagodili bezbednosne grupe potrebama. Tako balanser opterećenja možemo pristupiti samo na portovima 80 i 443, dok bastion host-u možemo pristupiti samo na portu 22.

Za potrebe baze podataka, koristili smo AWS RDS servis, gde smo kreirali PostgreSQL bazu podataka, koja je smeštena u izolovanoj pod mreži. Za potrebe upravljanja kredencijalima baze podataka koristili smo AWS Secret Manager, koji čuva i rotira kredencijale za pristup.

Serverski deo aplikacije se pokreću u kontejnerizovanom okruženju upravljanom AWS ECS servisom, koji se

nalaze u privatnoj pod mreži. Konektuju se na bazu podataka korišćenjem kredencijala iz tajni, dok se ispred servisa nalazi balanser opterećenja zadužen za raspoređivanje dolaznog saobraćaja.

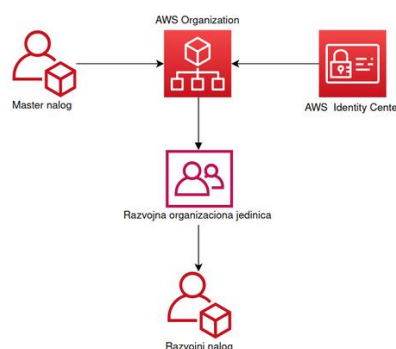
Klijentski deo aplikacije predstavlja statički veb sajt, čiji se fajlovi nalaze u S3 kanti, dok se isporučuju krajnjim korisnicima korišćenjem AWS Cloudfront distribucije, koja omogućava keširanje i brzu isporuku saobraćaja kroz mrežu rubnih tačaka. Ovakva konfiguracija klijentske aplikacije prikazana je na slici 2.



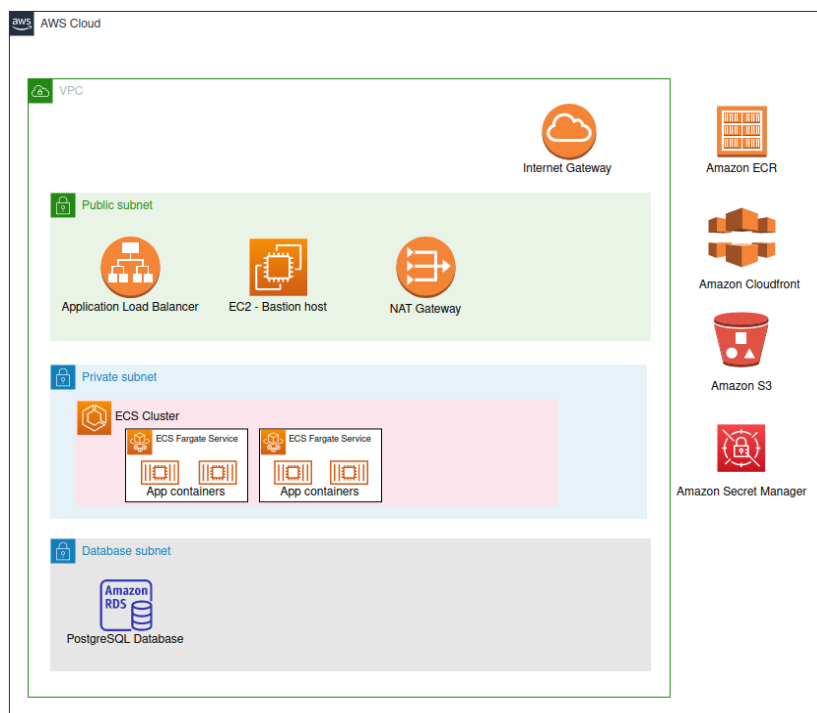
Slika 2. Prikaz dizajna rešenja za isporuku klijentske aplikacije

Da bismo obezbedili fleksibilnost i bezbednost sistema, koristili smo okruženje sa više naloga. U trenutnom obimu rada potrebni su nam samo razvojni i master nalog, ali je dizajn dovoljno fleksibilan da u buducnosti može podržati dodatna okruženja na različitim nalogima jednostavnim izmenama u Terraform konfiguracionim fajlovima.

Dijagram na slici 3. prikazuje rešenje implementirano sa dva AWS naloga: master i nalog za razvojno okruženje. Master nalog se koristi za upravljanje okruženjima sa više naloga pomoću servisa AWS Organization i AWS Identity Center. Za potrebe razvojnog okruženja kreiran je dodatni nalog koji je smesten u organizacionoj jedinici za razvojne naloge.



Slika 3. Prikaz dizajna okruženja sa više naloga



Slika 1. Prikaz dizajnirane infrastrukture za potrebe jednog okruženja

3.2 Konfiguracija Terraform okruženja

Kada je u pitanju organizacija konfiguracionih fajlova Terraform okruženja, na najvišem nivou koristili smo dva direktorijuma:

1. project-infrastructure - sadrži direktorijume za pozive modula, gde je opisano željeno stanje na AWS platformi
2. tf-modules - sadrži direktorijume sa kreiranim ponovno iskoristivim modulima

U project-infrastructure direktorijumu, podelili smo servise u zasebne direktorijume, gde smo definisali Terraform provajder koji koristimo, kao i konfiguraciju za čuvanje i enkripciju Terraform stanja.

Za potrebe čuvanja Terraform stanja, koristili smo udaljeni backend koji se nalazi na S3 kanti. Dodatno, omogućili smo zaključavanje stanja, i enkripciju datoteke za čuvanje stanja.

4. EVALUACIJA PROJEKTOVANE INFRASTRUKTURE KROZ PRIZMU DOBROG ARHITEKTONSKOG OKVIRA

Dobar arhitektonski okvir, ili AWS Well Architected Framework, predstavlja skup principa i najboljih praksi kreiranih od strane AWS, kako bi pomogao korisnicima da implementiraju sigurne, visoko dostupne, efikasne i robusne aplikacije u oblaku. Okvir se sastoji od pet stubova, koji uključuju principe koje treba slediti i smernice za njihovu implementaciju, kako bi osigurali da infrastruktura u oblaku bude izgrađena na najbolji mogući način.

Okvir se sastoji od 5 stubova:

1. Bezbednost (Security)
2. Operativna efikasnost (Operational Excellence)
3. Pouzdanost (Reliability)

4. Performanse (Performance Efficiency)
5. Optimizacija troškova (Cost Optimization)

Analizom svakog od ovih stubova, mogu se identifikovati potencijalni nedostaci i mogućnosti za unapređenje infrastrukture.

Fokus ovog rada su bila prva dva okvira: bezbednost i operativna efikasnost.

4.1 Bezbednost

Bezbednost predstavlja jedan od najbitnijih stubova i predstavlja temelj svake uspešne infrastrukture u oblaku. Bezbednost u oblaku obuhvata zaštitu celokupnog radnog okruženja, uključujući podatke, sisteme i sve resurse.

Zasniva se na 7 osnovnih principa:

1. Implementacija čvrste osnove identiteta
2. Održavanje praćenja
3. Primena bezbednosti na svim slojevima
4. Automatizacija najboljih bezbednosnih praksi
5. Zaštita podataka u tranzitu i u mirovanju
6. Održavanje ljudi daleko od podataka
7. Priprema za sigurnosne događaje

Evaluacijom projektovane infrastrukture u kontekstu stuba bezbednosti zaključili smo da projektovana infrastruktura u najvećoj meri zadovoljava preporuke navedenih principa.

Pre svega, osnova identiteta je implementirana praćenjem najboljih praksi. Odnosno, kreiranjem okruženja sa više naloga, sa centralnim upravljanjem identitetima i dozvolama, što smo postigli pomoću AWS Organization i AWS Identity Center servisa. Dodatno, dozvole smo u određenoj meri kreirali poštovanjem principa najmanjih privilegija i obezbedili smo privremene kredencijale za pristup.

Bezbednost na nivou okruženja je takođe uglavnom implementirana u skladu sa smernicama definisanim od strane AWS. Osnovu mrežne bezbednosti smo implementirali pomoću AWS VPC servisa, gde smo definisali bezbednosne grupe na nivou servisa.

Bezbednost podataka smo obezbedili enkripcijom podataka u bazi podataka, na S3 kantama kao i tajnama. Dodatno, razvojni tim nema direktan pristup bazi podataka i imamo implementirane sigurnosne kopije podataka.

Kompletna infrastruktura je implementirana u skladu sa praksom pisanja infrastructure kroz kod, koristeći Terraform alat.

Preporuke za unapređenje će biti iznete u zaključku ovog rada.

4.2 Operativna efikasnost

Operativna efikasnost predstavlja značajan stub za kreiranje uspešne infrastrukture. Operativna efikasnost uključuje optimizaciju operacija, kontinualno unapređivanje procedura, i održavanje visokog nivoa performansi i pouzdanosti.

Zasniva se na 5 osnovnih principa:

1. Izvršavanje operacija kroz kod
2. Pravljenje čestih, malih, revizibilnih promena
3. Često unapređivanje operativnih procedura
4. Korišćenje upravljivih servisa
5. Implementacija nadzora radi sticanja korisnih uvida

Za razliku od stuba bezbednosti, gde je infrastruktura u najvećoj meri zadovoljavala principe, u ovom slučaju su principi delimično zadovoljeni, pre svega iz razloga što u trenutnoj fazi projekta, neki principi ne mogu biti implementirani, dok za neke druge nemamo nikakvih prednosti za samu infrastrukturu.

Uprkos tome, veliki deo principa je zadovoljen. Infrastruktura je implementirana kroz kod, prilagođavanjem ponovno iskoristivih modula, što dodatno implicira da smo omogućili kreiranje malih promena na nivou servisa, bez uticaja na druge servise. Korišćenjem alata za kontrolu verzija Git, omogućeno je verzionisanje i praćenje promena.

Uglavnom su korišćeni upravljivi servisi, čime smo omogućili razvojnim timovima da budu efikasniji i da se oslobode dodatnih zadataka za održavanje infrastrukture.

Preporuke za unapređenje će biti iznete u zaključku ovog rada.

5. ZAKLJUČAK

Na osnovu projektovane infrastrukture koja predstavlja složeno ali istovremeno i dovoljno fleksibilno rešenje, dobijena je funkcionalna infrastruktura, koja je apsolutno operativna, bezbedna, i dovoljno skalabilna za opsluživanje velikog broja korisnika.

Evaluacijom implementirane infrastrukture kroz prizmu dobrog arhitektonskog okvira zaključili smo da infrastruktura ima jako dobro dizajniranu osnovu, korišćenjem AWS okruženja sa više naloga, i centralnim

servisom za upravljanje identiteta i dozvolama, što predstavlja najbolje prakse u vidu fundamentalne bezbednosti i buduće proširivosti. Bezbednost podataka i drugih komponenti je na visokom nivou, sledili smo najbolje prakse za konfigurisanje privatne mreže, i zaštitili sve slojeve. Dodatno, enkriptujemo podatke u tranzitu i u mirovanju gde je to bilo moguće, automatski rotiramo tajne, i pravimo redovne rezervne kopije podataka. Infracrtuktura je obezbeđenja pisanjem programskog koda, čime smo osigurali da je infrastruktura uvek u željenom stanju, i dodatno standardizovali kako operativne tako i sigurnosne postavke kreiranjem Terraform modula.

S obzirom da je infrastruktura implementirana korišćenjem najmodernijih tehnologija, i servisa koji se konstantno razvijaju i unapređuju, sasvim je legitimno reći da implementirano rešenje ima prostora za dalja usavršavanja, što je pokazala i evaluacija projektovane infrastrukture.

Jedan od segmenata za unapređenje je svakako zakupljivanje privatnog domena, čime bismo rešili problem sa kreiranjem SSL sertifikata na nivou balansera opterećenja. Rešenje za pristup bazi podataka može biti poboljšano uključivanjem AWS Session Manager-a i aktiviranjem funkcionalnosti za IAM autentikaciju na nivou AWS RDS baze podataka, koja obezbeđuje privremene kredencijale za korisnike koji imaju dozvole definisane politikama.

Dodatno se može obratiti paznja na reviziciju principa najmanjih privilegija, i sistem praćenja metrika i upozorenja.

6. LITERATURA

- [1] AWS - Amazon Web Services - <https://aws.amazon.com/> (pristupljeno u aprilu 2024.)
- [2] AWS Well-Architected Framework - <https://aws.amazon.com/architecture/well-architected/> (pristupljeno u aprilu 2024.)
- [3] Terraform - <https://www.terraform.io/use-cases/infrastructure-as-code/> (pristupljeno u aprilu 2024.)
- [4] James F. Kurose, Keith W. Ross, "Computer Networking: A Top-Down Approach", 2017.
- [5] Gauvar Agarwal, "Modern DevOps Practices", 2021.

Kratka biografija:



Nikola Kolović rođen je u Kraljevu 1998. god. Osnovnu i srednju školu završio je u Kraljevu, nakon čega je 2017. godine upisao Fakultet tehničkih nauka u Novom Sadu. Diplomirao je 2021. na modulu primenjene računarske nauke i iste godine upisao Master studije.
Kontakt: nikola.kolovic7@gmail.com

**РАЗВОЈ И ПРОГРАМИРАЊЕ ЕДУКАТИВНЕ СТАНИЦЕ ЗА ПРОУЧАВАЊЕ
КОМУНИКАЦИОНИХ ПРОТОКОЛА У ИНДУСТРИЈСКОМ ОКРУЖЕЊУ****DEVELOPMENT AND PROGRAMMING OF AN EDUCATIONAL STATION FOR
STUDYING COMMUNICATION PROTOCOLS IN AN INDUSTRIAL ENVIRONMENT**

Угљеша Поповић, Ивана Тодоровић, Дарко Марчетић, *Факултет техничких наука, Нови Сад*

**Област – ЕНЕРГЕТСКА ЕЛЕКТРОНИКА И
ЕЛЕКТРИЧНЕ МАШИНЕ**

Кратак садржај – У раду је дат опис едукативне станице (електроормара) посебно развијене за обуку студената у области индустријске аутоматизације, с посебним освртом на савремене комуникационе индустријске протоколе и елементе (уређаје) који се данас најчешће користе у индустрији. Стога, за студенте су осмишљени посебни задаци који од њих изискују примену знања из области примене програмабилних логичких контролера (PLC), као и примене различитих типова комуникационих протокола (Profinet, Profibus, Modbus TCP, USS) коришћењем предложене едукативне станице.

Кључне речи: PLC, Profinet, Profibus, Modbus TCP, USS

Abstract – The paper provides a description of an educational station (electrical cabinet) specifically developed for training students in the field of industrial automation, with a particular focus on modern industrial communication protocols and components (devices) that are most commonly used in the industry today. Therefore, special tasks have been designed for students that require them to apply their knowledge in the use of programmable logic controllers (PLCs), as well as the application of various types of communication protocols (Profinet, Profibus, Modbus TCP, USS) using the proposed educational station.

Keywords: PLC, Profinet, Profibus, Modbus TCP, USS

1. УВОД

Како индустријски комуникациони протоколи и генерално принципи индустријске аутоматизације имају велики значај у различитим инжењерским дисциплинама, јасно је колико су знања из ових области важна будућим инжењерима. Одавде и следи тема овог мастер рада, чији је циљ био израда електроормара аутоматског управљања, тј. његово пројектовање, шамирање и израда софтверског решења који ће се користити за обуку студената Факултета техничких наука у Новом Саду. Намера је била да се студентима прикаже савремени начин

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Дарко Марчетић, ред. проф.

пројектовања ормана и опрема која се данас користи у индустрији. Примењени софтвер је осмишљен тако да буде интерактиван и да студент добија задатке које треба да реши, решења се проверавају у позадини, а на крају се на HMI (енгл. Human Machine Interface) панелу исписују поруке о томе да ли су испуњени сви задати циљеви вежбе. Такође, нагласак је био на употреби различитих комуникационих протокола који се данас срећу у индустрији као што су Profinet, Modbus TCP, Profibus, Siemens S7 и Siemens USS. Осмишљено је девет вежби, а свака следећа вежба се ослања на знање стечено у претходној.

**2. КОРИШЋЕНА ОПРЕМА И
КОМУНИКАЦИОНИ ПРОТОКОЛИ**

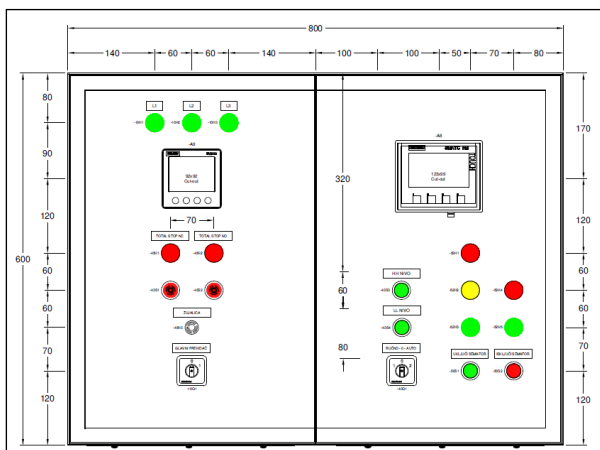
За израду едукативне станице (електроормара) коришћена је савремена опрема која се данас среће у индустрији. Значајан удео опреме за управљање процесима производи се од стране компаније Siemens, а може се рећи и да су њихови производи данас најзаступљенији у индустрији. У раду коришћени су њихови PLC уређаји новије генерације S7-1200 и модерни HMI панел осетљив на додир KTP400.

Додатно, водило се рачуна да се у раду са едукативном станицом користе данас најзаступљенији индустријски комуникациони протоколи. Стога, примењени су протоколи као што су Profinet, Profibus и Modbus TCP али и Siemens-ов USS комуникациони протокол, који је развијен за комуникацију са Siemens-овим Micromaster 440 инвертором. Иако овај протокол није најзаступљенији, ипак представља групу серијских протокола који се неретко јављају у практичним пројектима.

2.1. Коришћена опрема

За сам електроормар је изабран назидни ормар с двоје врата, с IP54 заштитом, чије су димензије 600x800x300 mm. На слици 1 је приказан распоред опреме на вратима електроормара, а на слици 2 распоред опреме унутар електроормара.

На левим вратима ормара налази се светлосна индикација за присуство напајања три фазе. Испод светлосне индикације се налази мрежни анализатор, а потом „тотал стоп“ тастери и светлосна индикација њихових стања. Један „тотал стоп“ тастер је радни, а други мирни, за потребе показне вежбе чији је циљ практична анализа утицаја типа контакта на рад



Слика 1. *Распоред опреме на вратима електроормара.*

система, одакле треба закључити зашто „тотал стоп“ тастере увек треба изводити са мирним контактом.

Испод „тотал стоп“ тастера, налази се зујалица за сигнализацију аларма и главни прекидач напајања.

На десним вратима се налази *HMI* панел. Испод панела налазе се две логичке групе тастера и светлосних индикација. На левој страни налазе се тастери са светлосном индикацијом који се користе у циљу симулације детектора максималног и минималног нивоа у резервоару. Испод ових тастера је преклопка за одабир режима рада пумпе. Тастери и преклопка користе се у вежбама 8 и 9. На десној страни налазе се светлосне индикације који представљају саобраћајни семафор и тастери. Тастери служе за укључење и искључење семафора и користе се у вежби број 4.

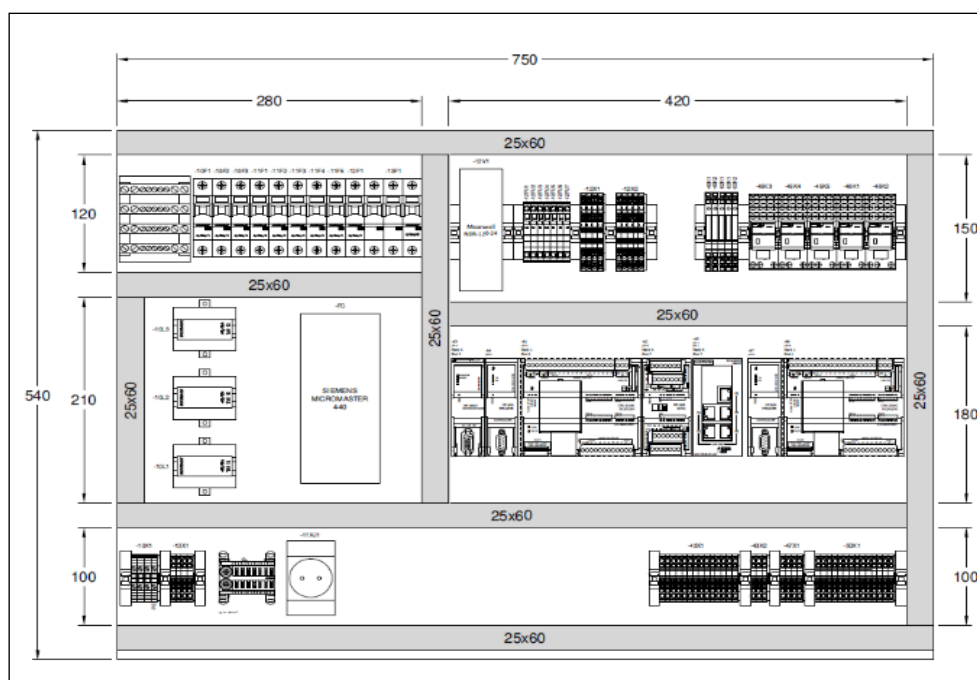
Унутрашњост електроормара се састоји из три нивоа. У самом дну налазе се стезаљке за напојни кабл, кабл мотора и за сигналне каблове, као и сервисна утичница. За потребе показне вежбе тестирања ормара, сигнали од значаја су повезани на *PLC* преко

стезалјки. Ови сигнали се у пракси директно повезују и нема потребе за индиректним повезивањем (користећи стезалјке). У средњем нивоу налазе се управљачки елементи. На левој страни смештен је инвертор који управља са повезаним мотором, као и струјни мерни трансформатори, а на десној страни два *PLC* уређаја са припадајућим картицама. На самом врху с леве стране налази се разводна кутија из које се дистрибуирају три фазе напајања и сигурносни елементи (аутоматски осигурачи) наизменичног напона, а на десној страни претварач напона, топлјиви осигурачи једносмерног напона и релеји.

Коришћена опрема произвођача *Siemens: Simatic S7-1200 CPU 1214C DC/DC/DC* (2 ком.), *Simatic HMI KTP400 Basic* (1 ком.), *Simatic S7-1200 SM1234* модул *4AI/2AO* (1 ком.), *Simatic S7-1200 CM1241 RS422/485* (2 ком.), *Simatic S7-1200 CSM1277 Ethernet unmanaged switch* (1 ком.), *Sentron PAC3220* мрежни анализатор (1 ком.), комуникациони модул за *Profibus DP V1* за *Sentron PAC3200/4200* (1 ком.) и инвертор *Micromaster 440*.

Коришћена опрема произвођача *Turck*: један температурни сензор (серијски бр. 9910624+6625013) и један индуктивни PNP сензор (серијски бр. 46070+6625013).

Сви тастери, тотал стоп тастер, помоћни релеји, гребенасте склопке и светлосне индикације набављене су од произвођача *Schrack*. Дистрибутивни блок за дистрибуцију фаза и неутралног вода набављен је од произвођача *Teknomega*. За стабилисани извор напајања 24 VDC/5 A изабран је *NDR-120-24* произвођача *Meanwell*. Ради размене дигиталних сигнала са *Micromaster* инвертором коришћени су *PLC* релеји *PLC-RSP-24DC/21* произвођача *Phoenix*.



Слика 2. *Распоред опреме унутар електроормара*

2.1. Коришћени комуникациони протоколи

У склопу едукативне станице користе се следећи комуникациони протоколи:

1. *Profinet* (*S7* комуникација),
2. *Profibus*,
3. *Modbus TCP*,
4. *USS* комуникација.

S7 (PUT/GET) комуникација се користи за размену података између *Master* и *Slave PLC* уређаја везаних за проверу софтверских решења које су развили студенти у склопу вежби. Овај комуникациони протокол је развио *Siemens* и може да се користи само између два *Siemens*-ова *PLC*-а. Овај протокол омогућава директан приступ меморији између два *Siemens*-ова *PLC* уређаја.

Profibus комуникација се користи за читање података са *Sentron PAC3220* мрежног анализатора. Подаци са овог уређаја користе се у вежбама 6 и 7. Да би ова комуникација била могућа, на *Sentron PAC3220* се поставља комуникациони модул за *Profibus DP V1* док се на *Master PLC* поставља додатни *Simatic S7-1200 SM1234* модул који подржава *Profibus* комуникацију.

Modbus TCP се користи у вежби број 7, где студенти треба да припреме податке на *Slave PLC*-у према дефинисаној табели *Modbus* регистара. Два *PLC* уређаја се налазе у *Ethernet* мрежи и повезани су преко *Simatic S7-1200 CSM1277 Ethernet unmanaged switch*-а.

USS комуникација се користи за читање тренутних вредности са *Siemens*-овог *Micromaster 440* инвертора. Ова комуникација се користи у вежбама 8 и 9 где је потребно имати информацију о тренутном стању инвертора као и у вежби 7 где се информације са инвертора очекују у неким *Modbus* регистрима.

3. ОДАБРАНИ ПРОБЛЕМИ ЗА РЕШАВАЊЕ

Постоји укупно 9 вежби, од којих су две показне и то вежба број 6 и вежба број 9. Вежбе су поређане тако да се свака наредна вежба ослања на претходно стечена знања приликом решавања претходних вежби. У зависности од вежбе, користе се различити елементи/уређаји електроормара и одговарајући комуникациони протоколи.

3.1. Вежба 1

У првој вежби се користе дигитални/аналогни улази/излази *Siemens*-овог *S7-1200 PLC* уређаја. Вежба се заснива на коришћењу основних логичких функција приликом рада са дигиталним улазима и излазима овог *PLC*-а. У склопу рада са аналогним улазима, студент ће имати прилику да користи температурни сензор *Turck* произвођача. Такође, у склопу рада са аналогним улазима студент ће слати аналогни сигнал на постојећи *Siemens*-ов инвертор *Micromaster 440* који представља референцу фреквенције обртања. Циљ вежбе је да студент стекне знање о основним логичким функцијама које се користе приликом рада са дигиталним улазима и излазима. Такође, студент ће научити да користи аналогне улазе и излазе и уочити разлику између

аналогног сигнала који се добија из спољашњег света и његове интерпретације у софтверу и *HMI* панелу.

3.2. Вежба 2

У другој вежби је дат акценат на рад са тајмерима. У овој вежби се не користи ниједан други уређај електроормара осим самог *PLC* уређаја.

Циљ је да се студент упозна са различитим врстама тајмера и начином њиховог рада.

3.3. Вежба 3

У трећој вежби потребно је искористити стечена знања из прве две вежбе и остварити импулсно ширинску модулацију на излазу *PLC* уређаја. Дат је фиксан период импулса, док се ширина импулса у процентима задаје преко *Siemens*-овог *HMI* панела.

Циљ ове вежбе је да се провери да ли су студенти схватили принцип рада тајмера и да се сусретну са једним видом управљања дозирања материјала или течности на основу неке величине у односу на коју се дефинише ширина импулса, односно дужина трајања дозирања.

3.4. Вежба 4

У овој вежби је потребно развити логику за рад саобраћајног семафора.

За потребе ове вежбе користе се светлосне индикације и тастери који се налазе на десним вратима електроормара. Светлосне индикације представљају светла семафора и контролишу се дигиталним излазима *PLC* уређаја док се стања тастера који покрећу и заустављају рад семафора шаљу на дигиталне улазе *PLC* уређаја.

Циљ ове вежбе је упознавање студената са начином управљања која се у теорији зове машина стања. Оно захтева коришћење *CASE* структуре и јасне кораке који морају да се извршавају један иза другог са тачним временским трајањима. Поред тога, студент ће уочити да је некада једноставније не користити тајмере када је потребно рачунати време. Научиће како се коришћењем системских сатова *PLC* уређаја могу рачунати секунде.

3.5. Вежба 5

У овој вежби предвиђен је рад са низовима.

Циљ ове вежбе је да се студент упозна са радом са низовима и да истражи постојеће алгоритме за сортирање низова али пре свега да се припреми за следећу вежбу у којој се за управљање потрошњом електричне енергије користе нивои због специфичности проблема приоритетних потрошача.

3.6. Вежба 6

Ова вежба је показна али захтева учествовање студента у њој. Предвиђено је да се у овом задатку студенту прикаже начин управљања електричном потрошњом у домаћинству.

За потребе ове вежбе користи се *Siemens*-ов *PAC3220* мрежни анализатор који комуницира са *PLC* уређајем преко *Profibus* мреже. Улога овог уређаја је да, између осталог, рачуна тренутну потрошњу електричне енергије и да ту информацију шаље на *PLC* уређај,

који даље према унапред дефинисаним условима прекида потрошњу појединих потрошача. Ови потрошачи се повезују на инсталационе шуко утичнице, произвођача *Schrack*, које се налазе на десној страни електроормара. Прекидање довода фаза на ове утичнице врши се преко одговарајућих релеја у електроормару, којима се управља преко одговарајућих дигиталних излаза *PLC* уређаја.

Улога студента у овој вежби је да подешава жељену потрошњу електроормара, приоритете потрошача и да шаље захтеве *PLC* уређају када је потребно вршити ограничење потрошње електричне енергије преко *HMI* панела.

Циљ ове вежбе је да се студенту прикаже значај коришћења низова у *PLC* програмирању, као и могућности управљања потрошњом електричне енергије употребом мрежних анализатора. Такође, студент ће се упознати и са величинама које један модерни мрежни анализатор мери и рачуна, а којима може да приступи *PLC* уређај коришћењем одговарајућег комуникационог протокола.

3.7. Вежба 7

У овој вежби предвиђен је рад с *Modbus TCP* комуникацијом. За потребе ове вежбе користе се два *PLC* уређаја (*Siemens S7-1200*), и разликујемо *Master PLC* који представља *Modbus* клијента и *Slave PLC* који представља *Modbus* сервера. Од студената се очекује да организују *Modbus* регистре на *Slave PLC* уређају како је дефинисано у задатку. Уз то, врши се провера да ли су пристигли подаци валидни.

Циљ ове вежбе је да се студент упозна са начином организације различитих типова података унутар блока података који треба да репрезентују дефинисану табелу *Modbus* регистара. Према дефиницији протокола, сви ови подаци су *word* типа. Такође, студент се упознаје са софтверском реализацијом *Modbus TCP* протокола унутар *PLC* уређаја.

3.8. Вежба 8

У овој вежби се студент упознаје са *ON/OFF* начином управљања, при чему се управља пумпом чија је улога да одржава ниво воде у резервоару. За потребе ове вежбе се користи *Siemens-ов* инвертор *Micromaster 440* који погони мотор који ради као пумпа. Како се ради о *ON/OFF* управљању, шаље се константна референца обртања од 50 Hz. Такође, поред инвертора, користе се и тастери који представљају детекторе критично ниског и критично високог нивоа течности у резервоару и који се воде на дигиталне улазе *PLC* уређаја.

Циљ ове вежбе је да се студент упозна са начином регулисања течности у резервоару коришћењем хистерезисног *ON/OFF* управљања и значаја коришћења хистерезиса у оваквим апликацијама.

3.9. Вежба 9

Иако је ова вежба начелно показна, ипак захтева учествовање студената. У вежби 9 развијено је решење које представља унапређење регулације нивоа течности у резервоару из претходног задатка. Слично

као и у претходној вежби, и овде се користи *Siemens-ов* инвертор *Micromaster 440* који погони мотор који представља пумпу. За разлику од претходне вежбе, у овој вежби се за регулацију течности у резервоару користи *PID* регулатор (Пропорционално–Интегрално–Диференцијални) који рачуна референцу обртања мотора у зависности од жељене и постигнуте висине течности у резервоару. Такође, поред инвертора, користе се и тастери који представљају детекторе критично ниског и критично високог нивоа течности и који се воде на дигиталне улазе *PLC* уређаја.

Улога студента у овој вежби јесте подешавање параметара *PID* регулатора преко *HMI* панела. Поред подешавања *PID* регулатора, студент може да шаље захтев за отварање или затварање одводног вентила, који би требао да представља поремећај у систему, који ће даље довести до промене управљања *PID* регулатора.

Циљ ове вежбе је упознавање студента са предностима *PID* регулације у односу на *ON/OFF* управљање из претходне вежбе и са самим начином подешавања *PID* регулатора у случају да није познат математички модел система управљања.

4. ЗАКЉУЧАК

У овом раду развијена је едукативна станица која ће допринети бољем стручном усавршавању будућих инжењера на студијском модулу Електроенергетика – Енергетска електроника и електричне машине и омогућити увид у уређаје и елементе који се данас користе у индустрији, као и у примену комуникационих протокола и начине њихове реализације.

Кратка биографија:



Угљеша Поповић рођен је у Новом Саду 1993. год. Дипломски рад на Факултету техничких наука из области Електротехнике и рачунарства – Аутоматика и управљање системима одбранио је 2016.год. контакт: uprovic93.up@gmail.com



Ивана Тодоровић рођена је 1993. године у Руми. Бави се научним истраживањима из области управљања претварањима енергетске електронике, претежно у применама оријентисаним ка електроенергетским мрежама.



Дарко Марчетић рођен је 1968. године у Новом Саду. И даље се релативно успешно бави научним истраживањима.

UPOTREBLJIVOST GRAPHQL BIBLIOTEKE ZA RAZVOJ NLP INTERFEJSA**USABILITY OF GRAPHQL LIBRARY FOR DEVELOPMENT OF NLP INTERFACES**Dragan Borkovac, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – *Konceptualno rešenje sistema koje korisnicima omogućava pristup podacima iz skladišta koristeći naredbe na prirodnom jeziku. Korisnici putem NLP interfejsa postavljaju upite na prirodnom jeziku kako bi dobili željene podatke. Sistem automatski prevodi upite na prirodnom jeziku u GraphQL upite kojim je pokriven celokupan API sistem, uključujući i skladište podataka.*

Ključne reči: NLP, GraphQL, API

Abstract – *Conceptual solution of a system that allows users to access data from the warehouse using natural language queries. Through the NLP interface, users make queries in natural language in order to get the desired data. The system automatically translates natural language queries into GraphQL queries that cover the entire API system, including the data warehouse.*

Keywords: NLP, GraphQL, API

1. UVOD

U današnjem digitalnom dobu, velika količina podataka je dostupna korisnicima širom sveta. Iz tog razloga sve češće dolazi do pojave problema efikasnog pristupa i pretraživanja podataka. Ovaj problem je najizraženiji kada je u pitanju raznolikost formata i struktura u kojima se podaci čuvaju. Tradicionalni načini pretrage, često zasnovani na predefinisanim ključnim rečima ili kompleksnim upitima, se ovde pokazuju kao neefikasni i zahtevaju od korisnika duboko razumevanje strukture podataka.

Upravo u cilju olakšavanja pretrage i pristupa podacima, nastaje ideja o sistemu koji korisnicima omogućava pristup podacima koristeći naredbe na prirodnom jeziku. Ovaj rad istražuje kako se putem interfejsa za obradu prirodnog jezika (NLP) mogu postavljati upiti na prirodnom jeziku kako bi se dobili željeni podaci iz skladišta podataka. Sama srž sistema leži u automatskom prevodenju ovih upita na prirodnom jeziku u GraphQL upite koji pokrivaju celokupni API sistem, uključujući i skladište podataka. Korišćenje GraphQL biblioteke omogućava precizno definisanje podataka koje korisnik želi da dobije, što čini ovaj pristup izuzetno fleksibilnim i efikasnim.

U okviru rada, razmotreni su osnovni koncepti NLP i GraphQL tehnologija i analizirano je trenutno stanje u

NAPOMENA:

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

praksi kroz pregled postojećih istraživanja koja povezuje ove dve tehnologije. Pored toga, istražene su osnove prepoznavanja jezika i tehnike semantičke analize koje su neophodne za uspešno mapiranje prirodnog jezika na GraphQL upite. Nakon toga, razmotren je dizajn i sama implementacija NLP sistema, kao i razvoj GraphQL interfejsa. Na kraju, demonstrirana je upotreba ovakvog sistema u realnom scenariju.

Cilj ovog rada jeste predstavljanje konceptualnog rešenja koje kombinuje prednosti NLP i GraphQL tehnologija kako bi omogućilo korisnicima intuitivno i efikasno pretraživanje podataka putem naredbi na prirodnom jeziku. Ovakvo rešenje može imati širok spektar primena, od olakšavanja pristupa kompleksnim bazama podataka do unapređenja korisničkog iskustva u različitim digitalnim okruženjima. Ono može smanjiti krivu učenja za programere kada je u pitanju GraphQL tehnologija, a može i predstavljati potencijalni alat za pristup podacima za netehnička lica.

2. OSNOVNI KONCEPTI I STANJE U PRAKSI

U ovom poglavlju biće predstavljani osnovni koncepti obrade prirodnog jezika i GraphQL biblioteke.

2.1. Uvod u NLP

Obrada prirodnog jezika (NLP) odnosi se na granu računarske nauke - konkretnije, granu veštačke inteligencije (AI) koja se bavi davanjem računarima sposobnosti da razumeju tekst i reči na sličan način kao i ljudi [1].

NLP kombinuje računarsku lingvistiku, zasnovanu na pravilima modelovanja ljudskog jezika, sa statističkim, mašinskim i dubokim učenjem [1]. Ove tehnologije omogućavaju računarima da obrade ljudski jezik u obliku tekstualnih ili glasovnih podataka i da "razumeju" njegovo potpuno značenje, uz nameru i emociju govornika ili pisca [1].

NLP predstavlja osnovu programskih sistema koji prevode tekst sa jednog jezika na drugi, odgovaraju na govorne komande i brzo sumiraju velike količine teksta [1]. Interakcija sa NLP sistemima sreće se i u obliku glasom upravljanih GPS sistema, digitalnih asistenata, softvera za pretvaranje govora u tekst, chatbotova za korisničku podršku i slično [1]. NLP takođe igra sve veću ulogu i u poslovnim rešenjima koja pomažu u optimizaciji poslovnih operacija, povećavaju produktivnost zaposlenih i pojednostavljaju razne poslovne procese [1].

2.2. Uvod u GraphQL

GraphQL je tip programskog interfejsa aplikacije (ili upitni jezik), ali i osnovni mehanizam za odgovaranje na postavljene upite [2].

GraphQL korisnicima pruža pojednostavljeni API idealan za mobilne aplikacije i implementacije sistema sa kompleksnim šemama gde su potrebni samo određeni podskupovi podataka [2]. Koristeći GraphQL programeri mogu precizno da definišu podatke koje žele da preuzmu [2]. Pored toga, GraphQL korisnicima omogućava preuzimanje podataka po zahtevu, a ne sve odjednom, trenutno primenjivanje promena i integraciju više izvora podataka [2].

3. MAPIRANJE PRIRODNOG JEZIKA NA GRAPHQL

U ovom poglavlju biće dat pregled različitih tehnika koje predstavljaju osnovu za precizno mapiranje prirodnog jezika na GraphQL upite.

3.1. Tokenizacija

Tokenizacija, u domenu obrade prirodnog jezika i mašinskog učenja, predstavlja proces pretvaranja tekstualnog niza u manje delove koji se zovu tokeni [3]. Ovi tokeni mogu biti mali poput karaktera ili dugi kao reči [3]. Bitnost ovog procesa ogleda se u tome što pomaže mašinama da razumeju ljudski jezik razbijajući ga na male komade, koji su lakši za analizu [3].

Primarni cilj tokenizacije je predstavljanje teksta na način koji je odgovarajući za mašine, a bez gubitka konteksta [3]. Pretvaranjem teksta u tokene, algoritmi mogu lakše da identifikuju obrasce što je ključno za razumevanje i odgovarajuće reagovanje na ljudski unos [3].

3.2. Parsiranje

Parsiranje, u oblasti obrade prirodnog jezika, je proces ispitivanja gramatičke strukture i odnosa unutar date rečenice ili teksta [4]. Ono podrazumeva analizu teksta da bi se utvrdile uloge određenih reči, kao što su imenice, glagoli i pridevi, kao i njihovi međusobni odnosi [4].

Ova analiza proizvodi strukturiranu reprezentaciju teksta, omogućavajući NLP procesima da razumeju kako se reči u frazi povezuju jedna sa drugom [4]. Parseri otkrivaju strukturu rečenice konstruisanjem stabala zavisnosti koja ilustruju hijerarhijske i sintaksičke odnose između reči [4].

3.3. Prepoznavanje imenovanih entiteta (NER)

Prepoznavanje imenovanih entiteta je komponenta ekstrakcije informacija koja ima za cilj da identifikuje i kategorizuje imenovane entitete unutar nestrukturiranog teksta [5]. NER podrazumeva identifikaciju ključnih informacija u tekstu i klasifikaciju u skup unapred definisanih kategorija [5]. Entitet je stvar o kojoj se konstantno govori ili na koju se poziva u tekstu, kao što su imena osoba, organizacija, lokacija, vremenskih izraza, količine, procenata i više unapred definisanih kategorija [5].

3.4. Analiza sentimenata

Analiza sentimenata predstavlja proces klasifikacije kojim se utvrđuje da li je blok teksta pozitivan, negativan ili neutralan [6]. Cilj koji rudarstvo sentimenata (eng. sentiment mining) pokušava da postigne je da se analiziraju mišljenja ljudi na način koji može pomoći preduzećima da se dalje razvijaju i šire [6]. Fokusira se ne samo na polaritet (pozitivan, negativan i neutralan), već i na emocije (srećne, tužne, ljute, itd.) [6].

4. DIZAJN I IMPLEMENTACIJA NLP SISTEMA

Koraci u razvoju NLP sistema:

1. Prikupljanje podataka za treniranje

Prvi korak je prikupljanje velikog skupa podataka rečenica na prirodnom jeziku i njima odgovarajućih GraphQL upita [7]. Ovi podaci mogu se prikupiti ručno ili koristeći već postojeće skupove podataka koji su dostupni na mreži [7].

2. Pretprocesiranje podataka

Sledeći korak je pretprocesiranje podataka čišćenjem, normalizacijom i tokenizacijom teksta [7]. Ovo može uključivati uklanjanje stop reči, interpunkcije i specijalnih znakova, kao i pretvaranje teksta u mala slova [7].

3. Treniranje jezičkog modela

Obučavanje jezičkog modela koristeći prethodno pretprocesirane podatke da generiše GraphQL upite na osnovu ulaznih rečenica [7]. Opcije su fino podešavanje pretreniranog modela ili obučavanje novog modela od nule [7].

4. Evaluacija modela

Kada je model obučan, trebalo bi proceniti njegove performanse koristeći skup testnih rečenica i njima odgovarajućih upita [7]. Mogu se koristiti metrike kao što su tačnost, preciznost i pamćenje da bi se izmerile performanse modela [7].

5. Primena modela

Kada je model obučan i evaluiran, može biti primenjen kao veb servis ili API kome korisnici mogu da pristupe kako bi generisali odgovarajuće GraphQL upite na osnovu zadatih upita na prirodnom jeziku [7].

6. Poboljšavanje modela

Tokom vremena model treba nastaviti poboljšavati prikupljanjem povratnih informacija od korisnika i obučavanjem na novim podacima [7]. Ova poboljšanja će učiniti da model vremenom postane tačniji i robusniji [7].

4.1. Treniranje jezičkog modela

BART

BART model, predstavljen 2019. godine, je model sekvenca-do-sekvence obučan kao autoenkoder za smanjenje šuma [8]. Ovo znači da fino podešen BART model može uzeti tekstualnu sekvencu (na primer, na engleskom jeziku) kao ulaz i proizvesti drugačiju tekstualnu sekvencu na izlazu (na primer, na francuskom jeziku) [9]. Ovaj tip modela je relevantan za mašinsko

prevođenje. odgovaranje na pitanja, sumiranje teksta ili klasifikaciju sekvenci (kategorizacija rečenica ili leksema ulaznog teksta) [9].

Pošto nenadgledano pretreniranje BART transformatora rezultuje jezičkim modelom, moguće je fino podesiti ovaj jezički model za određeni NLP zadatak [9]. Kako je model već pretreniran, za fino podešavanje nisu potrebni masivni označeni skupovi podataka (u odnosu na ono što bi bilo potrebno za obuku od nule) [9]. BART model se može fino podesiti na skupove podataka specifičnih za domen za razvoj aplikacija kao što su medicinski četbotovi, pretvaranje teksta na prirodnom jeziku u programski kod ili SQL upite, aplikacije za prevođenje jezika specifičnih za kontekst ili alat za parafraziranje istraživačkih radova [9].

BART transformator je obučen kao autoenkoder za uklanjanje šuma, tako da podaci za obuku uključuju „pokvaren“ tekst ili tekst sa šumom i njegovo mapiranje u čisti ili originalni tekst [9]. Format obuke je sličan obuci bilo kog autoenkodera za smanjenje šuma [9]. Autoenkoderi se obučavaju da uklone šum ili poboljšaju kvalitet slike tako što imaju slike sa šumom u podacima obuke mapirane sa čistim, originalnim slikama kao ciljem [9].

T5 model

Model T5, skraćeno od Text-to-Text Transfer Transformer, je model obrade prirodnog jezika koji je razvio Google [10]. Zasnovan je na transformer arhitekturi, koja je vrsta neuronske mreže za koju se pokazalo da je veoma efikasna u NLP zadacima [11]. Jedinstven je jer je dizajniran da bude veoma prilagodljiv i može se koristiti za širok spektar zadataka, uključujući klasifikaciju teksta, odgovaranje na pitanja i prevod jezika [11].

Sastoji se od enkoder-dekoder strukture, slično drugim sekvencija-do-sekvence modelima [11]. T5 model koristi skup slojeva transformatora u komponentama enkodera i dekodera, omogućavajući mu da uoči i obradi hijerarhijske reprezentacije ulaznih i izlaznih sekvenci [11].

Enkoder obrađuje ulaznu sekvencu, koja može biti kombinacija opisa zadatka i ulaznih podataka [11]. Primenuje mehanizme samo-pažnje kako bi uočio odnose između reči i kontekstualnih informacija [11]. Mehanizam samo-pažnje omogućava modelu da se bavi različitim delovima ulazne sekvence, uzimajući u obzir zavisnosti i odnose između reči [11].

Dekoder, s druge strane, generiše izlaznu sekvencu [11]. Uzima konačno skriveno stanje enkodera kao ulaz i vodi računa o relevantnim delovima ulazne sekvence tokom procesa dekodiranja [11]. Dekoder generiše izlaz korak po korak, koristeći mehanizam pažnje kako bi se fokusirao na različite delove ulaza dok predviđa sledeći token [11].

I slojevi ekodera i dekodera u T5 modelu koriste mehanizam samo-pažnje sa više glava, što omogućava modelu da razmotri više perspektiva i uhvati različite zavisnosti unutar ulaznih i izlaznih sekvenci [11]. Mehanizam pažnje poboljšava sposobnost modela da se nosi sa zavisnošću dugog dometa i efikasno uoči kontekstualne informacije [11].

T5 model takođe uključuje poziciono kodiranje za kodiranje informacija o položaju ulazne sekvence [11]. Ovo poziciono kodiranje pomaže modelu da razume redosled i poziciju leksema u nizu, što je ključno za uočavanje sekvencijalne prirode jezika [11].

5. DIZAJN I IMPLEMENTACIJA GRAPHQL API SISTEMA

Kreiranje GraphQL šeme

Šema je struktura podataka koje GraphQL klijenti mogu da zahtevaju.

Na listingu broj 1 prikazan je programski kod za kreiranje GraphQL šeme.

```
const { ApolloServer, gql } = require('apollo-server');

const typeDefs = gql`

  type Apartment {
    title: String
    description: String
    price: Float!
    floor: Int!
    area: Float!
    structure: Float!
    city: String!
    constructionYear: Int
  }

  type Query {
    apartments: [Apartment]
  }
`;
```

Listing 1. Kreiranje GraphQL šeme

Resolver funkcije

Dok šema definiše strukturu naše API biblioteke, *resolver* funkcije definišu odakle dolaze podaci. Koristeći GraphQL podaci se mogu preuzimati iz brojnih izvora podataka, poput SQL i NoSQL baza podataka, REST API biblioteka, drugih GraphQL API biblioteka, pa čak i iz samog statičkog JSON objekta).

6. DEMONSTRACIJA UPOTREBE SISTEMA

Prvi korak u upotrebi sistema jeste postavljanje upita na prirodnom jeziku. Ovaj upit postavlja korisnik sistema putem NLP interfejsa.

Primer upita:

“Pronađi mi stanove u Novom Sadu sa cenom ispod 150.000€ izgrađene najranije 2020. godine.”

Na osnovu upita na prirodnom jeziku, NLP sistem generiše odgovarajući GraphQL upit.

Primer odgovarajućeg GraphQL upita prikazan je na listingu broj 2:

```
query {
  apartments(input: {
    city: "Novi Sad"
    maxPrice: 150000
    minConstructionYear: 2020 })
  {
    title
    description
    price
    city
    constructionYear
  }
}
```

Listing 2. Primer izgenerisanog GraphQL upita

Nakon toga, generisani GraphQL upit se izvršava i dobijeni rezultati se prikazuju korisniku.

Primer dobijenih rezultata prikazan je na listingu broj 3:

```
{
  "data": {
    "apartments": [
      {
        "title": "Jednosoban stan na
Grbavici",
        "description": "Nov stan na odličnoj
lokaciji.",
        "price": 99.000,
        "city": "Novi Sad",
        "constructionYear": 2023
      }
    ]
  }
}
```

Listing 3. Primer rezultata izvršavanja GraphQL upita

7. PRIMENA I UPOTREBLJIVOST GRAPHQL BIBLIOTEKE U NLP SISTEMIMA

Prezentovano rešenje je od velike pomoći pri:

- konverziji složenih upita na prirodnom jeziku,
- rukovanju složenim filterima i agregacijama,
- povećanju efikasnosti upita uz smanjenje broja API poziva,
- jednostavnom rukovanju relacionim podacima i
- integraciji sa drugim tehnologijama i alatima.

8. ZAKLJUČAK

U ovom radu istražena je ideja povezivanja NLP interfejsa i GraphQL API sistema radi olakšavanja pretrage podataka koristeći upite na prirodnom jeziku. Predstavljani su osnovni koncepti ove dve tehnologije i napravljen je pregled trenutnog stanja u praksi. Diskutovani su brojni NLP pristupi i tehnike i neophodni procesi za mapiranje upita na prirodnom jeziku na GraphQL upite. U poglavlju o implementaciji sistema, pružen je pregled potrebnih koraka za uspešnu realizaciju NLP sistema i razmotreni su potencijalni jezički modeli koji bi mogli dovesti do najboljih rezultata, poput BART i T5 modela. Dostupan je i pregled kreiranja jednostavne GraphQL šeme neophodne za implementaciju GraphQL

API sistema. Na kraju, prezentovana je upotreba sistema i predstavljeni su dodatni primeri koji svedoče o upotrebljivosti GraphQL biblioteke za razvoj NLP interfejsa.

Kako je predstavljeno samo konceptualno rešenje dalji koraci bi podrazumevali realnu implementaciju predloženog sistema. Ovaj korak bi omogućio testiranje predloženih modela jezika i procenu tačnosti i efikasnosti rezultata koje oni pružaju.

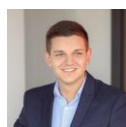
Kroz postojanje brojnih NLP sistema koji se fokusiraju na generisanje SQL upita na osnovu prirodnog jezika jasno je da potreba za ovakvim sistemima postoji. Kada se na to doda jedna od najvećih prednosti GraphQL biblioteke, odnosno mogućnost istovremenog korišćenja različitih izvora podataka, uključujući SQL i NoSQL baze podataka, REST API, mikroservise i slično, očekivan je još veći doprinos sistema koji integrišu NLP i GraphQL.

Sledeći korak u oblasti povezivanja NLP i GraphQL tehnologija jeste kreiranje javno dostupnog skupa podataka za treniranje jezičkih modela koji mapira upite na prirodnom jeziku na njima odgovarajuće GraphQL upite. Ovo bi, ne samo ubrzalo razvoj sistema koji kombinuju ove dve tehnologije već i velikoj meri povećalo tačnost korišćenih jezičkih modela.

9. LITERATURA

- [1] <https://www.ibm.com/topics/natural-language-processing> (pristupljeno u junu 2024.)
- [2] <https://graphql.org> (pristupljeno u junu 2024.)
- [3] <https://www.datacamp.com/blog/what-is-tokenization> (pristupljeno u junu 2024.)
- [4] <https://intellipaat.com/blog/what-is-parsing-in-nlp/> (pristupljeno u junu 2024.)
- [5] <https://www.geeksforgeeks.org/named-entity-recognition/> (pristupljeno u junu 2024.)
- [6] <https://www.geeksforgeeks.org/what-is-sentiment-analysis/> (pristupljeno u junu 2024.)
- [7] <https://ethglobal.com/showcase/the-graph-ai-z9cgp> (pristupljeno u junu 2024.)
- [8] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension", 2019
- [9] <https://www.projectpro.io/article/transformers-bart-model-explained/553> (pristupljeno u junu 2024.)
- [10] N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer", 2020
- [11] <https://databasecamp.de/en/ml-blog/t5-model> (pristupljeno u junu 2024.)

Kratka biografija:



Dragan Borkovac rođen je u Rumi 1996. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2024.god. kontakt: borkovac.dragan96@gmail.com

**PRIMENA OPERATIVNOG SISTEMA ZA RAD U REALNOM VREMENU U
INTERNETU STVARI****APPLICATION OF REAL TIME OPERATING SYSTEM IN THE INTERNET OF
THINGS**

Đorđe Novaković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MERNO-INFORMACIONI SISTEMI

Kratak sadržaj – U ovom radu realizovana je aplikacija koja prati očitavanja sa senzora i prikazuje ih na internet stranici. Senzori očitavaju temperaturu, prisustvo dima, osvetljenje i pritisak i vlažnosti vazduha. U projektu FreeRTOS operativni sistem omogućava organizaciju i upravljanje resursima, kao i sinhronizaciju i komunikaciju između različitih delova softvera.

Ključne reči: FREERTOS, IoT, STM32

Abstract – In this paper an application has been developed to monitor sensor readings and display them on a website. The sensors monitor temperature, smoke ambient light, pressure, and air humidity. In the project, the FreeRTOS operating system enables the organization and management of resources, as well as synchronization and communication between different software components.

Keywords: FREERTOS, IoT, STM32

1. UVOD

Ideja za projekat je nastala na osnovnim studijama kada se u okviru diplomskog rada dizajnirala štampana ploča za kontrolu robota i povezivanje pripadajućih senzora. Osnovna ideja je da robot prikuplja podatke sa senzora i da se samostalno kreće po prostoru, a da prikupljene podatke šalje na internet stranicu povezanu sa pridruženom bazom podataka. Robot bi prikupljao vrednosti sobne temperature, pa bi korisnik po potrebi preko aplikacije, mogao da kontrolom klima uređaja vrši regulaciju temperature. Takođe, zamišljeno je da se meri jačina osvetljenja u prostoriji kao i prisutnost gasa u slučaju curenja iz cevi ili detekciju požara u kući. Tema master rada je da se takva ideja realizuje i da se jedan deo sistema realizuje. Deo koji je urađen jeste očitavanje vrednosti sa senzora i njihov prikaz na internet stranici. Za senzore temperature, vlažnosti, pritiska i osvetljenja su upotrebljene *click* pločice koje se direktno stavljaju na ploču za proširenje. Projekat je realizovan u C programskom jeziku primenom FreeRTOS operativnog sistema. U projektu je korišten DISCOVERY F407 razvojni sistem sa pločom za proširenje.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Đorđe Novaković

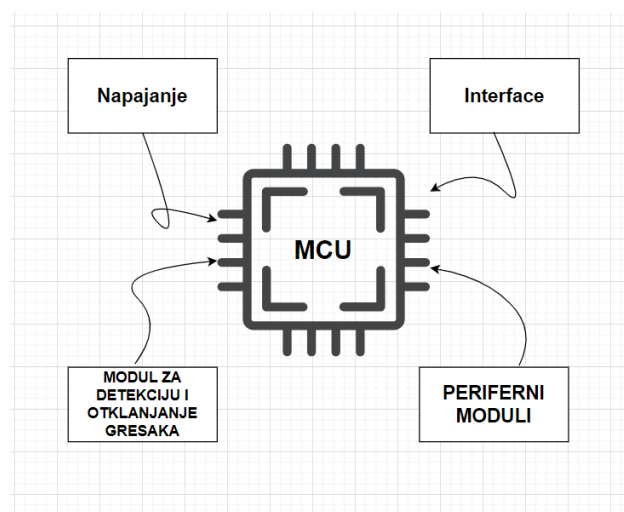
Za senzorske module korišteni su *Ambient click*, *Weather click*, *Wi-Fi (Wireless Fidelity) click* i MQ-135 senzor prisustva dima.

U drugoj sekciji biće opisan *hardware*-a razvojnog sistema kao i pojedinačni moduli.

U četvrtoj sekciji pojašnjeno je korišćenje i uloga FreeRTOS operativnog sistema u projektu. U petoj sekciji predstavljeno je okruženje u kome se prikupljaju podaci kao i neke karakteristike koje pruža samo okruženje.

2. HARDWARE

Razvojna ploča je DISCOVERY board F407 i glavne komponente razvojne ploče prikazane su na slici 1.



Slika 1. Blok šema
DISCOVERY F407 ploče

2.1. Mikrokontroler

Glavna komponenta razvojnog sistema je STM32F407VG mikrokontroler. Pored mikrokontrolera, na ploči se mogu pronaći i oscilator koji generiše spoljašnji takt signal za mikrokontroler, kao i regulatori napona (LD3985M25R i LD3985M33R) za obezbeđivanje stabilnog napajanja neophodnih za pravilno funkcionisanje mikrokontrolera uz *decoupling* kondenzatore koji služe kao dodatni filteri visokofrekventnih smetnji.

2.2. Periferni moduli

Na razvojnoj ploči se nalazi digitalni akcelerometar koji omogućava detekciju pokreta za sve tri ose (x,y,z). Pored

toga, ploča ima ugrađen digitalni mikrofoni koji omogućava snimanje zvuka, što je korisno za aplikacije kao što su prepoznavanje govora i audio snimanje. Takođe, ploča ima *Audio DAC (Digital to Analog converter)* sa integrisanim pojačavačem u klasi D, što omogućava reprodukciju zvuka visokog kvaliteta.

2.3. Napajanje

Koriste se čipovi za regulaciju napona i upravljanje napajanjem. Pošto različite periferije zahtevaju različite napone napajanja postoje komponente koje zahtevaju 2,5 V i 3 V i iz tih razloga postoje dva regulatora koji prilagođavaju napon 5 V jer se ploča napaja sa USB-a (*Universal Serial Bus*) računara.

2.4. Interfejsi

Mikrokontroler poseduje USB OTG (*On-The-Go*) čip i moguće je vršiti komunikaciju preko USB-a. Jedan od nedostataka razvojnog sistema je nepostojanje FT232 čipa kako bi se mogla vršiti komunikacija sa računarom preko UART-a (*Universal Asynchronous Receiver-Transmitter*) koja se zbog svoje jednostavnosti najčešće koristi za komunikaciju sa računarom. Zbog toga se koristi dodatna ploča na kojoj postoji FT232 čip i CAN (*Controller Area Network*) prijemnik, ako postoji potreba da se koristi takva komunikacija. Na slici 2. je prikazana ploča za proširenje koja omogućuje povezivanje senzora i ima pomenute konektore.



Slika 1. Ploča za proširenje

2.5. Otklanjanje grešaka

Ovaj sistem sadrži na sebi programator i *debugger* pomoću kojeg se može vršiti reprogramiranje mikrokontrolera kao i vršenje softverskog *debugging*-a samog sistema.

3. SENZORI

Za senzore temperature, vlažnosti, pritiska i osvetljenja su upotrebljene *click* pločice koje se direktno stavljaju na ploču za proširenje prikazanoj na slici 2. *Click* pločice predstavljaju standardizovano rešenje gde se senzori povezuju preko 16 pinova sa mikrokontrolerom posredstvom odgovarajućih konektora. Ovo omogućuje dodatnu fleksibilnost u razvijanju prototipa i povezivanju sa razvojnim sistemom jer se senzori mogu po potrebi menjati, a da se pri tome ne menja hardver sistema.

3.1 MQ-135

Na slici 3. prikazan je MQ-135 gasni senzor koji se koristi za detekciju različitih štetnih gasova u vazduhu, kao što su amonijak, dim, alkohol i druge supstance. Ovaj senzor je široko korišćen u raznim aplikacijama, uključujući merenje kvaliteta vazduha, bezbednost u domaćinstvu, kao i kontrolu industrijskih procesa.



Slika 3. MQ-135 gasni senzor

3.1.1 Princip rada

MQ-135 senzor koristi princip hemijske apsorpcije gasa kako bi se detektovalo prisustvo ciljanih gasova u vazduhu. U osnovi, senzor se sastoji od keramičkog elementa koji reaguje sa ciljanim gasom, izazivajući promene u električnom otporu senzora.

Promene u električnom otporu proporcionalne su koncentraciji gasa u vazduhu, što omogućava senzoru da detektuje prisustvo gasa i pruži odgovarajući izlazni signal.

3.2 Weather click

Na slici 4. prikazan je *Weather click* senzorski modul koji omogućava precizno merenje različitih meteoroloških parametara. Modul sadrži senzor za merenje temperature, vlažnosti i pritiska vazduha. Glavni komponenta koja se nalazi na pločici i koji meri sva tri parametra je senzorski čip BME280.



Slika 4. Weather click

3.2.1 Princip rada

1. Mrenje temperature

Temperatura se meri korišćenjem termistora. Termistor je elektronski uređaj čija električna otpornost varira sa promenom temperature. Termistor u BME280 senzoru se zagreva na temperaturu okoline, a zatim se meri promena električne otpornosti kako bi se odredila temperatura.

2. Merenje vlažnosti

Merenje vlažnosti vazduha se obično vrši korišćenjem senzora vlažnosti koji koristi princip apsorpcije vlage koji utiče na promenu kapacitivnosti kapacitivnosti, što je i realizovano u BME280 senzoru. Ovaj senzor ima dielektrični materijal koji apsorbuje vlagu iz okoline, čime se menja kapacitivnost senzora. Promena kapacitivnosti se zatim meri i pretvara u digitalni signal koji predstavlja relativnu vlažnost vazduha.

3. Merenje pritiska

Merenje vazdušnog pritiska se obično vrši korišćenjem senzora pritiska ili piezoelektričnog senzora tako što se njihove karakteristične vrednosti menjaju u zavisnosti od promene vazdušnog pritiska. BME280 senzor koristi piezoelektrični senzor pritiska, gde se promene u pritisku vazduha prenose na kristal koji generiše električni signal proporcionalan pritisku. Signal se zatim meri i pretvara u digitalni signal koji se može koristiti za određivanje atmosferskog pritiska.

3.3 Ambient click

Ambient 2 Click sa slike 5. je senzorska pločica koja se koristi za merenje ambijentalnog svetla. Na pločici se nalazi OPT3001 digitalni senzor jačine ambijentalnog osvetljenja.



Slika 5. *Ambient 2 click*

3.3.1 Princip rada

Ambient 2. OPT3001 meri nivo jačine ambijentalnog osvetljenja koristeći princip konverzije svetlosti u električni signal. Senzor poseduje fotodiodu koja apsorbuje svetlost iz okoline. Kada svetlost padne na fotodiodu, ona generiše struju proporcionalnu intenzitetu svetla. Generisani električni signal se zatim pretvara u digitalni signal unutar senzora. OPT3001 koristi integrisani analogni-digitalni konvertor. Digitalni signal se zatim obrađuje kako bi se dobio precizan podatak o nivou jačine ambijentalnog osvetljenja. Senzor ima ugrađene funkcije obrade signala koje omogućavaju precizno merenje osvetljenosti. Na osnovu digitalno obrađenog signala, senzor izračunava vrednost jačine osvetljenosti u luksima.

3.4 Wi Fi click

Wi-Fi ESP Click (slika 6.) je pločica koja pruža bežični prenos podataka. Ova pločica sadrži ESP- WROOM-02

potpuno integrisani Wi-Fi modul. Pruža bežičnu komunikaciju preko štampane antene u frekvencijskom opsegu od 2,4 GHz (802.11b/g/n). Modul podržava TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), HTTP (*Hypertext Transfer Protocol*) i FTP (*File Transfer Protocol*) mrežne protokole.



Slika 6. *Wi fi click*

3.4.1 Princip rada

Wi-Fi ESP Click radi na principu bežične komunikacije putem Wi-Fi mreže. Kada je Wi-Fi ESP *click* aktiviran sistem prenosi informacije koristeći frekvenciju od 2,4 GHz. Kada se Wi-Fi ESP *click* koristi kao pristupna tačka (*softAP* režim), on emituje Wi-Fi signal koji drugi uređaji mogu detektovati i povezati se sa njim. Kada se koristi kao Wi-Fi stanica, ESP-WROOM-02 se povezuje sa postojećom Wi-Fi mrežom kako bi omogućio pristup internetu ili lokalnoj mreži. Fizički sloj Wi-Fi komunikacije koristi radio frekvencijske signale za prenos podataka. Komunikacija sa čipom se odvija pomoću AT komandi koje se šalju putem UART-a. AT komande su naredbe koje se koriste za kontrolu i konfiguraciju uređaja koji podržavaju serijsku komunikaciju preko UART interfejsa.

Komande su standardizovane i često se koriste za upravljanje i konfiguraciju različitih uređaja kao što su Wi-Fi moduli, *Bluetooth* uređaji, i slično. U tabeli 1. prikazane su korišćene AT komande.

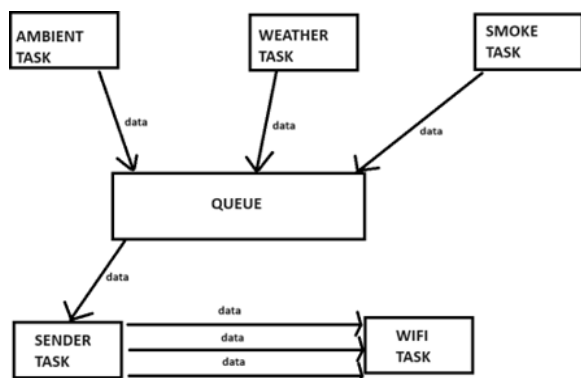
Tabela 1. *AT komande*

Komande	Opis njene uloge
AT+CIPSTART	Komanda se koristi za uspostavljanje TCP ili UDP veze sa udaljenim serverom ili uređajem
AT+CWMODE	Komanda se koristi za postavljanje režima rada Wi-Fi modula
AT+CWJAP	Komanda se koristi za povezivanje na Wi-Fi mrežu
AT+RST	Komanda se koristi za restartovanje Wi-Fi modula

4. ULOGA FREERTOS-A U APLIKACIJI

Ideja je da tri nezavisna zadatka očitavaju senzore i posredstvom zajedničkog reda upisuju očitane podatke tj. da prosleđuju podatke zadatku koji dalje prosleđuje podatke (*sender task*). *Sender task* prikupljene podatke iz tri nezavisna zadatka prosleđuje preko UART-a ka računaru koji će podatke ispisivati u konzolu i ukoliko se dogodi greška na srevskoj strani, da korisnik može brzo utvrditi da mikrokontrolerski sistem ispravno radi. Dva zadatka očitavaju dva senzora koja komuniciraju preko I2C (Inter-Integrated Circuit), oni pristupaju istom resursu. Stoga je potrebno da se obezbedi da u istom trenutku ne pristupaju I2C-u jer bi moglo doći do kolizije i zbog toga je uveden *Mutex*. *Red* koji je napravljen ima pet elementa i njegova veličina je struktura koja opisuje senzore.

Tajmer se koristi da omogućimo da se u ravnomernim periodima očitavaju podaci. Prilikom korišćenja *osDelay* može da dođe do vremenskog smicanja a sa *timer*-om se oslobađa pet semafora u istom trenutku ako svih pet čekaju na semafor u istom trenutku dobijaju semafor i vremenski su poravnati.



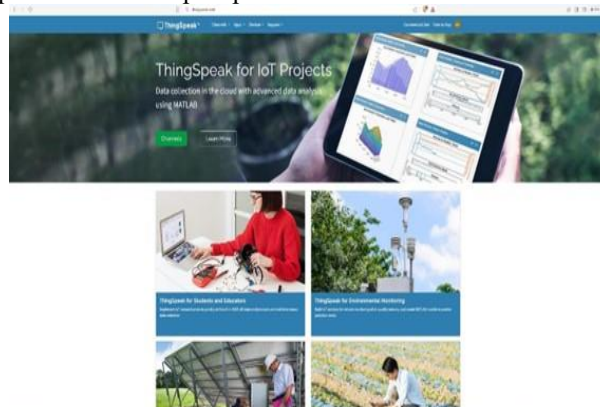
Slika 6. Blok šema FreeRTOS implementacije

Blok šema (slika 6.) predstavlja grafički prikaz implementirane FreeRTOS logike. Tri zadatka koji očitavaju vrednosti sa senzora pristupaju redu i upisuju podatke u njega. Zatim *sender task* kupi te podatke ispisuje ih putem UART-a i smešta očitane vrednosti u *buffer* i podaci se dalje šalju na sajt preko Wi-Fi *task*-a.

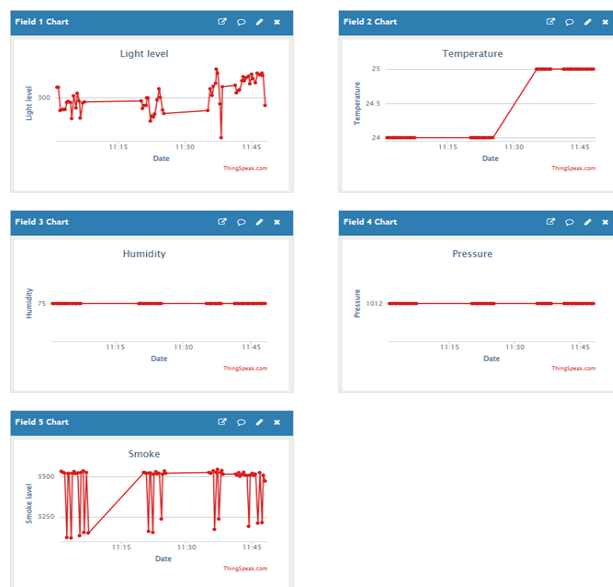
5. THINGSPEAK

ThingSpeak (slika 7.) je platforma koja pruža niz alata i funkcija prilagođenih potrebama za IoT (*Internet of Things*). U osnovi, implementirano je korisničko okruženje za prikupljanje, analizu i vizualizaciju podataka sa različitih IoT uređaja i senzora. *ThingSpeak* nudi alate za vizualizaciju podataka koji omogućavaju korisnicima da kreiraju grafikone, dijagrame i mape na osnovu podataka prikupljenih sa senzora. Pored vizualizacije podataka, sajt poseduje mogućnost analize podataka, omogućavajući korisnicima da se bave naprednom analitikom i da imaju značajne uvide u podatke koji se šalju na sajt. Na slici 7. dat je prikaz *ThingSpeak* okruženja, a na slici 8. je prikazano okruženje sa podacima koji se šalju sa mikrokontrolera. Takođe, na slici 8. prikazane su očitane vrednosti temperature, vlažnosti, pritiska, osvetljenja i prisutnosti dima. Podaci su očitavani na svakih petnaest sekundi jer *ThingSpeak* dolazi u besplatnoj verziji ali uz odgovarajuća ograničenja. Upravo jedno od ograničenja je

nemogućnost slanja više od jednog zahteva za upis na 15 s. Zbog toga se prilikom slanja podataka na *ThingSpeak*, podaci sa svih senzora objedinjuju u jedan zahtev i kao takvi se šalju u ekvidistantnim vremenskim intervalima. Korisnici se mogu po potrebi pretplatiti i na taj način povećati brzinu upisa podataka na *cloud*.



Slika 7. *Thingspeak* sajt



Slika 8. *Thingspeak* sajt sa graphicima očitanih podataka

6. ZAKLJUČAK

U radu je predstavljen projekat koji se bavi razvojem sistema za praćenje i kontrolu okoline koristeći senzore i slanjem podataka na *cloud*. Projekat uključuje integraciju različitih senzora za merenje temperature, vlažnosti, pritiska, osvetljenja i prisustva dima, sa mikrokontrolerom STM32 (*STMicroelectronics 32-bit ARM-based Microcontroller*) i upotrebom različitih komunikacionih protokola i interfejsa kao što su UART, I2C, TCP itd. Takođe, u projektu je korišćen operativni sistem za rad u realnom vremenu FreeRTOS. Detaljno su razmatrane implementacije funkcija u programskom kodu, kao i tehnike komunikacije, uključujući korišćenje AT komandi za komunikaciju sa Wi-Fi modulom ESP8266. Projekat imaza cilj kreiranje kompleksnog sistema koji omogućava nadzor okoline, prikupljanje podataka sa senzora, njihovo procesiranje i prenošenje putem interneta, sa mogućnošću

daljinskog upravljanja i praćenja podataka na internetu.

Dalji pravci razvoja mogu biti implementacija dela koji kontroliše kretanje robota. Robot može da se kreće na dva točka, gde bi implementirani algoritam za upravljanje mogao biti realizovan PID (*Proportional-Integral-Derivative*) kontrolerom, kako bi sam robot izgledao atraktivnije kada bi se plasirao na tržište. Trebalo bi dizajnirati mehaničke delove zatim projektovati i izraditi PCB (*Printed Circuit Board*) sa akcentom na minimizaciju same pločice. Potrebno je odabrati motore i baterije kako bi se oni uklopili u odgovarajuću težinu da bi robot bio funkcionalan, a sa druge strane da bude dovoljno dugo autonoman i da može samostalno raditi bez punjenja. Dalji razvoj takođe može biti usmeren da se postojeća aplikacija integriše sa *hardware*-om autonomnog usisivača gde bi se mogli pratiti parametri okoline.

7. LITERATURA

- [1] Richard Barry., "Mastering the FreeRTOS™ Real Time Kernel"
- [2] Steve Oualline, "Practical C Programming"
- [3] Samuel A. Rebelsky, "A Practical Introduction to C Programming Language"
- [4] Greg Perry, "C Programming Absolute Beginner's Guide"
- [5] Stephen G. Kochan, "Programming in C"
- [6] Warren Gay, "Mastering the FreeRTOS Real Time Kernel on STM32 microcontrollers: A Practical Guide"
- [7] Joseph Yiu, "The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors"
- [8] Robert Ward, "Beginning STM32: Developing with FreeRTOS, libopencm3 and GCC"
- [9] Samuel W. Lam, "STM32 Arm Programming for Embedded Systems: Volume 6"

Kratka biografija:



Dorde Novaković rođen je u Novom Sadu 1998. godine. Master rad na Fakultetu Merno-informacioni sistemi odbranio je 2024. godine.

kontakt: djordjenovakovic36@gmail.com

UTICAJ OBNOVLJIVIH IZVORA ENERGIJE NA ELEKTRODISTRIBUTIVNU MREŽU**THE IMPACT OF RENEWABLE ENERGY SOURCES ON THE ELECTRICAL DISTRIBUTION NETWORK**Nikola Sarkić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu se razmatraju procedure i propisi priključenja obnovljivih izvora energije (OIE) na distributivnu mrežu (DM). Prikazani su rezultati proračuna tokova snaga, kratkih spojeva, tranzijentne stabilnosti i kriterijuma za priključenje novog OIE na DM uz korišćenje softverskih alata PSS Sincal i ETAP.

Ključne reči: tokovi snaga, kratki spojevi, tranzijentna stabilnost, Hosting capacity, distributivna mreža, male solarne elektrane (MSE), male hidroelektrane (MHE), PSS Sincal, ETAP.

Abstract The paper discusses the procedures and regulations for connecting renewable energy sources (RES) to the distribution network (DM). The results of calculating power flows, short circuits, transient stability, and criteria for connecting the new OIE to DM using PSS Sincal and ETAP software tools are presented.

Keywords: power flows, short circuits, transient stability, Hosting capacity, distribution network, small solar power plants, small hydropower plants, PSS Sincal, ETAP.

1. UVOD

Na distributivnu mrežu (DM) se priključuje sve veći broj obnovljivih izvora za proizvodnju električne energije (energije). I pored svojih velikih prednosti, priključenje velikog broja ovih postrojenja može predstavljati problem po rad DM i operatora koji njome upravlja [1,2]. Samo neki od problema su zagušenje u mreži, problemi regulacije napona [13], povratak energije u prenosnu mrežu [14], itd.

1.1. Obnovljivi izvori energije

Neobnovljivi izvori energije negativno utiču na životnu sredinu. Pritom, rezerve goriva potrebne za njihov rad su na izmaku [3]. Sa druge strane obnovljivi izvori energije (OIE), koji ne samo da za proizvodnju energije koriste praktično neograničene resurse prirodne izvore već svojom proizvodnjom ne narušavaju životnu sredinu. U skladu sa tim, OIE imaju sve veći udeo u proizvodnji ukupne svetske energije. Pet osnovnih tipova OIE su elektrane koje koriste energiju sunca, vetra, biomase, vode i geotermalnu energiju [4,5]. Postrojenja za proizvodnju energije iz OIE nazivaju se distribuirani energetske resursi (eng. Distributed energy resources – DER) [6,7]. Proizvodnja i

skladištenje električne energije koju obavljaju različiti mali uređaji povezani na mrežu predstavlja "Distribuiranu proizvodnju".

2. POSTAVKA PROBLEMA

U ovom delu dati su osnovni tehnički podaci o DM, podaci o elektrani i kriterijumi za priključenje elektrane [8].

2.1. Osnovni tehnički podaci o distributivnoj mreži

Nazivni naponi DM (U_n) na koju može da se priključi elektrana su: $U_n = 0,4$ kV (1 kV), 10 kV, 20 kV i 35 kV. Snage kratkih spojeva zavise od nazivnog napona DM. Tipične vrednosti maksimalnih dozvoljenih struja (snaga) trofaznih kratkih spojeva:

- mreža 0.4 kV: 26 kA (18MVA) u kablovskoj mreži i 16 kA (11 MVA) u nadzemnoj mreži;
- mreža 10 kV: 14.5 kA (250 MVA)
- mreža 20 kV: 14.5 kA (500MVA)
- mreža 35 kV: 12 kA (750MVA)

Tipične vrednosti kapacitivne struje zemljospoja u izolovanim SN mrežama, zavisno od nazivnog napona DM, iznose 20, 15 i 10 A za mreže 10, 20 i 35 kV, respektivno.

Uzemljenje neutralne tačke DM izvodi se prema važećim tehničkim propisima, u zavisnosti od nazivnog napona neutralna tačka DM 0.4 i 1 kV su direktno uzemljenja, a neutralne tačke DM 10, 20 i 35 kV su izolovane, ili uzemljene preko niskoomske impedanse [8].

2.2. Osnovni tehnički podaci o elektrani

Elektrane se, u zavisnosti od vrste primarnog izvora energije i primenjene tehnologije proizvodnje energije, dele na [8]: hidroelektrane; vetroelektrane; solarne elektrane; kogenerativna postrojenja sa istovremenom proizvodnjom električne i toplotne energije; elektrane na biomasu, biogas, otpatke, gorivne ćelije, itd; i elektrane na konvencionalna goriva (ugalj, nafta, gas).

U odnosu na funkcionalne zahteve, proizvodne jedinice koje se priključuju na DM dele se na [8]:

- Tip A – proizvodna jedinica čija je nazivna snaga veća od 0,8 kW, a manja od 500 kW;
- Tip B – proizvodna jedinica čija je nazivna snaga veća od 500 kW, a manja od 10 MW;
- Tip C – proizvodna jedinica čija je nazivna snaga veća od 10 MW, a manja od 20 MW.

2.3. Osnovni kriterijumi za priključenje elektrane

Na DM može da se priključi elektrana koja ispunjava

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji je mentor dr Goran Švenda, red. prof.

sledeće kriterijume za priključenje [8]:

- dozvoljenog strujnog opterećenja elemenata DM;
- dozvoljene promene napona u stacionarnom režimu;
- dozvoljene promene napona u prelaznom režimu (kriterijum dozvoljene snage generatora);
- dozvoljenih flikera (vetroelektrane, hidroelektrane i solarne elektrane);
- dozvoljenih struja i napona viših harmonika (elektrane priključene preko invertora/pretvarača);
- snage kratkog spoja (elektrane snage veće od 1 MVA);
- nesimetrije napona.

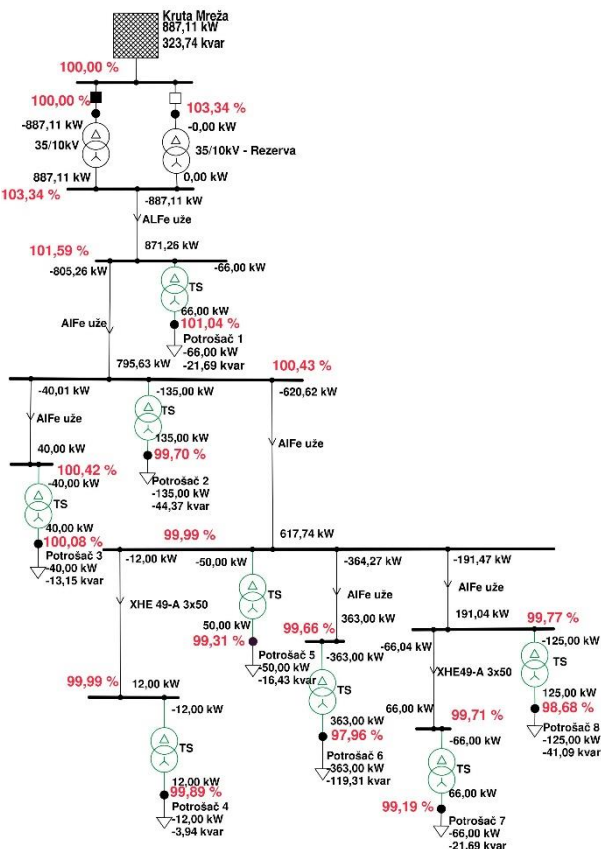
3. SOFTVERSKI ALATI

Za potrebe ovog rada, analiza uticaja OIE na rad DM, korišćeni su softverski alati ETAP i PSS Sincal. Generalno, ovi alati se koriste za modelovanje, simulaciju i analizu distributivnih i prenosnih elektroenergetskih mreža. Samo neke od njihovih mogućnosti su proračuni kratkih spojeva i tokova snaga, procenu kvaliteta napajanja, harmonijske analize, analize pokretanja motora, analize procene opasnosti luka kratkog spoja (ArcFlash), analiza tranzijentne stabilnosti i proračun podešenja i koordinacije zaštite [9,10].

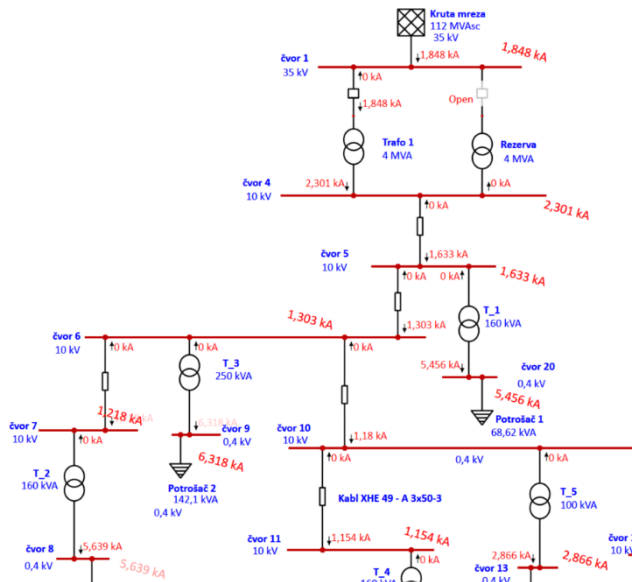
4. VERIFIKACIJA

Na primeru distributivne Test mreže sa 20 čvorova, slici 4.1, razmatrani su proračuni kriterijuma koji su potrebnih za priključenje OIE na DM. Razmatrana su tri režima:

- osnovni režim Test mreže bez OIE, slika 4.1;
- režim Test mreže sa priključenom MHE i
- režim Test mreže sa priključenom MSE.



Slika 4.1 – PSS Sincal: Osnovnom režimu (bez OIE)



Slika 4.2 – ETAP: Vrednosti struje 3pks na sabirnicama (Deo test mreža bez OIE)

Na osnovu prikazanih rezultata (PSS Sincal) može se uočiti da pre dodavanja OIE u Test mreži nema problema sa naponskim prilikama, i da je aktivna snaga na početku izvoda 887,11 kW, a reaktivna 323,74 kVAr, slika 4.1.

Smatra se da je trolpolni kratak spoj referentni za stabilnost sistema (ako je stabilnost zadovoljena pri ovom kvaru biće i za sve ostale tipove kvara). Vrednosti struja maksimalnog trolpnog kratkog spoja na sabirnicama prikazane su na slici 4.2. Može se uočiti da za Test mreže bez OIE nema kritično velikih struja kratkog spoja, odnosno da su po Tehničkoj preporuci sve vrednosti kratkog spoja u dozvoljenim granicama, manje od 14.5 kA.

4.1. Priključenje Male Hidroelektrane MHE SSuEE

U čvoru br. 11 priključena je Mala Hidroelektrana: instalirana snaga 470 kVA; maksimalna aktivna snaga: 450 kW; naponski nivo priključenja 10 kV i faktor snage: 0.95.

Project: MHE SSuEE Connection rule: VDE-AR-N
Title: Used guideline: 4110
Date:

Network Data: Generating Plant	
Network Element	MHE SSuEE
Connection point	
Max. apparent connection power	470.0 kVA
Max. active connection power	450.0 kW
Power factor	0.95
Rated current	27.14 A
Number of generator units	1
Highest active connection power of generator unit	450.0 kW
Converter manufacturer	Siemens
Converter model	Line commutated
Converter type	Typo A
Network Data: Considered Subnetwork	
0 generating plants in the subnetwork	
Subnetwork nodes	
Subnetwork elements	AlFe uže, 35/10kV, 35/10kV - Rezerva,...

Results

Last calculation: 2024-02-05 12:10:08, Input data: 2024-02-05 12:10:08

Criteria	State
Short Circuit Power	OK
Equipment limits	Not passed
Slow voltage change	Not passed
Fast voltage change	Not passed
Flicker	OK
Harmonics	OK

Slika 4.3 – Rezultat priključenja MHE SSuEE

Na slici 4.3 su prikazani rezultati kriterijuma priključenja MHE na Test mrežu. Može se uočiti da postoje problemi sa opremom, brzim i sporim promenama napona. Ostali kriterijumi ne predstavljaju probleme pri priključenju MHE. Na slici 4.4 prikazane su vrednosti rezultata kriterijuma kod kojih su uočeni problemi. Vrednosti brzih i sporih promena napona su 3.38% (dozvoljena granica za spore promene je 3%, a za brze promene je 2%).

Equipment Limits

Type	Name	Value	Limit	Remark
Continuous current load				
Max. continuous current load - Transformer	TS	97,1 %	< 100,0 %	OK
Max. continuous current load - Line	AlFe uže	14,57 %	< 100,0 %	OK
Max. continuous current load - Cable	XHE 49-A 3x50	15,4 %	< 100,0 %	OK
Continuous current short circuit (MAX)				
Max. short circuit utilization	---	NO RESULT---	< 100,0 %	Not performed
Max. clearing time cable/line	AlFe uže	0,1 s	> 0,5 s	Not passed

Voltage change

Type	Name	Value	Limit	Remark
Slow voltage change				
Connection point	čvor 11	3,38 %	< 3,0 %	Not passed
Connection point max. ΔV	čvor 11	3,38 %	< 3,0 %	Not passed
Fast voltage change				
Voltage change at the connection point (connect)	čvor 11	3,38 %	< 2,0 %	Not passed
Voltage change at the connection point (disconnect)	čvor 11	3,38 %	< 2,0 %	Not passed

Slika 4.4 – Vrednosti rezultata proračuna kriterijuma MHE SSuEE

Na slici 4.5 su prikazani rezultati kriterijuma za priključenje MHE na Test mrežu nakon što je na deonici 4–5 promenjen presek i tip provodnika. Zadovoljen je i kriterijum dimenzionisanja opreme nakon promene vrednosti referentne struje kratkog spoja (1s) [kA].

Results

Last calculation: 2024-02-05 18:28:36, Input data: 2024-02-05 18:28:35

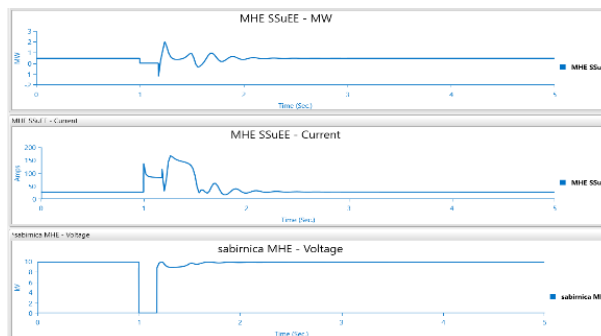
Criteria	State
Short Circuit Power	OK
Equipment limits	OK
Slow voltage change	OK
Fast voltage change	OK
Flicker	OK
Harmonics	OK

Slika 4.5 – Rezultati proračuna priključenja MHE SSuEE, nakon promene deonice 4–5

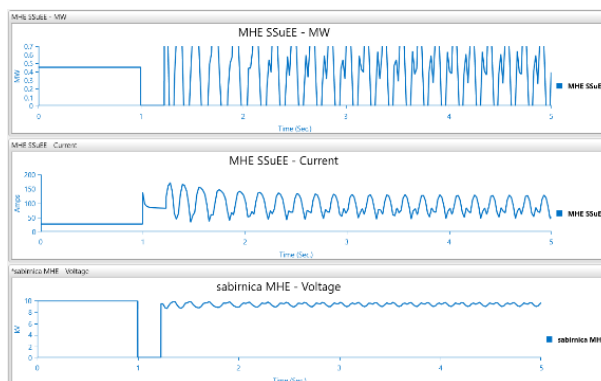
Tranzijentna stabilnost

Jedno od najvažnijih pitanja koje se nameće sa aspekta uklapanja MHE u EES jeste na koji način će one uticati na vrednosti struje kvara u priključnoj mreži. Pritom poznato je da usled kvara dolazi do propada napona u mreži. Udaljenost kvara od Male Hidroelektrane diktira kolika će biti dubina tog propada, a trajanje kvara određuje dužinu tog propada. U ovakvim analizama se kao referentni slučaj uzima tropolni kratak spoj. Ako se uzmu u obzir sve jači zahtevi za proizvodnju “čiste” energije izuzetno je bitno razmotriti stabilnost MHE pri mogućim kvarovima u EES-u. Kvarovi su neizbežni, ali je potrebno nakon njihovog eliminisanja uspostaviti održiv režim u kome će potrošači biti snabdevani dovoljnom količinom energije [11].

Na slici 4.6 vreme trajanja kvara je 180ms. Nakon što se kvar ukloni generator se vratio u normalan pogon. Na slici 4.7 su prikazane veličine za vreme trajanja kvara 240ms. Nakon te dužine trajanja kvara generator ulazi u nestabilan režim. Jasan pokazatelj nestabilnog rada je talasni oblik aktivne snage generatora na kom se vide oscilacije velike amplitude koje se ne prigušuju u vremenu.



Slika 4.6 – ETAP: Veličine snage, struje i napona za vreme trajanja kvara od 180 ms



Slika 4.7 – ETAP: Veličine snage, struje i napona za vreme trajanja kvara od 240 ms

4.2. Priključenje Solarne elektrane MSE SSuEE

U čvoru br. 10 priključena je Mala Solarna elektrana: instalirana snaga 450 kVA; maksimalna aktivna snaga: 450 kW; naponski nivo priključenja 10 kV i faktor snage: 1.00.

Na slici 4.8 su prikazani rezultati provere kriterijuma priključenja MSE. Može se uočiti da priključenje MSE neće imati za posledicu probleme u DM sa flikerima, harmonicima, brzim promenama napona i opremom. Ipak, uočeni su problemi na sporim promenama napona.

Results

Last calculation: 2024-05-24 16:20:26, Input data: 2024-05-24 16:20:26

Criteria	State
Short Circuit Power	Not performed
Equipment limits	OK
Slow voltage change	Not passed
Fast voltage change	OK
Flicker	OK
Harmonics	OK

Slika 4.8 – Rezultat proračuna za priključenje MSE SSuEE

Na slici 4.9 su prikazane vrednosti rezultata sporih promena napona. Problemi su u čvorovima 10 i 11. Ovaj problem zahteva promenu preseka provodnika na deonicama 5–6 i 6–10, npr. zamenom Al-Fe uže sa kablom XHE 49-A 3x1x150mm².

Voltage change

Type	Name	Value	Limit	Remark
Slow voltage change				
Connection point	čvor 10	3,19 %	< 3,0 %	Not passed
Connection point max. ΔV	čvor 11	3,87 %	< 3,0 %	Not passed

Slika 4.9 – Vrednosti rezultata sporih promena napona

Na slici 4.10 su prikazani rezultati kriterijuma za priključenje MSE na Test mrežu nakon što su na deonicama 5–6 i 6–10 zamenjeni preseki provodnika. Može se uočiti da su tada svi kriterijumi ispunjeni.

Results

Last calculation: 2024-05-24 16:48:16, Input data: 2024-05-24 16:48:16

Criteria	State
Short Circuit Power	Not performed
Equipment limits	OK
Slow voltage change	OK
Fast voltage change	OK
Flicker	OK
Harmonics	OK

Slika 4.10 – Rezultati proračuna priključenja MSE SSuEE nakon zamene preseka na deonicama 5–6 i 6–10

Results Hosting Capacity											
1/22 Results											
Node	[S] [MVA]	P [MW]	Q [Mvar]	vmin [%]	vmax [%]	ΔV [%]	ith [%]	ibmax [%]	ipmax [%]	Rev. Feed [MW]	State
čvor 15	0,614	0,614	0,000	101,425	103,424	1,506	80,641	0,000	0,000	0,000	Ok

Slika 4.11 – PSS Sincal: Rezultat proračuna Hosting kapaciteta na proširenoj Test mreži

5. ZAKLJUČAK

U radu su opisane:

- osnove OIE, i njena sve veća upotreba u svetu;
- procedure i propisi priključenja novog OIE na DM.
- kako i na koji način se vrši priključenje OIE, uz uslove i kriterijume koji su analizirani kroz softverske alate PSS SINCAL i ETAP.

Na osnovu prikazanih rezultata može se uočiti da su nakon priključenja OIE smanjeni gubici na mreži uz minimalna ulaganja, znatno popravljene naponske prilike i kvalitet napona i povećan stepen sigurnosti mreže.

6. LITERATURA

- "Zakon o električnoj energiji Republike Srpske" ("Službeni glasnik RS" br. 68/20)
- "Zakon o obnovljivim izvorima energije Republike Srpske" ("Službeni glasnik RS" br. 16/22)
- Neobnovljivi izvori energije https://hr.wikipedia.org/wiki/Neobnovljivi_izvori_energije
- Uloga obnovljivih izvora energije <https://www.industrija.rs/vesti/clanak/oie-energija-buducnosti>
- Istorija obnovljivih izvora energije [Istorija obnovljivih izvora energije - ecportal.me](https://ecportal.me/istorija-obnovljivih-izvora-energije-ecportal.me)
- Distribuirani energetski resursi https://en.wikipedia.org/wiki/Distributed_generation
- Podaci iz baze podataka elektrodistributivnog preduzeća MH "Elektroprivreda Republike Srpske" a.d. Trebinje.
- "Pravilnik o uslovima za priključenje elektrana na elektrodistributivnu mrežu Republike Srpske" MH "Elektroprivreda Republike Srpske" a.d. Trebinje, mart 2014. godine

4.3. Hosting capacity

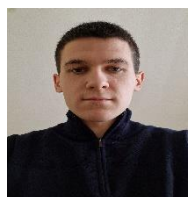
Hosting kapacitet se koristi kako bi se procenila sposobnost sistema da integriše nove OIE, kao što su solarni paneli, vetroelektrane i baterijske skladišne sisteme. Ovaj kapacitet predstavlja maksimalnu količinu dodatne energije koju sistem može prihvatiti bez da dođe do preopterećenja ili problema u kvalitetu napajanja.

Osnovna svrha hosting kapaciteta je procena koliko dodatne energije iz OIE može biti integrirano u postojeću mrežu bez skupe nadogradnje njene infrastrukture. Ovo je posebno važno u kontekstu tranzicije ka energetici s niskim emisijama CO₂ i povećane upotrebe OIE [12].

Na slici 4.11 je prikazano da se zahtevani uslovi kvaliteta napona mogu ispuniti samo na jednom mestu, i to u čvoru 15, koji je 0.4 kV naponskog nivoa. Moguće priključenje novog OIE je 614 kW zbog dodatnog uslova nemogućnosti injektiranja snage u 35 kV mrežu.

- Softverski alat PSS Sincal [Introduction - PSS@SINCAL - INDIELEC - Software CAD y CAE para ingenieria, ref-no-69-ps-c-pss-sincal-brochure-hires-intl-sept2018.pdf \(siemens.com\)](#)
- Softverski alat ETAP [ETAP | Schneider Electric Cpδnja \(se.com\)](#), [New Products for Engineers \(cfemedia.com\)](#), [Distribution System Software | Distribution Network Analysis | ADMS | ETAP](#)
- "Uticaj solarnih elektrana na elektroenergetski sistem BiH" – Parsons Brinckerhoff
- Hosting Capacity <https://www.nrel.gov/solar/market-research-analysis/advanced-hosting-capacity-analysis.html>
- G.Švenda, Z.Simendić: Adaptive On-Load Tap-Changing Voltage Control for Active Distribution Networks; Electrical Engineering, Vol. 104, April 2022, pp. 1041-1056; DOI: [10.1007/s00202-021-01357-8](https://doi.org/10.1007/s00202-021-01357-8)
- G.S.Švenda, I.Krstić, S.Kanjuh, M.Jajčanin, D.Vuletić: Volt Var Watt Optimization in Distribution Network with High Penetration of Renewable Energy Sources and Electric Vehicles; 2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Session 30: Industry Applications – Part II, Paper no. ID-21237, October 10-12, 2022, Novi Sad, Serbia; DOI: [10.1109/ISGT-Europe54678.2022.9960647](https://doi.org/10.1109/ISGT-Europe54678.2022.9960647)

Kratka biografija:



Nikola Sarkić je rođen u Doboju 1997. godine. Osnovne studije završio na FTN iz oblasti Elektrotehnike i Računarstvo – Elektroenergetski sistemi 2021. god. Iste godine upisao master studije na istom fakultetu.

POREĐENJE POKRIVENOSTI AKADEMSKIH BAZA COMPARISON OF COVERAGE OF ACADEMIC DATABASES

Aleksa Ivanić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PRIMENJENE RAČUNARSKE NAUKE I INFORMATIKA

Kratak sadržaj – Opis implementacije sistema za poređenje pokrivenosti akademskih baza „OpenAlex“ i „Scopus“ korišćenjem Java programskog jezika, Spring Boot radnog okvira, Elasticsearch pretraživača i React JavaScript biblioteke.

Ključne reči: *Akadske baze, poređenje, Elasticsearch.*

Abstract – Description of the implementation of the system for comparing the coverage of the academic databases "OpenAlex" and "Scopus" using Java programming language, Spring Boot framework, Elasticsearch search engine and the React JavaScript library.

Keywords: *Academic databases, comparing, Elasticsearch..*

1. UVOD

U ovom radu opisana je implementacija sistema za poređenje pokrivenosti akademskih baza koristeći Java programski jezik, Spring Boot radni okvir, Elasticsearch za skladištenje i pretragu naučnih radova i ReactJS biblioteke za implementaciju klijentske aplikacije.

Detaljno je opisan proces implementacije sistema za poređenje pokrivenosti akademskih baza. Opisana je konfiguracija Elasticsearch-a zatim korišćenje Elasticsearch-a zajedno sa Java programskim jezikom i Spring Boot radnim okvirom i na kraju sama implementacija indeksiranja naučnih radova i poređenje akademskih baza.

1.1. Elasticsearch

Elasticsearch pretraživač otvorenog koda (označava da je izvorni kod datog softvera dostupan i moguće ga je redistribuirati i modifikovati engl. *open-source*) i distribuiran je (softverski sistem u kojem komponente postavljene na povezane računare komuniciraju i koordiniraju svoje akcije slanjem poruka engl. *distributed*), a izgrađen je na vrhu Apache Lucene, biblioteke otvorenog koda, koja vam omogućava da implementirate funkciju pretraživanja u Java aplikaciji. Elasticsearch uzima ovu Lucene funkciju i proširuje je kako bi skladištenje, indeksiranje i pretraživanje učinili bržim, lakšim i, kao što ime kaže, elastičnim. Takođe, aplikacija ne mora da bude napisana u Javi da bi radila sa Elasticsearch-om;

NAPOMENA:

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

moguće je slati podatke preko HTTP-a u JSON formatu kako bi indeksirali, pretraživali i upravljali Elasticsearch klasterom.

U slučajevima kada je potrebno vraćanje relevantnih rezultata pretrage, dobavljanje statistike i sve to treba biti obaljeno brzo pretraživači poput Elasticsearch-a dolaze do izražaja jer su napravljeni da odgovore upravo na te izazove. Moguće je upariti pretraživač sa relacijom bazom podataka kako bi se kreirali indeksi i ubrzali SQL upiti. Ili, moguće je indeksirati podatke iz svog NoSQL skladišta podataka kako bi se omogućilo pretraživanje. Sve to moguće je postići sa Elasticsearch-om, a on dobro funkcioniše i sa dokument-orijentisanim bazama kao što je MongoDB jer su podaci u Elasticsearch-u takođe predstavljeni kao dokumenti. Moderni pretraživači kao što je Elasticsearch takođe su dobri u čuvanju podataka tako da ih je moguće koristiti kao NoSQL skladište podataka sa moćnim sposobnostima pretraživanja.

Elasticsearch je samo pretraživač i nikada neće biti korišćen samostalno. Kao i svako drugo skladište podataka, potreban je način kako da se u njega unesu podaci i verovatno je neophodno obezbediti interfejs za korisnike koji pretražuju te podatke.

Da bismo stekli predstavu o tome kako bi se Elasticsearch mogao uklopiti u veći sistem, razmotrimo tri tipična scenarija:

- *Elasticsearch kao primarni backend* (delovi aplikacije ili koda programa koji joj omogućavaju rad i kojima korisnik ne može pristupiti engl. *backend*) za veb-sajt — ukoliko na primer postoji veb-sajt koji omogućava ljudima da pišu objave na blogu, ali takođe želite mogućnost pretraživanja objava. Moguće je koristiti Elasticsearch za skladištenje svih podataka u vezi sa ovim objavama kao i postavljanje upita.

- *Dodavanje Elasticsearch-a postojećem sistemu*—U slučajevima kada već postoji sistem koji obrađuje podatke a potrebno je dodati pretragu, ovo je najčešći scenario korišćenja Elasticsearch-a.

- *Elasticsearch kao backend gotovog rešenja izgrađenog oko njega* — Budući da je Elasticsearch otvorenog koda i nudi jednostavan HTTP interfejs, veliki ekosistem ga podržava. Na primer, Elasticsearch je popularan za centralizovanje zapisa (zapis događaja ili radnji koje se dešavaju tokom izvršavanja programa engl. *log*); s obzirom na već dostupne alate koji mogu pisati i čitati sa Elasticsearch-a, osim konfigurisanja tih alata da rade onako kako želite, ne morate ništa dodatno da razvijate [1].

1.2. API

API-ji su mehanizmi koji omogućavaju da dve softverske komponente međusobno komuniciraju koristeći skup definicija i protokola. Na primer, softverski sistem meteorološkog zavoda sadrži dnevne podatke o vremenu. Aplikacija za vremensku prognozu na telefonu „razgovara“ sa ovim sistemom preko API-ja i prikazuje dnevne novosti o vremenu na telefonu.

API je skraćenica od engl. *Application Programming Interface*. U kontekstu API-ja, reč Aplikacija se odnosi na bilo koji softver sa jasnom funkcijom.

Interfejs se može posmatrati kao ugovor o usluzi između dve aplikacije. Ovaj ugovor definiše kako te dve aplikacije međusobno komuniciraju koristeći zahteve i odgovore. Njihova API dokumentacija sadrži informacije o tome kako programeri treba da strukturiraju te zahteve i odgovore.

API arhitektura se obično objašnjava terminima klijenta i servera. Aplikacija koja šalje zahtev naziva se klijent, a aplikacija koja šalje odgovor naziva se server. Dakle, u primeru vremenske prognoze, baza podataka meteorološkog zavoda je server, a mobilna aplikacija je klijent [2].

Postoje četiri različita načina na koje API-ji mogu da rade u zavisnosti od toga kada i zašto su kreirani:

- **SOAP API-ji** - koriste engl. *Simple Object Access Protocol*. Klijent i server razmenjuju poruke koristeći XML. Ovo je manje fleksibilan API koji je bio popularniji u prošlosti.
- **RPC API-ji** - engl. *Remote Procedure Calls*. Klijent kompletira funkciju (ili proceduru) na serveru, a server šalje izlaz nazad klijentu.
- **Vebsoket API-ji** - još jedan moderan razvoj veb API-ja koji koristi JSON objekte za prosleđivanje podataka. Vebsoket API podržava dvosmernu komunikaciju između klijentskih aplikacija i servera. Server može da šalje povratne poruke povezanim klijentima, što ga čini efikasnijim od REST API-ja.
- **REST API-ji** - ovo su najpopularniji i najfleksibilniji API-ji koji se danas nalaze na vebu. Klijent šalje zahteve serveru kao podatke. Server to koristi za pokretanje internih funkcija i vraća izlazne podatke nazad klijentu.

1.3. REST

REST je akronim za **REpresentational State Transfer** i arhitektonski stil za distribuirane hipermedijske sisteme. Roy Fielding ga je prvi put predstavio 2000. godine u svojoj čuvenoj disertaciji. Od tada je postao jedan od najčešće korišćenih pristupa za izgradnju veb-baziranih API-ja (Aplikacijski programski interfejsi).

REST nije protokol ili standard, to je arhitektonski stil. Tokom faze razvoja, API programeri mogu implementirati REST na različite načine. Kao i drugi arhitektonski stilovi, REST takođe ima svoje vodeće principe i ograničenja. Ovi principi moraju biti zadovoljeni kako bi servisni interfejs bio RESTful [3].

1.4. Pregled relevantne literature

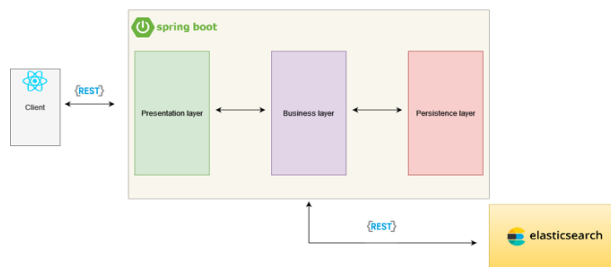
U radu [4] opisana je tehnika za merenje relativne veličine i preklapanja javnih veb pretraživača. Pretraživači su među najkorisnijim i najpopularnijim uslugama na Vebu. Korisnici su željni da saznaju kako se upoređuju. Koji ima najveću pokrivenost? Da li su indeksirali isti deo veba? Koliko stranica se tamo nalazi? Iako se o ovim pitanjima raspravljalo u popularnoj i u tehničkoj štampi, nije predložena objektivna metodologija evaluacije i nije se pojavilo dovoljno jasnih odgovora.

U ovom radu opisan je standardizovan, statistički način merenja pokrivenosti pretraživača i preklapanja putem nasumičnih upita. Ova tehnika ne zahteva privilegovan pristup nijednoj bazi podataka. Predstavljeni su rezultati obavljenih eksperimenata koji pokazuju procene veličine i preklapanja za „HotBot“, „AltaVista“, „Excite“ i „Infoseek“ kao i procenite njihove ukupne pokrivenosti sredinom 1997. i novembra 1997.

Ova metoda ne daje apsolutne vrednosti. Međutim, koristeći podatke iz drugih izvora, procenjeno je da od novembra 1997. godine broj stranica koje su indeksirali „HotBot“, „AltaVista“, „Excite“ i „Infoseek“ bio otprilike 77 miliona, 100 miliona, 32 miliona i 17 miliona, a zajednička ukupna pokrivenost iznosila je 160 miliona stranica. Dalje se pretpostavlja da je veličina statičnog, javnog Web-a od novembra bila preko 200 miliona stranica. Ono što je najviše iznenadilo jeste da je preklapanje veoma malo: manje od 1,4% od ukupne pokrivenosti, ili oko 2,2 miliona stranica indeksirano je od strane sva četiri pretraživača.

2. SPECIFIKACIJA SISTEMA ZA POREĐENJE POKRIVENOSTI AKADEMSKIH BAZA

U ovom poglavlju biće opisana arhitektura sistema za poređenje pokrivenosti akademski baza. Klijentska strana aplikacije implementirana je korišćenjem „React“ JavaScript biblioteke, dok je serverska aplikacija implementirana pomoću Java programskog jezika kao i Spring Boot radnog okvira. Spring Boot aplikacije zasnivaju se na slojevitoj arhitekturi (engl. *Layered Architecture*) gde hijerarhiski susedni slojevi međusobno komuniciraju, takva arhitektura zastupljena je i u backend-u ove aplikacije. Interakcija između Elasticsearch-a i backend-a aplikacije odvija se pomoću Spring Data Elasticsearch-a, koji u mnogome olakšava indeksiranje i pretragu prilikom korišćenja Elasticsearch-a. Komunikacija između klijentske i serverske aplikacije odvija se posredstvom API ulazne tačke uz poštovanje ograničenja REST-a.



Slika 1. Arhitektura sistema

3. POREĐENJE „OpenAlex“ I „Scopus“ BAZA NAUČNIH RADOVA

U ovom poglavlju biće opisan algoritam poređenja baza naučnih radova, koji se sastoji iz ciklusa. U svakom ciklusu će se nasumično birati određen broj dokumenata iz liste koja je dobijena pretraživanjem prethodno indeksiranih radova. Zatim je odabrane dokumente potrebno pronaći u bazi naučnih radova sa kojom poredimo trenutnu. Algoritam ciklusa će detaljnije biti opisan u nastavku:

- Odabir predefinisanoeg leksikona: potrebno je prvo pretražiti prethodno indeksirane radove kako bi se iz te inicijalne liste nasumično odabrali dokumenti za koje će se kasnije proveravati postojanje u drugoj bazi. Upit za pretragu sastavlja se od reči izabranih iz predefinisanoeg leksikona. Potrebno je izabrati jedan od tri postojeća leksikona: prvi je sastavljen od reči preuzetih iz ključnih reči naučnih radova, drugi sastavljen od reči iz apstrakta a treći od reči iz naslova.

- Pretraga naučnih radova: Nakon što je odabran leksikon sastavlja se upit nasumično birajući reči iz leksikona pa se zatim vrši pretraga radova odakle će se odabrati oni radovi čije će se postojanje proveravati u drugoj bazi. Vršiti se *boolean* pretraga gde je upit sastavljen disjunkciom prethodno odabranih reči. Za pretragu je korišćen `ElasticsearchRestTemplate` i `NativeQuery` koji nam omogućavaju maksimalnu fleksibilnost prilikom sastavljanja upita.

- Pretraga odabranog naučnog rada u drugoj bazi – Za svaki rad potrebno je proveriti da li postoji u drugoj bazi, pa je pre svega neophodno sastaviti upit koji jedinstveno oslikava dati naučni rad. U zavisnosti od odabira leksikona, u svrhu prezentaciju uzećemo da je odabran leksikon sa rečima iz apstrakta naučnih radova, sastavlja se upit od reči iz apstrakta datog rada tako što se proverava frekventnost svake reči prolazeći kroz leksikon. Na kraju se za upit uzimaju najmanje frekventne reči. Na taj način postizemo jedinstvenost upita za taj naučni rad.

- Algoritam za izračunavanje poklapanja radova – Nakon izvršene pretraga u drugoj bazi *Elasticsearch* može vratiti veliki broj rezultata pa će se stoga proveravati prvih 40 vraćenih radova. Kako bi rad zadovoljio kriterijume i smatrao se odgovarajućim neophodno je da po *default*-u zadovoljava sledeće kriterijume:

- Potpuno poklapanje identifikatora digitalnog objekta
- Poklapanje naslova 80%
- Poklapanje autora 40%

Korisniku je omogućeno da eksperimentiše i da menja ove kriterijume kao i da bira koji leksikon će se koristiti.

- Slanje podataka na klijentski deo aplikacije (engl. frontend) – nakon izvršenog kompleksnog

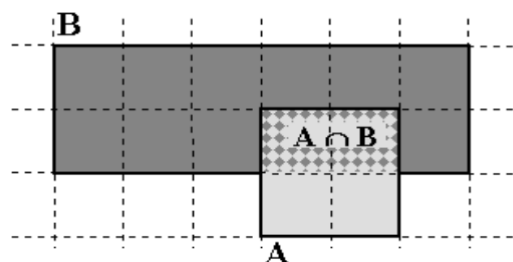
izračunavanja poklapanja radova, liste sa uzorkovanim naučnim radovima gde je svaki rad uparen sa istim takvim u suprotnoj bazi, bazi sa kojom se poredi, šalju se na frontend gde će se dalje vršiti detaljni prikaz i izračunavanje mera poklapanja dveju baza.

4. REZULTATI I DISKUSIJA

Sistem za izračunavanje poklapanja baza naučnih radova omogućava merenje relativne veličine i preklapanja baza. Za svaki par naučnih baza DB1 i DB2 moguće je izračunati:

- njihove relativne veličine:
 $Veličina(DB1):Veličina(DB2)$
- Deo DB1 baze koji je indeksiran u DB2, izražen kao procenat veličine baze DB1

Ideja koja stoji iza ovog pristupa je jednostavna: razmotrimo skupove A i B veličine 4 i 12 jedinica respektivno. Neka veličina njihovog preseka A i B bude 2 (slika 2). Pretpostavimo da ne znamo nijednu od ovih veličina, ali možemo jednolično uzorkovati iz bilo kog skupa i proveriti postojanje u bilo kom skupu. Ako uzorkujemo iz A i ustanovimo da je oko 1/2 uzoraka takođe u B. Otuda je veličina A otprilike 2 puta veća od preseka, A i B. Slično tome ćemo otkriti da je veličina B otprilike 6 puta veća od veličine A i B. Ovo će nas dovesti do zaključka da je B oko $6/2 = 3$ puta veći od A.



Slika 2. Izračunavanje odnosa veličine na osnovu procena preklapanja

U nastavku će biti prikazani estimacija preklapanja i relativnih veličina računato na sledeći način:

- Procena preklapanja: deo baze podataka DB1 indeksiran i u bazi DB2 procenjuje se na sledeći način:

Deo naučnih radova uzorkovanih iz DB1
pronađenih u DB2

- Poređenje veličine: Za baze podataka DB1 i DB2, odnos $Veličina(DB1)/Veličina(DB2)$ se procenjuje na sledeći način:

Deo naučnih radova uzorkovanih iz DB2 pronađenih u DB1
Deo naučnih radova uzorkovanih iz DB1 pronađenih u DB2

U nastavku će biti prikazani rezultati merenja na po 1000 uzorkovanih radova iz svake baze.

DB1 (izvor radova)	DB2 (baza sa kojom poredimo)	Procena preklapanja (radovi iz DB1 pronađeni u DB2)				Poređenje veličine (veličina DB1/veličina DB2)			
		1	2	3	4	1	2	3	4
OpenAlex	Scopus	42%	48%	43%	40%	1.21	1.17	1.26	1.27
Scopus	OpenAlex	51%	56%	54%	51%	0.82	0.86	0.80	0.78

Tabela 1. Rezultati merenja poređenja baza naučnih radova, korišćenjem leksikona sastavljenim od naslova radova

DB1 (izvor radova)	DB2 (baza sa kojom poredimo)	Procena preklapanja (radovi iz DB1 pronađeni u DB2)				Poređenje veličine (veličina DB1/veličina DB2)			
		1	2	3	4	1	2	3	4
OpenAlex	Scopus	38%	37%	38%	39%	1.18	1.13	1.13	1.15
Scopus	OpenAlex	45%	42%	43%	45%	0.84	0.88	0.88	0.87

Tabela 2. Rezultati merenja poređenja baza naučnih radova, korišćenjem leksikona sastavljenim od ključnih reči radova

DB1 (izvor radova)	DB2 (baza sa kojom poredimo)	Procena preklapanja (radovi iz DB1 pronađeni u DB2)				Poređenje veličine (veličina DB1/veličina DB2)			
		1	2	3	4	1	2	3	4
OpenAlex	Scopus	41%	39%	41%	42%	1.17	1.15	1.12	1.14
Scopus	OpenAlex	48%	45%	46%	48%	0.85	0.87	0.89	0.87

Tabela 3. Rezultati merenja poređenja baza naučnih radova, korišćenjem leksikona sastavljenim od apstrakta radova

5. ZAKLJUČAK

U ovom radu opisan je sistem za poređenje pokrivenosti akademskih baza „OpenAlex“ i „Scopus“ korišćenjem Java programskog jezika, Spring Boot radnog okvira, Elasticsearch pretraživača i React JavaScript biblioteke.

Na osnovu rezultata dobijenih korišćenjem opisanog sistema može se zaključiti da baza „OpenAlex“ indeksira veći broj radova. Korisniku je data mogućnost da konfiguriše proces poređenja ali kao što je prethodno prikazano u svim slučajevima poređenja dobija se približno ista razlika u pokrivenosti u korist „OpenAlex“ baze.

Pored poređenja pokrivenosti baza sistem pruža korisniku složen uvid u uzorkovane dokumente koji su korišćeni u procesu poređenja tako da je moguće detaljnije analizirati rezultate. Takođe je omogućena pretraga indeksiranih naučnih radova iz „OpenAlex“ i „Scopus“ baza.

5.1. Dalji razvoj sistema

Ovaj sistem sigurno ima veliki potencijal za dalji razvoj, jedan od načina kako se može unaprediti jeste da se omogući jednostavno dodavanje novih akademskih baza za poređenje što bi zahtevalo izmene u samoj arhitekturi, modelu i implementaciji sistema ali bi uloženo vreme u razvoj ove funkcionalnosti sigurno rezultiralo u poboljšanju sistema. Takođe, pretpostavka je i da bi veštačka inteligencija mogla poboljšati algoritam za sastavljanje upita koji najbolje opisuju uzorkovane radove čije postojanje je potrebno proveriti u drugoj bazi.

6. LITERATURA

- [1] Radu Gheorghe, Matthew Lee Hinman, and Roy Russo, „Elasticsearch in Action“, Novembar 2015.
- [2] What is an API (Application Programming Interface)? <https://aws.amazon.com/what-is/api/> (pristupljeno u aprilu 2024.)
- [3] What is REST <https://restfulapi.net/> (pristupljeno u aprilu 2024.)
- [4] Krishna Bharat and Andrei Broder, „A technique for measuring the relative size and overlap of public Web search engines“, 1998.

Kratka biografija:



Aleksa Ivanić rođen 7.8.1998. godine u Zrenjaninu. Smer računarstvo i automatika na Fakultetu tehničkih nauka u Novom Sadu upisao je 2017. godine. Osnovne studije završio je 2021. godine i iste godine upisao master akademske studije. kontakt: ivanicaaleksa98@gmail.com

MiniOptimizer: ПОЈЕДНОСТАВЉЕНИ РЕЛАЦИОНИ ОПТИМИЗАТОР УПИТА**MiniOptimizer: A SIMPLIFIED RELATIONAL QUERY OPTIMIZER**Милош Гравара, *Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

Кратак садржај – У овом раду је представљен *MiniOptimizer*, поједностављени релациони оптимизатор упита, са циљем да практично прикаже теоријске основе оптимизације упита. *MiniOptimizer*, развијен у програмском језику C#, подржава основне SQL операције и користи *MiniQL*, специјално дизајнирани упитни језик. Рад обухвата структуру алата *MiniOptimizer*, укључујући парсер, логички и физички план извршења, проценитељ кардиналности и каталог података. Процес оптимизације укључује парсирање *MiniQL* упита, креирање и трансформацију логичког плана, оптимизацију редоследа спајања и избор најповољнијег физичког плана извршења. Један од доприноса овог рада јесте разумевање техника оптимизације упита, кроз развој *MiniOptimizer*-а као едукативног алата који може да служи као основа за даља истраживања и развој напреднијих система за управљање базама података.

Кључне речи: Релациони оптимизатор упита, SQL, процена кардиналности, системи за управљање базама података

Abstract – This paper presents *MiniOptimizer*, a simplified relational query optimizer, with the aim of practically demonstrating the theoretical foundations of query optimization. Developed in the C# programming language, *MiniOptimizer* supports basic SQL operations and uses *MiniQL*, a specially designed query language. The paper covers the structure of the *MiniOptimizer* tool, including the parser, logical and physical execution plan, cardinality estimator, and data catalog. The optimization process includes parsing *MiniQL* queries, creating and transforming a logical plan, optimizing the join order, and choosing the most favorable physical execution plan. One of the contributions of this work is enhancing the understanding of query optimization techniques through the development of *MiniOptimizer* as an educational tool, which can serve as a foundation for further research and the development of more advanced database management systems.

Keywords: Relational query optimizer, SQL, cardinality estimation, database management systems

1. УВОД

Током седамдесетих година прошлог века, дошло је до значајног искорака у свету управљања над подацима развојем декларативних система. Претходно, процес претраживања и преузимања података претежно је био вођен процедуралним системима, који су захтевали од корисника да пажљиво специфицирају сваки корак у процесу добијања потребних података. Овакав приступ се често показао као непрактичан и неефикасан, што је представљало значајне изазове у руковању сложеним скуповима података. Са друге стране, декларативни системи омогућили су корисницима да, постављањем упита, специфицирају само шта желе да постигну, без детаљног описа како то да постигну. Овако постављени, декларативни системи су понудили интуитивнији и ефикаснији начин интеракције са базама података, превазилазећи ограничења својих процедуралних претходника, и омогућили корисницима да се фокусирају на пословну логику уместо на имплементационе детаље.

Ипак, како би се могле развијати апликације које користе велике количине података са циљем опслуживања корисничких захтева, декларативност није довољна, већ је неопходно да системи за управљање базама података (СУБП) буду ефикасни, ради смањења времена чекања на одговор захтева и пружања пријатног корисничког искуства. С тим на уму, оптимизација перформанси базе података постала је централна активност која омогућава кратко време одзива на корисничке захтеве за подацима. Главну улогу у овом процесу оптимизације игра оптимизатор упита, компонента задужена за анализирање постављених упита и одређивање оптималног начина њиховог извршавања. Узимајући у обзир различите факторе као што су структура упита, шема базе података и системски ресурси, оптимизатор упита смишља план извршења, којим употребу системских ресурса своди на минимум, истовремено пружајући најбоље перформансе.

Циљ овог рада јесте да се, на практичном примеру имплементације оптимизатора упита, пружи детаљан увид у улоге и механизме процеса оптимизације упита у релационим СУБП. Како би се то постигло, креиран је *MiniOptimizer*, алат написан у програмском језику C# који илуструје основне концепте оптимизације упита у релационим системима за управљање базама података. Пажња је усмерена на логички аспект оптимизације - генерисање ефикасних планова извршења применом техника оптимизације засноване на правилима и оптимизације засноване на трошку са

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Владимир Димитриески, ванр. проф.

избором оптималног редоследа спајања коришћењем алгорита динамичког програмирања.

2. АНАТОМИЈА ИЗВРШЕЊА УПИТА

Улазна тачка сваког упита представљеног СУБП-у је парсер. Задатак парсера је да, најпре, изврши лексичку и синтаксну анализу упита, како би проверио да упит подлеже правилима граматике упитног језика, а затим да га трансформише у апстрактно синтаксно стабло које представља хијерархијску структуру упита.

Након тога, на основу апстрактног синтаксног стабла се формира иницијални план упита, типично представљен усмереним ацикличним графом оператора релационе алгебре. Граф креиран на основу оператора релационе алгебре се назива логички план упита. Овакав логички план се, применом одређеног скупа правила, трансформише у еквивалентне логичке планове за које се очекује да ће захтевати мање времена за извршење [1].

Међутим, логички план је апстрахован од детаља физичког приступа подацима, као и алгоритама којима се реализују релациони оператори. Из тог разлога неопходно га је превести у физички план упита који СУБП може да изврши [1]. Физички план извршења упита дефинише избор алгоритама релационих оператора на нивоу приступа подацима и одређује редослед њиховог извршења. Поред тога, физички план садржи детаље о томе како треба да се приступа релацијама базе података, као и о томе да ли и када их је потребно сортирати. Као и логички план извршења, представљен је као усмерени ациклични граф.

У оквиру избора оптималног логичког и физичког плана, оптимизатор упита мора да донесе неколико одлука. Најпре, неопходно је одредити које алгебарски еквивалентне форме упита доводе до најефикаснијег избора алгоритама извршења упита [1]. Затим, за сваки релациони оператор потребно је изабрати најповољнији алгоритам имплементације. На крају, неопходно је донети одлуку о томе на који начин оператори треба да прослеђују податке једни другима, у виду проточне обраде, употребом бафера радне меморије или путем диска.

Након што се одабере најповољнији физички план упита, СУБП га извршава и враћа резултате кориснику који је упит специфицирао.

3. ПРЕГЛЕД СТАЊА У ОБЛАСТИ

Историја релационих оптимизатора упита обухвата развој неколико кључних система који су значајно унапредили технике оптимизације и поставили основе за савремене алате.

Селинцеров *R* оптимизатор [2], развијен у оквиру Систем *R* пројекта од стране *ИБМ*-а током 1970-их, уводи динамичко програмирање за генерисање оптималног плана извршења упита користећи статистичке информације о подацима.

Volcano оптимизатор [3], развијен крајем 1980-их од стране Геца Грефеа, користи генерализовани приступ оптимизацији, укључујући стратегију сепарације и евалуације (енгл. *branch and bound*) и подршку за паралелизам и процену трошкова на више нивоа.

Балса оптимизатор [4], заснован на моделу подржаног учења (енгл. *reinforcement learning*), представља

значајан помак од традиционалних метода. Балса континуирано ажурира своју базу знања из стварних повратних информација о извршењу упита и на тај начин се ефикасно прилагођава променама у дистрибуцији података и различитим шаблонима упита, стварајући тако ефикасније планове извршења.

4. ИМПЛЕМЕНТАЦИЈА АЛАТА *MiniOptimizer*

MiniOptimizer је поједностављени релациони оптимизатор упита чија је основна намена да илуструје основне теоријске концепте који стоје иза процеса оптимизације упита. Алат је написан у програмском језику *C#*, који је изабран због својих напредних механизма апстракције, што омогућава лакше моделовање концепата кључних за правилно функционисање оптимизатора упита, који је инхерентно логички оријентисан.

MiniOptimizer подржава основне *SQL* операције као што су филтрирање, спајање, пројекција и дистинкција. Ови конструкти су представљени путем *MiniQL* упитног језика развијеног за потребе овог пројекта.

Због сложености пројекта, *MiniOptimizer* не укључује компоненту СУБП-а која имплементира физичке операторе и директно извршава упите. Уместо тога, пажња је усмерена на логичком аспекту оптимизације, где се истражују различите стратегије и технике за генерисање ефикасних планова извршења упита.

4.1. Архитектура алата *MiniOptimizer*

Архитектура *MiniOptimizer* алата обухвата неколико компоненти које омогућавају избор најефикаснијих планова извршења упита и приказана је на слици 4.1.

Након што корисник представи упит *MiniOptimizer*-у, прва станица на коју стиже је парсер упита. Задатак парсера је да, најпре, изврши лексичку и синтаксну анализу, затим користећи каталог података да провери семантичку валидност упита и, на крају, да представљени упит написан у *MiniQL* језику претвори у иницијални логички план.

Иницијални логички план се прослеђује оптимизатору заснованом на правилима, који примењује скуп алгебарских закона чиме се логички план трансформише у ефикаснији план и затим прослеђује компоненти за оптимизацију операција спајања.

Компонента за оптимизацију операција спајања, уз употребу модела трошка, бира најповољнији редослед извршења операција спајања и такав преуређени логички план прослеђује оптимизатору заснованом на трошку.

Оптимизатор заснован на трошку, такође уз употребу модела трошка, који ће бити детаљно објашњен у поглављу 4.5, генерише оптималан физички план који у себи садржи детаље о начину приступа подацима, редоследу извршења операција и конкретним алгоритмима имплементације релационих оператора. Овако конструисан физички план се враћа кориснику који је алату представио *MiniQL* упит.

4.1. Каталог података

Каталог података у *MiniOptimizer* алату је централно складиште метаподатака о бази података.



Слика 4.1 – Архитектура алата *MiniOptimizer*

Он садржи податке о табелама, колонама, индексима и статистикама. Ови подаци укључују структуру табела, типове података колона, доступне индексе и статистичке податке као што су број редова и дистрибуција вредности.

4.2. Парсер и упитни језик *MiniQL*

Парсер *MiniOptimizer*-а развијен је употребом алата *ANTLR4* и његов основни задатак је анализа и обрада улазних упита писаних у *MiniQL* језику. Лексичка и синтаксна анализа се врши у оквиру алата *ANTLR4* спрам граматике *MiniQL* језика представљене на листингу 4.1.

```

query ::= SELECT [DISTINCT] attrList FROM relList [WHERE condition]
attrList ::= attribute [,attribute]*
relList ::= relation [,relation]*
condition ::= condition AND condition | attribute EQ attribute
attribute ::= identifier | constant
relation ::= identifier
  
```

Листинг 4.1 – Део граматике *MiniQL* упитног језика

Са друге стране, семантичка анализа се врши уз употребу каталога података и основни задатак овог корака је да се обезбеди да све табеле и колоне специфициране у упиту постоје у каталогу података, чиме се умањује потреба за провером на грешке у даљим корацима оптимизације упита.

Након што упит прође семантичку анализу, парсер генерише иницијални логички план рекурзивним обиласком апстрактног синтаксног стабла које је генерисао алат *ANTLR4*.

4.3. Логички план

Логички план представља алгебарски израз који описује начин извршења упита над подацима. У *MiniOptimizer*-у, представља структуру усмереног ацикличног графа моделовану кроз скуп међусобно повезаних логичких чворова, где сваки чвор поседује листу потомака и референцу на родитељски чвор.

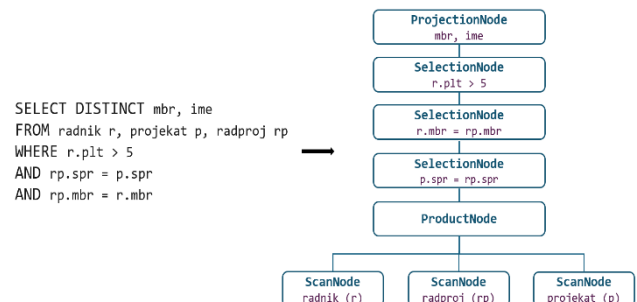
4.5. Модел трошка

Основни циљ модела трошка у *MiniOptimizer* алату је да пружи брзе и прецизне процене кардиналности и процене трошка извршења одређених оператора без њиховог физичког извршавања. Ова компонента садржи два основна задатка: Процена кардиналности и процена трошка. Оба поступка се заснивају на статистичким подацима садржаним у каталогу. Значај овакве структуре лежи у могућности за ефикасно премештање чворова са једног дела графа на други, чиме се подржавају хеуристике за побољшање логичких планова. Поред тога, логички

чворови поседују и информације о кардиналности резултата оператора које представљају, што је значајно током процена кардиналности међурелација приликом оптимизације редоследа операције спајања.

4.4. Оптимизатор заснован на правилима

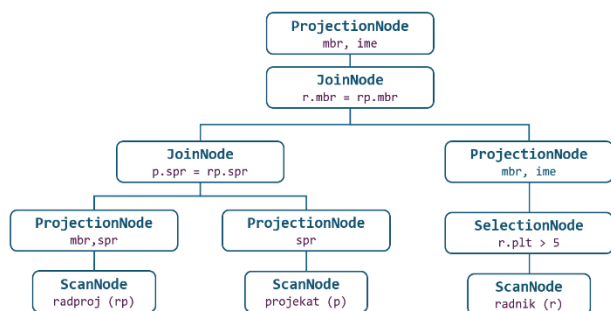
Иницијални логички план се састоји од релационог оператора пројекције који је настао на основу *SELECT* клаузуле и који као потомке садржи ланац оператора селекције који се формирају уколико је *MiniQL* упит садржао *WHERE* клаузулу. Ланац оператора селекције као потомка садржи оператор декартовог производа који се односи на све релације поменуте након *FROM* клаузуле упита. Пример иницијалног логичког плана приказан је на слици 4.2.



Слика 4.2 – Пример иницијалног логичког плана

Задатак оптимизатора заснованог на правилима је да трансформише иницијални логички план у еквивалентан али ефикаснији облик. Ово постиже применом неколико правила у одређеном редоследу. Најпре се врши креирање операција спајања где се за сваке две релације креира одговарајући релациони оператор споја, чиме се замењује мање ефикасни декартов производ. Затим се врши пропација селекције, при чему се оператори селекције померају ближе изворним релацијама. На крају, долази до креирања операција дистинкције и репликације оператора пројекције како би се смањио број података који пролази кроз систем. Резултат оптимизације засноване на правилима је ефикаснији логички план извршења који је спреман за фазу уређивања редоследа спајања. Слика 4.3 приказује логички план након оптимизације засноване на правилима.

Под проценом кардиналности се сматра поступак процене величине резултујуће релације након примене одређеног оператора релационе алгебре. Поступак процене кардиналности је неопходан приликом одређивања оптималног редоследа спајања.



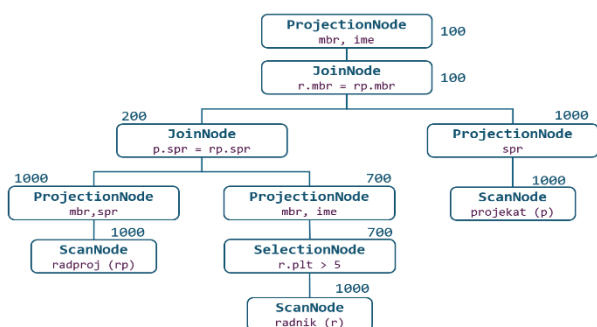
Слика 4.3 – Оптимизован логички план

Са друге стране, процена трошка представља поступак процене количине улазно-излазних операција ка диск јединици СУБП-а. Поступак процене трошка је неопходан приликом избора најповољнијих физичких оператора.

4.6. Оптимизатор операција спајања

Задатак оптимизатора операција спајања је да пронађе оптималан редослед спајања релација у упиту. Проблем проналазак оптималног редоследа у *MiniOptimizer* алату је решен применом алгоритма динамичког програмирања.

Алгоритам динамичког програмирања се састоји из неколико корака. Први корак представља иницијализацију табеле динамичког програмирања где се за сваке две релације процени кардиналност њиховог спајања. Затим се, итеративно, за сваки скуп релација који садржи више од две релације проналази њихов оптималан редослед спајања тако што се бира онај редослед чија процена даје најмању кардиналност. Овај процес се врши док се не дође до величине плана који одговара величини улазног плана, када се долази до оптималног редоследа. Слика 4.4 приказује оптимизован логички план са слике 4.3 уз процењене кардиналности.

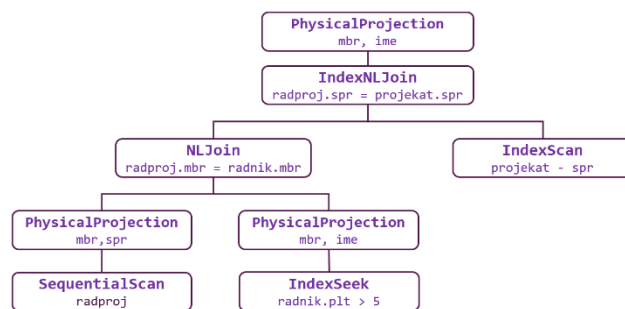


Слика 4.4 – Оптимизован редослед операција спајања

4.7. Оптимизатор заснован на трошку

Задатак оптимизатора заснованог на трошку је да за сваки релациони оператор, изабере најповољнији алгоритам његове имплементације. Поред тога, за сваку изворну релацију специфицирану у логичком плану, потребно је да изабере најповољнији приступни пут. На основу процене трошка, постојања индексних структура и структуре логичког плана, оптимизатор, најпре бира, најповољнији приступни пут, који може бити секвенцијално или индексно скенирање, а затим бира најповољнији алгоритам имплементације операција спајања, који може бити угњеждено спајање, индексно спајање или хеш спајање. Овим кораком се комплетира физички план и

враћа кориснику. Пример физичког плана приказан је на слици 4.5.



Слика 4.5 – Пример физичког плана

5. ЗАКЉУЧАК

MiniOptimizer алат је способан да генерише ефикасне планове извршења упита користећи основне поступке оптимизације упита попут скупа алгебарских закона за побољшање логичких планова, алгоритма динамичког програмирања за одређивање оптималног редоследа операција спајања и модела трошка за избор најповољнијих физичких оператора. Основна намена је да представи практичну имплементацију теоријских концепата које стоје иза поступка оптимизације упита и самим тим представља едукативни алат који служи као основа за даља истраживања и развој напреднијих система за управљање базама података.

Иако пружа солидну основу за оптимизацију упита, постоји низ праваца за даљи развој и унапређење алата. У будућим истраживањима, планира се додавање подршке за угњеждене упите, скуповне операције и ДМЛ (енгл. *Data Modification Language*) наредбе, што ће значајно проширити функционалности алата. Такође, интеграција извршиоца упита би омогућила валидацију перформанси алата и пружила детаљнију анализу ефикасности генерисаних планова извршења

6. ЛИТЕРАТУРА

- [1] Hector Garcia-Molina, Jeffrey D. Ullman. Database Systems: The Complete Book. Prentice Hall Press, USA, 2 edition, 2008.
- [2] Graefe, Goetz. "The Volcano optimizer generator: extensibility and efficient search." Proceedings of IEEE 9th International Conference on Data Engineering (1993): 209-218.
- [3] Selinger, Patricia G. et al. "Access path selection in a relational database management system." ACM SIGMOD Conference (1979).
- [4] Zongheng Yang et al. Balsa: Learning a query optimizer without expert demonstrations. In Proceedings of the 2022 International Conference on Management of Data, SIGMOD/PODS '22. ACM, June 2022.

Кратка биографија:



Милош Гавара рођен је 20. јануара 2000. године у Новом Саду. Завршио је Гимназију „Исидора Секулић“ 2019. године након чега уписује Факултет техничких наука, смер Рачунарство и аутоматика. Дипломски рад је одбрањено 2023. године када уписује мастер студије на студијском програму Рачунарство и аутоматика.

ФОРЕНЗИКА macOS-a**MacOS FORENSICS***Душан Миљковић, Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

Кратак садржај – Дигитална форензика представља важну област информатике, а један од најважнијих аспеката дигиталне форензике односи се на оперативне системе који представљају срце сваког рачунарског уређаја. У овом раду обрађена је тема дигиталне форензике macOS-a кроз анализу самог оперативног система, као и метода и алата за форензику оперативног система macOS. На крају је одрађена и студија случаја која приказује како се неке од метода и алата могу користити за дигиталну форензику.

Кључне речи: дигитална форензика, macOS

Abstract – Digital forensics represents an important field of computer science, and one of the most crucial aspects of digital forensics relates to operating systems, which form the heart of every computing device. This paper addresses the topic of macOS digital forensics by analyzing the operating system itself, as well as the methods and tools for macOS forensics. Finally, a case study is presented, demonstrating how some of these methods and tools can be used for digital forensics.

Keywords: digital forensics, macOS

1. УВОД

Дигитална форензика представља важну област информатике која се бави проналажењем, анализом и интерпретацијом дигиталних података како би се расветлили различити облици кривичних дела и инцидентата. У савременом друштву, где се дигитални уређаји и мреже налазе на сваком кораку и њихов број је све већи и већи, дигитална форензика игра неизоставну улогу у процесу разоткривања кривичних дела и осигуравању доказа неопходних за судске поступке.

У овом раду обрађена је тема дигиталне форензике MacOS-a. MacOS се истиче високим стандардима сигурности и безбедности, са нагласком на заштиту корисничких података и интегритета система.

Први део овог рада детаљније објашњава шта је дигитална форензика који су то принципи дигиталне форензике, процеси и области.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији је ментор био др Стеван Гостојић, ред. проф.

Након тога, описан је macOS, његова историја, архитектура, фајл систем и безбедносни механизми.

Затим следи поглавље о техникама за форензику macOS-a, са освртом на битне артефакте, њихово прикупљање и анализу. После тога следи и поглавље о алатима за форензику macOS-a.

За сам крај одрађена је студија случаја која приказује како се неке од претходно поменутих метода и алата могу користити у дигиталној форензици.

2. ДИГИТАЛНА ФОРЕНЗИКА

Дигитална форензика, која се понекад назива и компјутерска форензика, представља научну дисциплину чији предмет су идентификација, прикупљање, чување, прегледање, анализа и презентација дигиталних доказа коришћењем научно и правно ваљаних метода и алата [1].

2.1. Историја дигиталне форензике

Успон дигиталне форензике је започео осамдесетих година, када је FBI покренуо први званични програм за магнетне медије, а значајни напредак је постигнут током деведесетих и две хиљадитих због потребе за борбом против дечије порнографије и анализе дигиталних уређаја у војним операцијама. Прекретница у професионализацији дигиталне форензике догодила се 2006. године са увођењем обавезног режима за електронско откривање у Правилима о парничном поступку САД [2].

2.2. Примена дигиталне форензике

Дигитална форензика постала је изузетно значајна због савременог развоја информационих и комуникационих технологија које су довеле до пораста дигиталног криминала и до пораста безбедносних инцидентата.

Примена дигиталне форензике је веома велика, од решавања сајбер криминала, преко правосуђа па све до рекламација и различитих спорова [1].

2.3. Области дигиталне форензике

Дигитална форензика дели се на неколико грана (подобласти), у зависности од уређаја, система који преноси дигиталне доказе или од самог дигиталног доказа. Тако постоји дигитална форензика рачунара, форензика рачунарских мрежа, форензика мобилних уређаја, форензика мултимедијалних записа итд, али с обзиром на то да се информационе технологије јако брзо мењају, релативно брзо настају и нове области

дигиталне форензике као што су дигитална форензика уграђених система (енг. embedded systems), форензика IoT, форензика облака (енг. cloud) итд. [1].

2.4. Принципи дигиталне форензике

Како би се осигурала тачност, ефикасност и прихватљивост доказа, дигитални форензичар мора испоштовати основне принципе који се примењују у свим областима дигиталне форензике, а то су: користити научно прихваћене методе и алате, стриктно пратити протоколе, обезбедити ланац доказа, документовати процес и не мењати ништа на изворном медијуму [1].

2.5. Нивои апстракције

Докази се могу посматрати на различитим нивоима апстракције, за шта се користе различите технике и алати. Рад на најнижем нивоу апстракције је пожељан јер пружа највише информација, али може захтевати више времена, па се понекад праве изузеци када је потребна брза анализа [3].

2.6. Нивои волатилности (несталности) доказа

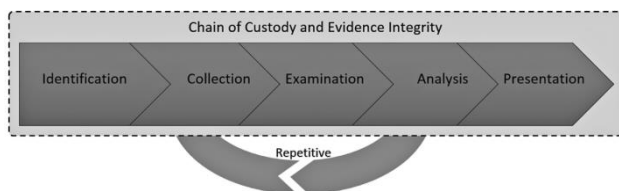
Волатилни (нестални) подаци односе се на информације смештене у привременој меморији система, као што је RAM или физичка меморија, и у активним процесима или сервисима. Ови подаци се карактеришу њиховом привременом природом; постоје само док је систем укључен и могу бити изгубљени или измењени при гашењу или поновном покретању система [1].

2.7. Процеси дигиталне форензике

Дигитални докази могу се наћи на различитим дигиталним уређајима и системима. Примена форензичког процеса осигурава исправност истраге кроз два основна принципа:

- **интегритет доказа**, који подразумева очување доказа у његовом изворном облику, и
- **ланац доказа**, који захтева документовање свих акција учињених на доказима како би се доказала њихова аутентичност и интегритет.

Цео истражни процес може се поделити у пет итеративних фаза: идентификација, прикупљање, прегледање, анализа и презентација доказа [1].



Слика 1. Процеси дигиталне форензике [1]

2.8. Дигитална форензика оперативних система

Форензика оперативног система представља процес прикупљања корисних информација из оперативног система рачунара или мобилног уређаја са циљем стицања емпиријских доказа против починиоца. Испитивање конфигурационих фајлова може помоћи у утврђивању догађаја на рачунару, за шта је потребно

разумевање оперативног система и његовог фајл система [4].

3. MacOS

Почеци MacOS-а датирају још од 1984. године, када је Apple представио Macintosh, први комерцијални рачунар са графичким корисничким интерфејсом и мишем. Ово је значајно олакшало коришћење машина и учинило их доступнијима за људе који нису толико заинтересовани за технологију. Apple је ову верзију назвао системски софтвер, или једноставно систем [5].

3.1. Класични MacOS

Софтвер се састојао од два фајла видљива корисницима: системског фајла и Finder-а који су се налазили у фолдеру са ознаком „системски фолдер“ (енг. system folder). Како ране верзије MacOS-а нису имале специфична имена, људи су почели да користе верзије системских фајлова како би разликовали различите верзије оперативног система, па је првобитна верзија позната и као систем 1. Прве верзије MacOS-а познате су и под називом „класични“ MacOS, а објављено је девет главних верзија класичног MacOS-а [6].

3.2. MacOS X

MacOS X, заснован на Unix архитектури, представља наследника класичног MacOS-а и представљен је 2001. године. Овај оперативни систем донео је велику стабилност, сигурност и нове функције у свет Macintosh рачунара. Са својим Aqua интерфејсом, систем је понудио естетски привлачан и интуитиван кориснички доживљај. Међу значајним карактеристикама су Spotlight за претрагу, Time Machine за бекап, и интеграција са iCloud-ом за синхронизацију података. Од 2012. године назив оперативног система је промењен у OS X, а од 2016. године, како би ускладио брендирање са брендирањем других примарних оперативних система компаније, Apple мења назив оперативног система у macOS [6].

3.3. MacOS (од 2020. године)

Од 2020. године, Apple је започео нову еру за MacBook рачунаре, увећавајући главну верзију оперативног система са 10 на 11 и доневши први редизајн корисничког интерфејса након 6 година. Велика измена је била прелазак на Apple Silicon чипове, што је донело знатна убрзања у раду и ефикасности система.

3.4. Архитектура macOS-а

MacOS направљен је за Macintosh рачунаре и изграђен је на основама базираним на UNIX-у. Архитектура оперативног система укључује неколико кључних компоненти.

Језгро macOS-а чини XNU кернел, он представља хибридни кернел чија су три основна дела Mach кернел развијен на Карнеги Мелон универзитету (енг. Carnegie Mellon University), BSD и C++ API за писање драјвера назван I/O-Kit.

APFS представља основни фајл систем за све оперативне системе унутар Apple-а, macOS-а, iOS, watchOS и tvOS. Од 2017. године он замењује HFS+ као

основни фајл систем, док је коначна верзија APFS-а објављена 2018. године. Неке од најзначајнијих његових карактеристика:

- Наносекундска временска ознака
- Делење простора
- Скоро тренутно копирање података
- Snapshots (read-only инстанце фајл система на волумену)
- Шифровање

Многи артефакти од интереса на macOS систему чувају се као property list или .plist фајлови. Постоје два типа property list-ова: обични текстуални, тј. XML plist и бинарни plist фајлови. Обични текстуални plist фајлови могу се директно прегледати или видети у било ком програму за приказ XML-а, док се бинарни морају претворити у обични текст пре анализе. Због тога што су компактнији од својих еквивалентних бинарних property list фајлови постају све чешће коришћени [7].

3.5. Безбедносни механизми

Apple дизајнира безбедност у сржи својих платформи. Сваки Apple уређај комбинује хардвер, софтвер и сервисе дизајниране да раде заједно како би постигли максималну безбедност и транспарентно корисничко искуство са циљем да личне информације остану сигурне.

Са првим издањем Apple Silicon чипа, значајно је унапређена безбедност и приватност корисника, уводећи функционалности као што су безбедно покретање, AES хардверски механизам за шифровање и Secure Enclave. Ови чипови осигуравају безбедно покретање уређаја од Boot ROM нивоа и имају Secure Enclave који штити осетљиве корисничке податке, укључујући и биометријске информације.

Механизам ажурирања софтвера и системска заштита интегритета (SIP) додатно доприносе повећању безбедности, спречавајући неовлашћени приступ и манипулацију системским компонентама, што додатно штити корисничке податке и оперативни систем [7].

4. ТЕХНИКЕ ЗА ФОРЕНЗИКУ MAC OS-A

Технике дигиталне форензике помажу да се провери неалоцирани простор на диску као и скривени фолдери за копије шифрованих, оштећених или обрисаних фајлова. Неке од најчешћих техника за форензику су: imaging, live forensics, претрага по кључним речима, враћање података, стегоанализа, анализа временске линије и друге [8].

4.1. Imaging

Приликом прављења дигиталне копије оригиналног диска (енг. imaging) macOS-а, најпре је потребно утврдити о ком моделу рачунара се ради. Уколико се ради о новијим верзијама система које користе Apple M серију чипова, неопходно је познавање корисничке шифре јер у супротном није могуће приступити подацима [8].

4.2. Анализа Apple метаподатака

У домену дигиталне форензике, метаподаци играју кључну улогу у откривању информација о фајловима, фолдерима и активностима које их окружују. Метаподаци, у суштини подаци о подацима, пружају драгоцене увиде у атрибуте фајлова, временске ознаке, порекло и интеракције корисника. У оквиру macOS екосистема, Apple метаподаци нуде богат извор форензичких доказа, а команда mdls служи као моћан алат за издвајање и анализу ових метаподатака [9].

4.3. Анализа .plist фајлова

Анализа .plist фајлова у дигиталној форензици пружа драгоцене увиде у активности корисника, обрасце коришћења апликација, конфигурације система и потенцијално злонамерне активности. Форензички истражитељи екстрахују и анализирају .plist фајлове помоћу различитих алата, откривајући корисничка подешавања, мрежна повезивања и конфигурације апликација. Метаподаци и временске ознаке у .plist фајловима помажу у успостављању временских линија активности корисника и системских догађаја. Форензичка анализа ових фајлова може наићи на изазове као што су рад са бинарним фајловима и идентификација релевантних података међу многобројним .plist фајловима [10].

4.4. Остали артефакти значајни за дигиталну форензику

Поред .plist фајлова, Apple има мноштво других артефаката који могу помоћи у бољем разумевању тока података. Knowledge C база података складишти разне типове података о интеракцијама и активностима корисника. Bash/Zsh историја садржи командну историју и информације о сесијама. FSEvents у корену партиције садржи локове и информације о фајл системским догађајима. QuarantinedEventsV2 база података прати и управља информацијама о фајловима преузетим са интернета и стављеним у карантин.

5. АЛАТИ ЗА ФОРЕНЗИКУ MAC OS-a

За дигиталну форензику је кључно да алати производе поуздане доказе који испуњавају стандарде прихватљивости на суду, захтевајући поновљивост и репродуктивност резултата тестова. Форензички алати морају бити тестирани по NIST стандардима и прихваћени на суду како би резултати били валидни. Постоји велики број комерцијалних и алата отвореног кода који се користе у дигиталној форензици [8].

5.1. Комерцијални алати

Apple редовно ажурира свој оперативни систем, унапређујући безбедност, што форензичарима отежава прикупљање података са MacBook рачунара и захтева стално ажурирање алата. Комерцијални форензички алати нуде могућности као што су блокирање уписивања, клонирање дискова, опоравак скривених и обрисаних фајлова, удаљено и live прикупљање доказа, анализа RAM-а, напредно претраживање и филтрирање података и метаподатака, као и аутоматско генерисање извештаја. Најпознатији алати укључују EnCase, BlackBag MacQquisition, Magnet AXIOM, Sumuri Recon LAB и Sumuri Recon ITR.

5.2. Алати отвореног кода

Коришћење алата отвореног кода у дигиталној форензици нуди бројне предности, укључујући бесплатно индивидуално образовање форензичара. Ови алати су преносиви и флексибилни, омогућавајући коришћење на различитим системима и на више начина, и могу потврдити налазе комерцијалног софтвера. Највећа предност је приступ њиховом коду, који се може уређивати и прилагођавати специфичним потребама корисника. Два најпознатија алата отвореног кода која се користе за форензику macOS-а су The Sleuth Kit и Autopsy.

6. СТУДИЈА СЛУЧАЈА

Анализа дигиталних уређаја постала је кључна компонента у истрагама савремених кривичних дела, посебно када је у питању злоупотреба истих. Студија случаја фокусира се на форензичку истрагу macOS-а у контексту злочина продаје података о кредитним картицама. Студија се спроводи са циљем да се открије да ли и у коликој мери је могуће идентификовати, сакупити и анализирати релевантне дигиталне доказе који би потврдили незаконите активности извршене на MacBook рачунару. Методе које су се користиле у истрази јесу imaging и претрага по кључним речима, а од алата је коришћен Sumuri-ев Recon ITR.

Анализирани су подаци са лаптопа који је коришћен за почињење кривичног дела продаје података о кредитним картицама.

Најпре, извршена је претрага по кључним речима којом се идентификовао извор комуникације, а затим детаљном анализом утврђено је да је починилац користећи GnuPG библиотеку извршио криптовање података, и тако криптоване послао назад кориснику. Такође, на основу порука, утврђена је и договорена цена за коју је починилац починио кривично дело, а да је тек након уплате новца и потврде трансакције послао енкриптоване податке. Анализа ових доказа омогућила је да се утврди тачно време и датум када су спорне активности извршене, као и да се идентификују корисници који су исте починили.

7. ЗАКЉУЧАК

Дигитална форензика представља неизоставан део савремених истрага кривичних дела, посебно у ери у којој су дигитални уређаји и мреже свеприсутни. Људи користе паметне уређаје на сваком кораку, непрестано остављајући дигиталне трагове, а веома често су и несвесни њиховог постојања. Сваки клик, претрага или комуникација оставља дигитални отисак који се може искористити у форензичкој истрази.

Овај рад пружио је свеобухватан преглед принципа и процеса дигиталне форензике, са посебним фокусом на оперативни систем macOS. Размотрени су његови безбедносни механизми, важни артефакти, методе, технике и алати који се користе за форензичке анализе.

Студија случаја показала је да је уз познавање корисничке шифре могуће извршити тријажу новије серије MacBook лаптопова и креирати форензичку слику.

8. ЛИТЕРАТУРА

- [1] Arnes André, Digital forensics: an academic introduction. Hoboken, NJ: John Wiley & Sons Inc., 2018.
- [2] <https://raf.edu.rs/citaliste/clanci/svastara/sta-je-to-digitalna-forenzika-i-kako-naci-posao-u-tom-mocnom-polju/> (приступљено у августу 2023)
- [3] <https://citeseerx.ist.psu.edu> (приступљено у августу 2023)
- [4] <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9678340> (приступљено у априлу 2024)
- [5] <https://www.macworld.com/article/793119/mac-os-history-video-system-0-97-to-macos-13-ventura.html> (приступљено у марту 2024)
- [6] https://www.youtube.com/watch?v=_K5e8dJtMgE (приступљено у марту 2024)
- [7] <https://developer.apple.com> (приступљено у марту 2024)
- [8] Thomas J. Holt, Adam M. Bossler, and Kathryn C. Seigfried-Spellar - Cybercrime and Digital Forensics, Routledge 2022
- [9] <https://www.youtube.com/watch?v=gcalmiLpZcc> (приступљено у априлу 2024)
- [10] <https://www.magnetforensics.com/resources/mac-os-forensic-artifacts-and-techniques-apr8/> (приступљено у априлу 2024)

Кратка биографија:



Душан Миљковић је рођен у Лесковцу 1995. године. Основне академске студије завршио је на Војној академији у Београду, смер „Војноелектронско инжињерство“, а мастер студије на Факултету техничких наука у Новом Саду, смер „Софтверско инжињерство и информационе технологије“

Контакт: dusanmiljkovic_95@hotmail.com

**ОПТИМИЗАЦИЈА ИЗБОРА ТРАНСФОРМАТОРА И НАПОЈНИХ КАБЛОВА ЗА
НАПАЈАЊЕ НОВЕ ГЛАВНЕ ЦРПНЕ СТАНИЦЕ 1 У НОВОМ САДУ****OPTIMIZATION OF TRANSFORMER AND FEEDER CABLE SELECTION FOR
SUPPLYING THE NEW MAIN PUMP STATION 1 IN NOVI SAD**

Зорица Чулић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду је обрађен избор трансформатора и расподела оптерећења по трансформаторима за напајање пумпи у Новој главној црпној станици 1, у Новом Саду. На нисконапонској страни трансформатора извршен је избор и димезионисање напојних каблова, као и прорачун за заштиту од преоптерећења и прорачун пада напона. Извршен је одабир батерија за компензацију реактивне снаге пумпи. Такође, у раду је обрађено одређивање нивоа заштите громобранске инсталације.

Кључне речи: Одабир енергетских трансформатора, избор проводника и заштите на нисконапонској страни трансформатора, компензација реактивне снаге.

Abstract – This paper addresses the selection of transformers and the distribution of loads among transformers for powering pumps at a pumping station in Novi Sad. On the low-voltage side of the transformers, the selection and sizing of supply cables were performed, along with calculations for overload protection and voltage drop. Batteries for reactive power compensation of the pumps were selected. Additionally, the paper covers the determination of the lightning protection level of the installation.

Keywords: Selection of power transformers, selection of conductors and protection on the low-voltage side of the transformer, Reactive power compensation.

1. УВОД

Вода је несумњиво један од најважнијих ресурса за опстанак сваког живог бића на планети. Она представља основу живота, а њена значајност обухвата широк спектар аспеката који утичу на људе, животиње, биљке и цели околни свет. Индустријска производња и енергетика такође захтевају воду у значајним количинама. Квалитет воде је битан, јер нечисте воде могу оштетити опрему и довести до загађења животне средине. Укратко, вода је животни важан ресурс који захтева посебну пажњу и заштиту

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Дејан Јеркан, ванр.проф.

Потребно је улагати у одрживо управљање водним ресурсима, промовисати ефикасно коришћење воде и посвећивати пажњу заштити водних екосистема.

2. ЦРПНА СТАНИЦА

Црпне станице имају значајну улогу у дистрибуцији воде, посебно у сушним периодима или у подручјима са специфичним захтевима за снабдевање водом. Канализациони систем града Новог Сада је током претходних деценија развијан као општи или заједнички систем одвођења отпадних вода односно као комбиновани канализациони систем. У Новом Саду предвиђена је изградња две нове главне црпне станице (НГЦ 1 и НГЦ 2). Пројектовани капацитет црпне станице НГЦ1 је 15,36 m³/s, а црпне станице НГЦ2 је 9,6 m³/s. У овом раду фокус је на новој главној црпној станици 1 (НГЦ 1)

2.1. Нова главна црпна станица 1 (НГЦ 1)

Нова главна црпна станица 1 се налази непосредно пре места изливања, на левој обали Дунава код Жежељевог моста, и има улогу расподеле протока.



Слика 2.1 Ситуациони приказ НГЦ 1

У функционалном смислу црпна станица (ЦС) је подељена у шест независних целина:

- Проточни канали са решеткама и табластим затварачима
- Црпна станица за отпадне воде (Ц пумпе)
- Црпна станица за први талас атмосферских вода (Д пумпе)
- Црпна станица за атмосферске воде (П пумпе)
- Систем унутрашњег транспорта
- Систем за елиминацију непријатних мириса

У ЦС се налазе три врсте пумпи. Све пумпе у ЦС раде у аутоматском режиму, према нивоима воде у црпилиштима, а по потреби пумпним агрегатима се може управљати и ручно. Црпна станица за отпадне воде и црпна станица за први талас атмосферских вода имају заједничко црпилиште, одвојено од црпилишта црпне станице атмосферских вода преливним зидом.

2.1.1 Црпна станица отпадних вода (Ц пумпе)

Црпна станица отпадних вода (Ц пумпе) служи за препумпавање отпадних вода које се примарно сакупљају са јужног и северног дела канализационог система Новог Сада. Налазе се четири Ц пумпе које имају снагу од 90kW. Три пумпе су радне, а четврта пумпа је резервна.

2.1.2 Црпна станица за први талас атмосферских вода (Д пумпе)

Пумпна станица за први талас атмосферских вода (Д пумпе) има за задатак да усмери први талас атмосферских вода у ретензиони базен, који се налази у оквиру пумпне станице НГЦ1. Номинална снага Д пумпи износи 250kW. У постројењу се налазе две пумпе и обе раде истовремено.

2.1.3 Црпна станица атмосферских вода (П пумпе)

П пумпе служе за препумпавање комбиноване отпадне воде директно у реку Дунав. Ово су растеретне пумпе. У постројењу се налазе четири пумпе називне снаге 360kW.

2.1.4 Остали потрошачи црпне станице

Инсталирана снага осталих електро потрошача у ЦС износи приближно 130kW. За напајање сопствене потрошње предвиђено је димензионисање посебног трансформатора.

3. ЕЛЕКТРОЕНЕРГЕТСКИ СИСТЕМ

Електроенергетски систем су енергетски објекти међусобно повезани тако да чине јединствен техничко-технолошки систем. Један од основних задатака система је квалитет испоруке електричне енергије. Према одредбама Уредбе о условима испоруке електричне енергије допуштена одступања од називног напона и фреквенције су: за напон $\pm 10\%$ и фреквенцију $\pm 0,5\text{Hz}$.

4. ВИСОКОНАПОНСКА ПОСТРОЈЕЊЕ

Високонапонска разводна постројења (ВНРП) играју виталну улогу као делови електроенергетског система (ЕЕС), уз електране, преносне и дистрибутивне мреже.

4.1. Високонапонско постројење за напајање предметних ТС

За предметну ТС предвиђа уградња опреме за 20 kV. Субтранзијентна снага трополног кратког споја на сабирницама 20 kV у ТС 110/20 kV износи 500 MVA,

а време трајања кратког споја је $t = 0,2 \text{ s}$. Високонапонско постројење се састоји од два 20 kV расклопна блока. Оба расклопна блока се налазе у истој просторији ТС "НГЦ-1".

Једно високонапонско постројење састоји се од две доводно одводне ћелије, једне мерне ћелије и једне доводно одводне ћелије. Овај расклопни блок је димензије 1900x1600x1020 mm (ширина x висина x дубина).

Други расклопни блок се састоји од једне доводно одводне ћелије и пет трансформаторских ћелија за трансформаторе са којих се напајају потрошачи црпне станице НГЦ1. Укупна димензија расклопног блока 4125x1220x1600/2050 mm

5. ТРАНСФОРМАТОРИ

Основна функција трансформатора је да промене електрични напон једног кола у виши или нижи електрични напон у другом колу, при чему остаје иста фреквенција струје. Кључни делови трансформатора су језгро трансформатора, намотаји и систем изолације. У погледу електричних карактеристика, трансформатори су дефинисани назначеним напоном, односом трансформације, назначеном снагом, назначеном струјом и другим параметрима који одређују њихову функционалност и способност преноса енергије.

За напајање НГЦ 1 предвиђа се изградња трансформаторске станице (ТС „НГЦ1“) , преносног односа 20(10)/0,4kV, у складу са издатим условима за пројектовање и прикључење.

5.1 Мерни трансформатори

У електроенергетским системима високог напона, где су напони обично веома високи, а струје велике, користе се мерни трансформатори који омогућавају мерење и контролу различитих величина и параметара. Постоје две врсте мерних трансформатора - струјни (СТ) и напонски (НТ).

5.2 Енергетски трансформатор

Карактеристичне вредности енергетских трансформатора су: назначени напон, назначени однос трансформације, назначена снага, назначена струја, могућност промене односа трансформације, напон кратког споја, губици у бакру, губици у гвожђу, начин хлађења и спрега трансформатора. Назначени однос трансформације m_n , је однос између назначеног напона примара и назначеног напона секундара. Спрежни број n је број који показује колико пута по 30° величине на нижој напонској страни касне за величинама на вишенпонској страни.

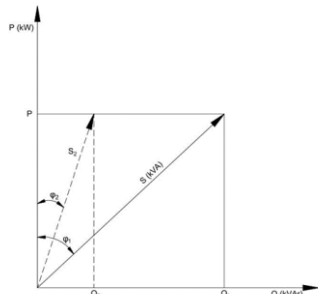
$$m_n = \frac{U_{1n}}{U_{2n}} e^{jn30^\circ}$$

5.3 Паралелан рад трансформатора

Паралелни рад трансформатора омогућава истовремени рад два или више трансформатора како би се подмириле додатне потребе или осигурала већа погонска сигурност.

6. КОМПЕНЗАЦИЈА РЕАКТИВНЕ ЕНЕРГИЈЕ

Оптерећење са малим фактором снаге вуче више струје од оптерећења са већом вредности фактора снаге за исту количину пренесене енергије. За одређену потрошњу активне снаге P (kW) уз фазни помак φ_1 имамо одговарајућу реактивну снагу Q_1 и да бисмо смањили ту реактивну енергију мрежа треба да добије фазни помак φ_2 , тј. мрежи треба додати реактивну снагу:



Слика 6.1 Компензација реактивне енергије

Са слике се може закључити да оптерећење са малим фактором снаге вуче више струје, него оптерећење са већим фактором снаге за исту количину пренесене корисне енергије. Због повећаних губитака, електродистрибутивна компанија наплаћује већу цену електричне енергије за потрошаче који имају фактор оптерећења мањи од 0.95, у Републици Србији.

Према хидрауличним симулацијама очекивано време рада Д пумпи је испод 100 часова током године, док је очекивано време рада П пумпи испод 40 часова током године што оправдава изостанак компензације реактивне енергије за ове пумпе. За Ц пумпе (90 kW, $\cos\varphi = 0,8$) је предвиђена фиксна компензација реактивне енергије од 75 kVAr батеријама кондензатора од 3x30 kVAr номиналног напона 440V. Како струја празног хода трансформатора 1600 kVA износи 1 % предвиђена је фиксна компензација реактивне енергије од 16 kVAr батеријом кондензатора од 20 kVAr номиналног напона 440 V. Ова компензација треба да подмири потребе магнетнења трансформатора.

7. ПРОРАЧУНИ

7.1 Прорачуни потребне снаге трансформатора

За постројење црпне станице су изабране четири врсте великих пумпи. У нормалном погонском стању раде три пумпе типа П, две пумпе типа Д и три пумпе типа Ц. Укупна привидна снага тада износи:

$$P_{an} = 3 \cdot 391,6 + 2 \cdot 269,4 + 3 \cdot 337,9 = 2727,4 \text{ kW}$$

$$Q_n = 3 \cdot 303,9 + 2 \cdot 188 + 3 \cdot 191,5 = 1862,4 \text{ kVAr}$$

$$S = \sqrt{P_{an}^2 + Q_n^2} = 3302,6 \text{ kVA}$$

Због тога се одабиру трансформатори снаге 1600kVA за напајање пумпних постројења и трансформатор снаге 160 kVA за напајање сопствене потрошње црпне станице. Укупна инсталирана снага за пумпно

постројење је 6400kVA. У следећој табели приказана је расподела напајања пумпи по трансформаторима:

Трансформатор	Пумпа			$S_n(kVA)$
ТР1	П1	Д1	Ц1	1210
ТР2	П2		Ц2	884
ТР3	П3		Ц3	884
ТР4	П4	Д2	Ц4	1210

7.2 Прорачун струје кратког споја

Прорачун струје кратког споја је важан због процене потребне опреме. Добијене вредности струје су 14,43kA на сабирницама 20kV и 38,49kA на сабирницама 0,4kV. Како је већа вредност струје у случају кратког споја на НН сабирници трансформатора она ће бити меродавна за механички прорачун сабирнице, као и за прекидну моћ прекидача.

7.3 Проводници за напајање инсталација

7.3.1 Избор проводника

Избор проводника према трајно дозвољеним струјама обављен је према стандарду SRPS HD 60364-4 уз примену: корекционог фактора за групна струјна кола k_n , корекционог фактора за температуру околине k_θ , и корекционог фактора за термичку отпорност тла k_λ .

$$I_z = k_\theta \cdot k_\lambda \cdot k_n \cdot I_{trdoz}$$

7.3.2 Заштита од струје преоптерећења

Заштитни уређаји морају бити предвиђени да прекидају сваку струју преоптерећења која протиче проводницима пре него што проузрокују повишење температуре штетно по изолацију, спојеве, стезаљке или околину. Радна карактеристика уређаја који штити електрични вод од преоптерећења мора да испуни два услова (SRPS HD 60364-4):

1. $I_B \leq I_n \leq I_z$
2. $I_2 \leq 1,45 \cdot I_z$

7.3.3 Провера систем заштите

Прописом SRPS N.B2.741/1989 су дефинисани захтеви за безбедност заштите од електричног удара (поглавље 5. Заштита од индиректног додира, глава 5.1 Заштита аутоматским искључењем напајања). Систем заштите TN изводи се: као четворопроводнички систем заштите TN-C у коме су неутрална и заштитна функција обједињене у једном проводнику кроз цео систем, као петопроводнички систем заштите TN-S, који кроз цео систем има раздвојене неутрални и заштитни проводник, или као комбинација ова два система - систем TN-C-S, у коме су неутрална и заштитна функција обједињене у једном проводнику само у једном делу система.

7.3.4 Прорачун пада напона

За прорачун пада напона за трофазне потрошаче који имају фактор снаге $\cos \varphi < 1$ коришћена је следећа формула:

$$u_{\%} = \frac{100 \cdot l \cdot P}{\sigma \cdot S \cdot U_l^2} \cdot (r + x \cdot \tan \varphi)$$

Дозвољен пад напона између тачке напајања електричне инсталације и било које друге тачке за струјно коло осветљења је 5%, а за струјна кола осталих потрошача 8%, ако се ел. инсталација напаја директно из ТС која је прикључена на високи напон. Највећи пад напона се јавља између трансформатора 1 (ТР1) и пумпе за атмосферску воду П2 и износи 0,88%, што задовољава захтев да пад напона буде мањи од 8%.

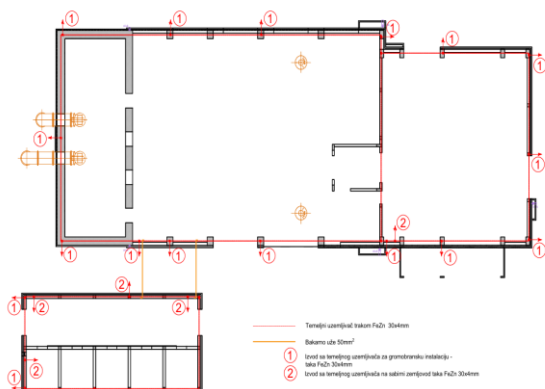
8. ГРОМОБРАНСКА ИНСТАЛАЦИЈА

За заштиту објекта од атмосферских пражњења предвиђена је громобранска инсталација у виду класичне громобранске инсталације- Фарадејевог кавеза. Фарадејев кавез се око објекта формира добрим електропроводним повезивањем хватаљки, спустова, мерно-раставних спојева, земљовода, темељног уземљивача и свих осталих електропроводних делова објекта у једну целину. Учестаност удара грома на територији града Новог Сада је $T_d=31$. Ефикасност громобранске инсталације се рачуна по обрасцу:

$$E_r \geq I - N_c/N_d$$
$$E_r \geq 0,9759651$$

За вредност $0,95 < E_r < 0,98$ громобранска инсталација има I ниво заштите.

Зграде трафостанице и црпне станице подижу се као засебни објекти, те је потребно да се ураде уземљивачи за сваки објекат. Уземљивач се састоји од поцинкованих трака FeZn 30x4mm у бетону темеља. Веза између ова два објекта се израђује бакарним ужетом Си 50mm². Вредност отпора здруженог уземљивача представља паралелну везу уземљивача ова два објекта и износи $R=0.804\Omega$.



Слика 8.1 Темељни уземљивач

9. ЗАКЉУЧАК

Изградња главне црпне станице у Новом Саду представља кључни корак ка унапређењу инфраструктуре града, са двонаменском функцијом за одвођење атмосферске и отпадне канализационе воде.

Правилни избор трансформатора за напајање пумпи и димензионисање каблова и заштите су од пресудног значаја за поузданост и ефикасност читавог система. Одговарајући трансформатори осигуравају стабилно напајање пумпи, чиме се спречавају прекиди у раду и потенцијалне штете на опреми. Прецизно димензионисање каблова гарантује адекватну дистрибуцију електричне енергије без губитака и прегревања, док адекватне заштитне мере омогућавају сигурност система и дуговечност опреме.

10. ЛИТЕРАТУРА

- [1] Војводинапројект, званични сајт. <https://www.vojvodinaprojekt.rs>
- [2] Јован Нахман, Разводна постројења, Електротехнички факултет Београд 2005.
- [3] Пројекат хидротехничких инсталација Нове главне црпне станице 1 у Новом Саду,
- [4] Пројекат електроенергетских инсталација Нове главне црпне станице 1 у Новом Саду,

Кратка биографија:

Зорица Чулић рођена у Бањалуци 1996. год. Завршни (Bachelor) рад на Фкултету техничких наука из области Електротехнике и рачунарства – Енергетика, електроника и телекомуникације одбранила је 2020. године.

**SEMANTIČKA SEGMENTACIJA POKRIVENOSTI ZEMLJIŠTA U GORNJEM
PODUNAVLJU NA SLIKAMA SENTINEL-2 SATELITA****SEMANTIC SEGMENTATION OF LAND COVER IN THE UPPER DANUBE REGION
ON SENTINEL-2 SATELLITE IMAGES**

Miloš Marinković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTMATIKA

Kratak sadržaj – U okviru ovog rada opisana je semantička segmentacija pokrivenosti zemljišta u Gornjem Podunavlju sa Sentinel-2 satelitskih slika. Segmentacija je izvršena u 6 klasa: krošnje drveća, vodene površine, travnate površine, tlo, poljoprivredno zemljište i vegetacija na vodenim površinama. Podaci predstavljaju multispektralne slike Sentinel-2 satelitskih slika koje obuhvataju područje Gornjeg Podunavlja. Za obuku su korišćeni XG-Boost klasifikator i Random Forest klasifikator. Dronske slike ovog područja služile su za kreiranje činjeničnog stanja. Kao dodatna obeležja prilikom klasifikacije korišćeni su vegetacioni indeksi. Maksimalna postignuta tačnost iznosila je 93%.

Ključne reči: semantička segmentacija, mašinsko učenje, Sentinel-2, Random forest, XG boost, vegetacioni indeksi

Abstract – This paper describes the semantic segmentation of land cover in the Upper Danube region using Sentinel-2 satellite images. The segmentation was performed into 6 classes: tree canopies, water bodies, grassland, bare soil, agricultural land, and vegetation on water surfaces. The data utilized multispectral Sentinel-2 satellite images covering the Upper Danube area. XG-Boost and Random Forest classifiers were employed for training. Drone images of the area served to establish ground truth. Vegetation indices were used as additional features during classification. Maximal accuracy was 93%.

Keywords: semantic segmentation, machine learning, Sentinel-2, Random Forest, XG Boost, vegetation indices

1. UVOD

Semantička segmentacija pokrivenosti zemljišta, zasnovana na podacima sa Sentinel-2 satelitskih slika zasniva se na klasifikaciji svakog pojedinačnog piksela nezavisno, upotrebom nekog algoritma mašinskog učenja [1]. Ovaj rad predstavlja primenu ovog pristupa na problem detekcije različitih slojeva koji pokrivaju oblast Gornjeg Podunavlja. Ti slojevi su: krošnje drveća, vodene površine, travnate površine, tlo, poljoprivredno zemljište i vegetacija na vodenim površinama.

Svaki uzorak, u ovom slučaju piksel, opisan sa više kanala različite rezolucije i talasnih dužina na kojima senzori satelita dobijaju informaciju.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Segedinac, vanredni prof.

Takođe, kao dodatna obeležja korišćeni su vegetacioni indeksi, dobijeni kombinovanjem već postojećih kanala. Vegetacioni indeksi se pokazuju kao vrlo značajni atributi za kvalitetnu klasifikaciju vegetacije [2]. Za klasifikaciju u ovom radu korišćeni su *Random Forest* i *XG Boost* algoritmi.

2. SENTINEL-2 SATELIT i L2A proizvod

Sentinel-2 satelit je deo evropskog programa za praćenje Zemlje, poznatog kao *Copernicus program*, koji pruža visokokvalitetne optičke slike Zemljine površine. Ovaj satelit ima mogućnost snimanja u više spektralnih kanala, što omogućava detaljno praćenje promena na Zemljinoj površini tokom vremena. Osim toga, Sentinel-2 slike su besplatno dostupne za upotrebu u istraživanju, razvoju i komercijalnim aktivnostima.

L2A proizvod u kontekstu daljinskog osmatranja i satelitskih snimaka se odnosi na nivo procesiranja sirovih satelitskih snimaka. U okviru programa Kopernikus, koji uključuje Sentinel satelite, podaci o slikama se obrađuju i distribuiraju u različitim nivoima, pri čemu svaki predstavlja različitu fazu obrade i unapređenja. Proizvod nivoa 2A je srednji nivo obrađenih podataka koji uključuje dodatne korekcije i unapređenja u poređenju sa podacima nivoa 1C.

Podaci nivoa 1C obično sadrže ortorektifikovane slike refleksije na vrhu atmosfere, dok podaci nivoa 2A prolaze kroz dodatnu obradu kako bi pružili refleksije na dnu atmosfere, korekciju atmosfere i dodatna unapređenja poput maskiranja oblaka i senki, korekciju vodene pare i moguću korekciju aerosola.

Proizvod nivoa 2A je posebno koristan za aplikacije koje zahtevaju tačne vrednosti refleksije površine, kao što su praćenje vegetacije, klasifikacija pokrivenosti zemlje i detekcija promena. U ovom radu korišćene su upravo slike sa ovim nivom obrade.

3. Opis baze podataka

Baza podataka predstavlja jednu satelitsku sliku u tri datuma: 2023.06.13, 2023.06.23 i 2023.07.03. Slika ima rezoluciju od 10 metara po pikselu i veličine je 10980x10980, dok je region od interesa veličine 400x400. Svaka slika je multispektralna i sadrži 12 kanala različitih delova elektromagnetnog spektra. Pored 12 kanala kao početnih obeležja koriste se i 22 vegetaciona indeksa kao dodatna obeležja.

Sa 34 obeležja u tri datuma dolazi se do ukupnog broja od 102 obeležja koji reprezentuju svaki uzorak odnosno

svaki piksel na slici. Skup podataka nije ujednačen, raspored klasa se može videti na slici 1.

Trening skup je sačinjen od 80% ukupnog uzoračkog skupa, a test skup predstavlja preostalih 20% podataka. Slika koja se obrađuje je slika iz Sentinel-2 putanje R036 sa oznakom T34TCR.



Slika 1. Zastupljenost klasa u skupu podataka

4. VEGETACIONI INDEKS

Vegetacioni indeksi su indeksi koji se računaju na osnovu kanala multispektralnih satelitskih slika i nose dodatnu informaciju o tome šta se nalazi na tim slikama.

Neki vegetacioni indeksi imaju različite vrednosti u različitim trenucima u vremenu za različite terene. Ovo su indeksi koji su se koristili:

NDVI (*Normalized Difference Vegetation Index*)

NDVIRE1 (*Normalized Difference Vegetation Index Red Edge 1*)

NDVIRE2 (*Normalized Difference Vegetation Index Red Edge 2*)

NDVIRE3 (*Normalized Difference Vegetation Index Red Edge 3*)

EVI (*Enhanced Vegetation Index*)

EVI2 (*Enhanced Vegetation Index 2*)

VARI (*Visible Atmospherically Resistant Index*)

SAVI (*Soil-Adjusted Vegetation Index*)

ARVI (*Atmospherically Resistant Vegetation Index*)

GAVI (*Green Atmospherically Resistant Vegetation Index*)

VDVI (*Visible Difference Vegetation Index*)

NDWI (*Normalized Difference Water Index*)

NDWI2 (*Normalized Difference Water Index 2*)

NLI (*Normalized Leaf Index*)

NLI2 (*Normalized Leaf Index 2*)

MNLI (*Modified Normalized Leaf Index*)

MNLI2 (*Modified Normalized Leaf Index 2*)

NDMI (*Normalized Difference Moisture Index*)

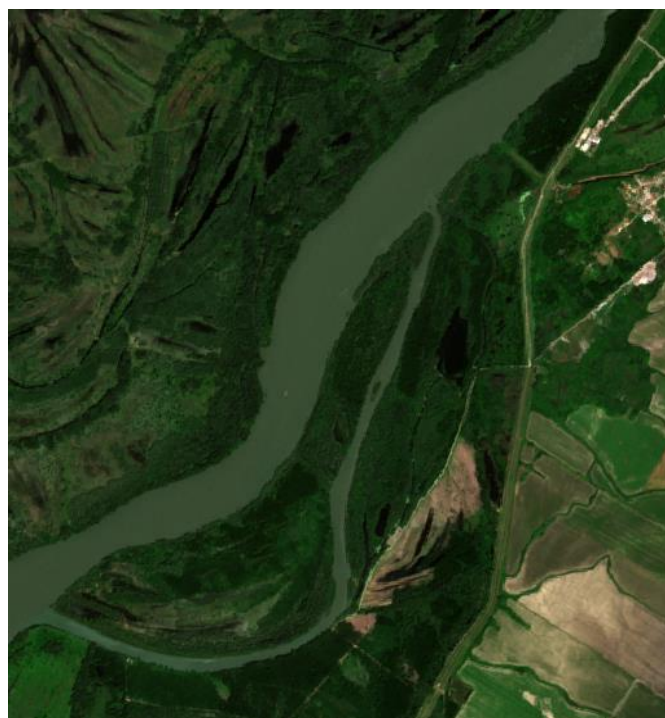
TG (*Triangular Greenness Index*)

GLI (*Green Leaf Index*)

ExG (*Excess Green Index*)

CIVE (*Color Index of Vegetation Extraction*)

Ovi indeksi su odabrani na osnovu problema koji se rešava i povezanosti između prediktovanih klasa i ovih obeležja.



Slika 2. Izgled Gornje Podunavlja Sentinel-2 slike

5. ALGORITMI MAŠINSKOG UČENJA

Ovaj rad predlaže dva pristupa za klasifikaciju pojedinačnih piksela. Kako se suštinski ovde radi o tabelarnim podacima, ideja je da se koriste standardni

algoritmi mašinskog učenja kao što su Random Forest i XGBoost algoritam i da se meri njihova makro usrednjen *F1-score*.

5.1. RANDOM FOREST

Random forest [3], koji je predstavio Leo Breiman 2001. godine, predstavlja metodu grupnog učenja koja se koristi za klasifikaciju i regresiju. Funkcionira tako što tokom obuke gradi više stabala odlučivanja i daje izlaz u vidu moda klasa (za klasifikaciju) ili prosečne vrednosti predikcija (za regresiju) pojedinačnih stabala. Ovaj pristup poboljšava tačnost predikcija i kontroliše prekomerno prilagođavanje (overfitting) prosečavanjem rezultata. Random forests su otporni na šum i mogu da se nose sa velikim skupovima podataka sa visokom dimenzionalnošću, što ih čini popularnim izborom za razne primene u mašinskom učenju.

5.1. XG BOOST

XGBoost (*eXtreme Gradient Boosting*) [4] je algoritam mašinskog učenja koji kombinuje stabla odlučivanja u sklopu metode pojačavanja gradijentom (*engl. gradient boosting*). Razvijen je od strane Tianqi Chen-a 2016. godine i popularan je zbog svoje visoke efikasnosti i tačnosti u širokom spektru problema klasifikacije i regresije. XGBoost radi iterativno, dodajući nova stabla u sekvenci, sa svakim novim stablom koje se fokusira na korekciju grešaka prethodnih stabala. Koristi gradijentni spust za minimizaciju gubitka prilikom dodavanja novih stabala, čime se postiže optimizacija performansi i brža konvergencija u odnosu na druge algoritme.

6. OBUKA I EVALUACIJA MODELA MAŠINSKOG UČENJA

Hipoteza koju ovaj radi ispituje jeste da li dodavanjem obeležja iz više datuma doprinosi tačnosti klasifikaciji. Konkretno postavlja se 6 eksperimenata u kom se svaki put model trenira na istom skupu podataka sa smanjenim brojem obeležja izbacivanjem po jednog datuma iz početnog skupa od tri datuma. Ovaj postupak se ponavlja i za Random Forest i za XGBoost. Takođe modeli se testiraju nad istim skupom podataka u svakom od ovih 6 eksperimenata.

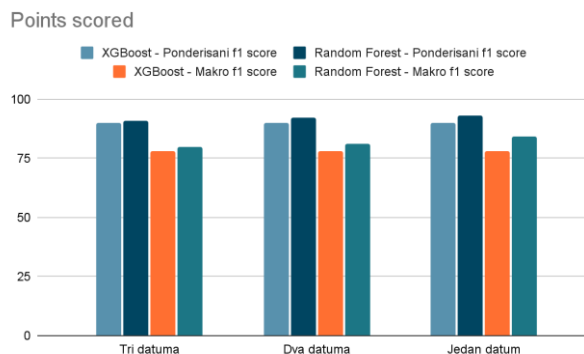
Tabela 1. Tabela trening i test podataka u svim eksperimentima

Skup podataka	Broj piksela	E1 - Broj obeležja	E2 - Broj obeležja	E3 - Broj obeležja
Trening	43217	34	68	102
Test	10806	34	68	102

Kako podaci nisu uravnoteženi za evaluaciju modela biće korišćeni ponderisana i makro F1 mera.

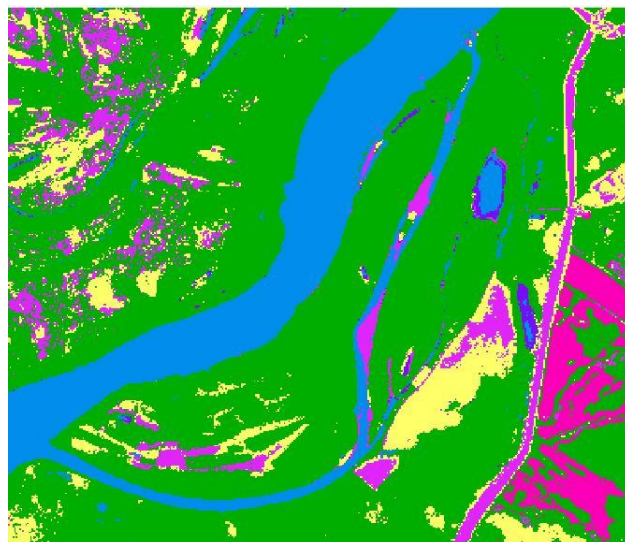
7. REZULTATI

Na osnovu rezultata dobijenih nakon izvršenih eksperimenata hipoteza nije potvrđena, naprotiv pokazuje se da model ima veću tačnost prilikom treniranjem sa obeležjima iz samo jednog datuma. Najbolje se pokazao Random Forest na samo jednom datumu i to je makro F1 mera iznosila 84%, dok je ponderisana F1 mera iznosila 92%. Najlošiji se pokazao XGBoost obučen na sva tri datuma sa 78% makro F1 mere i 90% ponderisane F1 mere.



Slika 3. F1 mera na osnovu broja obeležja

Takođe, sa slike se vidi da XGBoost se ne menjaju performanse nezavisno od broja podataka.



Slika 4. Dobijena mapa predikcija

8. ZAKLJUČAK

Istraživanje je pokazalo da je Random Forest bez dodatnih obeležja koji se menjaju u vremenu najbolji klasifikator i da vrlo uspešno rešava problem klasifikacije terena i vegetacije sa Sentinel-2 slika. Unapređenja ovog pristupa bi se ogledala u upotrebi dubokih rekurentnih [5] i konvolutivnih neuronskih mreža [6] koje bi imale veći kontekst o okolini piksela kao i povezanosti unutar vremenske serije. Ipak, kako u ovom istraživanju nije postojalo dovoljno obeleženih podataka, takvi pristupi ne bi imali smisla.

9. LITERATURA

- [1] Svoboda, J., Štych, P., Laštovička, J., Paluba, D.; Kobluk, N.: „Random Forest Classification of Land Use, Land-Use Change and Forestry (LULUCF) Using Sentinel-2 Data—A Case Study of Czechia“, *Remote Sens.* 14, 1189, 2022.
<https://doi.org/10.3390/rs14051189>
- [2] Xue J., Su B.: „Significant Remote Sensing Vegetation Indices: A Review of Developments and Applications“, *Journal of Sensors*, vol. 2017, Article ID 1353691, 17 pages, 2017.
<https://doi.org/10.1155/2017/1353691>

- [3] Breiman, L.: „Random Forests“, *Machine Learning* 45:5-32. 2021.
<http://dx.doi.org/10.1023/A:1010933404324>
- [4] Chen, T., Guestrin, C.: „XGBoost: A Scalable Tree Boosting System“, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794, 2016.
<https://doi.org/10.1145/2939672.2939785>
- [5] Marhon, S.A., Cameron, C.J.F., Kremer, S.C.: „Recurrent Neural Networks. In: Bianchini, M., Maggini, M., Jain, L. (eds) *Handbook on Neural Information Processing*“, *Intelligent Systems Reference Library*, vol 49. Springer, Berlin, Heidelberg, 2013.
https://doi.org/10.1007/978-3-642-36657-4_2
- [6] Yamashita, R., Nishio, M., Do, R.K.G. et al.: „Convolutional neural networks: an overview and application in radiology“, *Insights Imaging* 9, 611–629 2018.
<https://doi.org/10.1007/s13244-018-0639-9>

Kratka biografija:

Miloš Marinković rođen je u Novom Sadu 2000. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Sematička segmentacija pokrivenosti zemljišta u Gornje Podunavlju na slikama Sentinel-2 satelita odbranio je 2024.god.
kontakt: 1milosmarinkovic1@gmail.com

NAPREDNE TEHNIKE ZA SENTIMENT ANALIZU: STUDIJA KLASIFIKACIONIH I GENERATIVNIH MODELA NA ONLINE KOMENTARIMA**ADVANCED TECHNIQUES FOR SENTIMENT ANALYSIS: A STUDY OF CLASSIFICATION AND GENERATIVE MODELS ON ONLINE COMMENTS**

Cvijetin Mladenović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad se bavi sentiment analizom komentara korisnika koristeći različite tehnike obrade prirodnog jezika i algoritme dubokog učenja. Implementirani su modeli poput rekurentnih neuronskih mreža (RNN), konvolutivnih neuronskih mreža (CNN), Word2Vec, GloVe, BERT, kao i generativni llama-7b-hf model. Skup podataka je preuzet sa Kaggle platforme i sadrži komentare korisnika na različite proizvode. Evaluacija modela je izvršena korišćenjem F1 mere, omogućavajući detaljnu analizu performansi u kontekstu nebalansiranih klasa. Najbolji rezultati su postignuti upotrebom transformera i SVM klasifikatora.

Ključne reči: Sentiment analiza, RNN, CNN, Word2Vec, GloVe, BERT, llama-7b-hf, NLP, klasifikacija

Abstract – This paper explores sentiment analysis of user comments using various natural language processing techniques and deep learning algorithms. Implemented models include Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Word2Vec, GloVe, BERT, and the generative llama-7b-hf model. The dataset was obtained from the Kaggle platform and contains user comments on various products. Model evaluation was performed using the F1 score, enabling a detailed analysis of performance in the context of imbalanced classes. The best results were achieved using transformers and SVM classifiers..

Keywords: Sentiment analysis, RNN, CNN, Word2Vec, GloVe, BERT, llama-7b-hf, NLP, classification

1. UVOD

Razvoj interneta i digitalnih tehnologija značajno je promenio način na koji ljudi komuniciraju i kupuju. Sentiment analiza ima široku primenu, od poslovne inteligencije i marketinških istraživanja do unapređenja korisničkog iskustva. U e-commerce industriji, ona omogućava trgovcima da prate zadovoljstvo kupaca, identifikuju probleme i unaprede svoje proizvode i usluge. Komentari korisnika na online platformama predstavljaju bogat izvor podataka o njihovim stavovima i mišljenjima, što pruža mogućnost za dublju analizu i razumevanje ponašanja potrošača.

NAPOMENA:

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

Problem koji ovaj rad adresira je predikcija ocena (rejtinga) komentara korisnika na osnovu sentimenta izraženog u tekstualnom sadržaju tih komentara. Specifično, cilj je razviti modele koji mogu automatski analizirati tekstualne komentare i dodeliti im ocenu u rasponu od 1 do 5. Ovaj problem je izazovan zbog prirode tekstualnih podataka, koji su često neuredni, neformalni i kontekstualno zavisni.

Koriste se različiti pristupi dubokog učenja i NLP-a, uključujući RNN, CNN, Word2Vec, GloVe, BERT i generativni model llama-7b-hf. Podaci su preuzeti sa Kaggle platforme i uključuju 23,486 komentara korisnika [1]. Pretprocesiranje podataka obuhvata čišćenje teksta i pripremu za treniranje modela. Skup podataka je podeljen na trening, validacioni i testni deo za evaluaciju performansi modela.

Evaluacija je izvršena korišćenjem F1 mere. Najbolji rezultati postignuti su upotrebom transformera i SVM klasifikatora, dok su generativni modeli pokazali potencijal za dodatna istraživanja. Ovaj rad doprinosi unapređenju metoda za automatsku analizu sentimenta u komentarima korisnika, što može biti korisno za e-commerce platforme i druge aplikacije koje se oslanjaju na korisničke povratne informacije.

U prvom poglavlju dat je uvod sa opisom problema. U drugom poglavlju opisana su prethodna rešenja i najrelevantniji radovi. Treće poglavlje opisuje metodologiju predikcije, četvrto poglavlje eksperimente i njihove rezultate, a peto poglavlje daje zaključke rada.

2. PRETHODNA REŠENJA

Rad "Statistical Analysis on E-Commerce Reviews, with Sentiment Classification using Bidirectional Recurrent Neural Network" [2], istražuje vezu između recenzija kupaca i preporuka proizvoda, fokusirajući se na e-trgovinu. Autori su koristili 22,621 javni komentar. Implementirana je bidirekciona rekurentna neuronska mreža sa LSTM, uz korišćenje GloVe reprezentacije za mapiranje reči u vektorski prostor. Rejting od 3 ili više tretira se kao pozitivna ocena, dok se rejting manji od 3 tretira kao negativna ocena. Postignuta je F1 mera od 0.93 za klasifikaciju sentimenta, ali je uočena pristrasnost zbog nebalansiranosti pozitivnih i negativnih komentara.

U radu "Sentiment Analysis using machine learning algorithms: online women clothing reviews" [3], autor koristi različite metode za klasifikaciju sentimenta.

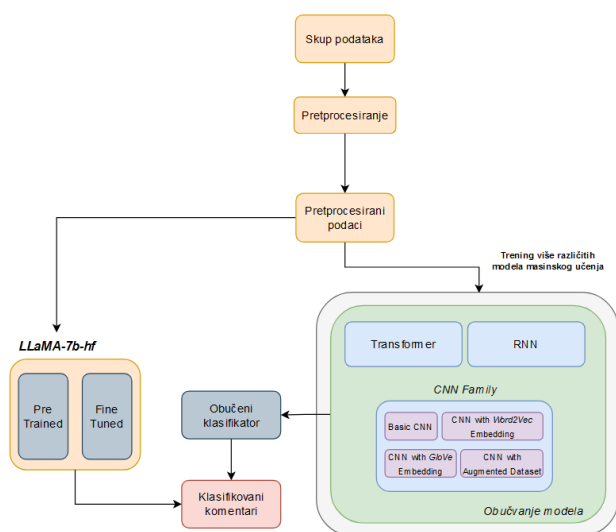
Korišćeni algoritmi uključuju *Support Vector Machine*, *Logistic Regression*, *Random Forest* i *Naive Bayes*. Modeli su evaluirani na osnovu tačnosti, preciznosti, odziva, F1 mere i AUC. *Naive Bayes* je postigao najveću preciznost od 93%.

Rad "Fine-grained Sentiment Classification using BERT" [4], predstavlja značajan doprinos u oblasti analize sentimenta, posebno kroz primenu *Bidirectional Encoder Representations from Transformers* (BERT) modela za precizno razlikovanje finih nijansi sentimenta. Eksperiment je izvršen nad *Stanford Sentiment Treebank fine-grained* (SST-5) skupom podataka, gde je BERT model postigao tačnost od 55,5%.

Rad "Self-Attention-Based BiLSTM Model for Short Text Fine-Grained Sentiment Classification" [5], uvodi model koji poboljšava analizu sentimenta kratkih tekstova fokusirajući se na nijanse i aspekte teksta. Model koristi BiLSTM i mehanizme samo-pažnje za bolje hvatanje konteksta i značaja reči. Evaluacija modela je izvršena nad tri skupa podataka sa *SemEval 2014 Task 4* takmičenja, gde je postignuta prosečna tačnost od 73% na prvom, 65% na drugom i 65% na trećem skupu podataka.

3. METODOLOGIJA

Metodologija korišćena u ovom radu za sentiment analizu komentara uključuje nekoliko ključnih koraka, koji su prikazani na slici 1.



Slika 1. Dijagram metodologije.

Prvo se primenjuje pretprocesiranje nad skupo podataka koji je preuzet sa sajta *Kaggle* i sadrži 23,486 redova u kojima su opisani komentari korisnika. Za potrebe projekta su korišćeni samo atributi 'Review Text' i 'Rating'. Ostala polja su izostavljena jer nisu relevantna za rešavanje problema kojim se ovaj rad bavi. Skup podataka je podeljen na trening, test i validacioni skup u razmeri 80/10/10, pri čemu trening skup sadrži 18,788 komentara, validacioni 2,349, a test skup 2,349 komentara. Podela je odrađena na osnovu atributa 'Rating', kako bi se očuvala ista distribucija ciljne labele u svakom od skupova podataka.

Da bi se poboljšale performanse modela, neophodno je očistiti tekst od nepotrebnih karaktera koji ne nose

nikakvo semantičko značenje. U tu svrhu primenjeni su sledeći koraci:

1. Uklanjanje praznih komentara koji ne sadrže korisne informacije
2. Izbacivanje duplikata kako bi se izbeglo višestruko učenje istih informacija
3. Zamenjena višestrukih razmaka (engl. *whitespace*) jednim razmakom za uniformnost teksta
4. Izbacivanje HTML tagova iz komentara koristeći *BeautifulSoup* [6] biblioteku
5. Zamenjena ili uklanjanje unicode karaktera koji mogu ometati procesiranje teksta
6. Uklanjanje nepoželjnih simbola koji mogu iskriviti analizu

Ove varijacije su implementirane kako bi se ispitalo koji pristup najbolje doprinosi poboljšanju performansi modela.

Pretprocesirani trening skup se koristi za obučavanje modela za klasifikaciju komentara. Pre nego što se tekstualni sadržaji mogu proslediti neuronskoj mreži za klasifikaciju, ključno je pretvoriti ih u vektore koji efikasno održavaju semantičku vrednost teksta. Dobijeni vektor teksta se zatim šalje na ulaz mreže. Implementacija konvolutivne mreže je odrađena upotrebom biblioteke *Keras* [7]. Tokenizacija teksta je realizovana korišćenjem *Keras*-ovog Tokenizera, dok su *Word2Vec* i *GloVe* korišćeni kao naprednije metode vektorizacije. Implementacija rekurentne mreže je odrađena upotrebom biblioteke *TensorFlow* [8]. Korišćena su dva bidirekionalna LSTM sloja, svaki sa 16 neurona, i dodatni *Dropout* slojevi kako bi se sprečilo preterano prilagođavanje. Tokom treniranja, vrednost funkcije greške na validacionom setu je pažljivo praćena kako bi se osiguralo da model ne gubi sposobnost generalizacije na novim podacima. Da bi se ublažila prisutna nebalansiranost u *dataset*-u, primenjen je pristup augmentacije podataka korišćenjem tehnike parafraziranja koristeći PEGASUS model iz biblioteke *Transformers*. Korišćen je *nli-mpnet-base-v2* transformer model iz *sentence-transformers* biblioteke [9]. Pretrenirani transformer model korišćen je kao sloj za vektorizaciju teksta, dok je na izlaz ovog modela dodat klasifikator koji se trenira i daje konačnu predikciju rejtinga za dati komentar. Klasifikatori kao što su SVM, *K-Nearest Neighbors*, *Random Forest*, *Multi-Layer Perceptron*, *Logistic Regression* i *Linear Regression* su trenirani na trening skupu i evaluirani na validacionom skupu podataka. Za implementaciju generativnog pristupa korišćen je *llama-7b-hf* model [10]. Implementacija uključuje kreiranje konteksta zadatka za model, prikazivanje test primera i prikupljanje odgovora koje model generiše za konačnu klasifikaciju. Koriste se *AutoModelForCausalLM* i *SFTTrainer* klase za implementaciju *few-shot learning* metode, gde se modelu dostavljaju reprezentativni primeri za svaku klasu sa informacijama o klasama.

4. EKSPERIMENTI I REZULTATI

U ovom poglavlju istraženi su različiti pristupi sentiment analizi koristeći kombinaciju tradicionalnih i savremenih metoda obrade prirodnog jezika. Eksperimenti su

podeljeni u dve glavne grupe, poređenje klasičnih NLP pristupa i poređenje generativnog pristupa. Eksperimenti su obuhvatili evaluaciju efikasnosti modela na temelju detaljno pripremljenog skupa podataka, prilagođavanja arhitekture modela i njihovih hiperparametara, kao i primenu generativnih modela. Komparativna analiza performansi modela bazirana je na standardnim metrikama, kako bi se identifikovali najefikasniji pristupi za rešavanje problema sentiment analize.

Komentari o ženskoj odeći, koji su prošli kroz proces čišćenja i lematizacije koristeći biblioteku *Spacy* [11]. Zbog dominacije komentara sa ocenom 5, primenjena je augmentacija podataka za uravnoteženje skupa.

Eksperimenti u prvoj grupi su sprovedeni koristeći različite pristupe:

1. Konvolutivne neuronske mreže (CNN): Korišćenje osnovnog CNN modela, kao i varijanti sa *Word2Vec* i *GloVe* reprezentacijama reči
2. Rekurentne neuronske mreže (RNN): Upotreba bidirekcionalnih LSTM slojeva za obradu sekvencijalnih podataka
3. Transformer modeli: Implementacija BERT modela za finu analizu sentimenta

Tokom eksperimenata, optimizovani su hiperparametri i arhitekture modela korišćenjem grafičkih kartica dostupnih preko *Google Colab*-a. Proces treniranja modela uključivao je tehnike ranog zaustavljanja kako bi se sprečilo prekomerno učenje.

Eksperimenti iz druge grupe za generativni pristup uključivali su testiranje *llama-2-7b-chat-hf* modela, u varijantama sa i bez finog podešavanja. *Fine tuning* je omogućio prilagođavanje modela specifičnostima skupa podataka, dok je upotreba *few-shot learning*-a pomogla u klasifikaciji komentara.

Evaluacija performansi modela izvršena je korišćenjem F1 mere kao glavne metrike zbog nebalansiranosti skupa podataka. Najbolje rezultate postigao je transformer model u kombinaciji sa SVM klasifikatorom. Augmentacija podataka pokazala je minimalan uticaj na poboljšanje performansi.

Tabela 1. Rezultati za transformer model u kombinaciji sa različitim klasifikacionim modelom

Klasifikator	Hiperparametri	Micro F1
SVM sa linearnim kernelom	<i>random_state</i> =1	0.6732
SVM sa linearnim kernelom	<i>class_weight</i> ='balanced'	0.6065
SVM sa rbf kernelom		0.6653
<i>K Neighbors Classifier</i>		0.5897
<i>Random Forest Classifier</i>		0.6153
<i>MLP Classifier</i>	<i>n_estimators</i> =300 <i>max_depth</i> =10	0.6697
<i>Logistic Regression CV</i>	<i>alpha</i> =0.01, <i>early_stopping</i> =True	0.6644
<i>Linear Regression</i>	<i>multi_class</i> ='multinomial'	0.6091

Tabela 2. Rezultati nad skupom sa uklonjenim emotikonima

Model	F1 mera
RNN	0.5298
CNN	0.5980
CNN + <i>GloVe pretrained embedding</i>	0.5979
CNN + <i>Augmented dataset</i>	0.5121
CNN + <i>Word2Vec embedding layer</i>	0.5592
Transformer + SVM	0.6623

Tabela 3. Rezultati za transformer model za tri različita skupa podataka

Skup podataka	F1 mera
Sa emotikonima	0.6628
Bez emotikona	0.6623
Zamenjeni emotikoni sa kratkim tekstom	0.6619

Tabela 4. Rezultati pretreniranog generativnog pristupa

Klasa	Preciznost	Odziv	F1 mera	Br. Primeraka
<i>Negative</i>	0.23	0.99	0.37	300
<i>Somewhat Negative</i>	0,00	0,00	0,00	300
<i>Neutral</i>	0,19	0,02	0,03	300
<i>Somewhat Positive</i>	0,00	0,00	0,00	300
<i>Positive</i>	0,78	0,41	0,54	300
Tačnost			0,28	1500
Makro prosek	0,24	0,28	0,19	1500

Tabela 5. Rezultati fine tuned verzije generativnog pristupa

Klasa	Preciznost	Odziv	F1 mera	Br. Primeraka
<i>Negative</i>	0.26	0.95	0.40	300
<i>Somewhat Negative</i>	0.00	0.00	0.00	300
<i>Neutral</i>	0.43	0.08	0.13	300
<i>Somewhat Positive</i>	0.00	0.00	0.00	300
<i>Positive</i>	0.73	0.82	0.77	300
Tačnost			0.37	1500
Makro prosek	0.28	0.37	0.26	1500

Rezultati eksperimenata pokazali su da konvolutivne i transformer mreže pružaju bolje performanse u odnosu na rekurentne mreže. Generativni pristupi pokazali su potencijal, ali su njihovi rezultati bili varijabilni i zavisili su od specifičnih postavki modela i broja epoha treniranja. Klasifikacija komentara sa srednjim ocenama predstavljala je izazov zbog njihove inherentne neodređenosti, što ukazuje na potrebu za daljim istraživanjima i unapređenjima modela.

Ovi rezultati potvrđuju da savremeni modeli dubokog učenja, naročito transformeri, mogu značajno unaprediti tačnost i pouzdanost sentiment analize u *e-commerce* recenzijama. Dalje istraživanje i optimizacija ovih modela mogli bi dodatno poboljšati rezultate, posebno u scenarijima sa nebalansiranim skupovima podataka.

5. ZAKLJUČAK

Klasifikacija korisničkih komentara prema sentimentu predstavlja složen izazov koji može znatno poboljšati interpretaciju i interakciju s korisnicima kada se adekvatno adresira. U ovom istraživanju analizirani su različiti metodološki pristupi za sentiment analizu komentara prikupljenih s različitih platformi za prodaju ženske odeće, preuzetih s veb stranice *Kaggle*. Podaci su očišćeni, obrađeni i kategorizovani u pet grupa za klasifikaciju u više klasa.

Skup podataka je podeljen na obučavajući, validacioni i testni segment. Zbog nebalansiranosti skupa podataka, korišćene su makro usrednjene metrike preciznosti, osetljivosti i F1 mere za ocenu modela.

U eksperimentima su testirani različiti pristupi:

1. Tradicionalni NLP metodi: Tekst je vektorizovan pomoću *Word2Vec*, *GloVe*, LSTM, CNN i Transformer tehnika
2. Augmentacija podataka: Korišćena za rešavanje problema nebalansiranosti skupa podataka
3. Generativni modeli: Klasifikacija pomoću *llama-7b-hf* modela, u pretreniranoj i *fine tuned* varijanti

Najbolje rezultate postigao je Transformer model sa F1 merom od 0.66, dok su konvolutivne mreže pokazale nešto slabije rezultate. Generativni model, iako nije pravljen za klasifikaciju, pokazao je sposobnost za ovu namenu. Postoji pretpostavka da generativni modeli mogu biti efikasni za klasifikaciju u kontekstu manjih skupova podataka zahvaljujući svom opštem znanju.

Ograničenja u resursima sprečila su dalja istraživanja i korišćenje modela sa većim brojem parametara. Iako su generativni modeli napredovali, klasični NLP pristupi su se pokazali efikasnim kada postoji dovoljna količina podataka za obuku.

Dalji razvoj rešenja bi se fokusirao na prikupljanje šireg seta podataka, posebno komentara s nižim ocenama, kako bi se adresirala nebalansiranost *dataset*-a i omogućila bolja generalizacija modela. Ovo bi uključivalo primenu LSTM mreža i CNN-a sa *Word2Vec* i *GloVe* vektorima, kao i usavršavanje BERT modela za detaljniju analizu konteksta i sentimenta. Razvoj raznolikijih *prompt*-ova za generativne modele i testiranje modela sa više parametara mogli bi dodatno poboljšati efikasnost.

Ovaj pristup bi omogućio modelima bolje razumevanje i interpretaciju suptilnih ili kompleksnih varijacija sentimenta, naročito u nepolarizovanim komentarima. Kroz kolaboraciju sa industrijskim partnerima i akademskim institucijama, moguće je dobiti pristup naprednijim tehnološkim resursima i znanju, što bi dodatno doprinelo preciznosti i efikasnosti modela.

6. LITERATURA

- [1] <https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews> (pristupljeno u martu 2022.)
- [2] Abien Fred M. Agarap, „Statistical Analysis on E-Commerce Reviews, with Sentiment Classification

using Bidirectional Recurrent Neural Network“, 2018.

- [3] Shuangyin Xie, „Sentiment Analysis using machine learning algorithms: online women clothin reviews“, 2019.
- [4] Manish Munikar, Sushil Shakya, Aakash Shrestha, “Fine-grained Sentiment Classification using BERT”, 2019.
- [5] Jun Xie, Bo Chen, Xinglong Gu, Fengmei Liang, Xinying Xu, “Self-Attention-Based BiLSTM Model for Short Text Fine-Grained Sentiment Classification”, 2018.
- [6] <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (pristupljeno u martu 2022.)
- [7] <https://keras.io/> (pristupljeno u martu 2022.)
- [8] <https://www.tensorflow.org/> (pristupljeno u martu 2022.)
- [9] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, Tie-Yan Liu, “MPNet: Masked and Permuted Pre-training for Language Understanding”, 2020.
- [10] <https://huggingface.co/meta-llama/Llama-2-7b-hf> (pristupljeno u februaru 2024.)
- [11] <https://spacy.io/> (pristupljeno u martu 2022.)

Kratka biografija:



Cvijetin Mladenović rođen je u Šapcu 1998. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Inteligentni sistemi odbranio je 2024.god.
kontakt: mladjenovic.cvijetin@gmail.com

**ESTIMACIJA KRVNOG PRITISKA NA OSNOVU FOTOPLETIZMOGRAMA
KORISTEĆI VEŠTAČKE NEURONSKE MREŽE****FEEDFORWARD NEURAL NETWORK APPROACH FOR BLOOD PRESSURE
ESTIMATION FROM PPG SIGNALS**

Igor Jorgovanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu se obrađuje postupak estimacije krvnog pritiska, na osnovu obeležja izvučenih iz PPG signala, pomoću veštačkih neuronskih mreža. Napravljeno je više modela neuronskih mreža sa različitim obeležjima na ulazu kako bi se našli najbitnije karakteristike PPG signala koje imaju vezu sa krvnim pritiskom. Na kraju su poređene performanse svih modela za isti set podataka za testiranje.

Ključne reči: Krvni pritisak, veštačke neuronske mreže, feedforward neuronska mreža, fotopletizmografija

Abstract – This article addresses the procedure for estimating blood pressure, based on features extracted from PPG signals, using artificial neural networks. Multiple models have been made so as to find the best combination of characteristics, of the PPG signal, that correlate to blood pressure the most. The results from all models are then directly compared.

Keywords: Blood pressure, artificial neural networks, feedforward neural network, photoplethysmography

1. UVOD

Krvni pritisak je osnovna fiziološka veličina čiji produženi abnormalni nivoi mogu izazvati razne kardiovaskularne bolesti. Svakodnevno praćenje istog kod pacijenata u dužim periodima je trenutna praksa koju primenjuju lekari kako bi se smanjili rizici i stopa smrtnosti zbog hipertenzije. Razvojem tehnologije se olakšava proces merenja, i samim tim i praćenje, krvnog pritiska, međutim, merenje krvnog pritiska jednom ili nekoliko puta dnevno još uvek ostavlja prostor za kasnu detekciju problema.

Jedan pristup neprekidnom praćenju krvnog pritiska je da se koriste signali koji mogu da se prikupljaju pasivno, npr. EKG signali ili PPG signali koji se dobijaju pomoću svetlosnih dioda, fotodetektora i prateći softver koji su svi implementirani u pametne satove. Akvizicija EKG-a ograničava pokretljivost pacijenta radi preciznog merenja, i nije pouzdano za kontinuirano praćenje [1]. Pored toga,

visoka impendansa kože uzrokuje fluktuacije bazne linije, značajan šum i, u nekim slučajevima, gubitak EKG traga. Sa druge strane, PPG uređaji rade na osnovu optički dobijenog signala koji može da detektuje promene volumena krvi u arteriji, i oni se već koriste radi detekcije nivoa kiseonika u krvi i srčanog pulsa.

2. FOTOPLETIZMOGRAFIJA

Fotopletizmografija (eng. *Photoplethysmography*, PPG) je neinvazivna tehnika koja se koristi za merenje volumenskih promena u krvi unutar perifernih krvnih sudova. Ova metoda koristi svetlosne senzore za detekciju promene u apsorpciji svetlosti, koja varira sa količinom krvi prisutnom u tkivima. Ima široku primenu u medicini za praćenje srčanog ritma, oksigenacije krvi i drugih vitalnih znakova [2]. Sa razvojem naprednih senzorskih tehnologija i nosivih uređaja, PPG postaje sve važnija u oblasti telemedicine i personalizovane zdravstvene zaštite.

2.1. Princip fotopletizmografije

Svetlost koja prolazi kroz biološko tkivo može biti apsorbovana od strane različitih supstanci, uključujući pigmente u koži, kostima, arterijskoj i venskoj krvi. Većina promena u protoku krvi se dešava u arterijama i arteriolama, ali ne i u venama. Na primer, arterije sadrže veću količinu krvi u toku sistolne faze srčanog ciklusa nego u toku dijasistolne faze [3].

PPG senzori optički detektuju promene u volumenu protoka krvi tj. promene u detektovanom intenzitetu svetlosti, u mikrovaskularno tkivu putem refleksije ili transmisije kroz tkivo.

3. METOD

U ovom radu je napravljeno pet različitih modela neuronskih mreža koji na osnovu različitih setova obeležja na ulazu estimiraju vrednost sistolnog i dijasistolnog pritiska na izlazu. PPG signali i odgovarajuće vrednosti krvnog pritiska su preuzeti iz baze podataka o kojoj se mogu videti detaljnije informacije u okviru rada [4]. Odabir obeležja koja se izvlače iz PPG signala je baziran na istraživanju [5].

NAPOMENA: Ovaj rad proistekao je iz master rada čiji mentor je bio dr Luka Mejić, docent

3.1. Obrada podataka

Podatke je, pre ulaska u neuronsku mrežu, potrebno obraditi tako što se uklone nebitne informacije, poput šuma, i izdvoje karakteristična obeležja čije vrednosti zavise od krvnog pritiska. Prosleđivanjem delova umesto celog signala neuronskim mrežama povećava preciznost mreže time što ne uzima u obzir informacije iz signala koje ne utiču na željeni izlaz.

Za ekstrakciju obeležja su pored samog PPG signala potrebni i prvi izvod PPG signala i njegov amplitudski spektar. Kako bi se precizno numerički odredio prvi izvod potrebno je filtrirati signal. Granične frekvencije filtera su empirijski određene na osnovu amplitudskog spektra PPG signala koji se filtrira. Korišćeni su *band-pass* filteri koji filtriraju i jednosmernu komponentu i više frekvencije (uglavnom preko 5Hz).

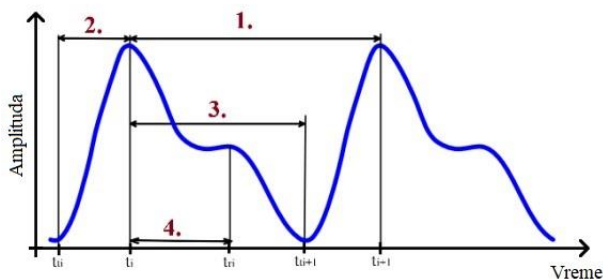
Nakon filtriranja se izračunao prvi izvod signala na osnovu kojeg se vrši ekstrakcija obeležja iz vremenskog domena, koja se prosleđuju na ulaz svih pet modela. Pored njih, na ulaze pojedinih modela se prosleđuju sledeća obeležja iz frekvencijskog domena: frekvencija prvog harmonika, odnos amplituda prva dva harmonika i vrednost amplitude prvog harmonika.

3.2. Ekstrakcija obeležja

Neki od najbitnijih parametara iz PPG signala koji se koriste u algoritmima za estimaciju krvnog pritiska su:

- Srčani ciklus (*eng. Cardiac period*)
- Vreme uspona sistolnog pritiska (*eng. Systolic upstroke time*)
- Dijastolno vreme (*eng. Diastolic time*)
- RPTT (*eng. Reflected Pulse Transit Time*)

Srčani ciklus je trajanje jednog otkucaja srca, vreme uspona sistolnog pritiska je vreme potrebno pulsu da dostigne maksimalan pik, dok je dijastolno vreme vreme kada je srce u stanju relaksacije. PTT se odnosi na vreme koje je potrebno da pritisni talas putuje između dva proizvoljno odabrana arterijska mesta, i u zavisnosti od kog do kog mesta se meri, postoje različiti tipovi PTT-a. RPTT je podtip PTT-a gde se meri vreme između detekcije prvog pika i detekcije reflektovanog pika. Spomenuta obeležja su prikazani na slici ispod.



Slika 1. Vremenska obeležja PPG signala – 1. Srčani ciklus, 2. Vreme uspona sist. pritiska, 3. Dijastolno vreme i 4. RPTT

Kako bi se dobile tačne vrednosti vrhova i “dolina” u signal (lokalni i globalni ekstremi) potrebno je izračunati prvi izvod i zatim pronaći njegove preseke sa nulom, ti trenuci odgovaraju ekstremima u originalnom signalu. Tako se dobijaju tri od četiri trenutaka koja su neophodna

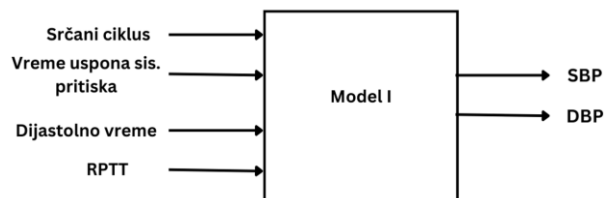
za računanje vrednosti vremenskih obeležja. Poslednji trenutak, vreme pojave reflektovanog pika, se nalazi na osnovu lokalnog maksimuma u prvom izvodu koji se nalazi između drugog i trećeg preseka sa nulom u prvoj periodu izvoda.

Pored vremenskih karakteristika, naučna istraživanja u poslednjih nekoliko godina ukazuju na korelaciju frekvencijskih osobina PPG signala i krvnog pritiska [5]. Primenom *Matlab*-ovog algoritma za detekciju pikova (*FindPeaks*) nad amplitudskim spektrima se dobiju vrednosti amplituda i frekvencija prva dva harmonika svakog PPG signala.

Ukoliko jedan pacijent ima više segmenata PPG signala koji su uspešno prošli kroz obradu signala, krajnje vrednosti obeležja su srednje vrednosti svih segmenata.

3.3. Modeli

Prvi model estimira krvni pritisak isključivo na osnovu obeležja iz vremenskog domena. Na ulaz se prosleđuju četiri vremenska obeležja koja su nabrojana na početku poglavlja 3.2. Prvi model je skiciran na slici ispod.



Slika 2. Prvi model

Drugi model pored ta 4 vremenska obeležja prima na ulazu vrednost frekvencije prvog harmonika filtriranog PPG signala. Cilj je da se poređenjem rezultata ovog modela sa rezultatima prvog modela ukaže na uticaj frekvencije rada srca na krvni pritisak, ukoliko je dovoljno značajan da se kvantifikuje.

Treći model, nalik drugom, sadrži sva 4 obeležja iz vremenskog domena i jedno obeležje iz frekvencijskog domena: odnos amplituda prva dva harmonika. Ovaj model uzima u obzir pretpostavku da vrednost amplituda prva dva harmonika ima korelaciju sa krvnim pritiskom [5], jedino umesto da se posmatraju vrednosti, posmatra se njihov odnos.

Četvrti model je kombinacija prethodna dva, gde se na ulaz prosleđuju svakako sva 4 obeležja iz vremenskog domena, frekvencija prvog harmonika i odnos amplituda prva dva harmonika. Ideja ovog modela je da se, ukoliko modeli sa samo frekvencijom ili odnosom amplituda na ulazu ne uspeju da ostvare vezu sa krvnim pritiskom, koriste oba tipa informacije dobijenih iz frekvencijskog domena zajedno sa obeležjima iz vremenskog domena.

Poslednjem modelu se prosleđuju na ulaz sva obeležja kao i prethodnom modelu i pored njih se dodatno prosleđuje i vrednost amplitude prvog harmonika. Na slici ispod je skiciran peti model.

3.4. Konfiguracije veštačkih neuronskih mreža

Kako bi se izbegli problemi poput preobučavanja i nedovoljne obuke mreže napravljeno je nekoliko konfiguracija za svaki model sa različitom širinom i/ili dubinom mreže. Napravljene su konfiguracije sa jednim, dva i tri skrivena sloja sa po 6, 8, 12, 16, 24 i 32 neurona po skrivenom sloju kako bi se pokrio što veći broj slučajeva.

Što se tiče aktivacionih funkcija, za sve kombinacije postoje 3 različite verzije u kojima je aktivaciona funkcija sigmoidna, ispravljena linearna funkcija i tangens hiperbolik redom. Pored toga, eksperimentisano je i sa različitim stopama učenja u vrednostima: 0.01, 0.001 i 0.0001. Promene stope učenja su pratile i promena broja epoha koja se prostirala od 100 do 300, kako bi se mreža sigurna stigla obučiti.

3.5. Statistička analiza i procena performansi modela

Nakon što su svi modeli obučeni i testirani, njihove performanse će se prvobitno proceniti i uporediti na osnovu srednje vrednosti njihove relativne i apsolutne greške, maksimalne vrednosti apsolutne greške i distribucije apsolutnih grešaka. Međutim, zbog velikog broja modela koji su napravljeni, dolazi do čestog preklapanja prethodno spomenutih kriterijuma procene performanse. Zbog toga je izvršena statistička metoda za poređenje dobijenih rezultata svih konfiguracija kako bi se utvrdilo da li postoji statistički značajna razlika između modela, na osnovu čega se mogu dokazati odnosi koji su pretpostavljeni poređenjem grešaka, ukoliko postoje razlike između performansi.

Za podatke koji su prošli test normalnosti upotrebljena je *Repeated Measures One-Way ANOVA* za inicijalno poređenje i ukoliko je dobijena *p-vrednost* bila manja od 0.05 vršila se *post-hoc* analiza zajedno sa *Bonferroni korekcijom* za dobijanje finalne *p-vrednosti*. Za podatke koji nisu prošli test normalnosti upotrebljena je *Friedmanova ANOVA* za poređenje.

4. REZULTATI I DISKUSIJA

U ovom poglavlju su prikazane tabelama srednje vrednosti relativnih i apsolutnih greške svih modela, maksimalne vrednosti apsolutne greške i distribucija apsolutnih grešaka koje ti modeli prave. Prikazani su i rezultati poređenja performansi modela pomoću statističke analize i diskutovani ti rezultati.

Što se tiče prvog i trećeg modela, koji su imali više od jedne konfiguracije, statistička analiza je pokazala da nema značajne razlike između tih konfiguracija, tako da su za dalje rezultate izabrane konfiguracije sa najmanjim relativnim greškama.

4.1. Relativne i apsolutne greške modela

U tabeli ispod su prikazane brojne metrike po kojima se mogu porediti performanse svih modela, koje su spomenute u prethodnom poglavlju. Pored njih, u nastavku ovog poglavlja je prikazana i tabela sa brojem estimacija sa apsolutnim odstupanjima manjim ili jednakim 5, 10 i 15 mmHg u odnosu na ukupan broj estimacija, u procentima, što je metrika koje će se koristiti za klasifikaciju modela.

Tabela 1. Performanse modela pri estimaciji sistolnog pritiska

Sistolni pritisak			
Model	Srednja relativna greška [%]	Srednja apsolutna greška [mmHg]	Maksimalna apsolutna greška [mmHg]
Model I	9.26	11.50	34.01
Model II	9.57	11.91	35.10
Model III	8.62	10.50	30.90
Model IV	8.06	9.97	33.66
Model V	9.78	12.01	35.97

Tabela 2. Performanse modela pri estimaciji dijastolnog pritiska

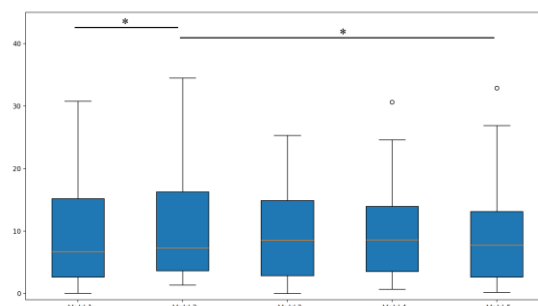
Dijastolni pritisak			
Model	Srednja relativna greška [%]	Srednja apsolutna greška [mmHg]	Maksimalna apsolutna greška [mmHg]
Model I	10.55	6.84	19.08
Model II	11.63	7.51	21.39
Model III	9.11	5.85	14.87
Model IV	10.28	6.56	16.23
Model V	9.99	6.41	17.42

Na tabelama 1. i 2. se može videti da je teško okarakterisati bilo koji model kao bolji ili gori od bilo kog drugog modela na osnovu datih kriterijuma zbog njihovog preklapanja. U sledećoj tabeli su prikazani rezultati statističke analize koji su pokazali statističku značajnu razliku u odnosu performansi svih modela.

Tabela 3. Rezultati statističke analize performansi modela sa *p-vrednostima* manjim od 0.05

Modeli koji se poredе	<i>p-vrednost</i> za sistolni pritisak	<i>p-vrednost</i> za dijastolni pritisak
1 i 2	0.670	0.033*
2 i 5	0.670	0.033*

Na osnovu rezultata poređenja može se zaključiti da nema statistički značajne razlike između svih 5 modela što se tiče estimacije sistolnog pritiska. Međutim, za estimaciju dijastolnog pritiska je pokazano da postoji razlika između prvog i drugog modela, i između drugog i petog modela, pošto su sve te *p-vrednosti* manje od 0.05. Ti rezultati zajedno sa slikom 4. pokazuju da su prvi i peti model bolji od drugog pri estimaciji dijastolnog pritiska.



Slika 4. Relativne greške – DBP

Treći i četvrti model imaju međusobno veoma slične performanse pri estimaciji dijastolnog pritiska i njihov odnos sa ostalim modelima ne može da se zaključi jer imaju veću srednju vrednost relativne greške ali manju standardnu devijaciju.

U sledećoj tabeli su prikazani prethodno spomenuti procenti estimacija sa određenim apsolutnim odstupanjima za dijastolni pritisak i odgovarajuće klase koje modeli uspevaju da dostignu.

Tabela 4. *Distribucija apsolutnih grešaka modela pri estimaciji DBP*

Model	% estimacija aps. greška ≤ 5 mmHg	% estimacija aps. greška ≤ 10 mmHg	% estimacija aps. greška ≤ 15 mmHg
Model I	50.0 %	77.3 %	100 %
Model II	50.0 %	68.2 %	81.8 %
Model III	45.4 %	78.3 %	95.6 %
Model IV	40.1 %	77.3 %	95.4 %
Model V	50.0 %	77.3 %	86.4 %

Na osnovu standarda klasifikacije uređaja za merenje krvnog pritiska BHS [7], svi modeli dostižu klasu C, dok prvi model dostiže čak klasu B, što znači da je moguća primena u medicini.

5. ZAKLJUČAK

Kako broj ljudi sa povišenim krvnim pritiskom i/ili drugim kardiovaskularnim problemima raste, tako je i potreba za konstantnim praćenjem krvnog pritiska veća, kako bi se na vreme uočio problem i samim tim znatno smanjio rizik od srčanih komplikacija ili smrti. Međutim, trenutna tehnologija ne omogućava nivo konzistentnosti praćenja krvnog pritiska koji je potreban kako bi se sa potpunom sigurnošću uočila abnormalnost na vreme, a aktivno merenje ume da uzrokuje povećan pritisak kod nekih ljudi, fenomen koji je poznat kao efekat belog mantila, što dovodi do neispravno očitanih vrednosti. Zbog toga je potrebna tehnologija koja će pasivno da prati stanje krvnog pritiska u toku dana i, ukoliko dođe do abnormalnih nivoa, da obavesti korisnika kako bi mogao na vreme da odreaguje ako ima potrebe.

Implementacija u vidu pametnog sata koji prikuplja PPG signal već postoji, i ta informacija može da se iskoristi kako bi se procenila vrednost krvnog pritiska uz pomoć veštačkih neuronskih mreža.

Kao što je prikazano u prethodnom poglavlju, svi modeli estimiraju krvni pritisak sa sveukupnom preciznošću od oko 89-92%, s tim da su u testovima bili prisutni podaci od ljudi sa znatno povišenim krvnim pritiskom u nedovoljnoj količini kako bi se model obučio za takve slučajeve u potpunosti. Većina rezultata kod ljudi sa normalnim krvnim pritiskom su estimirani sa relativnom greškom od 5-6%, što pokazuje da ovakav pristup može da nađe primenu u svakodnevnom praćenju kardiovaskularnih parametara kod sportista ili ljudi sa kardiovaskularnim problemima. Po standardnom protokolu klasifikacije uređaja za merenje krvnog pritiska, jedan model je klasifikovan klasom B, što znači da može da se koristi u kliničkim istraživanjima i u medicini, iako samo za dijastolni pritisak. Ovaj metod

klasifikacije je namenjen za uređaje koji direktno mere pritisak i samim tim što je metod estimacije uspeo da pokaže ovakav rezultat znači da, uz unapređenja, može se napraviti model koji zadovoljava uslove za upotrebu u medicini.

S obzirom da su modeli u ovom radu obučeni podacima pacijenata u stanju mirovanja, model je u stanju da estмира krvni pritisak takođe u stanju mirovanja. Moguća unapređenja uključuju proširivanje baze podataka za obuku kako bi estimirane vrednosti bile još vernije pravim vrednostima, i proširivanje baze podataka tako da uključi i podatke ljudi van stanja mirovanja, npr. u toku treninga, trčanja, plivanja i slično. Kada bi se mreža obučila sa podacima u različitim stanjima čoveka, primena bi mogla da se proširi na praćenje krvnog pritiska u toku treninga, što može biti veoma bitna informacija profesionalnim sportistima.

6. LITERATURA

- [1] M. Sharma, K. Barbosa, V. Ho, D. Griggs, T. Ghirmai, S. K. Krishnan, T. K. Hsiai, J.-C. Chiao, H. Cao, "Cuff-Less and Continuous Blood Pressure Monitoring: A Methodological Review," *Wearable Technologies*, 2017.
- [2] T. Tamura, Y. Maeda, M. Sekine, M. Yoshida, "Wearable Photoplethysmographic Sensors—Past and Present." *Electronics* 2014.
- [3] A. Poliński, J. Kot, A. Meresta, "Analysis of correlation between heart rate and blood pressure" *Federated Conference on Computer Science and Information Systems*, 2011.
- [4] Y. Liang, Z. Chen, G. Liu, M. Elgendi, "A new, short-recorded photoplethysmogram dataset for blood pressure monitoring in China," *Scientific Data (Sci Data)*, 2018.
- [5] J. H.-S. Wang, M.-H. Yeh, P. C.-P. Chao, T.-Y. Tu, Y.-H. Kao, R. Pandey, "A fast digital chip implementing a real-time noise-resistant algorithm for estimating blood pressure using a non-invasive, cuffless PPG sensor" *Microsystem Technologies*, 2020.
- [6] E. O'Brien, B. Waeber, G. Parati, J. Staessen, M.G. Myers, "Blood pressure measuring devices: Recommendations of the European Society of Hypertension." *BMJ*, 2001.

Kratka biografija:



Igor Jorgovanović rođen je u Novom Sadu 2000. godine. Diplomski rad na temu –*Modelovanje rada kalcinatora pomoću veštačkih neuronskih mreža*- iz oblasti Elektrotehnike i računarstva odbranio je na Fakultetu tehničkih nauka 2023. godine.

Kontakt: igor.jorgovanovic@gmail.com

**НАМЕНСКИ ЈЕЗИК И ОКРУЖЕЊЕ ЗАСНОВАНО НА ИНЖЕЊЕРСТВУ ВОЂЕНОМ
МОДЕЛИМА ЗА ОПИСИВАЊЕ И ГЕНЕРИСАЊЕ БАСНИ****A DOMAIN-SPECIFIC LANGUAGE AND MODEL-DRIVEN ENVIRONMENT FOR
FABLES SPECIFICATION AND GENERATION**

Сандра Бркић, Факултет техничких наука, Нови Сад

**Област – ПРИМЕЊЕНЕ РАЧУНАРСКЕ НАУКЕ И
ИНФОРМАТИКА**

Кратак садржај – У овом раду описан је графички наменски језик за моделовање басни за децу. Описане су технологије и окружење које је коришћено за имплементацију и дефинисање апстрактне синтаксе, са коришћеним концептима и значењима. Након тога дефинисана је конкретна синтакса графичког типа. Детаљно је описан коришћен алат и посупак креирања елемената и њихових алата. Приказан је и пример употребе свега дефинисаног у оквиру коришћених алата као и неопходна мапирања. Имплементирана су три генератора и начин њиховог функционисања, и то редом за сликовницу, текст и трећи који је комбинација претходна два. За реализацију рада коришћени су Eclipse Modelling Framework, Sirius и Xtend.

Кључне речи: басне, наменски језици, метамодел, развој софтвера базиран на моделу

Abstract – The paper describes a graphical domain-specific language for modeling fables for children. Used concepts and their meaning are presented, followed by concrete syntax and their graphic representations. The abstract syntax of this language, as well as detailed description of each of the concepts are described. The tools used and the process of creating elements and their tools are described in detail. A case study is presented. Three generators are implemented, for the picture book, the text, and the third one, which is a combination of the previous two. Technologies that are used are Eclipse Modelling Framework, Sirius and Xtend.

Keywords: fables, Domain-Specific Languages, metamodel, Model-Driven Software Development

1. УВОД

Басне су кратке приче за децу у којима су главни актери животиње. За њих су у главном везане људске особине као што је на пример лукавост, али и активности. Њихово понашање је типично људско.

Идеја је да прикажу карактерне особине, ситуације и односе међу људима, али тако да буду разумљиве деци у раном узрасту где је потребно да буду заинте-

ресовани необичностима, а животиње су увек интересантна тема.

Креатори оваквих прича често се сусрећу са проблемом креирања довољне количине различитих басни са сличним или разноликим улогама. Мотивација овог рада је да реши проблем смишљања што више оваквих прича за кратко време. Деца су константно заинтересована за још неку причу са њима драгим ликовима. Додатни проблем који се јавља је и културолошке природе, а тиче се тога да су неке животиње везане за одређену особине, док је на другом крају света то потпуно другачији случај. Ово доводи до креирања потпуно нове приче, а коришћењем овог наменског језика било би могуће само изменити тип животиње код једног лика.

Циљ рада је омогућење брже израде басни, а у случају недостатка маште или времена креатора, овај програмски језик намењен је ефикасном прављењу колекција басни са одређеним ликовима који би у њима учествовали. Додатно, овакав језик за циљ и могућност лаке модификације постојећих прича, као и креирање нових, а сличних постојећим. Он нуди брзо креирање, уз интуитивну употребу слика и предмета и уноса описа, како би текст касније био изгенерисан на основу њих.

Додатна предност настанка оваквог језика је и то, што би поред професионалних креатора, овако нешто могло да послужи и као додатна занимација деци одређеног узраста, која би била довољно велика да схвате коришћење доступних сличица.

Рад поред Увода и Закључка поседује још четири поглавља, од којих се у поглављу 2 говори о сличним решењима, поглављу три теоријским основама, док четврто представља коришћене технологије. У оквиру петог поглавља описан је процес имплементације наменског језика, а затим и самих генератора у поглављу шест.

**2. ПРЕГЛЕД ПОСТОЈЕЋЕГ СТАЊА У
ОБЛАСТИ**

Модел [1] представља издвајање битног од небитног, било да је у питању целокупни посматрани систем или само неки његов део. У току развоја софтвера они су се показали као јако корисни за пројектовање и даљу имплементацију. Модели се могу означити и као

НАПОМЕНА

Овај рад произтекао је из мастер рада чији ментор је био др Милан Челиковић, доцент.

поједностављење за разумевање неког проблема, или приказ потенцијалног решења.

Данас, најчешће употребљавани језик за моделовање је Unified Modeling Language (UML) [2], а могу се употребити на разне начине. Они могу послужити и као додатни начин комуникације између чланова тима, или било које заинтересоване стране, како би се на једноставнији начин представило функционисање система. За решавање проблема којим се бави овај рад до сад су сва постојећа решења заснована на AI(Artificial Intelligence) технологијама и не односе се на креирање комплетне приче већ на случајном избору имена и животиња за ликове или наслов, али не и целе приче.

3. ТЕОРИЈСКЕ ОСНОВЕ НАМЕНСКОГ ЈЕЗИКА И ТРАНСФОРМАЦИЈЕ МОДЕЛА

Језици наменски за домен (енгл. *Domain Specific Languages – DSL*) [1] су језици који дизајнирани тако да пруже решење за проблеме који припадају неком одређеном домену или области. Прилагођени су употреби у оквиру одређеног подручја, за разлику од општих програмских језика. Уколико су ови језици намењени за моделовање, онда се називају доменски специфични језици за моделовање.

Главне особине наменских програмских језика су:

- **Експресивност** — обликовани су тако да на најбољи могући начин искажу проблеме чијем решавању у одређеној области треба да допринесу.
- **Продуктивност** — употребом оваквих језика постиже се већа брзина у остваривању жељеног циља него употребом језика опште намене.
- **Поновна употребљивост** — имплементирано решење помоћу оваквих језика може се искористити поново.
- **Оптимизација** — пружају услуге оптимизације за специфичне задатке унутар домена ком су намењене.

Структура наменског језика прилагођена је домену за чију је употребу и намењена.

Основне компоненте наменског програмског језика су следеће:

1. **апстрактна синтакса** — служи да опише структуру језика и разумевање основних концепата који ће бити употребљавани. Треба да представи све неопходне компоненте у систему и проблему који треба језиком да се разреши, као и њихове могуће комбинације, везе и различите типове веза.
2. **конкретна синтакса** — представља репрезентацију језика моделовања. Ово је компонента која је намењена практичној употреби и која ће бити доступна крајњим корисницима. Постоје два типа ове синтаксе, а то су:
 - 1) **текстуална** — подаци се кодирају низом карактера, што је заступљено у програмским језицима и шире је распрострањено.

- 2) **графичка** — подаци се кодирају графичким елементима, њиховим распоредом, а могу се појавити и неки текстуални елементи.

3. **семантика** — ближе описује значење елемената који су дефинисани и начине на које их је могуће комбиновати и повезивати

Трансформација модела подразумева превођење из једног облика у други. Постоје два таква типа, а то су Model to Model(M2M) и Model to Text(M2T), који се користе зависно од тога шта је потребно урадити са моделима.

4. ОПИС КОРИШЋЕНИХ ТЕХНОЛОГИЈА

Синтакса која је одабрана је графичког типа на основу метамодела, а трансформација се ради из модела у текст, односно M2T.

4.1. Eclipse Modelling Framework

Eclipse Modeling Framework (EMF) је главна технологија коју нуди Eclipse за моделовање, креирање метамодела. За то постоји посебни језик метамоделовања који се назива Ecoge. Такође у оквиру овог алата, постоји и могућност за креирање динамичке инстанце, као и за генерисање кода. Могуће је и изгенерисати неке друге алате који ће бити у даљој употреби за развој.

4.2. OCL

Object Constraint Language [4], или скраћено OCL је језик који служи за дефинисање додатних ограничења. Разлог за то је постојање ограничења која је потребно дефинисати у наменском језику, али је то због оскудности алата немогуће на нивоу метамодела, тако да је ово језик за дефинисање ограничења објеката у буквалном преводу.

Ова правила потребно је дефинисати на нивоу апстрактне синтаксе и у главном се примењују на класе, атрибуте, регеренце или односе између неких дефинисаних класа.

4.3. Sirius

Sirius [3] је пројекат Eclipse-а који нуди креирање едитора за графички језик коришћењем EMF – Eclipse технологије за моделовање. Коришћење овог пројекта првенствено је намењено за графички приказ и управљање моделом, а примену налази у графичкој репрезентацији објеката дефинисаних апстрактном синтаксом. Односно, служи за дефинисање конкретне графичке синтаксе неког програмског језика.

Алат обезбеђује спецификацију графичке синтаксе у облику неких графичких елемената. У оквиру Sirius-а није омогућено генерисање кода. Што се тиче репрезентација оне могу бити прилагођене потребама, различитим додавањем иконица или слика, а могу се користити и предефинисани облици који долазе са алатом. Оно што Sirius обезбеђује је могућност дефинисања едитора помоћу дијаграма, интегрисање апликација као и прилагођавање изгледа графичких елемената. За корисника нуди интуитивне едиторе и синхронизацију међу различитим едиторима.

4.4 Xtend

Xtend је програмски језик који је настао као алтернатива за Java програмере. Он је језик вишег нивоа и може да се извршава на Java виртуелној машини. Његова главна намена је да олакша развој Java апликација и да пружи лакшу и читљивију синтаксу од њене. Што се тиче извршавања, XTend код се преводи у чист Java код, даље се компајлира у java бајткод што на крају доводи до тога да је његова, и употреба Java програмског језика скоро иста. Битне карактеристике Xtend-а су краћа синтакса, функционални стил програмирања са проширеним могућностима и интероперабилност са Java-ом.

5. НАМЕНСКИ ЈЕЗИК

Наменски језик је језик који није језик широке употребе. Његова намена јесте да реши конкретан проблем и буде понуђена циљној групи корисника која се тим проблемом бави.

5.1. Апстрактна синтакса

Коришћена технологија за дефинисање апстрактне синтаксе наменског језика је EMF.

За сваки концепт биће дат опис, са називом, обележјима, семантиком као и разлозима за употребу. На слици 1 приказан је изглед дефинисаног метамодела. Коренски елемент назива се колекција басни (KolekcijaBasni), која представља сет басни које се заснивају на истој базичној причи, али могу имати исте учеснике.

Што се тиче коренског концепта за њега је битно дефинисати само назив, даље свака од басни која припада сету поседује свој наслов (Naslov), што је други битни концепт. У току сваког наслова постоји више сцена (Scena), које ће касније бити мапирани на пасусе у басни. Свака сцена, односно пасус, садржи ток елемената (Element) који могу бити радња (Radnja) или нека врста разговора (Dijalog). Додатно за концепт радње могуће је везати и концепте који се тичу времена (Vreme), као и места (Mesto) где се радња дешава.

5.2. OCL ограничења

У систему постоје ограничења која нису могла бити имплементирана на нивоу метамодела, него је то

морало да се уради на други начин. То је урађено помоћу OCL правила. Пошто се изабрани домен за наменски језик тиче басни, ту постоји велика слобода, тако да оваквих случајева није било много.

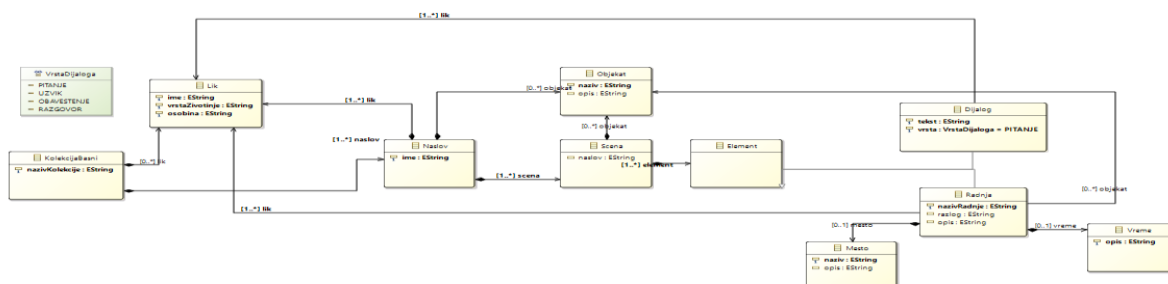
1. Оно што је путем OCL-а додатно дефинисано је да у оквиру колекције не могу постојати два Naslov-a, односно две басне с истим насловом.
2. Због случаја да сцене настају једна за другом, додато је правило и на самом концепту Scena, које се тиче провере јединствености њеног наслова.
3. Концепт Dijalog има посебно ограничење које се тиче броја учесника у разговору. Ограничење је такво да се он може водити између највише два учесника.

5.3. Конкретна синтакса

У оквиру Viewpoint-а предефинисано постоји један слој. Могуће је додати и више њих, али је за потребе овог језика био довољан један. На нивоу овог слоја могуће је дефинисати елементе као што су чворови и контејнери. Оно што је потребно урадити како би све функционисало како треба и да би се у пракси објекат појавио унутар свог контејнера је да се он дефинише угњеждено у контејнер. На овом нивоу потребно је и дефинисати све ивичне елементе који ће у графичкој синтакси бити репрезентовани као веза између нека два елемента.

Наредни корак био би дефинисање алата. За почетак је потребно додати секцију Section у оквиру овог Viewpoint-а и доделити јој неки назив, овде је то Alat. На овом нивоу потребно је за сваки претходно дефинисан елемент одабрати New Element Crea затим да ли је то чвор, контејнер или ивица.

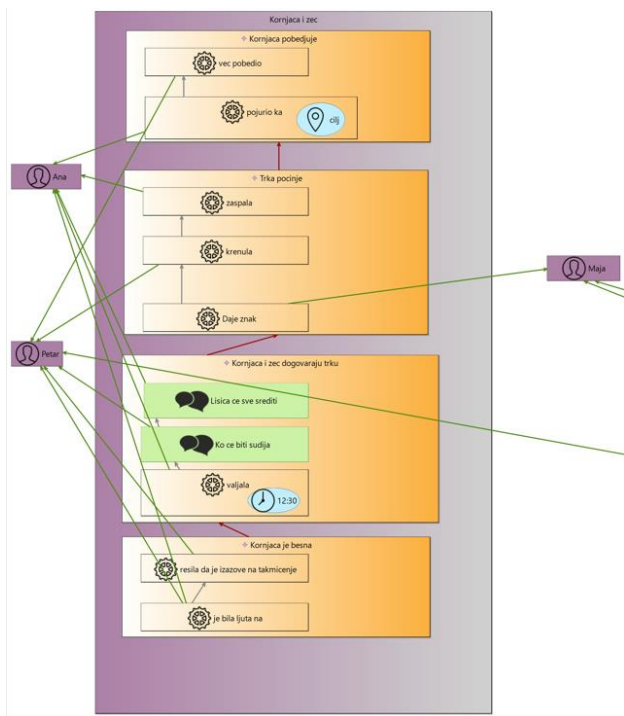
Иконице одабране за презентацију акција креирања нових инстанци горе побројаних концепата аналогне су онима којима се и репрезентују. С тим да су додате иконице и за контејнере за које у оквиру графичке представе није било потребе за приказивањем, као што су Naslov и Scena. На слици 2 приказан је пример написан дефинисаном конкретном синтаксом.



Слика 1 - Апстрактна синтакса

5.4. Пример употребе наменског језика

Пример са слике 2 приказује приче у оквиру које постоје ликови Маја, Ана и Петар. Представљена је једна приша из целокупне колекције која садржи четири сцене. У оквиру прве сцене налазе се две радње а за једну је дефинисано и место, друге три, док је трећа сцена испињена дијалозима и радњом з акоју је дефинисано време.



Слика 2 - Пример употребе

6. ОПИС ПРОЦЕСА ГЕНЕРИСАЊА ФАЈЛОВА

За генерисање фајлова који су у овом случају били Hypertext Markup Language (HTML) фајлови морало се извршити повезивање пројекта са Xtend форматом који је у могућности да препозна дефинисане концепте и приступи њиховим пољима. Покривено је генерисање сликовнице за млађе узрасте као и текстуалних прича за оне старије. Ово је извршено имплементацијом три генератора од којих је трећи комбинација претходна два.

Генератор функционише тако што пролази кроз дефинисане концепте и ради оно што се за њих специфицира. Потребно је конкретну синтаксу повезати са апстрактном да би генератор могао да их препознаје и генерише текст сходно типу на који наилази.

6.1. BasneGenerator имплементација

Прво се FOR петљом обезбедио пролазак кроз све наслове који у оквиру колекције постоје и за сваку је он исписан. За сваки наслов обезбеђен је пролаз кроз сцене које му припадају и такође исписан њихов назив. У овом случају није било потребе за приказом слике. Након тога било је потребно проћи кроз све елементе који се налазе у сцени, а који су могли бити радње и дијалози.

У овом случају било је потребно комплексније обрадити различите типове елемената и извршити потребне конверзије уколико су они припадали неком супертипу.

6.2. Print имплементација

Трећи генератор представља комбинацију претходна два. Његова сврха је да омогући раздвајање тако да свака прича има засебну нумерисану страницу у оквиру које ће бити приказан текст поред слике за коју је везан.

Измене које су биле потребне тичу се генерисања фолдера за колекцију и унутар њега засебних страница за приче. Што се тиче генерисања текста и слика, коришћен је исти приступ као у претходна два генератора.

7. ЗАКЉУЧАК

Мотивација за израду оваквог језика јесте лакоћа креирања велике количине прича тако да имају исте ликове, како би се на основу тога створила колекција. Овим је деци могуће пружити већу количину смисленог садржаја на лак начин.

Овакав наменски језик могао би бити унапређен на много начина. У наставку су наведена три примера. Први од њих је могућност листања сликовнице где би било потребно креирати HTML странице са могућношћу листања и позиционирања на одговарајућу страницу. Додатно било би добро да буде омогућено генерисање pdf-а како би се садржај могао експортирати и у физички облик. На крају, као решење проблема који се јавио, а тиче се измена језичких конструкција ка падежима предлог је увезивање са неким од AI алата који би изгенерисани “неправилни” текст средило у одговарајућу форму.

8. ЛИТЕРАТУРА

- [1] Brambilla, M., Cabot, J., & Wimmer, M., “Model-driven software engineering in practice”, *Morgan Claypool Publishers*, 2012
- [2] <https://www.uml.org/> (приступљено у јулу 2024.)
- [3] Sirius - <https://www.eclipse.org/sirius/> (приступљено у јулу 2024.)
- [4] <https://www.omg.org/spec/OCL/2.4/About-OCL> (приступљено у јулу 2024.)

Кратка биографија:



Сандра Бркић рођена је 9. маја 2000. године у Београду. Завршила је природно-математички смер „Шабачке гимназије“. Школске 2019/2020 уписује Факултет техничких наука у Новом Саду, на студијском програму Рачунарство и аутоматика. Основне студије завршава 2023 године, када уписује мастер програм Примењене рачунарске науке – Инжењеринг информационих система.

НАМЕНСКИ ЈЕЗИК И ОКРУЖЕЊЕ ЗАСНОВАНО НА ИНЖЕЊЕРСТВУ ВОЂЕНОМ МОДЕЛИМА ЗА ПРИСТУП РАЗЛИЧИТИМ ВЕКТОРСКИМ БАЗАМА ПОДАТАКА**DOMAIN-SPECIFIC LANGUAGE AND MODEL-DRIVEN ENVIRONMENT FOR ACCESSING DIVERSE VECTOR DATABASES**Елена Акик, *Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИЧКО И РАЧУНАРСКО ИНЖЕЊЕРСТВО**

Кратак садржај – У овом раду представљено је окружење за униформни приступ различитим векторским базама података, формирано кроз моделом вођен развој софтвера. У оквиру окружења подржано је повезивање са циљном базом података, дефинисање структура података, манипулација објектима базе података, постављање упита и векторска претрага сличности. Централни елемент окружења је наменски језик *uniVEC*, са текстуалном синтаксом. Важан део окружења су генератори кода, који трансформишу моделе креиране помоћу језика *uniVEC* у извршив програмски код. Генератори кода генеришу одговарајуће Python скрипте за подршку рада са одабраном циљном векторском базом података, као што су: *Pinecone*, *Milvus*, *Chroma*, *Weaviate* или *Qdrant*. Предложено окружење има циљ да поједностави рад са векторским базама података, смањи време потребно за учење концепата из домена и обезбеди миграције између различитих векторских база података.

Кључне речи: векторске базе података, наменски језици, развој софтвера вођен моделима, масивни подаци, машинско учење

Abstract – In this paper, a model-driven environment for uniform access to various vector databases is presented. The environment supports connection to the target vector database, definition of data structures, manipulation of database objects, querying, and vector similarity search. The central element of the environment is the domain-specific language *uniVEC*, with a textual syntax. An important part of the environment is code generators, that transform models created by using *uniVEC* into executable programming code. The code generators generate Python scripts to support work with one of the chosen vector databases, such as: *Pinecone*, *Milvus*, *Chroma*, *Weaviate*, or *Qdrant*. The proposed environment aims to simplify work with vector databases, reduce learning time and support migrations between different vector databases.

Keywords: Vector Database, Domain-Specific Language, Model-Driven Software Development, Big Data, Machine Learning

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији је ментор био др Милан Челиковић, доцент

1. УВОД

Масивни подаци (енгл. *Big Data*) мењају начин живота и рада људи и кључни су покретач развоја у различитим областима. У дигиталном свету, количина генерисаних података расте великим интензитетом. Масивни подаци користе се за генерисање знања и боље доношења одлука [1]. Пут од генерисаног знања до формирање одлуке је комплексан и дуготрајан, а кључну улогу имају предикције, које уочавају обрасце, на основу којих је могуће формирати одлуке.

Потенцијални изазови приликом управљања великим количинама података јесу где складиштити и како обрадити податке. Додатно, како људи нису у стању ефикасно обрадити масивне податке и на основу њих генерисати знање, као подршка, деценијама уназад, коришћена је вештачка интелигенција (енгл. *Artificial Intelligence*). Руковање масивним подацима може бити изазовно са становишта меморијског простора за складиштења података, као и перформанси читања и ажурирања података.

При одабиру начина складиштења структурираних и неструктурираних података, чест избор су векторске базе података, погодне за складиштење података датих типова, али и за семантичку претрагу и индексирање података, обезбеђење скалабилности система и добрих перформанси приликом извршавања упита над подацима у бази података [2].

Упркос предностима које пружају, сложеност употребе векторских база података један је од кључних изазова. Примарни изазов представља овладавање концептима векторских база података, због њихове сложености и разноврсности. Тренутно не постоји униформни језик за рад са различитим векторским базама података. Због тога, крајњи корисници морају овладати техникама приступа и начинима интерпретације жељених акција над базом података, специјализованим за приступ одабраној векторској бази података, реализујући их кроз програмски код.

У овом раду, предложено је окружење за униформну интеракцију са различитим векторским базама података, где је кључни елемент наменски језик *uniVec*. Помоћу језика *uniVec* могуће је креирати моделе за приступ векторским базама података у виду скрипти, док креиране моделе генератори кода трансформишу у извршив програмски код одабране векторске базе података. Предложено окружење има за циљ да обезбеди: поједностављење рада са циљном

векторском базом података кроз стандардизацију приступа; смањено време за учење концепата из домена и тестирање различитих векторских база података, као и редуковану сложеност приликом миграција између различитих векторских база података.

Након увода, у поглављу 2 дат је теоријски осврт на развој софтвера вођен моделима и наменске језике. Поглавље 3 усмерено је на преглед начина рада векторских база података. У поглављу 4 описан је наменски језик *uniVec*, а у поглављу 5 приказани су генератори кода. У поглављу 6 сумиран је општи допринос имплементираног окружења..

2. ТЕОРИЈСКЕ ОСНОВЕ РАЗВОЈА СОФТВЕРА ВОЂЕНОГ МОДЕЛИМА И НАМЕНСКИХ ЈЕЗИКА

Развој вођен моделима (енгл. *Model-Driven Development – MDD*) представља парадигму која користи моделе као примарне артефакте процеса развоја система, укључујући аутоматизовано или полу-аутоматизовано генерисање имплементационих артефакта из апстрактних модела [3]. Начин креирања и управљања моделима зависи од конкретних потреба које треба да разреши. Развој софтвера вођен моделима (енгл. *Model-Driven Software Development – MDS*) представља конкретизацију парадигме *MDD*, где су модели искоришћени за развој софтверских система.

Језгро парадигме *MDS* јесу језици који омогућавају крајњим корисницима да концептуализују стварност у експлицитном облику, кроз текстуалну или графичку представу. Две класе језика представљају језици опште намене (енгл. *General-Purpose Languages*) и наменски језици (енгл. *Domain-Specific Languages*). Језици опште намене применљиви су у било ком домену рада. Наменски језици формиран су за подршку специфичног домена, како би крајњи корисници користили концепте из домена приликом креирања модела. Наменски језици најчешће се користе у оквиру парадигме *MDS* [4].

Сваки наменски језик сачињен је из три компоненте: *апстрактне синтаксе*, где је приказана структура језика и начин комбиновања основних елемената; *конкретне синтаксе*, која пружа специфичну репрезентацију језика, а може бити текстуална или графичка; и *семантике*, којом је описано значење елемената или њихових комбинација, дефинисаних у оквиру језика.

3. ПРЕГЛЕД РАДА ВЕКТОРСКИХ БАЗА ПОДАТАКА

Векторске базе података служе као складишта за векторе, односно, излазне резултате алгоритама машинског учења. Такође, чувају мета-податке у вези са ускладиштеним векторима, омогућавајући ефикасну манипулацију подацима. Чувањем вектора, омогућен је поступак претраге сличности, усклађујући корисничке упите са одговарајућим векторима. Претраживање вектора подразумева израчунавање удаљености између вектора, користећи метрике попут косинусне сличности. Систем за управљање векторском базом података (енгл. *Vector DataBase*

Management System – VDBMS) [2] израчунава удаљености између вектора корисничког упита и вектора складиштених у бази података, те као одговор на упит враћа најсличније векторе, у складу са метрикама сличности. Рад са векторским базама података може се посматрати кроз неколико целина – дефинисање структура података, манипулацију подацима, векторизацију, те векторску претрагу и филтрирање података.

У овом поглављу, представљен је преглед начина рада векторских база података, где је поглавље 3.1 усмерено на разјашњењу структуре векторских база података, док су у поглављу 3.2 анализирани подржане операције над подацима у векторској бази података. Сви концепти описани у овом поглављу коришћени су приликом формирања апстрактне синтаксе наменског језика *uniVEC*, истакнуте у поглављу 4.

3.1 Структура векторске базе података

Пре уноса података, крајњи корисници треба да припреме податке, у циљу складиштења у формату релевантном за векторске базе података. Као подршка при формирању података, јавља се концепт *шеме*. Концептом шеме формирана је структурна корелација између података и објеката унутар базе података.

Концепт *колекције* служи као складиште за организацију и управљање ентитетима. Приликом креирања колекције, у одређеним векторским базама података, потребно је креирати шему колекције, која обезбеђује структуру колекције.

Концепт *поља* односи се на специфичан атрибут ентитета у колекцији. Поље може бити векторизовано или не векторизовано. Шеме се такође могу користити за опис структуре поља.

Ентитет је представљен скупом поља унутар колекције. Сваки ентитет колекције јединствено је идентификован примарним кључем.

Концепт *кластера* подразумева логичко груписање сличних података унутар векторске базе података. Овај механизам груписања организује тачке података на основу њихове близине или сличности у простору обележја, олакшавајући тако анализу података.

Након складиштења података, у циљу ефикасније обраде упита, врши се *индексирање* података. У векторским базама података, разликују се скаларни индекси, формиран за не векторизована поља, са циљем да убрзају упите са потпуним подударом и векторски индекси, формиран на основу векторизованих поља, ради убрзања претраге сличности.

Код индекса векторског типа, потребно је навести која *метрика* ће бити коришћена за израчунавање сличности између тачака у векторском простору, које су складиштене у векторској бази података.

3.2 Векторизација и манипулација подацима

Основни концепт векторских база података је *векторска уградња* (енгл. *vector embedding*), помоћу ког су подаци различитог типа структурираности

приказани кроз n -димензионалне векторе. Векторска уградња, у општем смислу, подразумева укореењен концепт у машинском учењу, где представља податке мапирани у високодимензионални простор, при чему семантички слични подаци егзистирају као физички блиске тачке података у векторском простору. Уобичајен приступ формирања векторских уградњи ослоњен је на дубоке неуронске мреже, где се користе предефинисани модели машинског учења, попут модела *BERT* (енгл. *Bidirectional Encoder Representations from Transformers*).

Поред векторске претраге сличности, могуће је вршити и претрагу на основу вредности обележја датих у оквиру постављеног упита. Додатно, *VDBMS* нуди уобичајене операције уноса, измене и брисања података.

Како не постоји стандардизовани језик за интеракцију са векторским базама података, корисници им углавном приступају помоћу програмског кода, што захтева стручност у интерпретацији циљних операција у оквиру одабраног програмског језика. Програмски језици који подржавају рад са векторским базама података су *Python*, *Java*, *JavaScript/TypeScript*, *Node.js* и *Go*, а приступ бази података могуће је остварити путем *REST API* принципа.

4. НАМЕНСКИ ЈЕЗИК *uniVEC*

Наменски језик *uniVEC* развијен је помоћу језика *Ecore* у окружењу *Eclipse Modeling Framework (EMF)* [5]. За потребе синтаксне и семантичке провере модела насталих помоћу језика *uniVEC*, формиране су софтверске компоненте за проверу ограничења

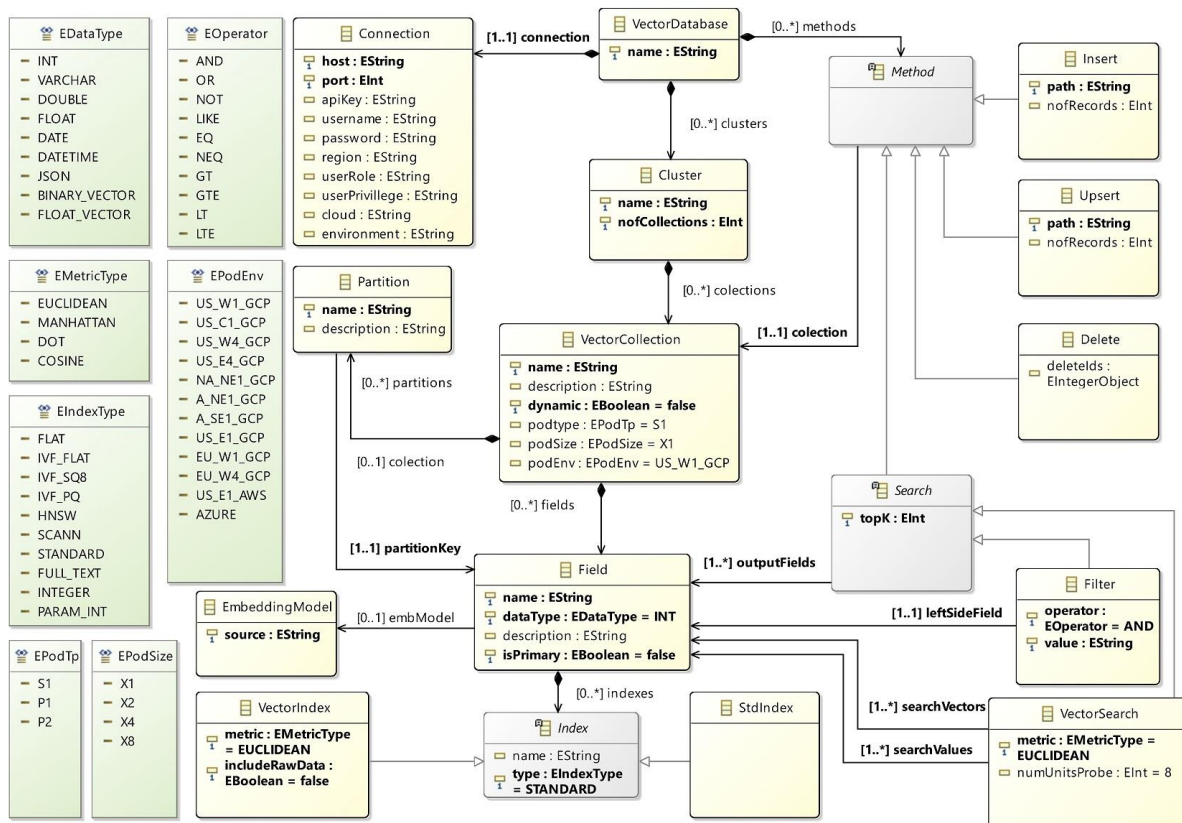
дефинисаних на нивоу мета-модела. Наменски језик подржава конкретну текстуалну синтаксу креирану у оквиру радног оквира *Xtext* [6]. Апстрактна синтакса језика *uniVEC* представљена је у поглављу 4.1, а у поглављу 4.2 приказан је пример модела насталог употребом дела конкретне синтаксе.

4.1 Апстрактна синтакса језика *uniVEC*

Коренски концепт апстрактне синтаксе приказане на *Слици 1* је векторска база података (*VectorDatabase*), која мора да има једну конекцију (*Connection*), како би веза ка циљној векторској бази података била успостављена. У векторској бази података може се наћи више кластера (*Cluster*), где сваки кластер може садржати више колекција (*VectorCollection*). Колекцију чини скуп поља (*Field*), која могу бити векторизована и неекторизована. За свако векторизовано поље неопходно је навести модел за генерисање векторских уградњи (*EmbeddingModel*). За поље је могуће дефинисати индекс (*Index*), векторизовани или неекторизовани, у складу са наведеним типом поља. За једну колекцију могуће је формирати онолико партиција (*Partition*) колико има поља, где је кључ партиције поље на основу ког је иста формирана. На нивоу векторске базе података, могуће је извршити следеће методе (*Method*): унос (*Insert*), измену (*Upsert*) и брисање (*Delete*) података, те филтрирање података (*Filter*) и векторску претрагу сличности (*VectorSearch*). Методи се извршавају на нивоу одређене колекције.

4.2 Конкретна синтакса језика *uniVEC*

За потребе коришћења језика *uniVEC*, формирана је текстуална конкретна синтакса језика, која је настала



Слика 1. Апстрактна синтакса наменског језика *uniVEC*

на основу мета-модела. У примеру представљеном у *Листингу 1*, приказано је формирање векторске базе података, са једним кластером, где је садржана колекција у вези са филмовима. Колекција има два поља, идентификатор филма – неекторизовано поље и примарни кључ, као и наслов филма – векторизовано поље. Од метода, примарно је илустрован унос података из датотеке типа *CSV* (енгл. *Comma-Separated Values*), те је извршена векторска претрага сличности, на основу поља наслова филма, где треба додати пет најсличнијих филмова који у себи садрже реч „Greek“.

```
CREATE VDB domijvd
WITH CONNECTION
  host = 127.0.0.1
  port = 19530
WITH CLUSTERS
CREATE CLUSTER domijcluster
  number of collections = 5
WITH COLLECTIONS
  CREATE COLLECTION MOVIES
    dynamic
    description = descr01
WITH FIELDS [
  CREATE FIELD id
    PRIMARY
    type = INT,
  CREATE FIELD movietitle
    type = VARCHAR ]
WITH METHODS
  INSERT INTO COLLECTION "domijcluster.MOVIES"
    data file path = "C:\Users\userXY\Desktop\data.csv"
    number of records = 10000;
  Vector Search ON COLLECTION "domijcluster.MOVIES"
    topK = 5
    metric = EUCLIDEAN
    search fields = [movietitle]
    value = "Greek"
    output fields = ["domijcluster.MOVIES.id",
"domijcluster.MOVIES.movietitle"];
```

Листинг 1. Пример конкретне синтаксе језика *uniVEC*

5. ГЕНЕРАТОРИ КОДА

Генератори кода генеришу извршив програмски код из модела, насталих помоћу језика *uniVEC*, у одговарајуће *Python* скрипте за подршку рада са одабраном векторском базом података, као што су: *Pinecone*, *Milvus*, *Chroma*, *Weaviate* или *Qdrant*. Генератори кода написани су у програмском језику *Xtend* [6]. У оквиру *Листинга 2*, илустрован је шаблон за генерисање индекса. Поред наведеног шаблона, имплементирани су и шаблони за генерисање концепата описаних у поглављу 3.

```
def generateIndex(Index index) '''
    «IF index instanceof StdIndex»
        indexType = «index.type.getName()»
        Index(name="«index.name»",
            type=IndexType.valueOf(indexType))
    «ELSEIF index instanceof VectorIndex»
        indexType = «index.type.getName()»
        metricType = «index.metric.getName()»
        indexParams = newHashMap[String, String]
        «IF index.includeRawData»
            indexParams.put("includeRawData", "true")
        «ENDIF»
        «FOR limit: index.indexLimits»
            indexParams.put("«limit.type.getName()»",
                "«limit.value»")
        «ENDFOR»
        VectorIndex(
            name="«index.name»",
            type=IndexType.valueOf(indexType),
            metric=EIndexType.valueOf(metricType),
            index_params=indexParams)
    «ENDIF» '''
```

Листинг 2. Део шаблона за генерисање концепта индекса

6. ЗАКЉУЧАК

Наменски језик *uniVEC* обједињује концепте векторских база података и омогућава униформни приступ различитим векторским базама података. Представљени језик и генератори кода имају циљ да буде: поједностављен рад са циљном векторском базом података; редуковано време неопходно за овладавање концептима у различитим векторским базама података; олакшано тестирање различитих векторских база података; и поједностављена миграција између одабране изворне и одредишне векторске базе података.

Будући правци истраживања и развоја представљеног окружења обухватају подршку рада са додатним векторским базама података, што би подразумевало имплементацију нових генератора кода и потенцијално проширење синтаксе наменског језика *uniVEC*. Додатно, могуће је интегрисати језик *uniVEC* у форми библиотеке у одређени програмски језик, како би било могуће користити *uniVEC* у окружењу програмског језика, што би инжењерима софтвера омогућило програмирање софтверског решења и приступ векторској бази података из једног окружења.

ЛИТЕРАТУРА

- [1] Z. Altan, Ed., *Applications and Approaches to Object-Oriented Software Design: Emerging Research and Opportunities*. in *Advances in Systems Analysis, Software Engineering, and High Performance Computing*. IGI Global, 2020. doi: 10.4018/978-1-7998-2142-7.
- [2] T. Taipalus, "Vector database management systems: Fundamental concepts, use-cases, and current challenges," *Cogn. Syst. Res.*, vol. 85, p. 101216, Jun. 2024, doi: 10.1016/j.cogsys.2024.101216.
- [3] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*, 2nd ed., San Rafael, USA: Morgan & Claypool Publishers, 2017.
- [4] M. Mernik, J. Heering, and A. M. Sloane, "When and how to develop domain-specific languages," *ACM Comput. Surv.*, vol. 37, no. 4, pp. 316–344, Dec. 2005, doi: 10.1145/1118890.1118892.
- [5] D. Steinberg, F. Budinsky, M. Paternostro, E. Merks, *EMF: Eclipse Modeling Framework*, 2nd ed., Upper Saddle River, NJ: Addison-Wesley, 2009.
- [6] L. Bettini and S. Efttinge, *Implementing domain-specific languages with Xtext and Xtend*, 2nd ed., Birmingham Mumbai: Packt Publishing, 2016.

КРАТКА БИОГРАФИЈА



Елена Акић рођена је 2000. године у Руми у Републици Србији. Школске 2019/2020. уписује се на Факултет техничких наука у Новом Саду, на студијски програм Информациони инжењеринг. Основне академске студије завршила је 2023. године и исте године се уписује на мастер академске студије, на студијском програму Рачунарство и аутоматика, модул – Инжењеринг информационалних система.

UPRAVLJANJE SINHRONIM MOTOROM SA PERMANENTNIM MAGNETIMA NA ROTORU U ELEKTROMOTORNIM POGONIMA BEZ DAVAČA NA VRATILU
SENSORLESS CONTROL OF PERMANENT MAGNET SYNCHRONOUS MOTOR ELECTRIC DRIVES

Nebojša Koledin, Vladimir Popović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Postoje mnogobrojni algoritmi za kontrolu SPMSM bez upotrebe davača brzine na vratilu baziranih na estimaciji EMS. Međutim, to nije slučaj i sa IPMSM jer se informacija o poziciji ne sadrži samo u EMS nego i u reluktanciji rotora. U ovom radu je predstavljen model IPMSM, a zatim predložena konfiguracija observera proširene elektromotorne sile. Za dobijanje informacije o poziciji i brzini rotora predložene su tri metode: PLL, direktna primena arctg funkcije i adaptivni zakon baziran na metodi Ljapunova. Numerička verifikacija algoritma je prvo izvršena u Matlab Simulink programskom okruženju, a zatim i na eksperimentalnoj postavi gde je kontrola napravljena u Typhoon HIL softverskom paketu.

Abstract – There are numerous sensorless algorithms for controlling SPMSM based on back EMF estimation. However, this is not the case with IPMSM, as position information is contained not only in the back EMF but also in the rotor reluctance. This paper presents a model of IPMSM and proposes a configuration for an extended back EMF. Three methods are proposed for obtaining rotor position and speed information: PLL, direct application of the arctg function, and an adaptive law based on the Lyapunov method. Numerical verification of the algorithm was first performed in the Matlab Simulink programming environment, and then on an experimental setup where control was implemented in the Typhoon HIL software package.

Ključne reči: IPMSM, observer proširene EMS, estimacije brzine i pozicije

1. UVOD

Sinhrona mašine sa stalnim magnetima na rotoru poslednjih decenija nalaze sve veću primenu čemu je veoma zaslužna i autoindustrija. U poređenju sa asinhronim motorima, sinhroni motori imaju veći stepen efikasnosti kao i manje dimenzije na čega se u današnje vreme, sa razvojem električnih vozila, sve više obraća pažnja. Delatnosti koje zahtevaju precizne radnje su sve brojnije, pa sa njima raste i broj pogona sa preciznom regulacijom pozicije vratila. Pogoni samo sa regulacijom brzine nisu više dovoljni.

NAPOMENA:

Ovaj rad je proistekao iz master rada čiji je mentor bio dr Vladimir Popović, docent

Pri projektovanju pogona od velikog je značaja jednostavnost instalacije i podešavanja, jednostavno povezivanje, mali broj provodnika, kablova i senzora. Korišćenje manjeg broja temperaturno i mehanički osetljivih naprava i provodnika znatno uvećava pouzdanost i robusnost pogona, a ujedno smanjuje njegovu cenu. Umanjenje broja senzora u pogonu je omogućeno masovnom proizvodnjom savremenih mikrokontrolera. Računska moć kontrolera sa vremenom raste eksponencijalno, što je posledica veoma brzog tehnološkog razvoja. Postoji mnoštvo algoritama za procenu pozicije i brzine primenljivih samo na mašinama kod kojih su stalni magneti na površini rotora. Ideja na kojoj su bazirani ti algoritmi je procena fluksa stalnih magneta ili elektromotorne sile, jer se pomoću njih može izračunati pozicija rotora, a iz nje i njegova brzina. Uspešnom implementacijom tih algoritama moguće je izbaciti davač brzine iz pogona i time povećati njegovu pouzdanost i robusnost, kao i smanjiti cenu. Problem nastaje pri primeni algoritma na sinhronu mašinu sa magnetima utisnutim u rotor. Matematički model SPMSM je samo specijalan slučaj matematičkog modela IPMSM kod kog su podužna i poprečna induktivnost jednake. Za razliku od modela SPMSM koji je jednostavan, IPMSM model informaciju o poziciji sadrži ne samo u fluksu i elektromotornoj sili, nego i u reluktanciji prouzrokovanoj razlikom induktivnosti podužne i poprečne ose. U ovom radu matematički model IPMSM će biti izveden u nekonvencijalnom obliku koji uvodi novu veličinu – proširenu elektromotornu silu koja se sastoji od dve komponente koje obe sadrže informaciju o poziciji rotora. Prva je dobro poznata indukovana elektromotorna sila, a druga komponenta je proizvod reluktancije rotora. Pored algoritma procene proširene elektromotorne sile potrebno je razviti i algoritam koji će iz nje izvući informaciju o poziciji i brzini rotora. U ovom radu predložene su tri metode: PLL (*phase locked loop*), adaptivni zakon baziran na metodi Ljapunova i upotreba arctg funkcije [1]-[2].

2. MATEMATIČKI MODEL IPMSM BAZIRAN NA PROŠIRENOJ ELEKTROMOTORNOJ SILI

Glavni problem kod konvencionalnih modela sinhrona mašine je zavisnost matrice induktivnosti od pozicije rotora, kod opšteg ($\gamma\delta$) modela, i potreba preciznog poznavanja položaja rotora, kod vektorski orijentisanog (dq) modela. Model koji navedene probleme nema je baziran na proširenoj elektromotornoj sili. Razlika ovog modela u odnosu na prethodne je ta što se on definiše u stacionatnom $\alpha\beta$ koordinatnom sistemu čime se postiže

jednostavnost matrice induktivnosti i oslobađa potrebe za poznavanjem položaja rotora.

2.1. Matematički model IPMSM baziran na proširenoj elektromotornoj sili

Jednačina naponske ravnoteže statora u dq domenu data je u (1), a jednačina u $\alpha\beta$ domenu data je u (2).

$$\begin{bmatrix} u_{ds} \\ u_{qs} \end{bmatrix} = R_s \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} + \begin{bmatrix} pL_d & -\omega L_q \\ \omega L_d & pL_q \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega\psi_{PM} \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \begin{bmatrix} R + pL_\alpha & pL_{\alpha\beta} \\ pL_{\alpha\beta} & R + pL_\beta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \omega\psi_{PM} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \quad (2)$$

gde su L_d i L_q podužna i poprečna induktivnost i:

$$\begin{aligned} L_0 &= (L_d + L_q)/2, & L_1 &= (L_d - L_q)/2 \\ L_\alpha &= L_0 + L_1 \cos 2\theta, & L_\beta &= L_0 - L_1 \cos 2\theta \\ L_{\alpha\beta} &= L_1 \sin 2\theta \end{aligned}$$

U (2) se jasno vidi da je informacija o poziciji rotora sadržana ne samo u EMS nego i u induktivnostima. Zavisnost induktivnosti od 2θ posledica je reluktancije motora. Jednostavnim matematičkim manipulacijama moguće je eliminisati zavisnost induktivnosti od položaja rotora. Za razliku od matrice induktivnosti kod SPMSM, kod IPMSM je ona nesimetrična. Ako se (1) preuredi tako da matrica induktivnosti bude simetrična, a ostatak pridružio članu koji modeluje EMS, dobija se (3).

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} R + pL_d & -\omega L_q \\ \omega L_q & R + pL_d \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ (L_d - L_q)(\omega i_d - \dot{i}_q) + \omega\psi_{PM} \end{bmatrix} \quad (3)$$

Prelaskom u $\alpha\beta$ domen pomoću inverzne Parkove transformacije jednačina naponske ravnoteže (3) dobija sledeći oblik:

$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = R \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \begin{bmatrix} pL_d & \omega(L_d - L_q) \\ -\omega(L_d - L_q) & pL_q \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \{ (L_d - L_q)(\omega i_d - \dot{i}_q) + \omega\psi_{PM} \} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \quad (4)$$

Drugi član sa desne strane jednačine (4) se definiše kao proširena EMS prikazana u (5). Ona se sastoji od konvencionalne EMS indukovane stalnim magnetima i neke vrste napona koji nastaje usled reluktancije rotora. U njoj je sadržana informacija o poziciji rotora koja potiče od EMS, kao i iz matrice induktivnosti. S tim u vezi, ukoliko se proširena EMS može proceniti, onda se iz nje može izvući informacija o položaju rotora [2].

$$e = \begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} = \{ (L_d - L_q)(\omega i_d - \dot{i}_q) + \omega\psi_{PM} \} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \quad (5)$$

3. OBSERVER PROŠIRNE ELEKTROMOTORNE SILE

U ovom poglavlju će biti opisana struktura adaptivnog observera proširene elektromotorne sile. On se bazira na linearnom matematičkom modelu IPMSM u prostoru stanja.

3.1. Linearni matematički model IPMSM u prostoru stanja

Matematički model sinhronne mašine sa stalnim magnetima utisnutim u rotor može da se predstavi kao linearni model u prostoru stanja (6). Promenljive stanja su struje statora i i proširene elektromotorne sile e . Ulazi u sistem su naponi statora u , a njegovi izlazi su struje statora i . Napon u i W su poremećaji sistema. Brzina rotora ω može da se posmatra kao konstantna ako se

uzme u obzir da je električna vremenska konstanta mnogo manja od mehaničke [2].

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} i \\ e \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} i \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ W \end{bmatrix} \\ i &= C \cdot \begin{bmatrix} i \\ e \end{bmatrix} \end{aligned} \quad (6)$$

gde su:

$$\begin{aligned} A_{11} &= -\frac{R}{L_d} I + \omega \frac{L_d - L_q}{L_d} J \\ A_{12} &= -\frac{1}{L_d} I = a_{12} I \\ A_{22} &= \omega J = a_{22} J \\ B &= \frac{1}{L_d} I = -a_{12} I, \quad C = [I \quad 0] \end{aligned}$$

$$W = (L_d - L_q)(\omega \dot{i}_d - \ddot{i}_q) \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

3.2. Observer proširene EMS

Član W u jednačini (6) je greška pri linearizaciji. On je prisutan samo kada se struje i_d i i_q menjaju. Međutim, pri brzinskoj kontroli to kratko traje zbog postojanja regulatora struje koji ubrzava njihov odziv, pa tranzijentni periodi traju veoma kratko. Sa argumentima koji su navedeni lako se može zaključiti da se član W može zanemariti u projektovanju observera. Zanemarenje neće uticati na stacionarno stanje sistema i njime se veoma pojednostavljuje observer, ali cena zanemarenja je da estimacija u tranzijentnim periodima neće biti precizna. U (7) je prikazan observer proširene elektromotorne sile.

$$\begin{aligned} \dot{\hat{i}} &= \tilde{A}_{11} \hat{i} + \tilde{A}_{12} \hat{e} + \tilde{B} u \\ \dot{\hat{e}} &= \tilde{A}_{22} \hat{e} + G(\hat{i} - i) \\ &= G\tilde{A}_{11} \hat{i} + G(\tilde{A}_{12} + \tilde{A}_{22}) \hat{e} + G\tilde{B} u + G\hat{i} \end{aligned} \quad (7)$$

gde je G korektivno pojačanje definisano kao:

$$G = g_1 I + g_2 J$$

Kako bi se izbeglo računanje izvoda struje i uvodi se nova promenljiva ξ [2].

$$\begin{aligned} \xi &= \hat{e} + G\hat{i} \\ \dot{\xi} &= \dot{\hat{e}} + G\dot{\hat{i}} \end{aligned} \quad (8)$$

Uvrštavanjem (8) u (7) dobija se ekvivalentan observer u kom ne postoji izvod struje i (9).

$$\begin{aligned} \dot{\xi} &= (G\tilde{A}_{12} + A_{22})\xi + G\tilde{B}u + (G\tilde{A}_{11} - G\tilde{A}_{12}G - \tilde{A}_{22}G)\hat{i} \\ \hat{e} &= \xi - G\hat{i} \end{aligned} \quad (9)$$

Ovako konstruisanom observeru može da se pridruži funkcija prenosa $S(j\omega)$ koja matematički određuje brzinu odziva observera, tj. brzinu uspostavljanja procenjene proširene elektromotorne sile:

$$S(j\omega) = (\alpha I + (\omega - \beta)J) \frac{(j\omega + \alpha)I + \beta J}{(j\omega + \alpha)^2 + \beta^2} \quad (10)$$

gde α i β predstavljaju polove observera i u vezi su sa pojačanjem observera G na sledeći način:

$$\alpha = -g_1 a_{12} \quad \beta = a_{22} + g_2 a_{12}$$

Primenom postupka koji je izložen u [2] dobijaju se vrednosti polova observera:

$$\alpha = \omega \cdot v, \quad \beta = \omega$$

gde je v parametar čija je preporučena vrednost 2.

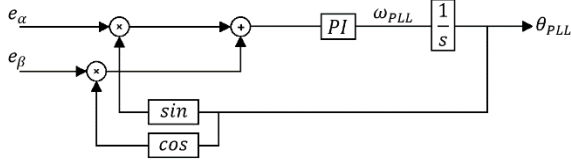
4. METODE ZA PROCENU POZICIJE I BRZINE IPMSM

Informacije o brzini i poziciji rotora su neophodne za vektorsku kontrolu električnih mašina. Pošto se ovaj rad bavi kontrolom sinhronne mašine sa stalnim magnetima bez upotrebe senzora brzine, u ovom poglavlju će biti opisane tri metode za procenu pozicije i brzine:

- arctg i difrenciranje;
- PLL petlja;
- Adaptivni mehanizam.

4.1. PLL algoritam za procenu pozicije i brzine

PLL (engl. Phase locked loop) se često koristi za procenu brzine i pozicije rotora. Svojom konfiguracijom petlja prirodno filtrira sve nepravilnosti i poremećaje u ulaznim signalima. To za posledicu ima eliminaciju direktne zavisnosti ulaznih i izlaznih signala, gde se pri bilo kom poremećaju u ulaznim signalima to direktno odražava i na izlaze. PLL je zatvorena petlja koja preko PI regulatora eliminiše grešku u proceni pozicije rotora. Izlaz iz regulatora je brzina rotora koja se integriše da bi se dobila pozicija po kojoj je zatvorena sprega. Konfiguracija PLL-a je prikazana na Slici 1. [3].



Slika 1. Konfiguracija PLL-a

Iz definicije proširene elektromotorne sile (5) očigledno je da ona direktno zavisi od pozicije rotora, pa se α i β komponenta proširene elektromotorne sile mogu napisati na sledeći način:

$$e_\alpha = -E \cdot \sin\theta \quad e_\beta = E \cdot \cos\theta$$

Pretpostavimo da imamo procenjen ugao iz PLL-a čiji su ulazi e_α i e_β . Ako se α komponenta pomnoži sa $\cos\theta_{PLL}$, a β sa $\sin\theta_{PLL}$ i saberu, za male greške pozicije dobija se:

$$e_{PLL} = -E \cdot \sin\theta \cdot \cos\theta_{PLL} + E \cdot \cos\theta \cdot \sin\theta_{PLL} = E \cdot \sin(\theta_{PLL} - \theta) \approx E \cdot (\theta_{PLL} - \theta) \quad (11)$$

4.2. Arctg i diferenciranje

Procena pozicije rotora može i da se izračuna primenom arctg funkcije na komponente proširene elektromotorne sile, (12), jer ona u sebi nosi informaciju o poziciji rotora.

$$\theta_{arctg} = \arctg(-e_\alpha/e_\beta) \quad (12)$$

Brzinu rotora, kada je poznata informacija o poziciji, jednostavno je proceniti njenim diferenciranjem (13).

$$\omega_{arctg} = \frac{d}{dt}(\theta_{arctg}) \quad (13)$$

Ovaj pristup je dobar jer ne unosi kašnjenje u sistem, tj. informacija o poziciji i brzini rotora se dobija trenutno. Cena koju je potrebno platiti za trenutnu procenu je ta da se svi poremećaji (šumovi, harmonici, itd.) direktno prenose i na poziciju. Ukoliko su ti poremećaji značajni može doći do veoma loše kontrole, pa i do ispada pogona iz rada zbog greške u proceni ugla.

4.3. Adaptivni mehanizam baziran na metodi Ljapunova

Treći način za procenu pozicije i brzine rotora je pomoću adaptivnog mehanizma zasnovanom na Model reference adaptive system (MRAS). Forma referentnog modela (15) se dobija iz (6), gledajući samo jednačinu proširene elektromotorne sile, ali se koristi procenjena elektromotorna sila dobijena iz observera, koja je potom

normalizovana (14); brzina rotora se posmatra kao konstantna.

$$\hat{e}_n = \left(1/\sqrt{\hat{e}_\alpha^2 + \hat{e}_\beta^2}\right) \cdot \begin{bmatrix} \hat{e}_\alpha \\ \hat{e}_\beta \end{bmatrix} \quad (14)$$

$$\dot{\hat{e}}_n = \omega J \hat{e}_n \quad (15)$$

Adaptivni model (16) ima istu formu kao i referentni sa razlikom člana koji ima funkciju popravke estimirane proširene elektromotorne sile.

$$\dot{\hat{e}}_n = \hat{\omega} J \hat{e}_n + G_1(\hat{e}_n - \hat{e}_n) \quad (16)$$

gde su:

$\hat{\omega}$ – procenjena brzina,

$G_1 = g_1 I = -100 \cdot I$ – korekciona matrica.

Za Ljapunovovu funkciju izabrana je (17) i može se videti da je ona pozitivna definitna funkcije. Kako bi njen izvod (18) bio negativan semidefinitan, potrebno je brzinu računati po zavisnosti (19) koja ujedno predstavlja i adaptivni zakon za estimaciju brzine rotora. Sa tim je Ljapunovov kriterijum stabilnosti zadovoljen i ostvarena je nulta greška između proširene elektromotorne sile referentnog $\hat{e}_n$ i adaptivnog $\hat{e}_n$ modela kao i konvergencija procenjene brzine $\hat{\omega}$ ka stvarnoj [4].

$$V(t) = \epsilon^T \cdot \epsilon + (\hat{\omega} - \omega)^2 / \lambda \quad (17)$$

$$\dot{V}(t) = \epsilon^T \cdot \dot{\epsilon} + \epsilon^T \cdot \dot{\epsilon} + 2\Delta\omega / \lambda \cdot \Delta\dot{\omega} \quad (18)$$

$$\hat{\omega} = (k_p + k_i/s) \epsilon^T J \hat{e}_n \quad (19)$$

Estimacija brzine primenom adaptivnog zakona baziranog na metodi Ljapunova je veoma dobra jer filtrira sve poremećaje u ulazima, bolje nego PLL, pa se procenjena i stvarna brzina neznatno razlikuju. Loša strana je što postojanje PI regulatora u adaptivnom zakonu unosi kašnjenje u estimaciji, kao i PLL, pa se informacija ne dobija trenutno kao kod arctg. Iako se pozicija rotora može proceniti kao integral procenjene brzine, ne treba je tako procenjivati jer ne postoji povratna sprega po poziciji. U slučaju da se u bilo kom, ma koliko kratkom, trenutku pojavi greška pri estimaciji brzine, to će se direktno oslikati i na grešku u poziciji.

Na kraju, zaključak je da za procenu pozicije i brzine rotora treba koristiti PLL ili kombinovati procenu pozicije preko arctg i procenu brzine pomoću adaptivnog zakona.

5. EKSPERIMENTALI REZULTATI

5.1. Konfiguracija eksperimentalne postavke

Model je testiran na postavci koja se sastoji od: jednosmernog naponskog izvora od 60V, trofaznog invertora i H-mosta koji dele jednosmerno međukolo, HIL 402 emulatora u kome se implementira model iz Typhoon HIL Schematic-a i ima ulogu kontrolera, sinhronne mašine sa stalnim magnetima i mašine jednosmerne struje (ima ulogu da nameće opterećenje sinhronoj mašini) čiji su rotori kruto spregnuti.

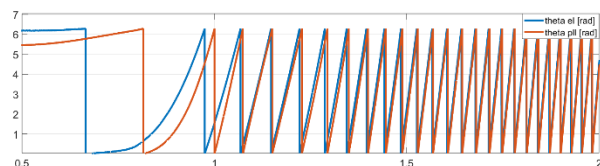
Nakon podešavanja parametara PLL-a i adaptivnog zakona, snimljeni su rezultati u offline radu observera i estimacije brzine i pozicije rotora pri zaletanju mašine do mehaničke brzine od 150 rad/s. Eksperiment je sproveden dva puta, jednom koristeći PLL, a drugi put koristeći arctg i adaptivni zakon za estimaciju. Parametri sinhronne mašine su dati u Tabeli I.

Tabela I. Parametri sinhronne mašine

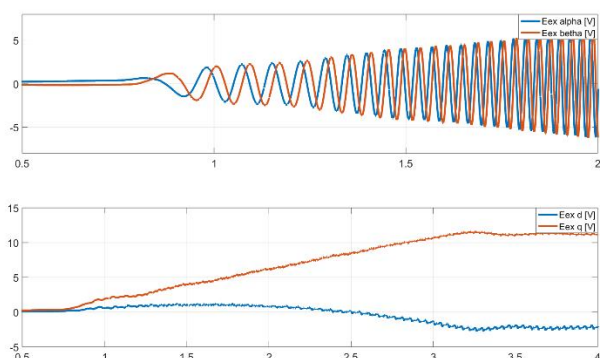
Parametar	Vrednost[veličina]
R_s	0.07[Ω]
$L_d = L_q$	0.2[mH]
ψ_{PM}	12.3[mWb]
p	3
J	0.0001kgm ²

5.2. Eksperimentalna analiza

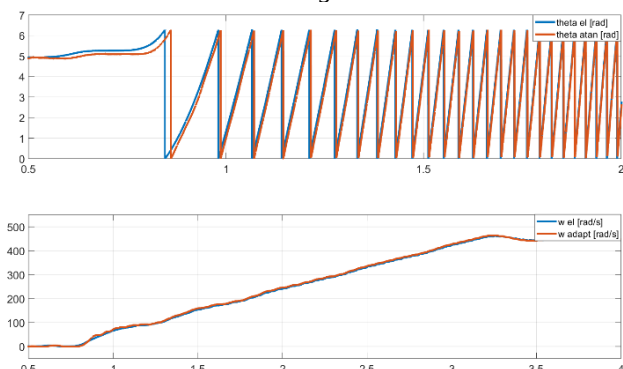
Na Slikama 2.-5. jasno se vidi da se sve veličine estimiraju veoma dobro u oba eksperimenta. Primetna je mala razlika u brzini sinhronizacije na položaj rotora gde je PLL sporiji u odnosu na adaptivni mehanizam.



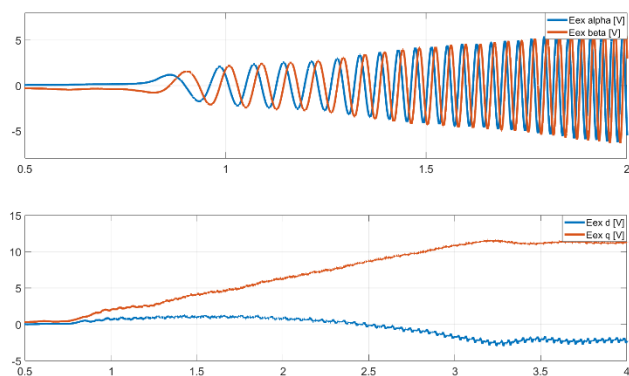
Slika 2. Stvarna i procenjena električna brzina i pozicija rotora– PLL kontrolni algoritam



Slika 3. Komponente proširene elektromotorne sile – PLL kontrolni algoritam



Slika 4. Stvarna i procenjena električna brzina i pozicija rotora – adaptivni mehanizam



Slika 5. Komponente proširene elektromotorne sile– adaptivni mehanizam

6. ZAKLJUČAK

Observer proširene elektromotorne sile se pokazao kao dobra opcija za kontrolu IPMSM bez davača na vratilu. Rezultati dva predložena načina za procenu brzine i pozicije rotora su pokazali da su slični, uz malu razliku u dinamičkom odzivu pri sinhronizaciji na ugao rotora.

7. LITERATURA

- [1] Slobodan N. Vukosavić, „Upravljanje električnim pogonima“, Akademska misao, Beograd, 2003.
- [2] Zhiqian Chen, Mutuwo Tomita, Shinji Doki, and Shigeru Okuma, „An Extended BEMF Model for Sensorless Control of IPMSM“, IEEE Trans. on Ind. Electr., vol. 50, no. 2, April 2003.
- [3] Darko P. Marčetić, „Mikroprocesorsko upravljanje energetskim pretvaračima“, Fakultet tehničkih nauka, Novi Sad, 2020.
- [4] Jean-Jacques E. Slotine, Weiping LI, „Applied nonlinear control“, Prentice Hall, New Jersey, 1991.

Kratka biografija:

Nebojša Koledin rođen je u Novom Sadu 2000. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2023. god.

Vladimir Popović rođen je u Somboru 1990. god. Doktorsku disertaciju na Fakultetu tehničkih nauka iz naučne oblasti Elektrotehnike i računarstva –odbranio je 2020. god, i u zvanju je docenta na Fakultetu tehničkih nauka. Oblast interesovanja su regulacija elektromotornih pogona i digitalna kontrola upravljačkih sistema.

ОДРЕЂИВАЊЕ ПАРАМЕТАРА ЕКВИВАЛЕНТНЕ ШЕМЕ АСИНХРОНОГ МОТОРА УПОТРЕБОМ СОФТВЕРСКИХ АЛАТА

DETERMINATION OF PARAMETERS OF EQUIVALENT CIRCUIT OF INDUCTION MOTOR USING SOFTWARE TOOLS

Никола Поповић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду поређене су стандардне методе добијања параметара еквивалентне шеме асинхронног мотора са методом одређивања параметара употребом софтверских алата. Као пример коришћен је асинхронни мотор називне снаге 11 kW.

Кључне речи: Асинхрона машина, еквивалентна шема, метода коначних елемената

Abstract – In this paper, standard methods of obtaining parameters of the equivalent circuit of an asynchronous motor were compared with a method of determining parameters using software tools. An asynchronous motor with a rated power of 11 kW was used as an example.

Keywords: Asynchronous Machine, Equivalent Circuit, Finite Element Method

1. УВОД

Асинхрона машина у савременој индустрији престава најважнији уређај за извршење механичког рада.

Коришћење асинхроне машине је економски исплативо и оправдано, одржавање је јефтино, отпорна је на грубе манипулације и једноставна је за коришћење, поуздана је и сигурна током рада, није великих габарита. У примени најчешће се сусреће као трофазни мотори.

Еквивалентна шема асинхроне машине представља параметре машине који омогућавају одређивање стања машине. Стање машине зависи од улазних параметара као што су напон статора, момент оптерећења вратила, брзина обртања вратила итд. Еквивалентна шема асинхроне машине је струјно коло формирано од еквивалентнираних параметара сведених на једну фазу.

Познавање параметара еквивалентне шеме омогућава се одређивање статичких и динамичких карактеристика машине.

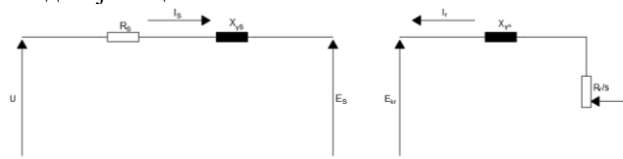
Стандардне методе за одређивање параметара еквивалентне шеме су нумеричка метода и експериментална метода.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Дејан Јеркан.

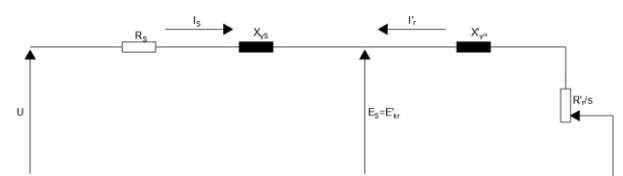
2. ФОРМИРАЊЕ ЕКВИВАЛЕНТНЕ ШЕМЕ АСИНХРОНЕ МАШИНЕ

Основ за добијање еквивалентне шеме представља пофазна представа асинхроне машине дата на следећој слици:



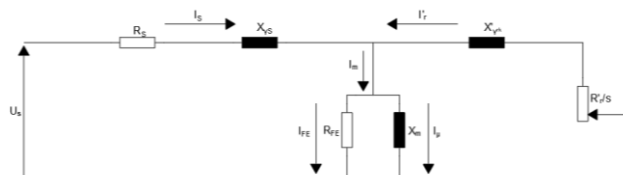
Слика 1. Пофазна представа асинхроне машине

Струјна кола приказана на претходној шеми су галвански одвојена, али постоји магнетна спрега између њих. Како би се добило еквивалентно коло неопходно је свести роторско струјно коло на статорско. Испитивањем специјалног случаја када је клизање $s=1$ у ком се машина понаша као трансформатор, може се доћи до следеће представе асинхроне машине:



Слика 2. Галвански повезана пофазна представа асинхроне машине

Изражавањем електромоторне силе статора преко струје магнећења и еквивалентирањем губитака у гвожђу машине омском отпорношћу може се одредити грана магнећења. Уврштавањем гране магнећења у еквивалентну шему добија се комплетна еквивалентна шема:



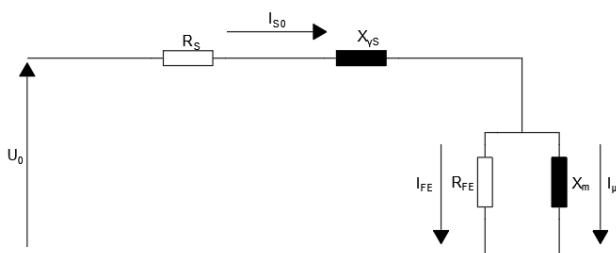
Слика 3. Комплетна еквивалентна шема асинхроне машине

3. НУМЕРИЧКА МЕТОДА ОДРЕЂИВАЊА ПАРАМЕТАРА ЕКВИВАЛЕНТНЕ ШЕМЕ

Нумеричка метода одређивања параметара еквивалентне шеме се своди на нумеричко одређивање параметара машине у току огледа кратког споја и празног хода. Комбинацијом добијених параметара се може добити комплетна еквивалентна шема.

3.1. Оглед празног хода

Празан ход асинхроне машине је оглед који подразумева да се на прикључке статора доводи номиналан напон, док се вратило ротора држи неоптерећеним. Уколико је вратило ротора неоптерећено тада ће и моменат оптерећења бити једнак нули што за последицу има следеће: $s=0$, $I_r=0$, $P_{gFE}=0$. Сада се еквивалентна шема асинхроне машине у току огледа празног хода може приказати на следећи начин:



Слика 4. Шема асинхроне машине у режиму празног хода

Машину у празном ходу карактериши следеће једначине:

$$S_0 = \sqrt{3} \cdot U_n \cdot I_0 = P_0 - jQ_0 \quad (1)$$

$$P_{gFE} = (P_0 - 3 \cdot R_s \cdot I_0^2) \cdot \frac{3}{4} \quad (2)$$

$$I_{FE} = I_0 \cos \varphi_0 \quad (3)$$

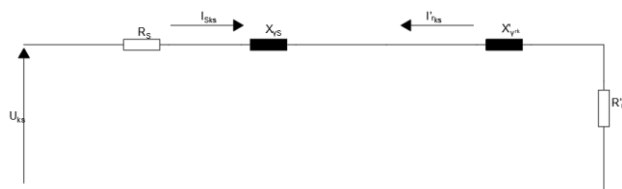
$$R_{FE} = \frac{P_{gFE}}{3 \cdot I_{FE}^2} \quad (4)$$

$$X_0 = \frac{Q_0}{3 \cdot I_0^2 \sin \varphi} \quad (5)$$

$$X_m = X_0 - X_{\gamma s} \quad (6)$$

3.2. Оглед кратког споја

Оглед кратког споја подразумева да се вратило ротора држи укоченим док се на прикључке статора доводи напон потребан за протицање номиналне струје у статору. Оглед кратког споја се изводи како би се одредили параметри редних грана статора и ротора еквивалентне шеме. Како је брзина обртања ротора једнака нули, клизање ће износити $s=1$. Ово за последицу има да ће статорска и роторска струја бити једнаке, па је из еквивалентне шеме могуће закључити да ће струја магнетнења бити једнака нули, односно да је могуће занемарити грану магнетнења. Еквивалентна шема огледа кратког споја се може исказати на следећи начин:



Слика 5. Шема асинхроне машине у режиму кратког споја

Машину у кратком споју карактеришу следеће једначине:

$$S_{ks} = \sqrt{3} \cdot U_{ks} \cdot I_n = P_{ks} - jQ_{ks} \quad (7)$$

$$U_{ks} = \frac{I_n}{I_{start}} \cdot U_n \quad (8)$$

$$R_{ks} = R_s + R'_r \quad (9)$$

$$P_{ks} = 3 \cdot R_{ks} \cdot I_n^2 \quad (10)$$

$$X_{ks} = \sqrt{\frac{U_{ks}^2}{I_n^2} - R_{ks}^2} \quad (11)$$

За одређивање параметара X_{ks} и R_{ks} могуће је користити следеће апроксимације у случају недостатка података:

$$\begin{aligned} X_{\gamma s} &= X'_{\gamma r} = \frac{X_{ks}}{2}, \\ R_s &= R'_r \end{aligned} \quad (12)$$

4. ОДРЕЂИВАЊЕ ПАРАМЕТАРА ЕКВИВАЛЕНТНЕ ШЕМЕ УПОТРЕБОМ СОФТВЕРСКОГ АЛАТА

У претходно приказаном прорачунском поступку за добијање вредности параметара еквивалентне шеме улазни подаци су били струја и напон статора, брзина обртања вратила и фактор снаге. У програмском алату *FEMM* улазни подаци су вредности струја по фази у посматраном моменту, фреквенција и број намотаја по фази.

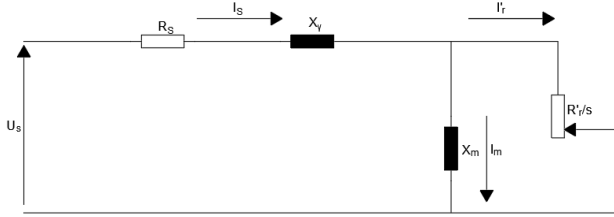
Програмски алат *FEMM* користи геометрију машине и магнетне особине материјала појединачних делова како би израчунао утицај магнетног поља произведеног од стране постојећих струја.

Након извршене симулације програм омогућава анализирање података. Међу подацима који се могу добити као резултат симулације налазе се напон, отпорност, струја, вектор јачине магнетног поља,... *FEMM* ради по принципу методе коначних елемената. Метода коначних елемената је метод нумеричког решавања диференцијалних једначина.

Принцип решавања једначина се своди на поделу проблема на мање подјединице зване коначни елементи.

Решења једначина добијају се извођењем једначина за све подјединице, коначне елементе, а затим израчунавањем свих добијених једначина у целокупном систему и апроксимацијом непознатих величина.

За одређивање појединих параметара еквивалентне шеме је неопходно прво извести адекватан математички модел. За извођење математичког модела користиће се следећа еквивалентна шема:



Слика 6. Γ -шема асинхроне машине

Из приказане еквивалентне шеме се може доћи до следећих израза:

$$X_\mu = \omega M \quad (13)$$

$$R'_r = \frac{M}{\tau} \quad (14)$$

$$L_\gamma = L_R - \frac{M}{1 + (\tau\omega_s)^2} \quad (15)$$

$$X_\gamma = \omega L_\gamma = 2\pi f L_\gamma \quad (16)$$

$$L_s = \left(L_\gamma + \frac{M}{1 + (\tau\omega_s)^2} \right) - j \left(\frac{\tau\omega_s M}{1 + (\tau\omega_s)^2} \right) \quad (17)$$

$$M_{meH} = 3pMi_s^2 \left(\frac{j\tau\omega_s}{1 + (j\tau\omega_s)^2} \right) \quad (18)$$

4.1. Израчунавање вредности у FEMM-у

Претходно добијени израз за укупну индуктивност статорског намотаја се може поделити на реалан и имагинаран део. На тај начин добијају се две једначине са три непознате. Те две једначине гласе:

$$Re[L_s] = L_R = L_\gamma + \frac{M}{1 + (\tau\omega_s)^2} \quad (19)$$

$$Im[L_s] = L_I = - \frac{\tau\omega_s M}{1 + (\tau\omega_s)^2} \quad (20)$$

Решавање наведених једначина ће се вршити методом најмањих квадрата. Метода најмањих квадрата је поступак који се користи у регресивној анализи. Метода се своди на одређивање криве која приближно задовољава сва решења, уз критеријум провере да је квадратна вредност разлике података и апроксимираних вредности на криви најмања. Како би се метода најмањих квадрата могла применити на горе задате вредности, неопходно је прво прилагодити добијене једначине. Прво ће се увести нове променљиве ради олакшавања записа:

$$c_1 = \tau M \quad (21)$$

$$c_2 = \tau^2 \quad (22)$$

Сада се претходни израз може записати као:

$$\omega_s c_1 + L_I \omega_s^2 c_2 = -L_I \quad (23)$$

Могуће је увести матрични запис претходног израза како би се представила различита решења добијена за различите кружне брзине клизања:

$$m = \begin{bmatrix} \omega_{s1} & L_{I1} \omega_{s1}^2 \\ \vdots & \vdots \\ \omega_{sk} & L_{Ik} \omega_{sk}^2 \end{bmatrix} \quad b = \begin{bmatrix} -L_{I1} \\ \vdots \\ -L_{Ik} \end{bmatrix} \quad (23)$$

$$m \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = b$$

Сада се добијени проблем може изразити на следећи начин:

$$m^T m \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = m^T b \quad (24)$$

Решење приказаног проблема ће бити:

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = (m^T m)^{-1} m^T b \quad (25)$$

Решење реалног дела укупне индуктивности статорског намотаја је сада могуће одредити на следећи начин:

$$L_\gamma = L_R - \frac{M}{1 + (\tau\omega_s)^2} \quad (26)$$

Захваљујући вредностима M и τ одређених у претходним корацима могуће је одредити вредност L_γ израчунавањем средње вредности:

$$L_\gamma = \frac{\sum_{n=1}^k L_{\gamma n}}{k} \quad (27)$$

5. СТУДИЈА СЛУЧАЈА СНАГЕ 11 kW

Рађења је студија случаја кавезне асинхроне машине снаге 11 kW са различитим навојним кораком секције. Асинхрони мотор чији ће параметри бити одређивани у наставку рада има следеће називне податке:

- Номинална снага $S_n = 11 \text{ kW}$
- Номинална брзина $n_n = 1440 \text{ o/min}$
- Номинални напон $U_n = 400 \text{ V}$
- Номинална фреквенција напона $f_n = 50 \text{ Hz}$
- Номинална линијска струја $I_n = 22 \text{ A}$

Пре него што се претходно описане методе примене на наведену машину, неопходно је кориговати улазне податке. Претпоставићемо да ће статорски намотај описане машине бити повезан у троугао. У даљем тексту ће се користити ознака f за фазну струју. Уколико је та ознака изостављена, ради се о линијској струји. Из номиналне брзине обртања можемо закључити да се ради о четворополној машини.

5.1. Параметри одређени употребом нумеричке методе

Применом претходно описаних оледа, нумеричком методом су одређени следеће вредности параметара еквивалентне шеме, дате у Табели 1.

Табела 1. Параметри еквивалентне шеме

R_s	1,34965 Ω
R'_r	1,34965 Ω
R_{FE}	570,549 Ω
$L_{\gamma s}$	0,00573 H
$L'_{\gamma r}$	0,00573 H
M	0,18262 H

5.2. Параметри еквивалентне шеме одређене употребом софтверског алата FEMM

За примењивање методе коначних елемената неопходно је познавати физичке карактеристике машине. Претпостављена дубина машине је 205 mm. Ротор машине је кавезног типа, што значи да су сви намотаји кратко спојени. Статор машине садржи 36 жлебова подељених на два дела. У један жлеб се поставља 26 навојака. Проводници статора су бакарни, док је кавез ротора од алуминијума. Језгро статора и ротора чини челик. Између статора и ротора се налази ваздушни процеп. На следећој слици приказан је део модела из софтверског алата FEMM на ком се види распоред материјала на моделу мотора.

За вредности статорских струја одабрана је номинална вредност амплитуде фазне струје. Проблем је симулиран са различитим фреквенцијама клизања да би се дошло до довољно прецизних резултата. Коришћене су вредности фреквенције клизања која износе од 0,25 до 3 Hz са кораком од 0,25 Hz.

Применом претходно описанаог поступка, овом методом су одређени следеће вредности параметара еквивалентне шеме, приказане у Табели 2.

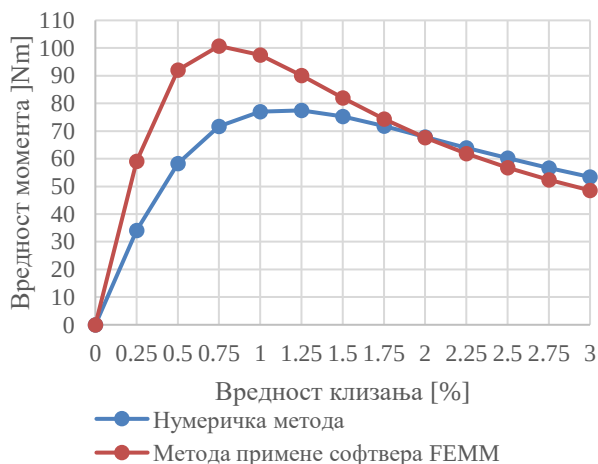
Табела 2. Параметри Г еквивалентне шеме

R'_r	1,00858 Ω
L_{ys}	0,00779 H
L'_{yr}	0,00779 H
M	0,20829 H

Приликом одређивања модела употребом ове методе изостављају се фактори отпорности статора и отпорности гране магнећења.

5.3. Поређење моментних карактеристика добијених различитим методама

Моментне карактеристике одређена употребом параметара добијених применом претходно описаних метода су приказане на следећем графику:



Слика 7. Упоредни приказ резултата

Називни моменат за описану машину се јавља при клизању од 2%. Са графика се може прочитати да се добија приближно једнака вредност момента за

називно клизање. Вредност превалног момента добијене приликом употребе методе примене софтверског алата FEMM (100 Nm) одступа од превалног момента добијеног применом нумеричке методе (80 Nm). Такође се може приметити да се вредност превалног клизања разликује за две поменуте методе (0,75% и 1%).

6. ЗАКЉУЧАК

На основу добијених резултата може се закључити да вредности добијене употребом обе методе дају релативно сличне резултате називног момента, као и да се моментне карактеристике подударају за вредности клизања блиске називном клизању. Одабир методе би требао да зависи од расположивих података о машини. Примењено је да су вредности превалног момента и превалног клизања различите за две методе, као и да се параметри еквивалентне шеме разликују. Одступања између резултата потичу од апроксимација и занемарења примењених у обе методе. Закључује се да је за прецизнију анализу података и проверу исправности метода неопходно познавати више карактеристика машине, или извршити испитивање машине и поредити добијене вредности наведених метода са стварним вредностима.

7. ЛИТЕРАТУРА

- [1] Веран Васић, „Скрипта увод у електричне машине“, Нови Сад 2021.
- [2] Бранко Митраковић, Никола Ј. Николић, „Асинхроне машине“, Научна књига, Београд, 1991.
- [3] Емил Леви, Владан Вучковић, Владимир Стрезорски, „Основи електроенергетике – Електроенергетски претварачи“, ФТН Издаваштво, Нови Сад, 2013.
- [4] Веран Васић, Борис Думнић, „Електроенергетски претварачи“, ФТН Издаваштво, Нови Сад, 2022.
- [5] Слободан Н. Вукосавић, „Електричне машине“, Академска мисао, Београд, 2010.
- [6] Веран Васић, „Електричне машине 2“, материјал са предавања, Факултет техничких наука у Новом Саду.

Кратка биографија:

Никола Поповић је рођен 1998. године у Новом Саду. Основне студије је успешно завршио на Факултету техничких наука у Новом Саду, из Електротехнике и рачунарства, на усмерењу Електричне машине и енергетска електроника.

контакт: 98.nikola@gmail.com

ПРИМЕНА ФЕДЕРАТИВНОГ УЧЕЊА У РЕШАВАЊУ ПРОБЛЕМА БИНАРНЕ КЛАСИФИКАЦИЈЕ**IMPLEMENTING FEDERATED LEARNING FOR BINARY CLASSIFICATION SOLUTIONS**Андрија Станишић, *Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИЧКО И РАЧУНАРСКО ИНЖЕЊЕРСТВО**

Кратак садржај – У овом раду је представљено решење које користи федеративно учење за решавање проблема бинарне класификације. Првенствено су анализирале теоријске основе дистрибуираних система и федеративног учења, након чега је представљена имплементација система за федеративну обуку неуронских мрежа. Развијени систем омогућава доделу и агрегацију тежина модела на више дистрибуираних чворова, чиме се елиминирају ризици повезани са централизованим складиштењем података. Рад обухвата преглед коришћених технологија, опис структуре федеративног система, као и евалуацију модела. Овај приступ пружа значајан допринос разумевању федеративног учења и може послужити као основа за даља истраживања и развој напреднијих дистрибуираних система.

Кључне речи: дистрибуирани системи, бинарна класификација, федеративно учење, неуронске мреже

Abstract – This paper presents a solution that employs federated learning to address binary classification problems. The theoretical foundations of distributed systems and federated learning are primarily analyzed, followed by the implementation of a system for federated neural network training. The developed system enables the allocation and aggregation of model weights across multiple distributed nodes, thereby eliminating risks associated with centralized data storage. The paper includes a review of the employed technologies, a description of the federated system's structure, and an evaluation of the model. This approach significantly contributes to the understanding of federated learning and can serve as a foundation for further research and the development of more advanced distributed systems.

Keywords: distributed systems, binary classification, federated learning, neural networks

1. УВОД

Савремени свет је незамислив без рачунара, који су од средине 20. века доживели експоненцијални раст

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији је ментор био др Душан Гајић, ванр. проф.

захваљујући микропроцесорима и рачунарским мрежама. Локалне мреже (енгл. *local-area networks - LAN*), метрополитанске мреже (енгл. *metropolitan-area networks - MAN*) и мреже широког подручја (енгл. *wide-area networks - WAN*) омогућиле су напредак у телекомуникацијама, образовању, здравству и финансијама.

Овај развој је створио основу за сложене системе који ефикасно управљају и размењују информације на глобалном нивоу. Данас, један од најубедљивијих трендова у развоју рачунарских система је свакако вештачка интелигенција. Алгоритми вештачке интелигенције омогућавају рачунарима да уче из података и самостално доносе одлуке. Многе индустрије почињу да користе вештачку интелигенцију за извршавање послова који могу варирати од релативно једноставних до изузетно комплексних. Међутим, ништа од овога не би било могуће да претходно није развијена моћна рачунарска инфраструктура која чини базу вештачке интелигенције. Велики удео ове инфраструктуре чине савремени дистрибуирани системи (енгл. *distributed systems*). Дакле, системи чије се компоненте налазе на различитим, неретко географски удаљеним рачунарима који међусобном комуникацијом чине кохерентну целину. Поред неопходне компутационе снаге, алгоритми вештачке интелигенције захтевају и мноштво релевантних података како би креирали примењиве производе. Међутим, велики обим података и законска ограничења довели су до потребе за новим методама обраде и складиштења података.

Федеративно учење (енгл. *federated learning - FL*) представља револуционарни приступ који омогућава обуку неуронских мрежа на децентрализованим скуповима података. Овај приступ смањује ризике и трошкове повезане са преносом и складиштењем података на централизованим серверима, омогућавајући ефикаснију употребу података и ресурса. Федеративно учење такође омогућава коришћење различитих варијација података који су раније били недоступни због законских и техничких ограничења.

Циљ овог рада је да кроз практичну имплементацију објасни најбитније аспекте федеративног приступа у обучавању неуронских мрежа. Разматрају се кључни аспекти дистрибуираних система и федеративног учења, њихове предности и недостаци, као и технологије и методе коришћене у имплементацији система за бинарну класификацију. На крају, рад

предлаже правце за даљи развој овог иновативног приступа.

2. ТЕОРИЈСКЕ ОСНОВЕ

Савремени рачунарски системи се посматрају као колекција рачунара међусобно повезаних у рачунарску мрежу. Почетна идеја умрежавања рачунара произашла је из потребе за повезивањем постојећих система ради побољшања перформанси и ефикасности. Први приступ умрежавања је интегративни, где се постојећи системи повезују како би формирали моћан систем који омогућава колаборативно решавање комплексних проблема. Други приступ је проширујући, који укључује додавање нових рачунара постојећим системима, чиме се постиже боља отпорност на отказе и повећава доступност.

Централизовани системи имају једну тачку контроле и управљања, што олакшава одржавање и обезбеђује високу поузданост. Међутим, како се комплексност задатака повећава, прелазак на дистрибуиране или децентрализоване системе постаје неопходан.

Дистрибуирани системи распоређују процесе и ресурсе на више рачунара, што омогућава већу поузданост, скалабилност и отпорност на отказе. Децентрализовани системи, пак, елиминишу централну тачку контроле, при чему сваки чвор у мрежи има значајну улогу у обављању функција система, обезбеђујући висок степен сигурности и независности од појединачних тачака отказа.

2.1. Основни принципи дистрибуираних система

Дистрибуирани систем је систем у којем компоненте егзистирају на мрежно повезаним рачунарима, који су неретко и географско удаљени, док је целокупна комуникација и координација система заснована на међусобној размени порука [1]. Из ове дефиниције могуће је издвојити неколико значајних карактеристика дистрибуираних система:

- *Конкурентност* - Дистрибуирани системи омогућавају истовремено, ефикасно и безбедно коришћење ресурса од стране више корисника.
- *Независни отказ компоненти* - Отказ једног дела система не сме утицати на остале делове. Систем треба да настави рад без прекида, аутоматски откривајући и отклањајући отказе.
- *Непостојање глобалног сата* - Географска удаљеност чворова онемогућава коришћење глобалног сата. Стога се за синхронизацију користе логички сатови и размена порука.

2.2. Класификација дистрибуираних система

У односу на начин пројектовања и њихову намену дистрибуирани системи се могу поделити на три основне категорије: дистрибуирани рачунарски системи високих перформанси (енгл. *High-performance distributed computing*), дистрибуирани информациони системи (енгл. *Distributed information*

systems) и прожимајући системи (енгл. *Pervasive systems*) [3].

- *Дистрибуирани рачунарски системи високих перформанси* - Ови системи су дизајнирани за решавање захтевних проблема који захтевају употребу мултипроцесорских рачунара са заједничким меморијским простором. Захваљујући дистрибуцији процеса и ресурса, они могу извршавати комплексне рачунарске задатке са високом ефикасношћу. Ова категорија обухвата GRID рачунаре, кластере и рачунарство у облаку.

- *Дистрибуирани информациони системи* - Ова класа дистрибуираних система повезује велики број мрежно увезаних апликација ради ефикасног управљања подацима. Углавном се састоји од сервера и база података, са изазовом интероперабилности.

- *Прожимајући системи* - Интегрисани у свакодневни живот, прожимајући системи комбинују карактеристике децентрализованих и дистрибуираних система. Пример ових система су сензорске мреже, свеприсутни рачунарски системи, као и мобилни рачунарски системи.

2.3. Архитектуре дистрибуираних система

За успешну реализацију било ког дистрибуираног система неопходно је одредити његову софтверску и хардверску архитектуру. Софтверска архитектура (енгл. *software architecture*) се односи на логичку апстракцију компоненти система. Она дефинише начин представљања компоненти и њихову међусобну комуникацију. Са друге стране, хардверска архитектура (енгл. *hardware architecture*) одговорна је за распоређивање софтверских компоненти на физичке рачунаре чиме се постиже иницијализација дистрибуираног система.

Архитектура софтвера представља структуру дистрибуираног система у смислу одвојено специфицираних компоненти и њихових међусобних односа. Главни циљ је да се обезбеди да структура испуни садашње и будуће захтеве. Кључни аспекти су поузданост, управљивост, прилагодљивост и економичност система [1]. Неке од најзначајнијих архитектура обухватају:

- *Клијент-сервер архитектура* - Основни елементи су клијент и сервер, где сервер пружа услуге, а клијент их потражује слањем захтева. Ова архитектура је изузетно примењива и једноставна за имплементацију, али има проблеме са скалабилношћу и јединственом тачком отказа.

- *P2P мреже* - Централна идеја ових система је равноправност и равномерно укључивање свих процеса у решавање одређеног задатка. Сви чворови покрећу исти програм и нуде идентичне интерфејсе, што омогућава отпорнију и скалабилнију обраду.

Дистрибуирани системи играју кључну улогу у обради великих скупова података, омогућавајући паралелну обраду и обуку комплексних модела машинског учења. Њихова способност паралелизације убрзава процес обраде и повећава поузданост, јер

отказ једног сервера не утиче на целокупан систем. Скалабилност омогућава додавање нових сервера како расту потребе за обрадом. Осим тога, дистрибуирани системи омогућавају рад са хетерогеним подацима на различитим локацијама, интегришући их без премештања на централизован сервер, чиме се подстиче колаборација и иновација у машинском учењу.

2.4. Федеративно учење

Федеративно учење се ослања на дистрибуирану архитектуру обуке у којој је процес тренирања модела подељен између K клијентских уређаја са локалним подацима, а координише га централни сервер. Локалност података побољшава њихову безбедност и минимизује трошкове складиштења. Осим тога, примена федеративног учења омогућава бољу искоришћеност разноликих података, што доводи до креирања робуснијих и прецизнијих модела.

Процес обуке обухвата више федеративних рунди које укључују локално рачунање градијената модела на клијентским уређајима и глобалну агрегацију на централном серверу. На почетку сваке рунде t , случајно се бира скуп од S клијената којима сервер шаље глобално стање алгорита, односно тренутне параметре глобалног модела које су означене са ω_t . Сваки изабрани клијент локално рачуна нове параметре модела на основу глобалног стања и приватног скупа података. Клијенти затим шаљу измене назад централном серверу који их агрегира у ново глобално стање.

Процес агрегације параметара локалних модела на централном серверу назива се федеративна оптимизација. Проблем федеративне оптимизације може се математички формулисати на следећи начин:

$$\min_{w \in \mathbb{R}^d} f(w) \text{ где } f(w) = \frac{1}{n} \sum_{i=1}^n f_i(w)$$

Типично, за проблем машинског учења, функција циља се може дефинисати као $f_i(w) = L(x_i, y_i; w)$, што представља функцију губитка модела на основу узорка (x_i, y_i) из локалног скупа података клијента и параметара модела w [2].

Уз претпоставку да постоји K клијената са локалним подацима, целокупан скуп података величине n је подељен на партиције при чему партиција P_k представља скуп индексираних података на клијенту k . С тим на уму, функција циља федеративне оптимизације може се реформулисати на следећи начин:

$$f(w) = \sum_{k=1}^K \frac{n_k}{n} F_k(w) \text{ где } F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w)$$

У овом изразу $F_k(w)$ представља локалну функцију циља клијента k . Ова функција обухвата рачунање функције циља за сваки појединачни податак у скупу података клијента k . Вредност n_k представља величину скупа података P_k . Тежински фактор n_k / n

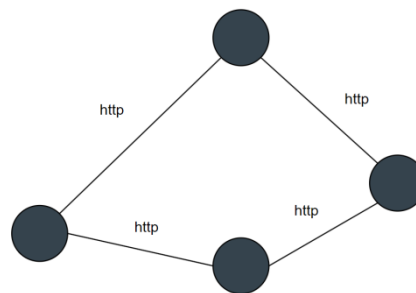
одређује допринос клијента k глобалној функцији циља у складу са величином његовог скупа података.

3. ИМПЛЕМЕНТАЦИОНЕ ТЕХНОЛОГИЈЕ

У процесу имплементације коришћени су програмски језици R, Python и Go, сваки са специфичним разлозима. Програмски језик R је коришћен за иницијалну визуализацију и припрему података због своје извршне подршке за статистичко рачунање и напредну манипулацију подацима, што је омогућило ефикасну обраду и анализу података. Python је изабран за развој и тестирање алгоритама машинског учења захваљујући својој флексибилности и богатом екосистему библиотека као што су NumPy, Pandas и TensorFlow, који омогућавају лако и брзо креирање и експериментисање са моделима. Програмски језик Go је коришћен за имплементацију серверских компоненти и комуникацију између чворова захваљујући својој способности за паралелну обраду великих количина података, што је обезбедило високе перформансе и поузданост система. Ове технологије су заједно омогућиле ефикасну и прецизну обраду података, испуњавајући комплексне захтеве система федеративног учења.

4. ИМПЛЕМЕНТАЦИЈА ФЕДЕРАТИВНОГ СИСТЕМА

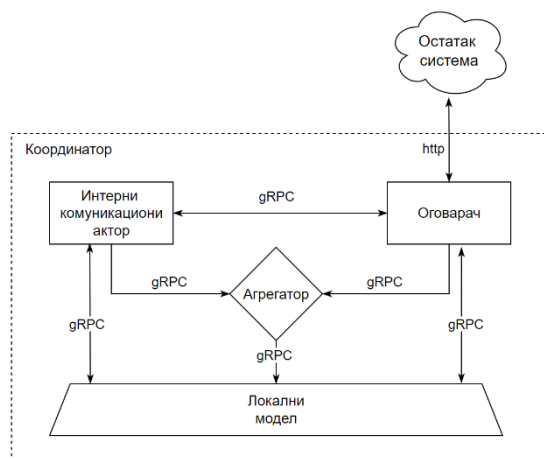
Имплементирани систем је заснован на федеративном учењу. Централна идеја система је извршавање бинарне класификације над улазним подацима. Конкретно, систем на основу обележја, која су у овом случају симптоми пацијената, даје предикцију о томе да ли је пацијент заражен вирусом или не. Систем је имплементиран као скуп чворова који међусобно комуницирају слањем HTTP захтева. Архитектура система, приказана на слици 1, представља класичну P2P мрежу.



Слика 1 – Архитектура система

Сваки чвор је имплементиран као колекција актора који међусобно комуницирају путем gRPC комуникације. Актори може бити комуникациони, агрегатор или координатор. Додатно, у оквиру сваког чвора налази се имплементација Flask сервера који покреће неуронску мрежу и нуди интерфејс за добављање кључних података везаних за стање модела. Имплементирана неуронска мрежа представља облик вишеслојног перцептрона (енгл. *Multilayer Perceptron - MLP*). Поред пропагације унапред, имплементирана је и пропагација уназад (енгл. *backpropagation*), ради постизања бољих

перформанси. Интерна структура чвора приказана је на слици 2, док се у наставку налази опис најзначајнијих карактеристика ове структуре.



Слика 2 – Интерна структура чвора

1. Координатор је задужен за дефинисање тока реализације федеративне обуке. Он посредује у интерној комуникацији између актора, управља креирањем и уништавањем актора у зависности од параметара као што су обим посла и оптерећеност мреже. У контексту система, координатор осигурава синхронизовани тренинг модела, складишти међурезултате обуке и иницијализује тежине модела насумичним вредностима које дистрибуира свим чворовима.

2. Агрегатор је одговоран за прикупљање и комбиновање тежина модела са различитих чворова у систему. Он прима локалне моделе са клијентских уређаја, агрегира их применом агрегационе функције и ажурира глобални модел. Ажуриране тежине модела се затим пропагирају назад чворовима за наредну епоху обуке. Агрегатор осигурава да модели буду правилно синхронизовани и ажурирани током целокупног процеса обуке.

3. Комуникациони актор је кључна компонента федеративног учења, одговоран за координацију и комуникацију током процеса обуке модела. Постоје два типа актора: оговарач и интерни актор. Оговарач омогућава размену информација између чворова система. Шаље и прима информације о тежинама модела и другим подацима, омогућавајући синхронизовану обуку модела. Интерни актор комуницира са сервером који обрађује податке и врши обуку модела. Функционише као интерфејс између чворова и сервера модела, прикупља тежине и бајасе из неуронске мреже и шаље их агенту агрегатору.

5. ЕВАЛУАЦИЈА МОДЕЛА

Добијене тежине из процеса федеративног обучавања се користе за тренирање модела над тестним скупом података. За евалуацију модела одабрана је F1 метрика, која представља меру перформанси бинарног класификатора и комбинује прецизност (енгл. *precision* - p) и одзив (енгл. *recall* - r). Прецизност дефинише тачност позитивних

предикција, док одзив мери ефикасност у утврђивању позитивних инстанци. Висока вредност F1 метрике означава да је модел ефикасан приликом правилне идентификације свих позитивних инстанци и минимизирању броја лажно позитивних и лажно негативних предвиђања. Обука модела над тестним скупом података, као и евалуација, реализоване су покретањем R скрипте. Модел је показао задовољавајуће перформансе са F1 метриком од 0.89, што указује на добру равнотежу између прецизности и одзива.

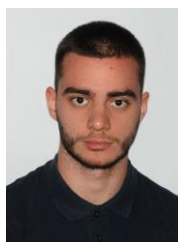
6. ЗАКЉУЧАК

У овом мастер раду истраживана је примена федеративног учења у решавању проблема бинарне класификације. Објашњени су основни принципи дистрибуираних система и њихова примена у машинском учењу, као и проблеми са којима се суочавају приликом рада са великим скуповима података. Федеративно учење је уведено као нови приступ обуци модела над децентрализованим скуповима података. Имплементирани систем успешно је применио федеративно учење за бинарну класификацију података. Истраживања су показала могућности за даља унапређења кроз обуку на релевантнијим скуповима података, оптимизацију алгоритама и унапређење алгоритама агрегације.

7. ЛИТЕРАТУРА

- [1] George F Coulouris, Jean Dollimore, and Tim Kindberg. Distributed systems: concepts and design. pearson education, 2005.
- [2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüey-Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, pages 1273--1282. PMLR, 2017.
- [3] Maarten Van Steen and Andrew S Tanenbaum. Distributed systems. Maarten van Steen Leiden, The Netherlands, 2017.

Кратка биографија:



Андреја Станишић рођен је 25. јула 2000. године у Београду. Завршио је Гимназију „Исидора Секулић“ 2019. године након чега уписује Факултет техничких наука, смер Рачунарство и аутоматика. Дипломски рад је одбранио 2023. године када уписује мастер студије на студијском програму Рачунарство и аутоматика.

**ANALIZA PRELAZNIH REŽIMA NA MODELU 20[kV] MREŽE PRILIKOM
AUTOMATSKOG PONOVOG UKLJUČENJA****ANALYSIS OF TRANSIENT REGIMES IN A 20kV NETWORK MODEL DURING AUTO-
RECLOSING**

Milan Gvozdenović, Stevan Cvetičanin, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U Matlab/Simulink okruženju nepraviljen je model dvostrano napajanog 20 [kV] voda na kome je implementirana i testirana prekostrujna zaštita sa automatskim ponovnim uključenjem (APU), a s obzirom na ograničenja koja model ima izvršeno je praviljenje modela 20[kV] izvoda baziranog na realnom izvodu 20[kV] mreže koji se može naći u praksi u okviru elektroenergetskog sistema. Model baziran na realnom izvodu 20[kV] mreže uzima u obzir stvarne karakteristike komponenti i parametre mreže. Na ovako razvijenom modelu vršene su simulacije kratkih spojeva i analizirani su rezultati koji se razlikuju od tipa, mesta i dužine trajanja kratkog spoja. Cilj rada je analiza prelaznih procesa prilikom automatskog ponovnog uključenja (APU).

Ključne reči: Automatsko ponovno uključenje (APU), distributivne mreže, prekostrujna zaštita

Abstract – In the Matlab/Simulink environment, a model of a 20 kV double-fed feeder with implemented and tested overcurrent protection and auto-reclosing (AR) was created. Considering the limitations of the model, a model of a 20 kV feeder based on a real 20 kV network feeder, which can be found in practice within the power system, was created. The model based on the real 20 kV network feeder takes into account the actual characteristics of the components and network parameters. On this developed model, short circuit simulations were performed and the results were analyzed, which vary based on the type, location, and duration of the short circuit. The aim of the work is to analyze the transient processes during automatic reclosing (AR).

Keywords: Auto-reclosing, distribution networks, overcurrent protection

1. UVOD

Distributivne mreže igraju ključnu ulogu u osiguravanju pouzdane isporuke električne energije. Kako bi se osigurala sigurnost distributivnih mreža, ključno je primeniti adekvatne zaštitne mehanizme. Jedan od ključnih aspekata zaštite distributivnih mreža jeste prekostrujna zaštita.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Stevan Cvetičanin

Prekostrujnu zaštitu koristimo za detekciju i ograničenje prekomernih struja koje mogu nastati kao posledica kratkih spojeva ili drugih kritičnih događaja tokom eksploatacije. U ovom radu modelovan je dvostrano napajan vod u Matlab/Simulink okruženju na kome vršimo simulacije kratkih spojeva te upoređujemo njegove s rezultatima dobijenim iz modela baziranog na realnoj distributivnoj mreži takođe razvijene u Matlab/Simulink okruženju. Fokusiramo se na evaluaciju performansi prekostrujne zaštite i automatskog ponovnog uključenja releja te simuliramo kratke spojeve radi procene stabilnosti sistema.

**2. TEORIJSKI OKVIR I OSNOVNE
PRETPOSTAVKE**

Da bi počeli sa izradom modela moramo da krenemo od osnovnih teorijskih razmatranja i pretpostavki koje su nam ključne za rad.

2.1. Konfiguracija mreža srednjeg napona

Potrebno je da razumemo da distributivne mreže prema konfiguraciji mogu da se podele na: radijalne, prstenaste i upetljane, model koji se razmatra u radu prema konfiguraciji pripada radijalnim mrežama [1].

2.2. Kratki spojevi

Metalni kratak spoj je međusobno direktno povezivanje tačaka različitih faza sa ili bez povezivanja sa zemljom odnosno neutralnim provodnikom. Postoje četiri osnovna kratka spoja a to su: jednopolni, dvopolni, dvopolni sa zemljom i trolpolni kratak spoj, kao i trolpolni sa zemljom koji se ne analizira zasebno jer ima istu prirodu kao i trolfazni kratak spoj. Prilikom izrade master rada simulirani su svi tipovi kratkih spojeva, dok je na na slici 3 prikazan samo trolpolni kratak spoj [2].

2.3. Automatsko ponovno uključenje (APU)

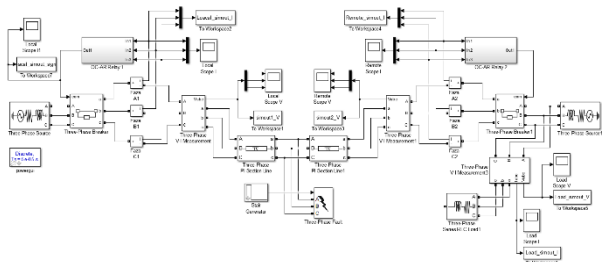
Prilikom nastanka kratkotrajnih prolaznih kvarova dolazi do delovanja zaštitom i određeni vod na kome se desio kvar se isključi, nakon isteka prethodno definisanog vremena pauze vod se ponovo automatski uključi, pošto je električni luk ugašen vod može nastaviti sa normalnim radom. Ovaj ciklus od nastanka kvara, isključenja, pauze i ponovnog uključenja traje veoma kratko (od 0,5s) tako da potrošači u većini slučajeva i ne primete prekid napajanja, takođe od prekidača se zahteva mogućnost za dva ponovna automatska uključenja. Zato je važno da kod modernih postrojenja imamo primenu savremenih prekidača koji imaju mogućnost isključenje i automatskog ponovnog uključenja (APU) [3].

3. RAZVOJ MODELA DVOSTRANO NAPAJANOG VODA

Matlab – SimPowerSystems je računarski program koji nam omogućava simulaciju sa već izgrađenim modelima ali i izradu sopstvenog modela mreže ili sistema koji je moguće analizirati. U Matlab – SimPowerSystems razvijen je model od ponudjenih blokova u koje su uneseni odgovarajući parametri. Na slici 1 je prikazan simulink model elektroenergetske mreže na kome je simulirano APU nakon isključenja voda uzrokovano zadatim kratkim spojem. Prilikom pronalaženja rešenja za način upravljanja prekidačima, podešavanjem prekostrujne zaštite i APU, bilo je potrebno krenuti od uprošćenog modela mreže (slika 1), sa manje elemenata. U ovom slučaju izabran je dvostrano napajani vod sa jednim potrošačkim područjem. Cilj je da manipulacijama na mreži u slučaju kratkog spoja potrošačko područje bude izolovano od kvara i napojeno preko drugog izvora napajanja koje nije zahvaćeno kvarom [4].

3.1. Ograničenja modela dvostrano napajnog voda

Ograničenja prilikom simuliranja APU na ovakvom modelu su brojne. Ograničeni smo sa brojem čvorova na kojima možemo simulirati kratke spojeve na modelu zbog jednostavne konfiguracije modela, ograničeni smo sa brojem ciklusa APU, teže je podesiti parametre rejela, teže je uneti potrebne vrednosti za struju isključenja, vremensko kašnjenje prekidača i podešenje ponovnog uključenja prekidača.



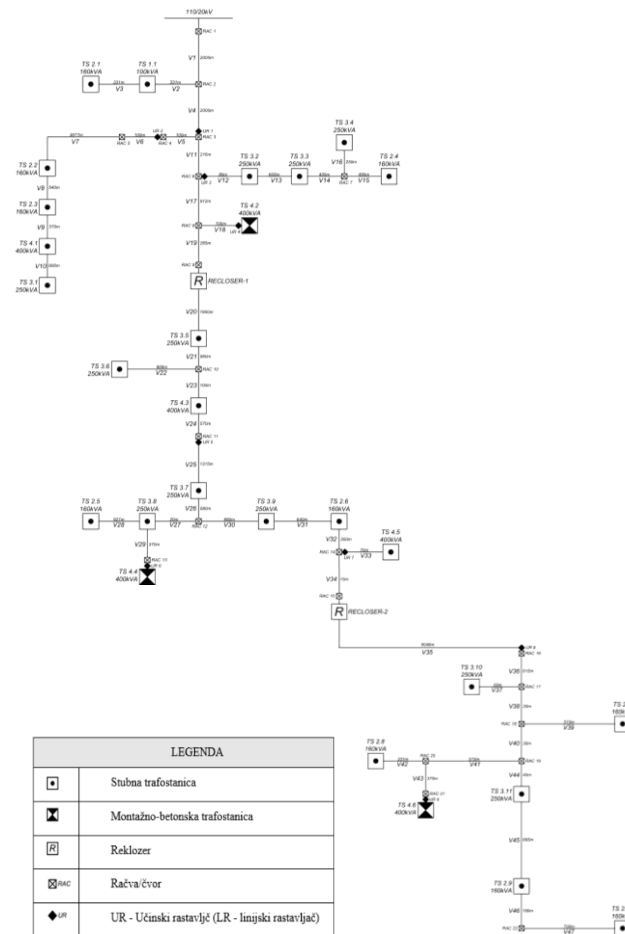
Slika 1. Model dvostrano napajnog voda sa implementiranom prekostrujnom zaštitom i APU

3. RAZVOJ MODELA BAZIRANOG NA REALNOM IZVODU 20[kV] MREŽE

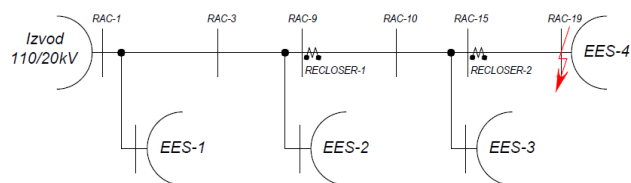
Zbog navedenih ograničenja kod modela dvostrano napajnog 20[kV] voda postoji potreba za razvojem modela baziranog na realnom izvodu 20[kV] distributivne mreže u Matlab/Simulink okruženju, uzimajući u obzir stvarne karakteristike komponenti i parametara mreže. (slika 2). Model mreže koji analiziramo se sastoji iz 22 čvora, 47 vodova, 28 trafostanica sa 28 potrošačka područja i 2 reklozere. APU je podešeno tako da ima radnu sekvencu prema izrazu:

$$O - 0,5s - CO - 5s - CO \quad (1)$$

Gde je O–Isključenje, C–Uključenje, CO–Ponovno uključenje. Druga pauza od 5s je izabrana zbog bolje preglednosti i kvalitetnije analize odziva iako je u praksi vreme podešenja druge pauze podešeno na 20s. Izgled mreže je prikazan na slici 2 dok je zbog jednostavnijeg prikaza modela mreže služimo se uprošćenim modelom prikazanim na slici 3.



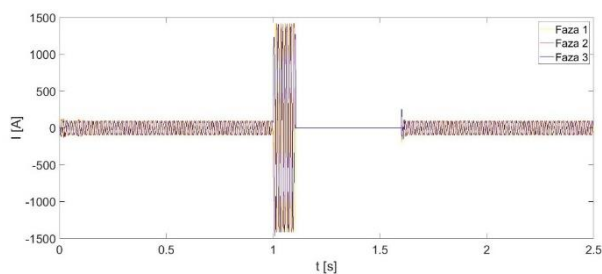
Slika 2. Blok šema 20kV mreže prema kojoj je napravljen Matlab/Simulink model



Slika 3. Uprošćen model voda sa implementiranom prekostrujnom zaštitom i APU

4. ANALIZA APU NAKON TROPOLNOG KRATKOG SPOJA

Prilikom simulacije tropolnog kratkog spoja na čvoru RAC19 prikazanog na slici 3, kvar je simuliran sa početkom u 1s i završetkom u 1,5s dok celokupna simulacija traje 2,5s. Kvar je simuliran na kraju voda što dovodi do reagovanja zaštitom, odnosno najbliža prekostrujna zaštita sa APU će reagovati 0,1s od nastanka kvara posle čega dolazi do isključenja voda zahvaćenog kvarom. Prema izrazu (1) nakon isključenja voda zadata je pauza u trajanju od 0,5s nakon koje se automatski ponovo uključuje vod i obzirom na situaciju da nakon APU ne postoji registrovan kratak spoj, sistem se vraća u noramalan režim rada što je prikazano na slici 4, dok su rezultati dati u tabeli 1.



Slika 4. Struja merena u čvoru RAC15 prilikom simulacije topolnog kratkog spoja u čvoru RAC19

Naziv čvora	Režim rada	Vreme [s]	I [kA]		
			L1	L2	L3
RAC15	normalan	0-1	0,099	0,106	0,097
	kr. spoj	1-1,1	14,20	14,20	14,10
	prva pauza	1,1-1,6	0	0	0
	prel. proces	1,6-1,7	0,227	0,115	0,254
	normalan	1,7-2,5	0,097	0,096	0,096
RAC9	normalan	0-1	0,149	0,149	0,148
	kr. spoj	1-1,1	14,10	14,50	14,20
	prva pauza	1,1-1,6	0,062	0,078	0,064
	prel. proces	1,6-1,7	0,200	0,209	0,220
	normalan	1,7-2,5	0,147	0,147	0,148
RAC1	normalan	0-1	0,226	0,235	0,259
	kr. spoj	1-1,1	14,20	14,67	14,27
	prva pauza	1,1-1,6	0,129	0,152	0,134
	prel. proces	1,6-1,7	0,272	0,274	0,294
	normalan	1,7-2,5	0,212	0,210	0,210

Tabela 1. Maksimalne vrednosti struje merene u čvoru RAC15, RAC9 i RAC1 prilikom simulacije topolnog kratkog spoja u čvoru RAC19

Sa slike 4 i iz tabele 5 analizom rezultata dobijene su očekivane vrednosti struja kratkog spoja u odnosu na zadatu situaciju. Grafik sa slike 4 podeljen je u nekoliko vremenskih celina radi lakše analize:

- 0s - 1s – Normalan režim rada
- 1s - 1.1s – Registrujemo topolni kratak spoj
- 1.1s - 1.6s – Isključenje dela voda (EES-4)
- 1.6s- 5s – APU i uspostavljanje normalnog režima rada

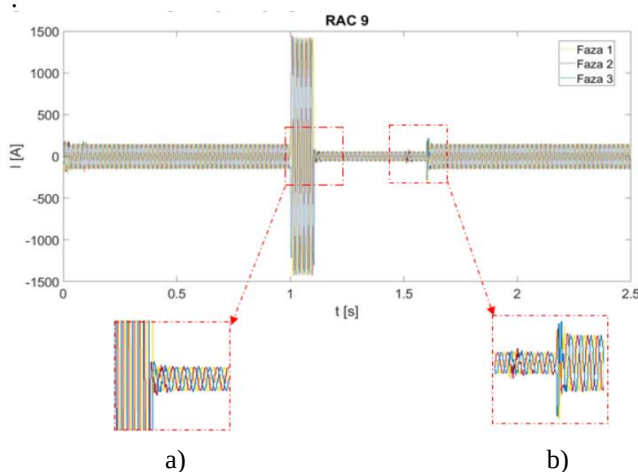
Odabrano vreme trajanja kratkog spoja je takvo da kvar bude otklonjen u prvom ciklusu iako je ostavljena mogućnost za testiranje kratkih spojeva na modelu tako da dobijemo situaciju da kvar bude otklonjen u drugom ciklusu, ali isto tako je ostavljena mogućnost za kreiranje situaciju gde se vrši demonstracija neuspešnog APU, ove situacije su analizirane u master radu.

4. ANALIZA PRELAZNIH PROCESA PRILIKOM APU VODA

Na slici 4 prikazan je karakterističan oblik signala koji se javlja prilikom uključanja voda u elektroenergetskom sistemu. Ovaj signal predstavlja prelazne procese - odnosno, promene koje se događaju dok sistem ne dođe do svog stabilnog stanja nakon perturbacije, kao što je uključanje voda.

Kada se vod uključi, u elektroenergetskom sistemu se javljaju prelazni fenomeni koji su rezultat naglih promena u naponu i struji. To je vidljivo kao oštar vrh na početku grafikona. Nakon tog vrha, signal se stabilizuje u obliku sinusoidalnih talasa koji predstavljaju stabilan rad sistema.

Uzrok ovih prelaznih procesa je reaktansa komponenta sistema kao što su transformatori, prekidači, i kablovi. Oni imaju induktivne i kapacitivne komponente koje se "protive" iznenadnoj promeni struje, što rezultira oscilacijama dok se sistem ne stabilizuje. Prilikom uključanja, energija se naglo oslobađa u sistemu, što uzrokuje oscilacije struje i napona. Dodatni faktori koji mogu doprineti prelaznim procesima uključuju karakteristike izvora napajanja, vrstu opterećenja na mreži, kao i brzinu i način na koji se vod uključuje.



Slika 5. a) prelazni procesi prilikom isključenja voda b) prelazni procesi prilikom APU voda nakon simulacije topolnog kratkog spoja u čvoru „RAC19“

Prilikom analize izabran je slučaj gde se simulira topolni kratak spoj u čvoru RAC19, a merenje je vršeno u čvoru RAC9, ali treba napomenuti da se ovi prelazni procesi dešavaju nakon uključanja voda bez obzira na tip, mesto i vreme trajanja kratkog spoja nakon kog ponovo uključujemo vod.

3. ZAKLJUČAK

Ovim radom pružamo dublji uvid u važnost prekostrujne zaštite i automatskog ponovnog uključanja u osiguravanju stabilnosti i pouzdanosti snabdevanja električnom energijom distributivnih mreža. Analizirajući rezultate dobijene iz teorijskog i realnog modela distributivne mreže, primetili smo razlike koje ukazuju na ograničenja teorijskog modela.

Ograničenja teorijskog modela proizlaze iz činjenice da smo ograničeni brojem ciklusa automatskog ponovnog uključanja, kao i brojem čvorova na kojima možemo simulirati kratke spojeve. Suprotno tome, realni model distributivne mreže pruža nam realističnije podatke i omogućava nam simulaciju svih vrsta kratkih spojeva na svim čvorovima u sistemu.

Kroz ovu analizu, ističemo važnost korištenja realnih modela distributivnih mreža u istraživanjima i analizama, jer nam oni omogućuju preciznije rezultate i bolje razumevanje ponašanja sistema u stvarnim uslovima rada. Takođe, realni modeli pružaju nam mogućnost simuliranja trajanja kratkog spoja, što je ključno za uspješno automatsko ponovno uključanje u prvom ili drugom ciklusu, kao i prepoznavanje situacija u kojima automatsko ponovno uključanje može biti nespešno.

Stoga, daljnja istraživanja i implementacija prekostrujne zaštite i automatskog ponovnog uključanja trebaju uzeti u obzir ova ograničenja teorijskih modela i preferirati korištenje realnih modela distributivnih mreža kako bi se osigurala optimalna sigurnost i pouzdanost električne mreže.

Na temelju upoređivanja rezultata iz teorijskog i realnog modela distributivne mreže možemo zaključiti o razlici između predviđenih i stvarnih performansi sistema. Ova analiza pruža uvid u važnost preciznosti modeliranja i simulacije pri dizajniranju i analizi distributivnih mreža.

4. LITERATURA

- [1] Vladimir C. Strezoski, "Osnovi elektroenergetike", FTN Izdavaštvo, Novi Sad, 2015.
- [2] Strahil J. Glušavac, "Osnovni principi projektovanja u mrežama srednjeg i niskog napona", FTN Izdavaštvo, Novi Sad, 2014.
- [3] Milenko Đurić, Zoran Stojanović, "Relejna zaštita", KIZ „Centar“, Beograd, 2014.
- [4] <https://www.mathworks.com/products/matlab.html>

Kratka biografija:



Milan Gvozdenović rođen je u Beogradu 25.10.1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Distribuirani elektroenergetski resursi odbranio je 2024.god.

kontakt: milgvo733@gmail.com



Stevan Cvetičanin rođen je u Bačkoj Palanci 04.10.1986. god. Od 2023. radi kao vanredni profesor na Katedri za elektroenergetiku i primenjeno softversko inženjerstvo Fakulteta tehničkih nauka, Univerziteta u Novom Sadu.

kontakt: stevan.cveticanin@uns.ac.rs

**PREDIKCIJA HRONIČNE BOLESTI BUBREGA NA OSNOVU SCINTIGRAFSKIH
SNIMAKA UPOTREBOM METODA MAŠINSKOG UČENJA****PREDICTION OF CHRONIC KIDNEY DISEASE BASED ON SCINTIGRAPHIC IMAGES
USING MACHINE LEARNING METHODS***Kristina Vajda, Fakultet tehničkih nauka, Novi Sad***Oblast – Veštačka inteligencija i mašinsko učenje**

Kratak sadržaj – U ovom radu istražena je primena mašinskog učenja za predviđanje stadijuma hronične bolesti bubrega na osnovu podataka ispitanika i slika dobijenih scintigrafijom. Analizirani su klinički i demografski podaci kao i podaci dobijeni iz slika, a primenjeni modeli su: metod k - najbližih suseda, stablo odluke, metod slučajne šume i gradijentnog pojačavanja. Rezultati su pokazali da metod k - najbližih suseda ima najnižu tačnost od 24%, dok su stablo odluke i metod slučajne šume pokazali znatno bolje performanse sa tačnošću od 96%. Model gradijentnog pojačavanja se najbolje pokazao postigavši tačnost od 100%.

Ključne reči: Mašinsko učenje, scintigrafija, hronična bolest bubrega

Abstract – This paper investigates the application of machine learning data for predicting the stage of chronic kidney disease based on examinee data and images obtained by scintigraphy. Clinical and demographic data, as well as data obtained from images, were analyzed, and the following models were applied: k-nearest neighbors, decision tree, random forest and gradient boosting. The results showed that the k-nearest neighbors method had the lowest accuracy of 24%, while the decision tree and random forest methods showed significantly better performance with an accuracy of 96%. The gradient boosting model performed the best, achieving an accuracy of 100%.

Keywords: Machine learning, scintigraphy, chronic kidney disease

1. UVOD

Veštačka inteligencija i mašinsko učenje i njihova primena predstavljaju jedan od najvećih izazova XXI veka. Na prvi pogled primena deluje poprilično jednostavno i tiče se automatizacije osnovnih čovekovih delatnosti, kao što je rad u fabrici, rad na otvorenom pa čak i prevoz materijala, ali danas mašinsko učenje i veštačka inteligencija sve više i više uspevaju da u pojedinim industrijama zamene čoveka mašinom. Poslednjih godina, veštačka inteligencija je dobila široku primenu u medicini korišćenjem specijalizo-

vanih alata za pronalazak veza unutar velikih skupova podataka. Otkriće teških bolesti u ranim stadijuma, automatizovano prepoznavanje bolesti koje u nekim slučajevima prevazilazi veštine medicinskih stručnjaka govori o važnosti naučnih radova baziranih na veštačkoj inteligenciji i mašinskom učenju u medicini.

Scintigrafija je dijagnostička metoda koja koristi radioaktivne izotope za generisanje slika unutrašnjih organa. Ove slike su važne za identifikaciju različitih zdravstvenih stanja, međutim njihova analiza može biti izazovna. Jedan od najvećih izazova u radu sa medicinskim podacima jeste osiguravanje tačnosti i pouzdanosti rezultata, što zahteva dobre tehnike preprocesiranja i analize podataka. U ovom radu istražuje se kako se mašinsko učenje može koristiti za prevazilaženje ovih izazova i poboljšanje dijagnostičkih procesa.

Predmet istraživanja rada predstavlja metode i tehnike pomoću kojih, na osnovu kliničkih i demografskih podataka kao i podataka dobijenih iz slike, se može predvideti stadijum hronične bolesti bubrega.

Cilj ovog rada je da pruži sveobuhvatan pregled primene mašinskog učenja u analizi scintigrafskih podataka i rešenja u ovom polju.

2. TEORIJSKE OSNOVE I DEFINICIJE

U ovom poglavlju biće predstavljene teorijske osnove algoritama mašinskog učenja koji su korišćeni za predikciju stadijuma hronične bolesti bubrega.

2.1. Metod k – najbližih suseda

Metod k - najbližih suseda (eng. k – nearest neighbors – kNN) spada u grupu metoda kasnog učenja, koje podrazumevaju odlaganje obrade uzoraka za obuku do trenutka kada treba klasifikovati neobeleženi uzorak. Algoritam k - najbližih suseda je jedan od najjednostavnijih algoritama mašinskog učenja za probleme klasifikacije. Zbog svoje jednostavnosti, on je često i jedan od prvih algoritama mašinskog učenja sa čijom primenom se pokušava rešiti određeni problem. U pitanju je intuitivan algoritam koji klasifikuje nepoznati uzorak na osnovu klase pripadnosti susednih uzoraka iz skupa za obuku. Prilikom klasifikacije neobeleženog uzorka, na osnovu odabrane metrike se meri njegova udaljenost od svakog uzorka iz skupa za obuku. Klasa neobeleženog uzorka predviđa se na osnovu klasne

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dunja Vrbaški, red. prof.

pripadnosti k uzoraka iz skupa za obuku koji su najbliži neobeleženom uzorku, jer se pretpostavlja da je dati uzorak sličniji uzorcima iz skupa za obuku koji su mu bliži u prostoru obeležja [1,2].

2.2. Stablo odluke

Stablo odluke predstavlja jednu od često korišćenih metoda nadgledanog učenja. Može da se koristi za rešavanje i klasifikacionih i regresionih problema i može da radi i sa numeričkim i sa kategoričkim obeležjima. Koren stabla predstavlja čvor koji sadrži skup svih uzoraka, i od njega se stablo grana, odnosno vrši se sukcesivna particija skupa uzoraka na dva disjunktne podskupa. Particija se formira na osnovu vrednosti obeležja koje se promeni kao najznačajnije u datom trenutku. Prema odgovoru na postavljeno pitanje, odnosno prema vrednosti datog obeležja za svaki pojedinačni uzorak, skup pridružen čvoru deli se na dva podskupa, čime se ujedno formiraju i dva nova čvora. Algoritam pronalazi optimalno pitanje na osnovu kojeg vrši particiju čvora prolaskom kroz skup mogućih pitanja i procenom u kojoj meri će čvorovi formirani na osnovu svakog od tih pitanja biti čisti. Čistoća određenog skupa, odnosno čvora, podrazumeva što izraženiju dominaciju jedne od klasa u njemu. Kriterijum za zaustavljanje grananja može biti dobijanje potpuno čistog čvora, a može se zasnivati i na ograničenju u pogledu maksimalnog broja uzoraka u krajnjem čvoru ili maksimalne dubine stabla. Klasa test uzorka kod klasifikacionih problema najčešće se određuje kao klasa koja preovladava među uzorcima u listu [1].

2.3. Metod slučajne šume

Metod slučajne šume (eng. random forest) je metoda ansambalskog učenja koja koristi stabla odluke kao jednostavne klasifikatore. Ideja je da se obuci mnoštvo stabala odluke, a donošenje krajnje odluke o klasi ili vrednosti nepoznatog uzorka vrši se glasanjem u slučaju klasifikacije. Na početku se, primenom metode ponovnog uzorkovanja iz skupa za obuku, formira M novih skupova. Metoda ponovnog uzorkovanja podrazumeva kreiranje novih skupova za obuku nasumičnim izvlačenjem uzoraka sa vraćanjem iz originalnog skupa. Kada je svaki od novodobijenih skupova za obuku iste veličine kao originalni skup, on tipično sadrži oko $2/3$ jedinstvenih uzoraka. Svako od stabala odluke u okviru metode slučajne šume zatim se obučava, odnosno formira na osnovu jednog od M novodobijenih skupova za obuku. Kod ovog algoritma greška se procenjuje na osnovu tačnosti predviđanja za uzorke koji u postupku ponovnog uzorkovanja nisu nijednom izvučeni iz polaznog skupa [1].

2.4. Gradijentno pojačavanje

Po pitanju preciznosti, metode gradijentnog pojačavanja (eng. gradient boosting) su među najboljim metodama pojačavanja i mašinskog učenja uopšte. Obučavanje se takođe može smatrati relativno efikasnim. Zbog toga su vrlo popularne u primenama. Osnovna ideja dolazi iz gradijentnih metoda optimizacije, koje počivaju na popravljanju tekućeg rešenja optimizacionog problema dodavanjem vektora proporcionalnog negativnoj vrednosti gradijenta funkcije koja se minimizuje. Ovo ima smisla, pošto negativna vrednost gradijenta pokazuje smer

opadanja funkcije. U kontekstu pojačavanja, ova ideja se realizuje tako što se model F_m kojim se dopunjuje ansambl F_{m-1} odredi tako da aproksimira gradijent greške po funkciji F_{m-1} [3].

3. METODOLOGIJA

U ovom poglavlju biće predstavljen proces implemetacije sistema za klasifikaciju stadijuma hronične bolesti bubrega koji se kreće od 1 (početni stadijum) do 5 (napredni stadijum).

Skup podataka je javno dostupan [4] i uključuje demografske i kliničke podatke ispitanika kao što su starost, pol, težina, visina, sistolni i dijastolni krvni pritisak pre i posle dinamičke renalne scintigrafije, BMI, BSA – H (površina tela prema Haycock metodi) i BSA – D (površina tela prema DuBois metodi). Sastoji se od 107 ispitanika od kojih je svaki pojedinačni ispitanik imao minimum 4, a maksimum 6 slika. Slike koje su snimane su posteriorne i anteriorne projekcije dinamičke renalne studije, posteriorne i anteriorne projekcije flood izvora odnosno transmisijski podaci i snimci nakon mokrenja u posteriornim i anteriornim projekcijama.

Stadijum hronične bolesti bubrega je procenjen na osnovu brzine glomuralne filtracije (eng. glomerular filtration rate - GFR). Brzine glomuralne filtracije je procenjena regresijom od klirensa etilendiamintetrasirćetne kiseline (EDTA) dostupnog kod nekih ispitanika i merkaptacetiltriglicin (MAG3) klirensa dostupnog kod svih ispitanika.

Kako bi se podaci iz ovih skupova podataka pripremili za modele mašinskog učenja, kategoričko obeležje koje se odnosi na pol ispitanika je zamenjeno numeričkim, odnosno kategorička vrednost M menja se numeričkom vrednošću 0, dok se F menja sa 1.

Uzevši u obzir da je srednja vrednost osetljiva na vrednosti koje odstupaju od ostalih, u ovom radu nedostajuće vrednosti su zamenjene medijanom.

Skup podataka proširen je novim atributima koji bi bili zasnovani na podacima dobijenih iz slike. Pre samog izvlačenja obeležja iz slika izvršeno je uklanjanje šuma iz slike i normalizacija intenziteta piksela. Takođe je bilo potrebno primeniti promenu veličine svih slika, da bi slike bile istih dimenzija, što je važno prilikom korišćenja algoritama mašinskog učenja.

Sledeći korak u pripremi podataka je dobijanje segmentiranih slika.

U radu [5], Otsu-ova metoda se analizira kao jedna od najvažnijih globalnih metoda za određivanje praga slike. Osim što je ova metoda istaknuta kao jedna od najpopularnijih i najčešće korišćenih metoda za određivanje praga slike zbog njene jednostavne implementacije i dobrih performansi, pokazala se efikasnom za slike sa dobrim kontrastom između objekta i pozadine. Ova metoda je ocenjena kao robusna i efikasna tehnika za određivanje praga slike u mnogim situacijama, stoga će biti primenjena za dobijanje segmentirane slike u ovom radu. Otsu-ov prag se koristi za automatsko određivanje praga, a prag se izračunava na osnovu

histograma slike i traži se vrednost koja maksimizira razliku između dve klase piksela [6].

Taj prag se zatim primenjuje na sliku i dobija se binarna slika u kojoj pikseli veći od praga imaju vrednost True, a pikseli manji od praga imaju vrednost False. Na ovaj način je moguće u slici razlikovati objekte od pozadine. Uklonjeni su svi binarni objekti koji se nalaze na ivicama slike da bi se poboljšala tačnost segmentacije. Zatim su obeleženi svi povezani segmenti istom oznakom.

Naredni korak podrazumeva izračunavanje srednje vrednosti i standardne devijacije svih piksela u slici, kao i asimetriju (eng. skewness) i špicastost (eng. kurtosis) rasporede piksela. Ova statistička obeležja pružaju osnovne informacije o distribuciji piksela u slici, koje su korisne za razumevanje globalnih karakteristika slike. Izračunata su svojstva označenih segmenata u segmentiranoj slici i iz svakog segmenta se za obeležja uzimaju površina regiona, obim regiona, solidnost (odnos površine i konveksnog omotača regiona) i ekscentričnost regiona (mera odstupanja regiona od idealne elipse). Kombinovanjem svih ovih izračunatih vrednosti sa kliničkim podacima generiše se konačan skup obeležja.

Podaci su podeljeni na trening i test skup u odnosu 80:20.

Na ovaj način skup podataka od 578 uzoraka podeljen je na trening i test skup, tako da trening skup sadrži 462 uzorka, dok test skup sadrži 116 uzoraka.

Nakon podele na trening i test skup, vrši se dodavanje nula gde je potrebno kako bi se osiguralo da svi skupovi obeležja odnosno svi uzorci imaju istu dužinu.

S obzirom da skup podataka relativno mali, izvršena je augmentacija podataka da bi se proširio trening skup podataka koji će se koristiti za obučavanje modela.

Augmentacija podataka izvršena je rotacijom svake slike za 90, 180 i 270 stepeni. Za svaku rotiranu sliku ponovljen je postupak ukanjanja šuma, normalizacije intenziteta piksela, promena veličine slike, segmentacije slike i izvlačenje obeležja iz slike. Rotacija je oblik augmentacije podataka koja povećava raznolikost skupa podataka, što može poboljšati generalizaciju modela pri upotrebi za neviđene podatke.

Od ukupnih 462 uzorka iz trening skupa nakon izvršene augmentacije podataka dobijeno je 1848 uzoraka.

U ovom radu za optimizaciju hiperparametara korišćena je pretraga po mreži (eng. grid search). Pretraga po mreži je najjednostavnija tehnika koja se koristi kada broj hiperparametara i njihov opseg nisu preveliki [7]. GridSearchCV() pretražuje zadati skup hiperparametara i pronalazi najbolju kombinaciju. Za svaku kombinaciju hiperparametara, GridSearchCV() vrši unakrsnu validaciju i izračunava performanse modela [8].

4. REZULTATI I DISKUSIJA

Prvi korišćeni model je kNN odnosno model k - najbližih suseda. Prema tabeli 1 mogu se videti mere kvaliteta modela i zaključiti da se ovaj model nije baš najbolje pokazao na ovom skupu podataka. Dobijena je tačnost od 24% što je prilično nisko. Preciznost i osetljivost su

takođe niske za većinu klasa. Najviša F1-mera je za klasu 3, ali i to je samo 34% što takođe nije zadovoljavajuće.

Nakon ovog modela primenjen je model stabla odluke koji daje znatno bolje mere kvaliteta, što se može videti u tabeli 2. Tačnost od 96% je vrlo visoka. Preciznost, osetljivost i F1-mera su visoki za sve klase, osim za klasu 5 gde je osetljivost 69%, što je još uvek prihvatljivo. Generalno, stablo odluke daje veoma pouzdane i dobre rezultate.

Tabela 1. Rezultati modela k – najbližih suseda na test skupu

	PRECIZNOST <i>T</i>	OSETLJIVOST	F1 - MERA
1	0.83	0.16	0.27
2	0.25	0.15	0.19
3	0.29	0.42	0.34
4	0.15	0.26	0.19
5	0.07	0.08	0.07
TAČNOST			0.24
MAKRO PROSECI	0.32	0.22	0.21
TEŽINSKI PROSECI	0.38	0.24	0.24

Tabela 2. Rezultati modela stabla odluke na test skupu

	PRECIZNOST <i>T</i>	OSETLJIVOST	F1 - MERA
1	0.97	1.00	0.98
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	0.82	0.95	0.88
5	1.00	0.69	0.82
TAČNOST			0.96
MAKRO PROSECI	0.96	0.93	0.94
TEŽINSKI PROSECI	0.96	0.96	0.96

Tabela 3. Rezultati metode slučajne šume na test skupu

	PRECIZNOST <i>T</i>	OSETLJIVOST	F1 - MERA
1	0.97	1.00	0.98
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	0.82	0.95	0.88
5	1.00	0.69	0.82
TAČNOST			0.96
MAKRO PROSECI	0.96	0.96	0.94

TEŽINSKI PROSECI	0.96	0.96	0.96
-------------------------	------	------	------

Tabela 4. Rezultati modela gradijentnog pojačavanja na test skupu

	PRECIZNOST	OSETLJIVOST	F1 - MERA
T			
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	1.00	1.00	1.00
5	1.00	1.00	1.00
TAČNOST			1.00
MAKRO PROSECI	1.00	1.00	1.00
TEŽINSKI PROSECI	1.00	1.00	1.00

Sledeći model je metod slučajne šume koji je postigao identične rezultate kao i stablo odluke, sa tačnošću od 96%, što je pokazano u tabeli 3.

Nakon modela slučajnih šuma, testiran je i model gradijentnog pojačavanja koji je postigao savršene rezultate sa tačnošću od 100%, prikazan u tabeli 4. Sve klase su tačno klasifikovane sa preciznošću, osetljivošću i F1 - merom od 100%. Ovo može ukazivati da je gradijentno pojačavanje previše prilagođen na trening skupu, što može biti znak natprilagođenja, ali ovde je očigledno model tačno klasifikovao sve primere iz test skupa.

5. ZAKLJUČAK

U ovom radu istražena je primena mašinskog učenja u analizi podataka dobijenih scintigrafijom sa ciljem predviđanja stadijuma hronične bolesti bubrega.

Nakon preprocesiranja podataka i odabira atributa, kreirani su modeli. Trenirani su metod k – najbližih suseda, stablo odluke, metod slučajne šume i gradijentno pojačavanje.

Zatim su evaluirane su njihove performanse.

Rezultati su pokazali da metod k - najbližih suseda nije dao zadovoljavajuće performanse, sa tačnošću od samo 24% i niskim vrednostima preciznosti, osetljivosti i F1-mere za većinu klasa. S druge strane, stablo odluke i metod slučajne šume pokazali su značajno bolje rezultate sa tačnošću od 96%, visokim preciznostima i osetljivostima za sve klase osim klase 5, gde je osetljivost bila nešto niža, ali i dalje prihvatljiva.

Model gradijentnog pojačavanja dao je savršene rezultate sa tačnošću od 100% za sve klase.

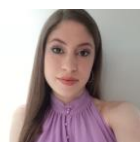
Analizom ovih rezultata zaključuje se da su stablo odluke, metod slučajnih šuma i gradijentno pojačavanje veoma efikasni modeli za klasifikaciju scintigrafskih podataka i predviđanje stadijuma hronične bolesti bubrega na osnovu podataka dobijenih scintigrafijom. Ovi modeli mogu značajno doprineti poboljšanju dijagnostičkih procesa i pružiti podršku medicinskim stručnjacima u donošenju preciznijih odluka.

Primena mašinskog učenja u analizi scintigrafskih podataka pokazuje veliki potencijal za unapređenje medicinske dijagnostike. Dalja istraživanja bi trebalo da se fokusiraju na testiranje ovih modela na većem i raznovrsnijem skupu podataka. Takođe, bilo bi korisno istražiti kombinaciju ovih modela sa drugim tehnikama mašinskog učenja i integrisati ih u šire sisteme za podršku dijagnostici.

6 LITERATURA

- [1] T. Nosek, B. Brkljač, D. Despotović, M. Sečujski, T. Lončar-Turukalo, Praktikum iz mašinskog učenja, 1st ed., Novi Sad: Univerzitet u Novom Sadu, 2020, pp. 122.
- [2] P. Cunningham, S. J. Delany., "k-Nearest neighbour classifiers.", ACM Computing Surveys, vol. 54(6), pp. 1-25, April, 2007.
- [3] M. Nikolić, A. Zečević, Mašinsko učenje, Beograd: Matematički fakultet Univerziteta u Beogradu, 2019.
- [4] <https://dynamicrenalstudy.org/> (pristupljeno u maju 2024.)
- [5] M. Sezgin, B. Sankur., "Survey over image thresholding techniques and quantitative performance evaluation." Journal of Electronic Imaging, vol. 13(1), pp. 146-165, 2004.
- [6] N. Otsu., (1979). "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics, vol. 9(1), pp. 62-66, 1979.
- [7] D. Vidaković, Tehnologije i alati u mašinskom učenju, Novi Sad: Fakultet tehničkih nauka Univerziteta u Novom Sadu, 2023.
- [8] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (pristupljeno u junu 2024.)

Kratka biografija:



Kristina Vajda je rođena 1999. godine u Vrbasu. Završila je gimnaziju u Bečeju, 2018. godine. Nakon završene srednje škole, upisuje Fakultet tehničkih nauka u Novom Sadu, smer Biomedicinsko inženjerstvo. Nakon završenih osnovnih akademskih studija, upisuje 2022. godine master akademske studije na Fakultetu tehničkih nauka, smer Veštačka inteligencija i mašinsko učenje, modul Veštačka inteligencija u medicini. Ispunila je sve obaveze i položila je sve ispite predviđene studijskim programom sa prosečnom ocenom od 9,76.

kontakt: vajda.kristina2@gmail.com

KLASIFIKACIJA TIPA RADARSKOG OBJEKTA KORISTEĆI LSTM NEURONSKE MREŽE SA KONVOLUCIJAMA NA SKUPOVIMA TAČAKA**RADAR OBJECT TYPE CLASSIFICATION USING KERNEL POINT CONVOLUTION LSTM NEURAL NETWORKS**

Vladimir Lunić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – Primenjene računarske nauke i informatika

Kratak sadržaj – Rad predstavlja model neuronskih mreža za klasifikaciju tipa radarskog objekta na osnovu skupa tačaka koje definišu objekat. Arhitektura koristi konvolucije na neuređenim skupovima tačaka u okviru kratkotrajno-dugotrajne memorijske ćelije da enkodira geometriju i osobine tačaka kroz vreme. Model pokazuje visoke mere performansi na skupu podataka View of Delft sa 94% tačnosti i 0.94 F1 merom na klasama pešaka, automobila i biciklista.

Ključne reči: Veštačka inteligencija, Neuronske mreže, Radarska klasifikacija

Abstract – This paper presents a neural network model for type classification of radar objects based on associated point clouds. The architecture uses kernel point convolutions on unordered point sets within a long-short term memory cell to encode the geometric data and features of the point cloud. The model shows high performance scores on the View of Delft dataset, with 94% accuracy and 0.94 F1 Score for car, pedestrian and bicycle classes.

Keywords: Artificial Intelligence, Neural Networks, Radar type classification

1. UVOD

Radari su sistemi koji koriste elektromagnetne talase u radio spektru kako bi odredili distancu objekta u odnosu na mesto gde se radar nalazi. Takođe određuju smer u kojem se objekat nalazi spram mesta gde je radar postavljen koristeći azimutni ugao i ugao elevacije. Ključna prednost radara u odnosu na druge sisteme koji mogu da odrede gde se objekat nalazi jeste i mogućnost radarskih sistema da odrede i radijalnu brzinu objekta u odnosu na mesto gde je radarski sistem postavljen, koristeći Doplerov efekat.

Radarski sistemi, za razliku od kamera, su velikim delom otporni na vremenske uslove u kojima kamera ne bi mogla da vrši svoj zadatak. Osim mogućnosti da radi u vremenskim uslovima kao što su kiša, magla i sneg, radarskim sistemima nije potreban stalni izvor svetlosti, stoga mogu da rade i u mraku ili tokom noći.

Radarski sistemi imaju i svoje mane. Prostor koji radar vidi je mnogo manje gust u odnosu na LiDAR koji radi na sličan način, osim toga, radarski sistemi zahtevaju da objekat ka kojem je poslat elektromagnetni talas, bude reflektivan kako bi ga radarski sistem uopšte mogao videti. Sem nedostatka refleksivnosti nekih objekata, čest problem jeste i prevelika količina refleksivnosti, koja može da zbuni radarski klasifikator.

Kako bi se radarski senzori koristili u okviru automobilske sistema, potrebna je velika pouzdanost da senzor tačno klasifikuje objekat, kako bi se izbegla materijalna ili fizička šteta. Kroz istoriju, zbog ograničenja radarskih senzora, ali i ograničenja računarske moći, algoritmi za klasifikaciju su bili prosti statistički modeli koji razaznaju da li se nešto kreće ili ne. Povećanjem računarske moći i prelaskom na mašinsko učenje i duboko učenje, modeli su mogli da klasifikuju različite učesnike u saobraćaju u nameri da se razviju napredni sistemi za pomoć pri vožnji, kao i sistemi za autonomnu vožnju.

Radarski senzori u automobilske industriji vide svet kao skup tačaka u prostoru, koje se nazivaju detekcije. Detekcije koje su međusobno blizu i imaju slične osobine se grupišu u grupe detekcija koje se nazivaju klasteri ili liste detekcija. Liste detekcija se dalje koriste kako bi se kreirali objekti u prostoru.

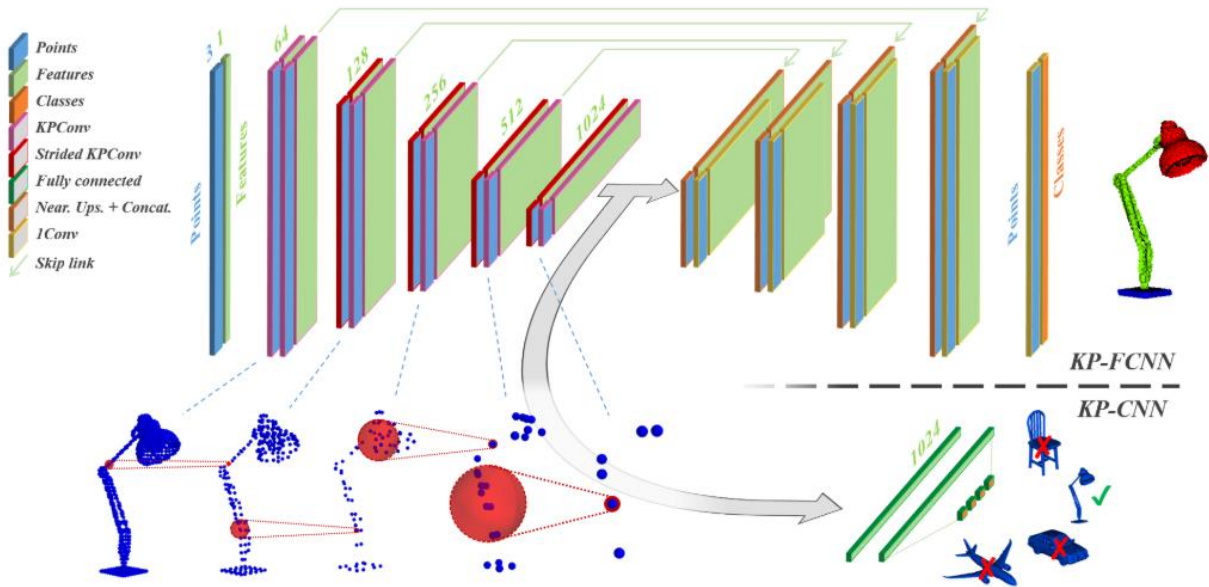
Ovaj rad će predstaviti model dubokog učenja, baziran na rekurentnim mrežama, koji će koristiti liste detekcija kako bi klasifikovao tip objekta, odnosno klasifikovao vrstu učesnika u saobraćaju.

2. PREGLED STANJA U OBLASTI

Prvobitni pokušaji klasifikacije radarskih objekata u domenu dubokog učenja bili su bazirani na dubokim neuronskim mrežama i klasifikaciji na osnovu jednog primera (eng. single shot classification). Koristili su informacije iz objektnog nivoa apstrakcije kao što su brzina objekta, dimenzije objekta i radarski presek objekta. Ovakvi modeli nisu pokazali visoke mere performansi usled činjenice da se izgubio veliki deo informacija prelaskom na viši nivo apstrakcije. Pokušaji da se produbi mreža nisu doprineli velikom prosatu metrika, kao ni promene u arhitekturi sa običnih dubokih mreža na konvolucione mreže.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Rapaić, red. prof.



SLIKA 1: ARHITEKTURA KERNEL POINT KONVOLUTIVNE MREŽE

Nakon pokušaja sa klasičnim dubokim arhitekturama, modeli su počeli da koriste rekurentne mreže kako bi primenili latentne informacije iz vremenskog domena. Rekurentnost je napravila prvi veći skok u performansama modela, pošto je sada bilo lakše razaznati pešake, bicikliste i vozila usled razlika u mikrodopleru. Ali i dalje postoji problem što su modeli radili na nivou apstrakcije objekata. U ranijim radarima ovakvi modeli su predstavljali vrhunac zbog činjenice, da, na kraju razvoja modela, model treba da vrši inferenciju na samom radaru u realnom vremenu, što je veliko usko grlo dubokog učenja na radarskim sistemima usled skupih procesorskih operacija.

Prvi rad koji se smatra savremenim pokušajem klasifikacije i segmentacije neuređenog skupa tačaka je PointNet [1]. PointNet rezonuje nad trodimenzionim geometrijskim podacima kao što su skupovi tačaka, odnosno oblaci tačaka. Arhitektura PointNet neuronske mreže se sastoji iz tri glavne komponente koje predstavljaju rešenje za ograničenja sa radom na neorganizovanim skupovima tačaka. Prvom komponentom realizuje se embedding ulaznog skupa tačaka u globalni vektor. Druga komponenta se odnosi na dalje korišćenje embedovanog skupa tačaka za klasifikaciju ili segmentaciju. Poslednja komponenta rešava problem transformacija ulaznog skupa, tako što pomoću manjih neuronskih mreža, transformiše ulazne skupove tačaka u referentni sistem, koji se uči, uporedo sa klasifikatorom.

PointNet++ [2] arhitektura je predstavljena kao unapređenje koje specifično cilja da reši problem zavisnosti tačke od lokalne strukture susedstva. Arhitekturu predstavljenu u radu [2] čini komponenta za hijerarhijsko učenje osobina skupa tačaka kojom se ostvaruje i prikaz lokalnih susedstava skupova tačaka, umesto samo globalni vektor.

Rad [3] uzima inspiraciju iz klasičnih konvolutivnih mreža i predstavlja *kernel point* konvolucije (nadalje KPC) i konvolutivne modele za segmentaciju i

klasifikaciju neuređenih skupova tačaka. Primer segmentacionog i klasifikacionog KPC modela predstavljen je na slici 1. Rad uvodi pojam kernela tačaka, kao modifikaciju kernel matrica koje koriste konvolutivne mreže. Centriranjem kernela na pojedinačne tačke, operatorom konvolucije, računaju se izlazne osobine te tačke, kreirajući 1 – 1 mapiranje između ulaznog i izlaznog skupa tačaka.

3. METODOLOGIJA

Za rad je korišćen programski jezik *Python* verzije 3.11.2. Za implementaciju modela korišćene su isključivo funkcije prisutne u *Tensorflow* i *Keras* bibliotekama uz hardverska ubrzanja korišćenjem anotacija `@tensorflow.function()`. Minimalna verzija biblioteke *Tensorflow*, kako bi model radio kako je namenjeno, je verzija 2.16. Za *Keras* je potrebna minimalno verzija 3.0, dok je u radu korišćena verzija 3.3

3.1. Postavljanje kernel tačaka

Kreiranje kernela se svodi na proces optimizacije pozicije tačaka u okviru lopte koja predstavlja kernel. Ako pretpostavimo da imamo K kernel tačaka $\tilde{x}_k \in R^d$ u procesu optimizacije želeli bismo da postavimo tačke tako da im je međusobna distanca maksimalna i da su i dalje u okviru lopte poluprečnika r , za koji se obično uzima vrednost $r = 1$. Stoga svakoj tački dodeljujemo odbojnu energiju spram tačke $\tilde{x}_k$, $E_k^{rep}(x)$, i privlačnu energiju, $E_k^{att}(x)$, ka centru lopte, koja je centrirana na $\vec{0}$. Optimizacioni proces teži da minimizuje ukupnu energiju sistema koji se sastoji od K tačaka. Dodatno, u optimizacionom procesu se mogu postaviti fiksne tačke koje ne učestvuju u optimizacionom procesu ali pridodaju ukupnoj energiji. Optimizacioni proces se izvršava iterativno pomoću gradijentnog spusta.

$$E_k^{rep}(x) = \frac{1}{||x - x_k||} \quad (1)$$

$$E^{att}(x) = ||x|| \quad (2)$$

$$\text{minimize } E^{tot} = \sum_{k \leq K} \left(E^{att}(\tilde{x}_k) + \sum_{l \neq k} E_k^{rep}(\tilde{x}_l) \right) \quad (3)$$

3.2. Operatori kernel point konvolucije

Nakon definicije kernela, operator konvolucije može da se izvrši. KPC nad skupom osobina $\mathcal{F}$ kernelom g u tački x je definisana sledećim izrazom (4).

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x) f_i \quad (4)$$

Tačke x_i predstavljaju tačke ulaznog skupa tačaka $\mathcal{P} \in R^{N \times d}$, gde N predstavlja broj tačaka skupa i d predstavlja dimenzionalnost tačaka. Tačke f_i predstavljaju vektore osobina tačaka skupa $\mathcal{P}$ i pripadaju svom odvojenom skupu $\mathcal{F} \in R^{N \times D}$, gde D predstavlja dimenzionalnost prostora osobina. Cilj KPC je onda da transformiše skup osobina $\mathcal{F}$ u novi skup osobina, konvolucijama nad tačkama skupa $\mathcal{P}$.

Izraz (4) dodatno definiše susedstvo tačke x , $\mathcal{N}_x$, skup tačaka koji će učestvovati u konvoluciji tačke x . Susedstvo $\mathcal{N}_x$ je definisan kao $\{x_i \in \mathcal{P} \mid ||x_i - x|| \leq r\}$, odnosno svaka tačka skupa $\mathcal{P}$ čija distanca od tačke x spada u loptu poluprečnika r .

Kernel funkcija g predstavlja ključnu funkciju pri KPC. Funkcija g prima kao ulaz tačke susedstva tačke x , centrirane oko same tačke x , nadalje definisane sa $y_i = x_i - x$. Definisano susedstvo onda predstavlja loptu poluprečnika r , $\mathcal{B}_r^d = \{y \in R^d \mid ||y|| \leq r\}$, te je domen kernel funkcije lopta $\mathcal{B}_r^d$. Ideja kernel funkcije je da primeni različite težine tačkama $\mathcal{B}_r^d$, nalik konvolucijama uređenih skupova tačaka. Stoga potreban je utvrditi način na koji će se izračunati koje težine treba primeniti na koje tačke lopte.

Definišimo sada tačke kernela $\{\tilde{x}_k \mid k \leq K\} \subset \mathcal{B}_r^d$ i težine asocirane za svaku od tačaka kernela sa $\{W_k \mid k \leq K\} \subset R^{D_{in} \times D_{out}}$, gde D_{in} predstavlja dimenzionalnost prostora osobina ulaznog skupa osobina, a D_{out} dimenzionalnost prostora osobina izlaznog skupa osobina. Sa tako definisanim tačkama i težinama definišemo kernel funkciju g za svaku tačku $y_i \in \mathcal{B}_r^d$ izrazom (5).

$$g(y_i) = \sum_{k \leq K} h(y_i, \tilde{x}_k) W_k \quad (5)$$

Funkcija $h(y_i, \tilde{x}_k)$ definiše korelaciju između tačke susedstva y_i i tačke kernela $\tilde{x}_k$ i njena vrednost ukazuje na vednost korelacije ove dve tačke. Mera korelacije se može izračunati na veliki broj načina, imajući na umu da se radi o tačkama u realnom svetu, dobra mera korelacije bi bila rastojanje između te dve tačke. Dodatno mera korelacije bi trebalo da raste sa smanjenjem rastojanja između dve tačke i da konvergira ka 0 što je rastojanje veće. Stoga funkciju korelacije $h(y_i, \tilde{x}_k)$ definišemo izrazom (6).

$$h(y_i, \tilde{x}_k) = \max\left(0, 1 - \frac{||y_i - \tilde{x}_k||}{\sigma}\right) \quad (6)$$

Parametar σ izraza (6) definiše snagu kernela. Parametar se može protumačiti kao zona unutar koje je korelacija sa kernelom velika. Odabir parametra snage kernela zavisi od definicije drugih parametara KPC sloja.

3.1. Kernel point LSTM ćelija

LSTM ćelija nastoji da premesti KPC u vremenski domen. To vrši zamenjujući operaciju linearne kombinacije u kapijama sa operacijom KPC. Izrazi (7-13) definišu nove preračune kapija KPC LSTM ćelije.

$$\hat{X}_t = KPC(X_t, K^X) \quad (7)$$

$$F_t = \sigma\left(KPC([\hat{X}_t, H_{t-1}], K^F)\right) \quad (8)$$

$$I_t = \sigma\left(KPC([\hat{X}_t, H_{t-1}], K^I)\right) \quad (9)$$

$$O_t = \sigma\left(KPC([\hat{X}_t, H_{t-1}], K^O)\right) \quad (10)$$

$$\hat{C}_t = \sigma\left(KPC([X_t, H_{t-1}], K^C)\right) \quad (11)$$

$$C_t = F_t \odot C_{t-1} + I_t \odot \hat{C}_t \quad (12)$$

$$H_t = O_t \odot \tanh(C_t) \quad (13)$$

$KPC(X, K)$ predstavlja KPC nad X kernelima K , gde X predstavlja uređeni par skupa tačaka i tenzora osobina. X , predstavljen uređenim parom $(\mathcal{P}_X, \mathcal{F}_X)$, je jedan element niza uređenih parova skupova tačaka i tenzora osobina koji predstavljaju vremensku seriju. $\mathcal{P}_X$ je definisan matricom $N \times d$, gde N predstavlja broj tačaka, a d dimenzionalnost tačaka, dok je $\mathcal{F}_X$ definisan tenzorom $N \times D^{in} \times c$, gde D^{in} predstavlja dimenzionalnost ulaznog prostora osobina, a c broj kanala tenzora osobina. Sa druge strane, H predstavlja skriveno stanje LSTM ćelije. Skriveno stanje je takođe definisano uređenim parom tačaka i osobina, $(\mathcal{P}_H, \mathcal{F}_H)$. $\mathcal{P}_H$ je matrica veličine $N' \times d$, dok $\mathcal{F}_H$ predstavlja tenzor veličine $N' \times D^{out} \times c'$, gde N' predstavlja broj tačaka skrivenog stanja, D^{out} predstavlja dimenzionalnost skrivenog prostora osobina, koji je ujedno i dimenzionalnost izlaznog prostora osobina, a c' broj kernela korišćen u kapijama KPC. Ćelijsko stanje C , nalik H , je definisano uređenim parom $(\mathcal{P}_C, \mathcal{F}_C)$, istih dimenzionalnosti kao $(\mathcal{P}_H, \mathcal{F}_H)$. $[x, y]$ definiše operaciju vertikalne konkatenacije, dok operator $\odot_m$ predstavlja modifikovano Adamarovo množenje.

Adamarovo množenje zahteva da oba operanda budu matrice iste veličine, ali u ovom slučaju, izlazi kapija predstavljaju uređene parove matrica i tenzora veličina $\hat{N} \times d$ i $\hat{N} \times D^{out} \times c'$, dok C_{t-1} i $\hat{C}_t$ predstavljaju uređene parove matrica i tenzora veličina $N' \times d$ i $N' \times D^{out} \times c'$. Usled toga, Adamarovo množenje se ne može izvršiti zbog neslaganja dimenzija $\hat{N}$ i N' . Stoga, operator Adamarovog množenja je izmenjen da može da radi sa operandima različitih veličina. Operator za

operande prima dva uređena para tačaka i njihovih vektora osobina. Broj tačaka ne mora biti isti, ali dimenzionalnost tačaka i osobina tačaka mora biti ista. Rezultat operacije vraća uređeni par tačaka i izlaznih osobina. Prvi element uređenog para prima vrednost matrice tačaka levog operanda, dok drugi element uređenog para nakon računa ostaje iste dimenzionalnosti kao tenzor osobina levog operanda. Izrazi (14-16) definišu modifikovani operator Adamarovog množenja.

$$h(\mathcal{P}_x, \mathcal{P}_y) = \max(0, 1 - \|\mathcal{P}_x - \mathcal{P}_y\|) \quad (14)$$

$$\mathcal{F} = \mathcal{F}_x \odot (h(\mathcal{P}_x, \mathcal{P}_y) \circ \mathcal{F}_y) \quad (15)$$

$$(\mathcal{P}_x, \mathcal{F}_x) \odot_m (\mathcal{P}_y, \mathcal{F}_y) = (\mathcal{P}_x, \mathcal{F}) \quad (16)$$

4. SKUP PODATAKA

Skup podataka korišćen za sve eksperimente potiče iz rada [4], univerziteta u Delftu i naziva se *View of Delft* (nadalje VoD). Skup podataka je kreiran vožnjom kroz kampus, predgrađa i centar Delfta. Radar korišćen za kreiranje skupa podataka je *ZF FRGen21 3+1* dimenzioni radar, frekvencije 13Hz, postavljen iza prednjeg branika automobila. Dodatno za prikupljanje podataka korišćena je stereo kamera, frekvencije 30Hz postavljena na vetrobran i *Velodyne HDL-64 S3 LiDAR*, frekvencije 10Hz. Pozicije senzora predstavljene su na slici 2.

Frejmovi svakog od senzora su sinhronizovani, tako da svaki frejm sadži tačke koje je LiDAR izmerio, tačke koje je radar izmerio, sliku stereo kamere i kalibraciju senzora za trenutni frejm. LiDAR je uzet kao glavni frejm spram kojeg su traženi najbliži frejm radara i kamere, maksimalne tolerancije razlike vremenskih odrednica frejmova od 0.05 sekundi. Stoga frejmovi predstavljaju sekvence snimljene na svakih 10Hz.

Skup podataka je organizovan što je bliže moguće KITTI formatu, koji definiše kako se skladišti jedan frejm. Podaci su podeljeni na trening i test frejmove. VoD nudi labele samo za trening frejmove, te test frejmovi nisu mogli biti korišćeni za svrhe rada, bez da se cela implementacija rada pošalje univerzitetu u Delftu.



SLIKA 2: POSTAVKA SENZORA KORIŠĆENIH U "VJU OF DELFT" SKUPU PODATAKA

Ukupan broj trening i test frejmova je 8693, od čega, 26% predstavljaju test frejmove, ostavljajući samo 6433 frejmova koji su labelirani. Trening je dalje podeljen u direktorijume za LiDAR frejmove, i tri vrste radar frejmova koji predstavljaju 1, 3 ili 5 frejmova spojenih u jedan frejm. Za svrhe rada, korišćeni su radar podaci sačinjeni od jednog frejma detekcija, jer takva vrsta podataka najbliže predstavlja realne podatke.

5. REZULTATI I DISKUSIJA

Eksperimenti na arhitekturi mreže će biti podeljeni u tri kategorije. Prva kategorija je uticaj veličine i broja kernela na rezultate. Druga vrsta eksperimenata će se baviti uticajem koraka na rezultate. Treća kategorija eksperimentiše sa količinom ćelija u rekurentnom bloku. Sve arhitekture, osim ako nije dodatno navedeno, će koristiti isti skup podataka, sačinjen od najvećeg podskupa podataka koji sadrži iste količine labela automobila, biciklista i pešaka. Biće trenirane u 100 epoha, koristeći *Adaptive Moment Estimation* (ADAM) optimizacionu metodu, sa invarijantnom stopom učenja od 1.0×10^{-3} , funkcijom greške unakrsne entorpije (CEL), metrikama tačnosti i makro F1 mere i bez mehanizma za ranije zaustavljanje. Evaluacija će biti odrađena na celokupnom test skupu podataka, odvojenom u fazi preprocesiranja, i biće učitana kontrolna tačka sa najmanjom validacionom greškom.

Prvi model za evaluaciju se sastojao od minimalne arhitekture koja prati preporuke za KPC predstavljene u radu [3]. Arhitektura se sastoji od jedne KPC LSTM ćelije. Ćelija na ulazu prima RCS detekcije i absolutnu vrednost kompenzovane Dopler brzine, te je ulazni prostor osobina dvodimenzijoni. Dimenzionalnost skrivenog prostora osobina je dva puta veći od ulaznog po preporuci [3] koja nalaže dupliranje dimenzionalnosti između slojeva. Broj kernela je 1 kao i kod običnih KPC mreža predstavljenih u [3], dok je broj tačaka u kernelu 15. Korak je u ovoj vrsti eksperimenta fiksiran i ima vrednost 1.0. Drugi model prati istu arhitekturu kao i prvi model, sa razlikom da sadrži 3 kernela. Dok treći model sadrži 1 kernel i 7 tačaka. Rezultati su predstavljeni u tabeli 1.

TABELA 1: REZULTATI EKSPERIMENATA KERNEL TAČAKA

Oblik Kernela	Tačnost	Makro F1 Mera
[1, 15]	90%	90%
[3, 15]	91.5%	91.5%
[1, 7]	91.6%	92%

Povećanje broja kernela je donelo usporenje od 10%, dok je smanjenje ubrzalo treniranje za 30%. Povećanje broja kernela je učinilo proces treniranja manje stabilnim, sa mnogo više skokova u grešci za razliku od modela sa manjim brojem jezgara, usled činjenice da mreža ima veliki broj parametara. Fenomen eksplodirajućeg gradijenta je stoga bio primećen u određenim epohama, što se može rešiti praćenjem performansi modela za

potrebe ranijeg zaustavljanja i biranjem najboljeg sačuvanog modela. Model sa jednim jezgrom i sedam tačaka je imao najstabilniji proces učenja, sa minimalnim oscilacijama u grešci, koje najverovatnije potiču usled upotrebe algoritma stohastičkog gradijenta.

Sledeća grupa eksperimenata se fokusira na parametar koraka. Mreže korišćene u ovim eksperimentima imaju fiksiranu arhitekturu na jednu ćeliju sa 4 dimenzije u skrivenom prostoru osobina, sa jednim kernelom od 14 tačaka i uticajem kernel tačaka od 4.0. Mreže su testirane sa korakom 1.0, 0.25, 0.5 i 0.75. Rezultati eksperimenata su predstavljeni u tabeli 2.

TABELA 2: REZULTATI EKSPERIMENTA KORAKA

Korak	Tačnost	Makro F1 Mera
1.0	90%	90%
0.75	94.3%	94.3%
0.5	90.7%	90.8%
0.25	89.6%	89.7%

Smanjenjem koraka se mreža očekivano kraće trenirala, pitanje koje ovaj eksperiment hoće da odgovori jeste da li uopšte korak utiče na performanse. Dodatno, ako se korak pokaže da pozitivno utiče na proces treniranja, koja vrednost je najbolja, odnosno za koliko smanjivati prostor svakom sledećom KPC. Pozivajući se na puling funkcije običnih konvolutivnih mreža, očekivana vrednost bi bila 0.5, ali usled skupa koji nije gust kao što su slike, vrednost se očekuje da bude veća. Korak između 1.0 i 0.75 se pokazao kao najbolji u očuvavanju relevantnih tačaka i prostora u kojem se kernel kreće. Uz smanjenje prostora koji kerneli pokušavaju da embeduju, proces učenja ispod koraka 0.5 postaje haotičniji i vrednost greške češće skače. Ako korak skрати broj centara konvolucije previše, susedstva će često biti retka i uglavnom prazna, što naravno, ne pomaže procesu učenja.

Poslednja grupa eksperimenata trenira mrežu sa dve rekurentne ćelije. Prva arhitektura enkodira 4 dimenzije skrivenog prostora stanja, ima 1 kernel od 14 tačaka, korak od 1.0 i uticaj od 4.0. Dok druga ćelija, kao što [3] nalaže, duplira dimenzionalnost skrivenog prostora osobina, u ovom slučaju sa 4 na 8. Kernel ostaje isti, ali se uticaj kernel tačaka menja sa porastom enkodiranih dimenzija, sa 4.0 na 8.0, opet po preporuci [3]. Poređenje ovakve dve mreže je predstavljeno tabelom 3.

Mreža koja je sadržala veći broj ćelija nije bila u stanju da generalizuje pravilno na problemu. Proces treniranja je trajao duplo duže i bio je podložan pojavi fenomena eksplodirajućih gradijenata, gde u poslednjem slučaju kada je gradijent eksplodirao, mreža nije mogla da se vrati na logaritamski pad greške kako su epohe tekle i zatekla se u lokalnom minimumu iz kojeg nije mogla da izađe.

TABELA 3: REZULTATI EKSPERIMENTA BROJA ĆELIJA

Broj Ćelija	Tačnost	Makro F1 Mera
1	90%	90%
2	74%	74%

Pretpostavka je da je nova dimenzionalnost uvedena drugim slojem negativno uticala na mnogostrukost koji deli klasifikacioni prostor, te mreža nije mogla da nađe adekvatne parametre koji bi izmenili mnogostrukost na bolje usled većeg izbora dimenzija koji je donela nova ćelija.

6. LITERATURA

- [1] Qi, C.R., Su, H., Mo, K. and Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 652-660).
- [2] Qi, C.R., Yi, L., Su, H. and Guibas, L.J., 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems, 30.
- [3] Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F. and Guibas, L.J., 2019. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 6411-6420).
- [4] Palffy, A., Pool, E., Baratam, S., Kooij, J.F. and Gavrila, D.M., 2022. Multi-class road user detection with 3+ 1D radar in the View-of-Delft dataset. IEEE Robotics and Automation Letters, 7(2), pp.4961-4968.
- [5] Nobis, F., Fent, F., Betz, J. and Lienkamp, M., 2021. Kernel point convolution LSTM networks for radar point cloud segmentation. Applied Sciences, 11(6), p.2599.
- [6] Fent, F., 2020. Machine Learning-Based Radar Point Cloud Segmentation.
- [7] Fan, H. and Yang, Y., 2019. PointRNN: Point recurrent neural network for moving point cloud processing. arXiv preprint arXiv:1910.08287.
- [8] Schumann, O., Hahn, M., Dickmann, J. and Wöhler, C., 2018, July. Semantic segmentation on radar point clouds. In 2018 21st International Conference on Information Fusion (FUSION) (pp. 2179-2186). IEEE.

Kratka biografija:

Vladimir Lunić rođen je u Novom Sadu 2000. god. Master rad na Fakultetu tehničkih nauka iz oblasti Primenjene računarske nauke i informatika – Inteligentni sistemi odbranio je 2024.god.
kontakt: vladimir.lunic2000@gmail.com

РАЗВОЈ ВЕБ АПЛИКАЦИЈЕ ЗА ПРЕЗЕНТОВАЊЕ ПОДАТАКА О ФАКУЛТЕТИМА И ДАВАЊЕ ПРЕДЛОГА УПОТРЕБОМ "CHAT GPT AI" МОДЕЛА**DEVELOPMENT OF A WEB APPLICATION FOR PRESENTING DATA ABOUT FACULTIES AND GIVING SUGGESTIONS USING THE "CHATGPT AI" MODEL**

Миле Кајтез, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИЧКО И РАЧУНАРСКО ИНЖЕЊЕРСТВО

Кратак садржај – Рад описује развој "Faculty Info" веб апликације која ће помоћи будућим студентима у процесу доношења одлуке који факултет уписати и којом професијом се бавити у будућности, употребом вештачке интелигенције (у овом случају коришћењем "Chat GPT AI" модела). Такође, корисник има могућност увида у информације о факултетима, департманима, курсевима и предметима, као и постављање питања о истим.

Кључне речи: систем за управљање знањем / ChatGPT / React / .NET CORE 6

Abstract – The work describes the development of an application that will help future students in the decision-making process of which faculty to enter and which profession to pursue in the future, using artificial intelligence (in this case using the "Chat GPT AI" model). Also, the user has the opportunity to view information about faculties, departments, courses and subjects, as well as ask questions about them.

Keywords: knowledge management system / ChatGPT / React / .NET CORE 6

1. УВОД

У претходних десетак до двадесет година, све је већа употреба друштвених мрежа као главног извора информација у готово свим областима, почевши од дневних вести, временске прогнозе, најаве разних догађаја, до информација у областима као што су медицина, образовање и мноштво других друштвено корисних активности. Највећи број корисника друштвених мрежа су млади људи, пре свега они који су у средњим школама и на факултетима. Изложеност великој количини информација, често доводи до тога да млади не могу самостално и на прави начин извући правовремене и добре закључке. Иза тога стоје и многи други разлози, као што су презентовање информација на начин који није попуно разумљив, половичност, делимична тачност информација и слично.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Александар Селаков, ванр. проф.

Једно од решења јесте коришћење вештачке интелигенције, која ће уместо човека прикупљати, селектирати, обрађивати и претварати информације у, за човека, разумљив и погодан облик. Конкретно "Faculty Info" апликација ће користити вештачку интелигенцију која ће будућим студентима, на основу њихових животних преференција и интересовања, помоћи у доношењу одлуке који факултет уписати и чиме се бавити у будућности.

2. КОРИШЋЕНЕ ТЕХНИКЕ И ТЕХНОЛОГИЈЕ

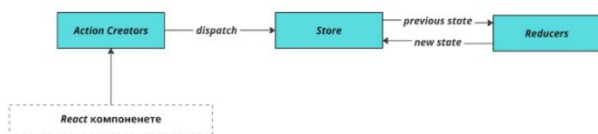
Анализом сличних решења чији је примарни задатак информисање студентата и оних који ће то тек постати, закључак је да свако решење има свој опсег употребе. Оно што је заједничко за све јесу њихова усмереност да корисници на ефикасан и тачан начин, у што краћем временском периоду, дођу до жељених информација. Поред самих перформанси, битно је нагласити да апликације могу бити коришћене од стране великог броја корисника. Некада, велики број функционалности може бити контрапродуктиван јер доводи до збуњености корисника по питању начина коришћења апликације. Као у свему, и у овом аспекту развоја је потребно пронаћи баланс између пружања услуга, једноставности и перформанси коришћења. Уколико се пронађе тај баланс, често те апликације доживљавају велику експанзију и употребу, па самим тим и успех. Као предуслов за испуњење претходно наведених услова, у самом процесу развоја се морају донети одлуке по питању које технологије и технике користити. Пошто је "Faculty Info" систем, по свом типу веб апликација, технике и технологије су подељене на два дела, тј. оне које се користе за израду клијентске апликације, као и технике и технологије за израду серверског дела и комуникацију и складиштење података у оквиру базе података.

2.1. Технологије клијентске апликације

Могућности за креирање клијентске апликације су данас велике због постојања великог броја развојних окружења који омогућавају флексибилност у развоју. Конкретно, у случају "Faculty Info" апликације, ради се о "SPA (Single Page Application)" [1], креираном употребом "React JS" библиотеке [2], "Redux" механизма [3] и "CSS (Cascading Style Sheets)" [4] језика, који се користи за дефинисање самог изгледа апликације. За разлику од ранијих, данашње веб апликације су окренуте ка клијентској страни, тј.

тежи се ка томе да клијентска апликација буде што „тежа“ тј. да се део операција извршава одмах на клијентској страни, без беспотребног слања захтева ка серверу. Овај приступ омогућава лакше конструисање апликација које ће бити намењене за велики број корисника, као и њихов бржи и поузданији рад. У сврху тога, клијентска апликација, поред компоненти, садржи и “*Redux*” библиотеку, која омогућава чување дела података на клијентској страни, што скраћује време добављања података и самим тим резултује бржим радом и одзивом у реалном времену, док сам корисник стиче осећај правовременог и брзог одговора. “*Redux*” је state контејнер за “*JavaScript*” [5] апликације и састоји се из три дела (Слика 2.1.1), и то:

1. “*Store*” – објекат у оквиру којег се налазе сви подаци и уз помоћ којег се могу пратити све промене података у току рада апликације. Једини начин за промену *state* објекта, јесте емитовање акције. Такође, променом *state* објекта извршава се поновно учитавање свих компоненти.
2. “*Action*” – акција која позива промену *state* објекта (описује промену стања)
3. “*Reducer*” – врши промену *state* објекта (спроводи акцију)



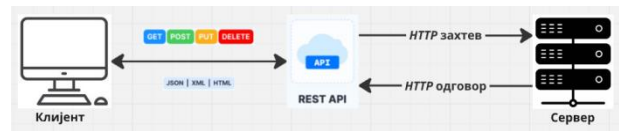
Слика 2.1.1. - “*Redux*” начин рада

Што се тиче имплементације самог дизајна клијентске апликације, CSS стилови су писани коришћењем стилизованих компоненти (“*Styled Components*”) [6], што се показало као једно од бољих решења за велике веб апликације. Употребом ове врсте компоненти омогућава се креирање многих дизајнерских функционалности попут промене теме, лакшег коришћења стилова на више места, што резултира лаку прилагодивост и проширивост већ постојећих стилова, итд.

2.2. Технологије серверске апликације

Да би систем, као целина могао да корисницима пружи тачне и правовремене информације, потребно је да исти има серверску апликацију која може да задовољи претходно наведене услове, да има стабилан и поуздан рад, могућност лаког опоравка од отказа, гаранцију да подаци који се обрађују буду валидни и заштићени од спољних утицаја, итд. Да би се дошло до одговора на питање који архитектурални стил користити у изради овог дела апликације, прво се морају сагледати идеје коришћења апликације, скуп функционалности које систем треба да задовољи, број корисника који ће потенцијално да користе систем, који тип корисника ће користити систем, итд. Што се саме архитектуре тиче, узимајући у обзир да је сам развој апликације у почетној фази, донета је одлука да се користи монолитна архитектура, која у

почетним фазама производа нуди много већу брзину развоја и флексибилност измена. Што се тиче саме комуникације са клијентском страном, она је имплементирана употребом “*HTTP*” протокола [7], док сама серверска апликација ради по “*REST*” принципу [8] (Слика 2.2.1.).



Слика 2.2.1. - “*REST*” принцип комуникације клијентске и серверске апликације

Инфраструктура и организација логичких целина серверске апликације, дефинисана је коришћењем “*Onion*” архитектуре (Слика 2.2.2.) [9], која омогућава многе бенефите као што су:

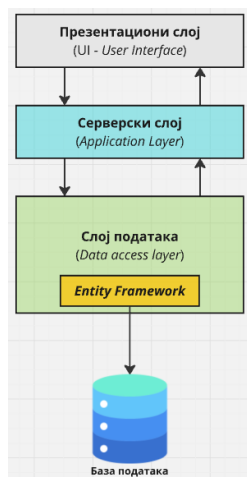
1. Лакши потенцијални прелазак са монолитне на микросервисну архитектуру
2. Лакше одржавање кода
3. Флексибилније креирање, модификација и тестирање компоненти и функционалности
4. Лакша и независнија комуникација са екстерним сервисима



Слика 2.2.2. - Илустрација “*Onion*” архитектуре

У склопу “*Onion*” архитектуре користи се и “*Command and Query Responsibility Segregation (CQRS)*” образац [10], који раздваја операције читања и ажурирања базе података. Пружа додатну скалабилност и сигурност због омогућавања лакшег развијања система током времена и спречавања да наредбе ажурирања узрокују конфликте на нивоу домена. Команде (“*Commands*”) се користе за ажурирање базе података, док се упити (“*Queries*”) користе за добављање података. Што се тиче технологије која се користи у имплементацији претходно наведене архитектуре, користи се “*C#*” програмски језик [11] и “*ASP .NET Core 6*” [12] који представља радно окружење за развијање серверских апликација. Употребом “*ASP .NET Core 6*” окружења пружа се могућност лаког креирања серверске апликације, логике за комуникацију са базом податка, употребом објектно-релационог мапера, као што је “*EF (Entity Framework)*” [13], као и другим екстерним сервисима, али и употребом “*CRUD (Create, Read, Update, Delete)*” операција [14]. “*Entity Framework*” је објектно-релациони мапер који пружа опцију лаког и

брзог начина дефинисања и коришћења базе података (Слика 2.2.3.).



Слика 2.2.3. Приказ позиције “Entity Framework” мапера

Сама база података је дефинисана употребом “PostgreSQL” [15], објектно-релационог система за управљање базама података који садржи моћан објектно-релациони модел података, богат избор врста података, лаку надоградивост, као и надограђени скуп наредби “Structured Query Language (SQL)” језика [16].

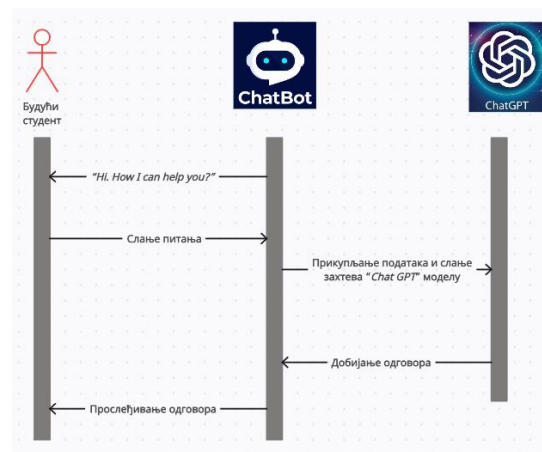
Поред наведених делова система, систем користи и екстерне сервисе, као што су “Sendgrid” [17] за слање електронске поште, “Amazon Simple Storage Service (AWS S3)” [18] за посебно складиштење ресурса попут слика и докумената, “Geoapify” [19] за одређивање географских локација, као и “ChatGPT OpenAI API” [20] који омогућава коришћење моћних модела језика као што су “GPT-3.5” и “GPT-4” путем једноставног “API (Application Programming Interface)” [21] позива.

3. ФУНКЦИОНАЛНОСТИ И ДЕМОНСТРАЦИЈА ИМПЛЕМЕНТАЦИЈЕ

Апликација је конципирана као дигитална платформа која омогућава приступње подацима о факултетима, курсевима и предметима, као и манипулисање подацима у циљу лакшег доношења одлука у вези будућег студирања и професионалне каријере. Систем подржава три врсте корисника:

1. *Главни администратор* који је одговоран за администрацију целокупног система и подешавања (манипулација факултетима и администраторима факултета).
2. *Администратор факултета* одговоран за администрацију информација везаних за свој факултет, одговарање на питања, дефинисање курсева, предмета и анкета, као и увид у изглед страница факултета ком је придодат.
3. *Будући студент* који има могућност прегледа факултета, курсева и предмета, манипулација профилних података, постављање питања, одговарање на питања из анкете, као и комуникација са *chatbot* механизмом.

Сваки од корисника, при отварању апликације, има приступ почетној страници, где се може извршити логовање или регистрација, у зависности од тога да ли је корисник већ регистрован или је пак будући студент. У наставку ће бити описан начин рада апликације у процесу комуникације будућег студента и *chatbot* механизма, у интеграцији са “ChatGPT” моделом (Слика 3.1.), што представља главну функционалност апликације.



Слика 3.1. Кораци комуникације будућег студента и *chatbot*-а

После успешног одговарања на сва или пак на већину питања из анкете, корисник може отићи на страницу (Слика 3.2.) на којој има могућност комуникације путем *chatbot* механизма, који се ослања на “ChatGPT” модел. Комуникацију започиње *chatbot*, док је задатак корисника да постави питања која жели. У року од пар секунди добиће одговор и на тај начин се формира комуникација. Како се комуникација развија, тако систем све више препознаје корисникова интересовања и чиме би потенцијално могао да се бави у будућности и на основу тих информација шаље будуће одговоре. Корисник у просеку на одговор чека до 3 секунде, што представља сасвим солидну брзину комуникације и добијања нових информација.



Слика 3.2. Приказ комуникације корисника са *chatbot*-ом

4. ЗАКЉУЧАК

Проласком кроз рад, описан је систем који се састоји из три целине, које функционишу и међусобно комуницирају и на тај начин чине један, јединствен систем. Показало се да сама веб апликација поседује особине које дају предуслов за брзо и ефикасно коришћење, као и одзив у реалном времену. Ова тврдња може се потврдити постојањем “*Redux*” логике на клијентској апликацији, која чува податке и омогућава смањење броја захтева ка серверској страни. Коришћењем “*Opinion*” архитектуре на серверској страни, постоји могућност лаког проширења и измене делова логике. Што се тиче базе података, користи се релациона база података која омогућава лакше манипулисање односима међу ентитетима система. Сама апликација обезбеђује интуитивно и поуздано искуство, омогућавајући ефикасно и јасно управљање информацијама, као и интеракцију са “*ChatGPT*” моделом ради добијања персонализованих препорука. Очекује се да ће апликација пружити корисницима свеобухватне ресурсе и подршку у процесу доношења одлука о образовању. Иако апликација има и друге добре особине, као што су једноставан и прегледан дизајн, тачно и правовремено одговарање сервисне стране, висок ниво апстракције кода, свакако да постоји још простора за побољшања и увођење нових решења, као што су креирање мобилне верзије апликације која ће проширити број корисника који користи систем, претварање апликације у друштвену мрежу на којој ће корисници моћи међусобно делити информације и материјале који ће им бити од помоћи у даљем образовању, као и увођење нових функционалности попут увида у резултате испита, унос оцена, приказ статистике, постојање нотификација, могућности креирања анкета и других садржаја, итд.

5. ЛИТЕРАТУРА

- [1] <https://developer.mozilla.org/en-US/docs/Glossary/SPA>, преузето Јуна 2024.
- [2] <https://reactjs.org/>, преузето Јуна 2024.
- [3] <https://redux.js.org/>, преузето Јуна 2024.
- [4] <https://www.tutorialrepublic.com/css-tutorial/>, преузето Јуна 2024.
- [5] <https://www.javascript.com/>, преузето Јуна 2024.
- [6] <https://styled-components.com/>, преузето Јуна 2024.
- [7] <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>, преузето Јуна 2024.
- [8] <https://restfulapi.net/>, преузето Јуна 2024.
- [9] <https://medium.com/expedia-group-tech/onion-architecture-deed8a554423>, преузето Јуна 2024.
- [10] <https://www.techtarget.com/searchapparchitecture/definition/CQRS-command-query-responsibility-segregation>, преузето Јуна 2024.
- [11] <https://docs.microsoft.com/en-us/dotnet/csharp/>, преузето Јуна 2024.
- [12] <https://www.tutorialsteacher.com/core/dotnet-core>, преузето Јуна 2024.
- [13] <https://docs.microsoft.com/en-us/ef/>, преузето Јуна 2024.

[14] <https://www.codecademy.com/article/what-is-crud>, преузето Јуна 2024.

[15] <https://www.postgresql.org/>, преузето Јуна 2024.

[16] <https://www.w3schools.com/sql/>, преузето Јуна 2024.

[17] <https://sendgrid.com/en-us>, преузето Јуна 2024.

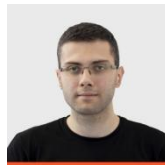
[18] <https://aws.amazon.com/s3/>, преузето Јуна 2024.

[19] <https://www.geoapify.com/>, преузето Јуна 2024.

[20] <https://openai.com/index/chatgpt/>, преузето Јуна 2024.

[21] <https://aws.amazon.com/what-is/api/>, преузето Јуна 2024.

Кратка биографија:



Миле Кајтез је рођен 30.06.1997. године у Новом Саду. Дипломирао је на Факултету техничких наука у Новом Саду, на смеру Примењено софтверско инжењерство 2022. године. Потом, те године, уписује мастер академске студије на истом смеру и завршава их 2024. године бранећи тему “*Развој веб апликације за презентовање података о факултетима и давање предлога употребом ChatGPT AI модела*”.

SOFTVERSKO REŠENJE ZA AUTOMATIZOVANU RAZMENU ZDRAVSTVENIH KARTONA ZASNOVANO NA TEHNOLOGIJI BLOKČEJN**A SOFTWARE SOLUTION FOR AN AUTOMATED MEDICAL RECORD EXCHANGE BASED ON THE BLOCKCHAIN TECHNOLOGY**

Vladimir Jovanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj - U ovom radu predstavljeno je softversko rešenje za automatizovanu razmenu medicinskih kartona zasnovano na tehnologiji Blokčejn. Softversko rešenje obezbeđuje bezbednu razmenu podataka između zdravstvenih institucija i istraživačkih organizacija, uz očuvanje anonimnosti pacijenata. Korišćenjem pametnih ugovora, pacijentima se omogućava potpuna kontrola nad dozvolama za pristup njihovim medicinskim kartonima. Softversko rešenje koristi tehnologije Hyperledger Fabric i MongoDB, čime se postiže skalabilnost i efikasnost u realnom vremenu, pri čemu su transparentnost i bezbednost ključni prioriteti.

Ključne reči: Blokčejn, pametni ugovori, Hyperledger Fabric, MongoDB, elektronski zdravstveni karton

Abstract - This paper presents a software solution for automated medical record exchange based on blockchain technology. The system ensures secure data exchange between healthcare institutions and research organizations while preserving patient anonymity. Smart contracts enable patients to have full control over access permissions to their medical records. The software solution uses Hyperledger Fabric and MongoDB technologies, achieving scalability and real-time efficiency, with transparency and security as key priorities.

Keywords: Blockchain, Smart contracts, EHR (Electronic Health Record)

1. UVOD

U savremenom društvu, razmena podataka ima ključnu ulogu u mnogim sektorima, od finansija i proizvodnje do logistike i zdravstvene zaštite. Efikasna razmena podataka omogućava unapređenje procesa, povećanje produktivnosti i donošenje odluka zasnovanih na relevantnim informacijama, što sve zajedno doprinosi inovacijama i razvoju novih tehnologija.

U zdravstvu, razmena podataka je značajna zbog svoje uloge u spasavanju života i unapređenu zdravstvenih usluga. Uvođenje elektronskih zdravstvenih kartona (EZK), predstavlja revoluciju u načinu upravljanja i

razmene medicinskih podataka. EZK omogućavaju digitalno čuvanje svih bitnih informacija o pacijentu, što poboljšava kvalitet zdravstvenih usluga.

Napretkom nauke o podacima i ulogom analitike u obradi i tumačenju medicinskih informacija, podaci postaju izvor za inovacije u medicini. Sve veća usmerenost ka inovacijama stvara potrebu za intenzivnijom razmenom podataka između medicinskih ustanova i istraživačkih centara. Pristup kvalitetnim zdravstvenim podacima omogućava razvoj novih terapija i tehnologija, uz potrebu za sigurnim sistemima koji štite privatnost pacijenata.

Glavni izazovi u razmeni podataka uključuju obezbeđivanje bezbednosti i privatnosti medicinskih informacija, kao i automatizaciju pristupa tim podacima. Pored toga, zdravstvenim radnicima je potrebna tehnička podrška i obuka kako bi se efikasno koristili novi sistemi za razmenu podataka. Implementacija i održavanje sistema za razmenu podataka takođe zahtevaju značajne resurse, što može predstavljati prepreku, posebno za manje ustanove.

Cilj ovog rada jeste da se prethodno navedeni izazovi prevaziđu kroz primenu tehnologije blokčejn i pametnih ugovora u sistemima za razmenu medicinskih podataka. Tehnologija blokčejn obezbeđuje neizmenjivost, decentralizaciju i transparentnost podataka. Dok sa druge strane pametni ugovori omogućavaju automatizaciju ključnih procesa, poput dobijanja dozvola pristupa medicinskim kartonima i anonimizacije podataka. Na taj način, ovaj pristup osigurava efikasno upravljanje dozvolama i visoku sigurnost procesa razmene podataka.

2. PREGLED TRENUTNOG STANJA U OBLASTI

U ovom poglavlju predstavljeni su neki od najkorišćenijih sistema za razmenu medicinskih kartona.

U radu [1] predstavljena je platforma otvorenog koda *PopMedNet*, koja omogućava implementaciju i upravljanje distribuiranim mrežama zdravstvenih podataka. Platforma omogućava vlasnicima podataka da zadrže punu kontrolu nad svojim podacima, dok istraživači mogu da šalju upite za analizu podataka. Sistem koristi distribuiranu mrežnu arhitekturu koja osigurava privatnost i poverljivost podataka. Značaj ovog rada je u tome što je pomenuta platforma iskorišćena i primenjena u radu [2]. U radu [2] predstavljen je sistem *MDPH* koji trenutno obuhvata tri velike ordinacije u Masačusetsu. Pomenuti sistem omogućava kontrolisan pristup zdravstvenim kartonima iz više nezavisnih ordinacija zbog efikasnog praćenja bolesti i procene

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji je mentor bio dr Vladimir Dimitrieski, vanredni profesor.

zdravstvene i epidemiološke situacije u regionu gde je sistem primenjen.

U radu [3] predlaže se platforma *BBDS* koja koristi privatnu blokčejn mrežu za unapređenje kontrole pristupa osetljivim medicinskim podacima. Platforma omogućava da pristup podacima imaju isključivo ovlašćeni korisnici, čime se obezbeđuje odgovornost i privatnost. Tehnologija blokčejn osigurava nepromenljivost podataka i detaljno beleži sve korisničke akcije. Medicinski kartoni se čuvaju na *cloud*-u, dok se blokčejn koristi za trajno beleženje podataka o pristupu, čime se kombinuju prednosti sigurnosti blokčejna i ekonomičnosti *cloud* skladištenja.

Rad [4] predlaže upotrebu tehnologije blokčejn i pametnih ugovora na platformi *Ethereum* za bezbedno deljenje medicinskih podataka. Indeksi zdravstvenih kartona, kreirani na osnovu složenih logičkih izraza koji opisuju podatke o pacijentima, čuvaju se na blokčejn mreži, dok su stvarni podaci šifrovani i smešteni na skladištu u oblaku. Korisnici mogu pretraživati ove indekse na blokčejnu, a nakon autentifikacije i plaćanja putem pametnog ugovora, dobiti dozvolu za pristup i dekrpciju podataka, čime se obezbeđuje siguran pristup medicinskim informacijama..

Predloženo softversko rešenje koristi prednosti privatne blokčejn mreže, obezbeđujući veću kontrolu bezbednosti i integriteta podataka u odnosu na javne mreže. Za razliku od rada [3], implementiran je mehanizam za specifikaciju podataka putem logičkih uslova. Osim toga, rad uvodi mehanizam anonimizacije privatnih podataka, što dodatno povećava bezbednost sistema.

3. SPECIFIKACIJA ZAHTEVA

U okviru predloženog softverskog rešenja postoje tri vrste korisnika:

Pacijent koji nakon registracije može svoje medicinske kartone predati sistemu na obradu i čuvanje, zadržavajući potpunu kontrolu nad pristupom podacima. Pacijent može pregledati kartone, odobravati ili odbijati zahteve za pristup, pratiti istoriju pristupa i ažurirati podatke.

Zdravstvena ustanova deluje kao posrednik između pacijenata i softverskog rešenja. Kada ustanova preda karton u sistem, preuzima vlasništvo i odgovornost za dalje upravljanje podacima. Ustanova ima iste mogućnosti kao i pacijent, ali je komunikacija sa pacijentom van okvira ovog rešenja. Smatra se da ustanova ima sva potrebna ovlašćenja za dalji prenos kartona.

Istraživačka ustanova je registrovani korisnik koji zahteva pristup medicinskim podacima zbog obavljanja naučnog istraživanja. Zahtevi se šalju putem pametnih ugovora sa specifičnim logičkim uslovima. Nakon odobrenja, ustanova preuzima anonimizirane podatke prilagođene za istraživanje.

3.1 Funkcionalni zahtevi

Funkcionalni zahtevi sistema podeljeni su u tri glavne kategorije: osnovni zahtevi, rukovanje medicinskim kartonima i kontrola pristupa podacima.

Osnovni zahtevi uključuju registraciju korisnika, koja je ključna za omogućavanje pristupa svim ostalim

funkcijama sistema. Registracija korisnika je korak koji osigurava da samo verifikovani i autorizovani korisnici mogu pristupiti sistemu i koristiti njegove funkcionalnosti. Proces registracije obavlja se automatski, uz proveru identiteta korisnika i dodeljivanje odgovarajućih pristupnih prava.

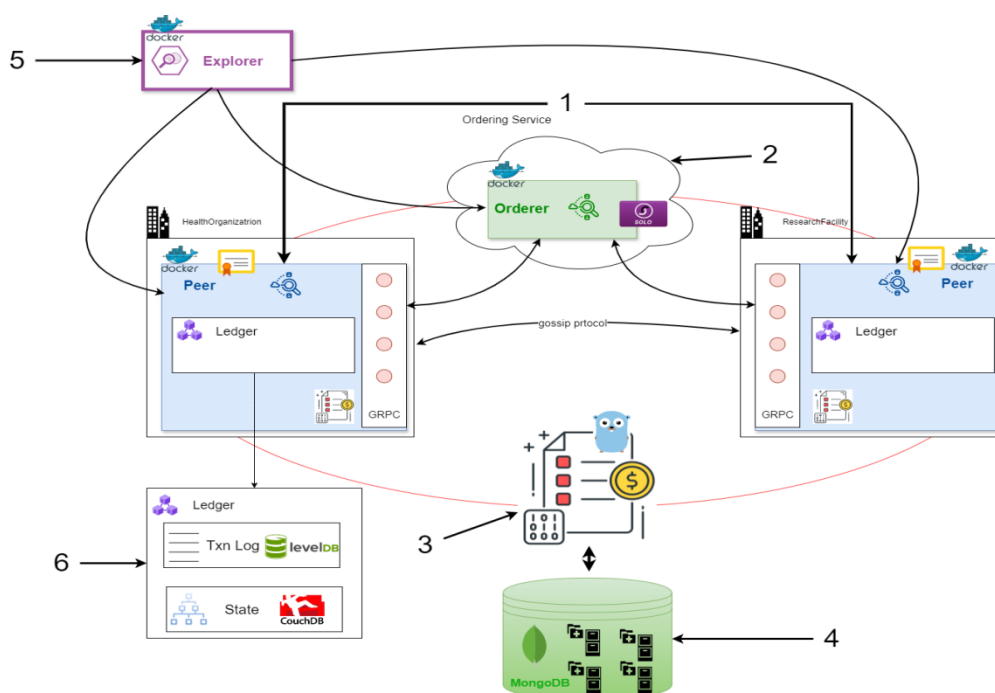
Rukovanje medicinskim kartonima obuhvata unos i ažuriranje medicinskih kartona od strane pacijenata i zdravstvenih ustanova. Prilikom unosa kartona, podaci se automatski anonimiziraju kako bi se zaštitila privatnost pacijenata. Anonimizovani podaci zatim se sigurno čuvaju, čime se obezbeđuje da su svi uneti podaci pouzdani i nepromenjeni. Ova funkcionalnost omogućava pacijentima i zdravstvenim ustanovama da zadrže kontrolu nad svojim podacima, dok istovremeno omogućava istraživačkim ustanovama pristup ažuriranim i tačnim informacijama.

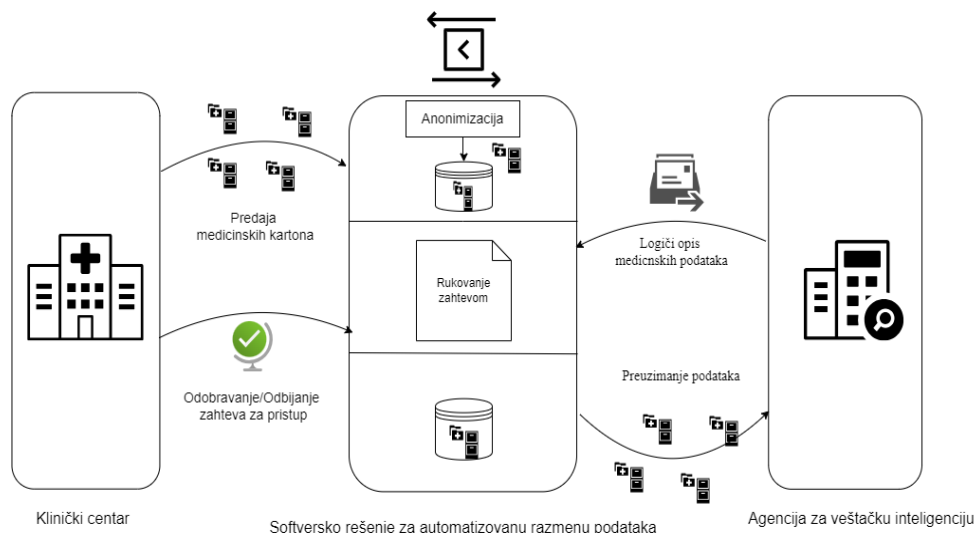
Kontrola pristupa podacima omogućava pacijentima i zdravstvenim ustanovama da upravljaju pravima pristupa svojim medicinskim kartonima. Pacijenti imaju mogućnost da odobre ili odbiju zahteve za pristup podacima koje podnose istraživačke ustanove. Zahtevi za pristup se formiraju putem logičkih upita, koji definišu podatke za istraživanje. Nakon odobrenja zahteva, istraživačke ustanove dobijaju pristup anonimiziranim podacima, prilagođenim za istraživačke potrebe, čime se obezbeđuje zaštita privatnosti pacijenata i sigurnost podataka.

4. ARHITEKTURA SISTEMA

Predloženo softversko rešenje za automatizovanu razmenu medicinskih kartona sastoji se od nekoliko glavnih komponenti. Celokupna arhitektura sistema prikazana je na slici 4.1, gde su obuhvaćene glavne komponente ovog rešenja, uključujući pir čvorove (engl. *Peer Nodes*), servis za naručivanje (engl. *Ordering Service*), pametne ugovore, skladište podataka i alat za nadgledanje sistema. Sve komponente sistema su implementirane i orkestrirane korišćenjem *Docker*-a i *Docker Compose*-a, što omogućava laku instalaciju, konfiguraciju i upravljanje celokupnom infrastrukturom.. U narednim paragrafima dat je opis komponenti.

Pir čvorovi (označeni brojem 1 na slici 4.1) su centralne komponente softverskog rešenja za automatizovanu razmenu medicinskih kartona i ključni elementi distribuirane mreže. Njihova glavna uloga je validacija transakcija, pri čemu proveravaju ispunjenost uslova transakcija i autentičnost učesnika pre nego što se one dodaju u distribuirani registar (engl. *Ledger*). Takođe, pir čvorovi izvršavaju pametne ugovore koji definišu pravila za odobravanje ili odbijanje transakcija, te održavaju kopiju distribuiranog registra, čime se osigurava dosledan i tačan prikaz stanja podataka u mreži. Distribuirani registar, koji pir čvorovi održavaju, sastoji se od transakcionog dnevnika, koji beleži promene podataka, i baze podataka stanja, koja prikazuje trenutno stanje svih objekata u sistemu. U arhitekturi, po jedan pir čvor je kreiran za svaku organizaciju, s mogućnošću horizontalnog skaliranja po potrebi. Za implementaciju distribuiranog registra korišćeni su *LevelDB* za transakcioni dnevnik i *CouchDB* za bazu podataka stanja, čime se osigurava pouzdanost i efikasnost sistema.





Slika 5.1. Dijagram toka

Klinički centar može ažurirati podatke, što je značajno za dugoročna istraživanja, i omogućava korišćenje ažuriranih podataka za validaciju modela veštačke inteligencije.

6. ZAKLJUČAK

Ovaj rad predstavlja softversko rešenje za automatizovanu razmenu medicinskih kartona zasnovano na tehnologiji Blokčejn, koje omogućava sigurnu i pouzdanu razmenu medicinskih podataka između zdravstvenih i istraživačkih ustanova. Integracijom *Hyperledger Fabric*-a i *MongoDB*-a razvijen je sistem koji omogućava anonimizaciju podataka, upravljanje dozvolama za pristup, i praćenje aktivnosti u realnom vremenu. Glavni doprinosi rada uključuju unapređenje sigurnosti i privatnosti kroz primenu tehnologije blokčejn, automatizaciju dobijanja dozvola i fleksibilno upravljanje pristupom medicinskim kartonima.

U poređenju sa postojećim rešenjima, predloženi sistem premošćuje nedostatke kao što su nedostatak automatizacije i ograničene mogućnosti pretrage podataka. Korišćenje privatne blokčejn mreže obezbeđuje veću kontrolu nad bezbednošću i integritetom podataka, dok pretraga putem logičkih uslova olakšava istraživačima pronalazak relevantnih podataka. Softversko rešenje omogućava ažuriranje podataka u realnom vremenu, što je ključno za dugoročna istraživanja.

Rezultati ostvareni kroz ovaj rad pokazali su da je moguće razviti sigurno i skalabilno softversko rešenje koje omogućava automatizovanu razmenu medicinskih kartona uz obezbeđenje visokog nivoa sigurnosti i privatnosti. Iako su postignuti značajni rezultati, postoji prostor za dalje unapređenje sistema, posebno u domenu poboljšanja načina pristupa softverskog rešenja kao i lakoće korišćenja.

U budućim istraživanjima, predlaže se unapređenje sistema kroz integraciju sa postojećim zdravstvenim sistemima i standardima, uključujući razvoj algoritama za konverziju različitih formata medicinskih kartona, čime bi se obezbedila kompatibilnost i šira primena. Takođe, integracija u okviru platforme eUprava mogla bi da

poboljša korisničko iskustvo, omogućavajući pacijentima lakše upravljanje dozvolama za pristup njihovim podacima i praćenje statusa zahteva. Kao dodatni pravac razvoja, predlaže se uvođenje kriptovalute unutar platforme koja bi podstakla pacijente da dobiju nagrade za dozvolu korišćenja njihovih podataka u istraživanjima, što bi moglo značajno unaprediti dostupnost podataka..

7. LITERATURA

- [1] DEE, Daniel, et al. PopMedNet: A scalable and extensible open-source informatics platform designed to facilitate the implementation and operation of distributed health data networks. *Journal of Open Source Software*, 2022, 7.71: 4062.
- [2] VOGEL, Joshua, et al. MDPHnet: secure, distributed sharing of electronic health record data for public health surveillance, evaluation, and planning. *American journal of public health*, 2014, 104.12: 2265-2270
- [3] XIA, Qi, et al. BBDS: Blockchain-based data sharing for electronic medical records in cloud environments. *Information*, 2017, 8.2: 44.
- [4] CHEN, Lanxiang, et al. Blockchain based searchable encryption for electronic health record sharing. *Future generation computer systems*, 2019, 95: 420-429.

Kratka biografija:



Vladimir Jovanović rođen je 6. jula 1999. godine u Bijeljini, Bosna i Hercegovina, gde je stekao osnovno obrazovanje u osnovnoj školi „Sveti Sava“. Dalje školovanje nastavio je u Bijeljini gde je završio Gimnaziju „Filip Višnjić“. Školske 2018/2019 godine upisuje se na Fakultet tehničkih nauka Univerziteta u Novom Sadu, smer Računarstvo i automatika. Osnovne akademske studije završio je 2022. godine i iste godine upisuje se na master akademske studije na studijskom programu Računarstvo i automatika Fakulteta tehničkih nauka. U toku master studija izabran je na poziciju Saradnika u nastavi na Fakultetu tehničkih nauka.

ANALIZA DELOVANJA TERMIČKE ZAŠTITE KOD OPASNIH POGONSKIH STANJA ASINHRONIH MOTORA

ANALYSIS OF THERMAL PROTECTION TRIPPING IN ABNORMAL OPERATION CONDITIONS OF ASYNCHRONOUS MOTORS

Biljana Milošević, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – DISTRIBUIRANI
ELEKTROENERGETSKI RESURSI I PAMETNE
MREŽE**

Kratak sadržaj – Ovaj rad bavi se analizom delovanja termičke zaštite kod opasnih pogonskih stanja asinhronih motora. Za tu analizu korišćen je matematički model asinhronog motora, a zatim su napravljene simulacije opasnih pogonskih stanja u Simulinku. Posmatrani su naponi, brzina i struje u ovim stanjima sa ciljem da se proceni da li će se delovati termičkom zaštitom.

Ključne reči – Asinhroni motor, opasna pogonska stanja asinhronih motora, termički relej.

Abstract – This paper deals with the analysis of the effect of thermal protection in abnormal operating conditions of induction motors. For this analysis, a mathematical model of an induction motor was used, and then simulations of dangerous operating conditions were made in Simulink. Voltages, velocities and currents in these states were observed in order to determine whether thermal protection will be applied.

Keywords – Asynchronous motor, abnormal operating conditions of asynchronous motors, thermal protection.

1. UVOD

Ovaj rad se bavi analizom delovanja termičke zaštite asinhronog motora u opasnim pogonskim stanjima. Za te analize je korišćen odgovarajući matematički model asinhronog motora zajedno sa Simulink paketom.

U drugoj glavi dat model asinhronog motora potreban za analize u ovom radu. U trećoj glavi su data opasna pogonska kao i ona koja se razmatraju u ovom radu. Dat je i opis termičke zaštite od ovakvih stanja. U četvrtoj glavi je analiziran delovanje termičke zaštite za izabrana opasna pogonska stanja asinhronog motora pomoću programskog paketa Simulink. U petoj glavi je zaključak, a u šestoj literatura koja je korišćena u izradi ovog rada.

2. ASINHRONI MOTOR

Za potrebe analize opasnih pogonskih stanja korišćen je matematički model asinhronog motora. Praktično standardni postupak podrazumeva da polazi od matematičkog

modela asinhronog motora u faznom domenu koji se zatim primenom transformacija svodi u dq domen (rotorske veličine svedene na statorsku stranu) [1]:

$$u_d = R_s i_d + \frac{d\psi_d}{dt} - \omega_s \psi_q, \quad (1)$$

$$u_q = R_s i_q + \frac{d\psi_q}{dt} + \omega_s \psi_d, \quad (2)$$

$$0 = R_r i'_d + \frac{d\psi'_d}{dt} - \omega_K \psi'_q, \quad (3)$$

$$0 = R_r i'_q + \frac{d\psi'_q}{dt} + \omega_K \psi'_d, \quad (4)$$

$$\psi_d = L_s i_d + L_m i'_d, \quad (5)$$

$$\psi_q = L_s i_q + L_m i'_q, \quad (6)$$

$$\psi'_d = L_r i_d + L'_r i'_d, \quad (7)$$

$$\psi'_q = L_r i_q + L'_r i'_q, \quad (8)$$

$$\frac{d\omega}{dt} = \frac{p}{J} (m_e - m_m - \frac{k_{tr}}{p} \omega), \quad (9)$$

gde su:

u_d, u_q – naponi na d i q namotajima statora,
 i_d, i_q – struje u d i q namotajima statora,
 ψ_d, ψ_q – fluksni obuhvati d i q namotaja statora,
 i'_d, i'_q – struje u D i Q namotajima rotora,
 ψ'_d, ψ'_q – fluksni obuhvati u D i Q namotajima rotora,
 R_s, R_r – otpornosti namotaja statora i rotora,
 ω_s – ugaona učestanost napajanja i
 ω_K – ugaona učestanost klizanja,
 L_s – sopstvena induktivnost statora,
 L_r – sopstvena induktivnost rotora,
 L_m – međusobna induktivnost statora i rotora,
 p – broj pari polova,
 J – momenat inercije,
 ω – električna ugaona brzina rotora,
 k_{tr} – koeficijent trenja.

Ovakvo dobijeni model asinhronog motora se koristi za simulacije i proveru delovanja termičke zaštite kod opasnih pogonskih stanja asinhronog motora.

3. OPASNA POGONSKA STANJA ASINHRONIH MOTORA

U opasna pogonska stanja asinhronog motora se ubrajaju [2]:

- Preopterećenje asinhronog motora – ovakvo stanje nastaje kada je momenat opterećenja asinhronog motora veći od nazivnog momenta.

NAPOMENA:

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

- Nestanak i ponovni povratak napona – do ovog stanja dolazi usled planiranih i neplaniranih isključenja mreže, pregrevanja, preopterećenja i kratkih spojeva, nestabilnog napajanja mreže.
- Pojava inverznog redosleda faza – ovo stanje uglavnom je posledica ljudske greške usled zamene faznih provodnika prilikom priključenja.
- Rad trofaznog asinhronog motora bez jedne faze – rad asinhronog motora bez jedne faze je situacija kada u toku rada dođe do kvara na jednoj fazi.
- Predugo vreme zaletanja asinhronog motora – u slučaju kada pokretanje asinhronog motora traje dugo i kada su struje pri pokretanju velike može doći do termičkog preopterećenja asinhronog motora.

Ako se ovakva stanja ne bi na vreme eliminisala, dovela bi do pregrevanja i proboja izolacije, odnosno, uništenja samog asinhronog motora.

U ovom radu su analizirana sledeća pogonska stanja asinhronog motora:

1. Zalet asinhronog motora u normalnim uslovima.
2. Prekid faze sa visokonaponske (VN) strane transformatora.
3. Start sa povećanim momentom opterećenja.

Poslednja dva stanja su opasna pogonska stanja asinhronog motora za koje je analizirano da li će termička zaštita delovati. Provera zaštite rađena je tako što je na grafike struje starta asinhronog motora dodata karakteristika termičkog releja. Ukoliko u stanju kvara karakteristika termičkog releja (isprekidanom crvenom linijom je obeležene vrednosti vrha trenutne vrednosti podešenja ove zaštite) preseca vrednost struje asinhronog motora znači da zaštita deluje.

3.1. Zaštita motora termičkim relejom

Rad termičkog releja zasnovan je na praćenju struje koja teče kroz štice objekta – u ovom slučaju asinhroni motor. Kroz ovu zaštitu protiče struja asinhronog motora i isključuje napajanje ukoliko struja pređe zadati prag. Ove zaštite su najčešće bimetalne zaštite prilagođene datom tipu asinhronog motora kako bi se postiglo brže zagrevanje ove zaštite u odnosu na štice asinhroni motor.

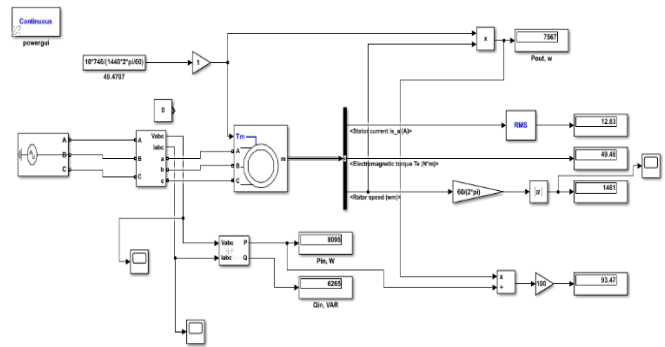
Standardno podešena termička zaštita podrazumeva da je njena karakteristika delovanja u ustaljenom stanju za 5% viša u odnosu na nominalnu struju asinhronog motora, dok u periodu zaletanja motora ne sme presecati krive vrednosti struja asinhronog motora.

4. PROVERA DELOVANJA TERMIČKE ZAŠTITE

U ovom radu modelovan je asinhroni motor snage 7.5kW koji je priključen na izvor napajanja preko transformatora 10/0.4 kV/kV – prikaz ove šeme u Simulinku je na slici 1. Sve simulacije u ovom radu su napravljene korišćenjem ove šeme.

4.1. Rad motora u normalnim uslovima

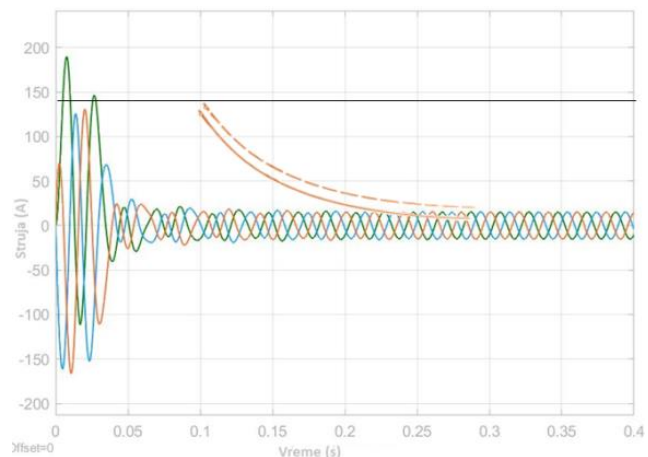
Ovo je polazni slučaj u odnosu na koji se razmatraju ostala opasna pogonska stanja. Na slici 2. predstavljena je zavisnost struje u vremenu i na njoj je dodata karakteristika termičkog releja koji se koristi za zaštitu asinhronog motora.



Slika 1. Razmatrani Simulink model motora i transformatora

Na slici 2. se može primetiti da su struje u fazama B i C asimetrične na početku zaleta. Ovo je posledica postojanja aperiodične komponente struje u tim fazama koja se vremenom prigušuje, pa tako nakon 0.15 s, ta asimetrija se gubi. Sve tri struje u ustaljenom stanju imaju maksimalnu vrednost od 17A.

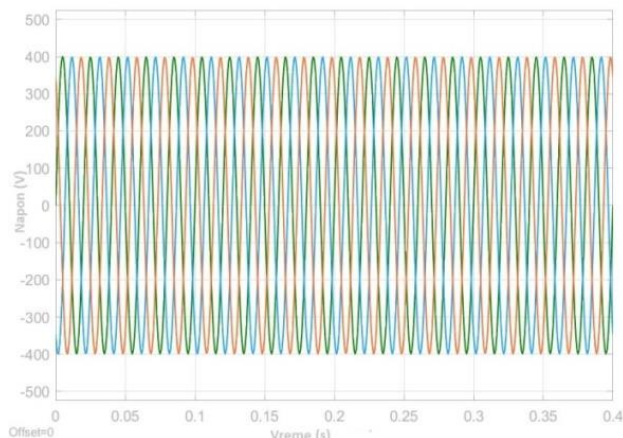
Sa slike 2. primećuje se da karakteristika termičkog releja (isprekidana crvena linija) ne preseca struju, odnosno, relej neće delovati.



Slika 2. Zavisnost struja u vremenu za slučaj zaleta u normalnim uslovima

Na slici 3. predstavljeni su naponi kada motor radi u stanju bez kvara. Vrednost napona u sve tri faze je 400V i fazno su pomereni za 120°.

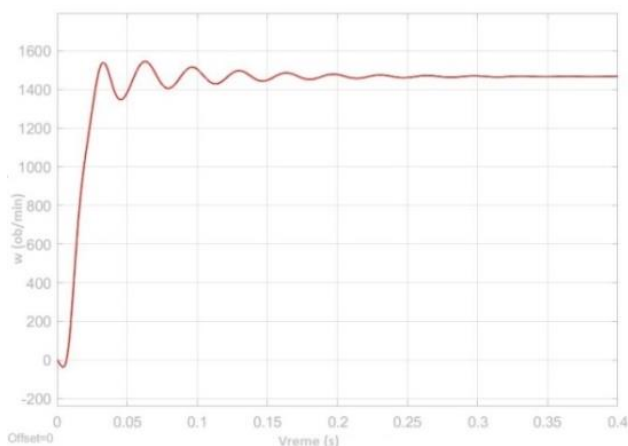
Iz praktičnih razloga je strujno vremenska karakteristika termičkog releja nacrtana nesrazmerno. Do preseka karakteristika termičkog releja i struje asinhronog motora bi došlo obično posle više desetina sekundi. Pošto u razmatranjima koja slede se podrazumeva da je termička zaštita podešena po pravilima podešavanja i saglasno datom tipu motora, u razmatranjima koja slede razmatra se samo da li će doći do delovanja termičkom zaštitom ili ne. Karakteristika termičkog releja prikazana je sa sažetom vremenskom osom da bi se mogla prikazati u graficima struja asinhronog motora neposredno posle ustaljenja prelaznih procesa po završetku zaleta asinhronog motora. To je urađeno sa ciljem da se ne bi nepotrebno produžavao dinamički proračun od nekoliko minuta ili par desetina minuta. Ovo je korišćeno na svim strujnim graphicima sa karakteristikom termičkog releja.



Slika 3. Zavisnost napona u vremenu za slučaj zaleta u normalnim uslovima

Na slici 4. data je zavisnost brzine obrtanja asinhronog motora u vremenu.

Sa slike 3. može se uočiti da se brzina asinhronog motora ustali nakon 0.35s na vrednost od 1461 obr/min.



Slika 4. Zavisnost brzine obrtanja u vremenu pri zaletu u normalnim uslovima

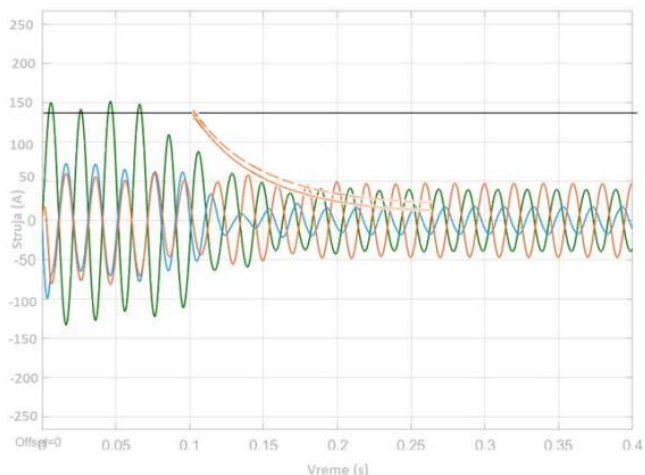
4.2. Prekid faze sa VN strane transformatora

Asinhroni motor je preko transformatora 10/0.4 kV/kV povezan na izvor napajanja. Može se desiti da u toku rada asinhronog motora dođe do prekida faze sa VN strane transformatora (npr. pregorevanje jednog osigurača). Na slici 5. predstavljene su struje sa niskonaponske (NN) strane transformatora kada se desi takav prekid.

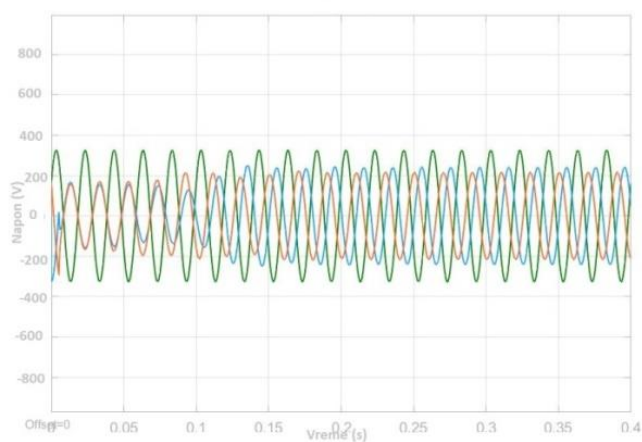
Sa slike 5. se uočava da se struja u fazi C sa NN strane ustaljuje na vrednost od 49A (pri prekidu u fazi C sa VN strane). Struja u fazi A je nešto manja u odnosu na normalno stanje i ustaljuje se na 41A, a u fazi B najmanja i iznosi 14A.

Sa ove slike uočava se da karakteristika termičkog releja preseca krive struja. Dakle, ukoliko se desi prekid faze sa VN strane transformatora termički relej će delovati i asinhroni motor će biti zaštićen.

Na slici 6. date su vrednosti napona kada se desi prekid sa VN strane transformatora. Napon faze A u kojoj se desio prekid je 320V, dok je u ostale dve faze manji (u fazi B je 250V i fazi C je 216V).



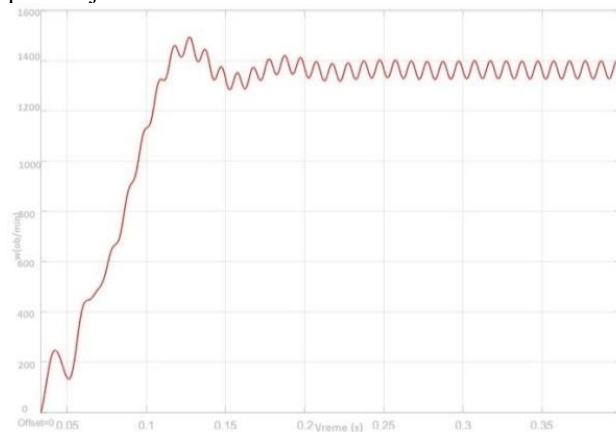
Slika 5. Struje starta asinhronog motora pri prekidu faze sa VN strane transformatora



Slika 6. Napon pri prekidu faze sa VN strane transformatora

Na slici 7. prikazana je brzina obrtanja asinhronog motora kada se desi prekid faze sa VN strane transformatora.

Sa slike 7. se uočava da se brzina asinhronog motora ustaljuje na 1400ob/min. Ta brzina je manja od nominalne brzine obrtanja asinhronog motora. Brzina u ustaljenom stanju ima oscilacije koje imaju za rezultat povećanje gubitaka u asinhronom motoru. Zbog ovih gubitaka nije prihvatljivo da motor radi u ovakvom režimu.

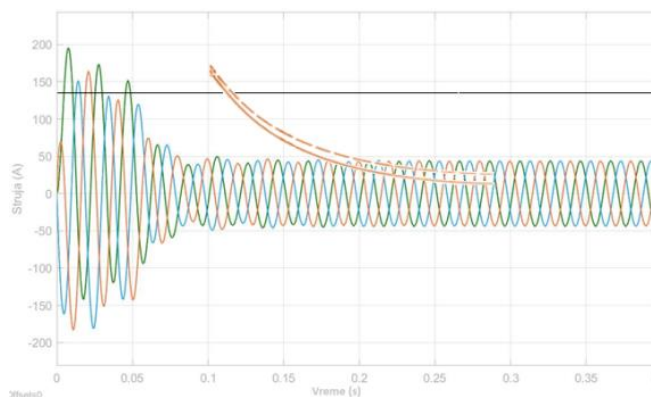


Slika 7. Zavisnost brzine obrtanja u vremenu pri prekidu faze sa VN strane transformatora

4.3. Start asinhronog motora sa povećanim momentom opterećenja

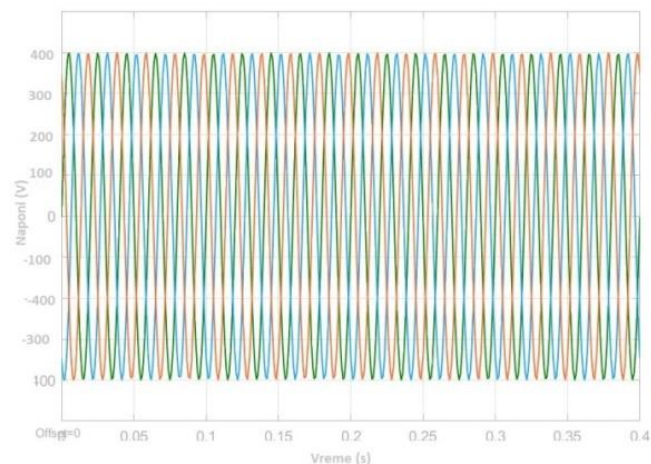
Momenat opterećenja za asinhroni motor na kome su vršene simulacije iznosi 49Nm. Neka se pretpostavi da se momenat opterećenja poveća za 100Nm, odnosno, da je 149Nm. Pri takvom startu asinhronog motora dobijaju se struje koje su prikazane na slici 8.

Struje se ustaljuju nakon 0.13s uz porast struja u ustaljenom stanju u odnosu na struje sa nominalnim momentom opterećenja za 30A. Vrednosti struja su iste u sve tri faze u ustaljenom stanju – slika 8. Sa ove slike se uočava da se karakteristika termičke zaštite preseca struju starta asinhronog motora, dakle, zaštita će delovati ukoliko dođe do povećanja momenta opterećenja.



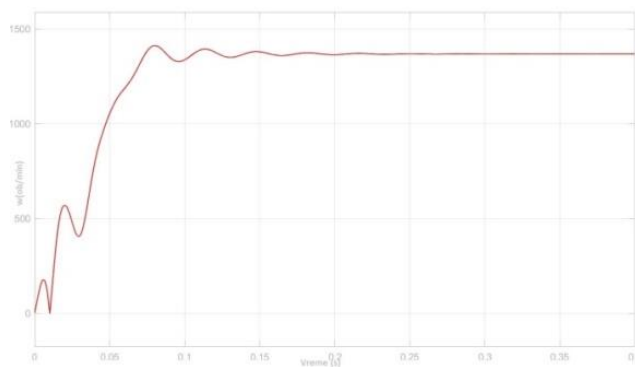
Slika 8. Struje starta pri povećanom momentu opterećenja

Na slici 9. dati su naponi za slučaj sa pomenutim povećanjem momenta opterećenja asinhronog motora. Naponi su nepromenjeni u odnosu na zalet asinhronog motora u normalnim uslovima, što je i očekivan rezultat.



Slika 9. Naponi pri povećanom momentu opterećenja

Na slici 10. data je brzina obrtanja pri pomenutom povećanom momenta opterećenja koja je očekivano manja u odnosu na nominalnu.



Slika 10. Brzina obrtanja pri povećanom momentu opterećenja

5. ZAKLJUČAK

U radu je analizirano delovanje standardno podešene termičke zaštite za izabrane slučajeve opasnog pogonskog stanja. Dobijeni rezultati pokazuju da standardno podešena termička zaštita deluje kod svih razmatranih opasnih pogonskih stanja.

6. LITERATURA

- [1] Karlo Zamljačenec, „Zaštita asinhronog kaveznog motora“, Osijek, 2019.
- [2] Duško Bekut, „Relejna zaštita“, Fakultet tehničkih nauka, Novi Sad, 2009.

Kratka biografija:



Biljana Milošević rođena je u Loznici 1997. godine. Osnovne studije završila na Fakultetu tehničkih nauka u Novom Sadu 2021. godine iz oblasti Energetika, elektronika i telekomunikacije, smer Elektroenergetski sistemi. Master rad, na istom fakultetu, je odbranila 2024. godine. Kontakt: biljanavasiljevic399@gmail.com

PRIMARNA I SEKUNDARNA KONTROLA ZA KOMPENZACIJU NESIMETRIJE NAPONA ZAJEDNIČKE SABIRNICE SADEJSTVOM PRETVARAČA U MIKROMREŽI

PRIMARY AND SECONDARY CONTROL FOR VOLTAGE UNBALANCE COMPENSATION AT THE POINT OF COMMON COUPLING IN A MICROGRID

Aleksandar Dzever, Marko Vekić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovaj rad predstavlja metodu kompenzacije nesimetrije napona zajedničke sabirnice sadejstvom pretvarača energetske elektronike u mikromreži koja radi u ostvrskom režimu. Prikazan je simulacioni model mikromreže sa hijerarhijskom arhitekturom upravljanja, njen rad kada se priključi neuravnotežen potrošač a zatim i dejstvo sekundarne kontrole.

Ključne reči: Energetski pretvarači, hijerarhijska arhitektura, centralizovana sekundarna kontrola, kompenzacija neuravnoteženosti napona, mikromreže

Abstract – This paper showcases a voltage unbalance compensation method at the point of common coupling using power converters in an islanded microgrid. A simulation model of the grid with hierarchical control architecture is presented, followed by its operation when an unbalanced load is connected and the subsequent compensation effect from the secondary control.

Keywords: Power converters, hierarchical architecture, centralized secondary control, voltage unbalance compensation, microgrids

1. UVOD

Usled kontinuiranog razvoja upravljačkih algoritama, decentralizacije elektroenergetskih sistema, ekonomskih podsticaja i opadanja investicionih troškova vezanih za izgradnju obnovljivih izvora energije mikromreže postaju sve zastupljenije [1]. Njihova integracija u distributivne mreže unosi nove kvalitete koji transformišu tradicionalnu topologiju na srednjem i niskom naponskom nivou u aktivne distributivne mreže, ali unose i potencijalne rizike i probleme, kojima se mora pravilno pristupiti u cilju održanja stabilnosti mikromreža i distributivne mreže, vodeći računa da se ne naruši kvalitet električne energije i ekonomska isplativost rada istih. Kako je kvalitet električne energije disciplina koja pokriva širok opseg problema, u ovom radu će pažnja biti usmerena ka smanjenju uticaja režima nesimetričnih napona po amplitudi, čije dejstvo ostavlja posledice po potrošače, iako su oni neretko uzrok tog režima.

NAPOMENA:

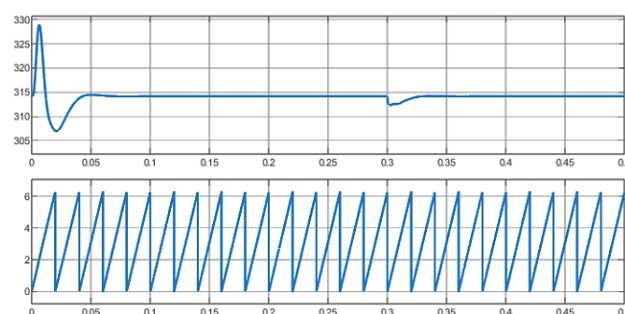
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Marko Vekić, vanr. prof.

1.1. Zadatak rada

Pristup kojim se vrši kompenzacija nesimetrije napona po amplitudi na zajedničkoj sabirnici zasnovan je na upotrebi centralizovane sekundarne kontrole koja merene fazne napone transformiše u direktne i poprečne komponente napona direktnog i inverznog redosleda, vrši proračun faktora neuravnoteženosti napona a on se zajedno sa referentnom vrednošću dovodi na PID regulator, na čijem izlazu se generiše korekcionni signal napona i prosleđuje naponskim petljama svakog energetskog pretvarača u datoj mikromreži. Prvo je prikazan rad jednog pretvarača sa poremećajem i bez kompenzacije, zatim rad više pretvarača sa droop kontrolom bez poremećaja i dejstvo sekundarne kontrole za kompenzaciju nesimetrije napona.

2. RAD PRETVARAČA SA POREMEĆAJEM I BEZ KOMPENZACIJE NESIMETRIJE NAPONA

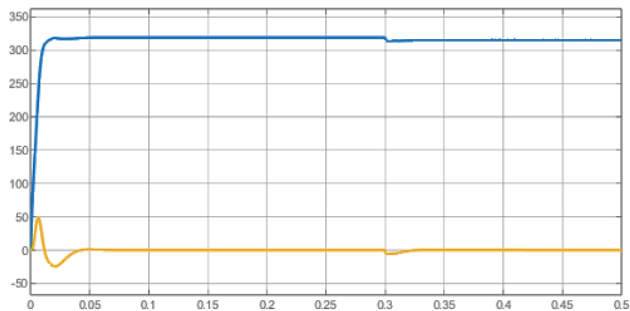
Inicijalni model sastoji se od idealnog naponskog izvora jednosmernog napona, energetskog pretvarača, LC filtra, uravnoteženog trofaznog i neuravnoteženog, jednofaznog potrošača. Primarna kontrola sastoji se od sklopa fazno zaključane petlje sa dvostrukim uopštenim integratorom drugog reda, to jest *DSOGI – PLL* i regulacionih petlja napona i struje sa proporcionalno-rezonantnim regulatorima. Po priključenju neuravnoteženog potrošača trenutku $t = 0.3s$, *DSOGI – PLL* nakon kratkotrajnog propada učestanosti, nastavlja sa pravilnim radom i sinhronizacijom sa učestanošću mreže (slika 2.1.).



Slika 2.1. Učestanost i ugao mrežnog napona *DSOGI PLL* sklopa pri naponskom poremećaju

To potvrđuje i slika 2.2. na kojoj se vidi direktna i poprečna komponenta napona dobijena Parkovom transformacijom nad proračunatim $\alpha\beta$ komponentama sa izlaza *SOGI-QSG* estimatora koji su eliminisali oscilacije na dvostrukoj učestanosti mreže. Ovom simulacijom je prikazan uticaj poremećaja bez kompenzacije i sposobnost *DSOGI – PLL* sklopa da efikasno radi uprkos nesimetrije napona, što će

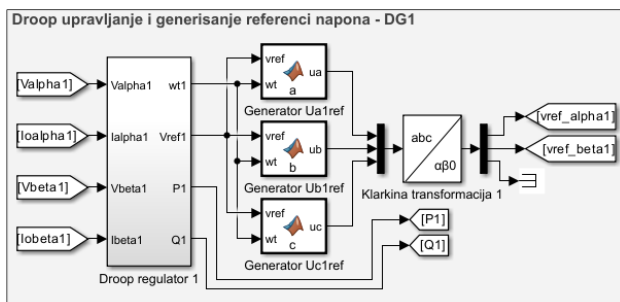
biti od značaja kada se upravljanje simulacione mreže proširi sa *droop* regulacijom i sekundarnom kontrolom.



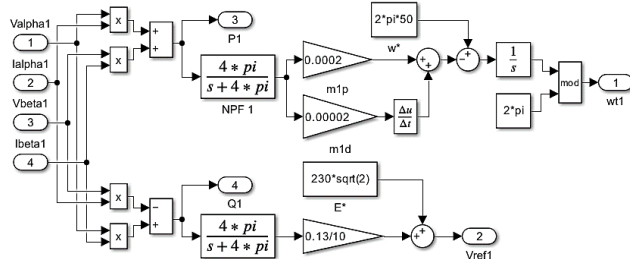
Slika 2.2. Direktna i poprečna komponenta napona sa SOGI – QSG estimatorima pri naponskom poremećaju

3. DVA PRETVARAČA U MIKROMREŽI SA DROOP UPRAVLJANJEM BEZ POREMEĆAJA

Proširenje modela dodatnim pretvaračem, *droop* regulacijom (slike 3.1., 3.2.) i virtuelnim impedansama je izvršeno kako bi se dobio referentni model mreže za proveru delovanja sekundarne kontrole. Korišćenje *droop*

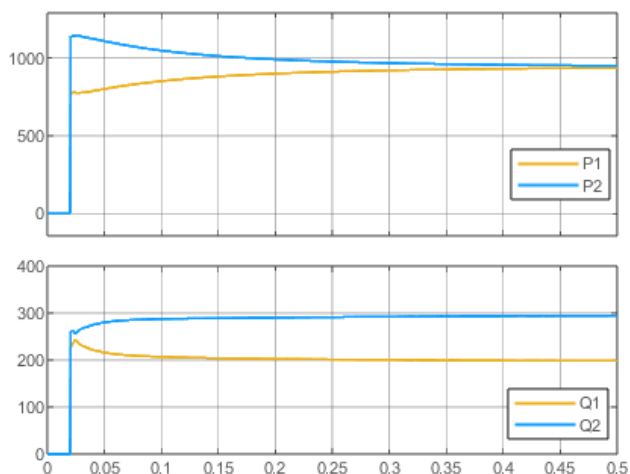


Slika 3.1. Droop upravljanje i generisanje referenci napona jednog pretvarača



Slika 3.2. Unutrašnja struktura Droop regulacije jednog pretvarača

regulacije u mikromrežama je dobro prihvaćena metoda i pruža dobru regulaciju napona i učestanosti uz postojanje male greške u ustaljenom stanju, što se može kompenzovati uvođenjem struktura u sekundarnu kontrolu [2]. Dodavanjem virtuelnih impedansi u primarnu kontrolu, postignuto je poboljšanje dinamike i preciznija podela aktivne i reaktivne snage između pretvarača. (slika 3.2.). Ipak, i sa virtuelnom impedansom, precizna podela proizvodnje reaktivne snage može biti problematična pri pojavi nesimetričnog režima, te se virtuelna impedansa može poboljšati nekim naprednijim algoritmom poput [3] gde je na *droop* regulaciju primenjena adaptivna virtuelna impedansa sa konsenzusnom kontrolom reaktivne snage.



Slika 3.1. Aktivne i reaktivne snage pretvarača DG₁ i DG₂

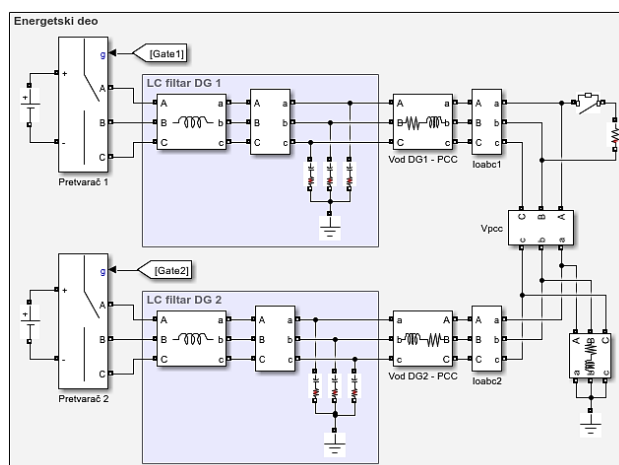
4. KOMPENZACIJA NESIMETRIJE NAPONA ZAJEDNIČKE SABIRNICE SADEJSTVOM DVA PRETVARAČA SEKUNDARNOM KONTROLOM

Model mikromreže sa primarnom i sekundarnom kontrolom, prikazan je na slikama (4.1.), (4.2.), (4.3.). Filtrirani naponski signali dobijaju se iz direktne i poprečne komponente napona direktnog i inverznog redosleda koji se dovode na kaskadno vezane pokretne usrednjavajuće filtre (MAF) (slika 4.4.) i filtre nepropusnika opsega učestanosti (*Notch filter*) (slika 4.5.) i oni su ulazne veličine za blok kompenzacije nesimetrije napona [4], koji jednačinama 4.1. i 4.2. a zatim njihovim deljenjem i množenjem sa 100 (jednačina 4.3.) daje koeficijent neuravnoteženosti napona (*VUF*).

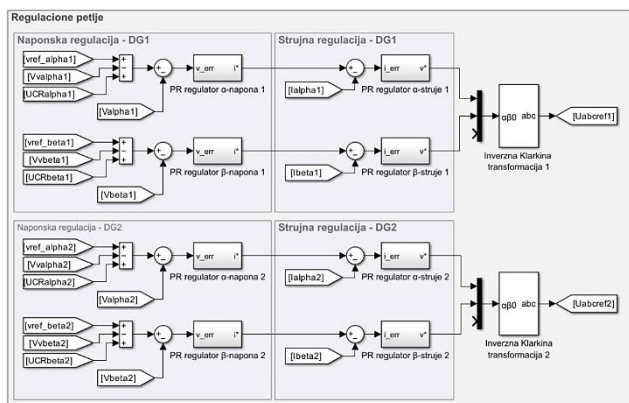
$$V_{dq+} = \sqrt{(V_{d+})^2 + (V_{q+})^2} \quad (4.1.)$$

$$V_{dq-} = \sqrt{(V_{d-})^2 + (V_{q-})^2} \quad (4.2.)$$

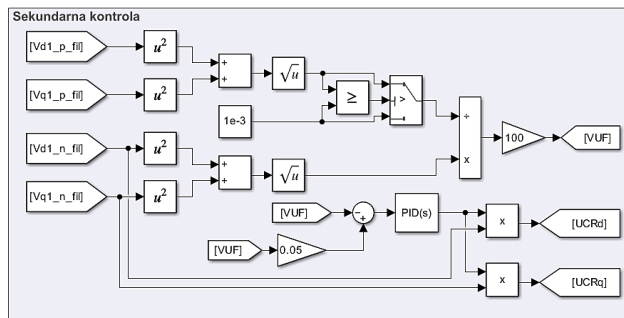
$$VUF = \frac{V_{dq-}}{V_{dq+}} 100 \quad (4.3.)$$



Slika 4.1. Energetski deo modela

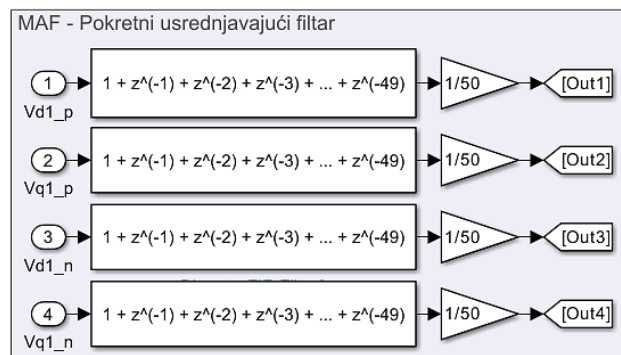


Slika 4.2. Regulacione petlje primarne kontrole

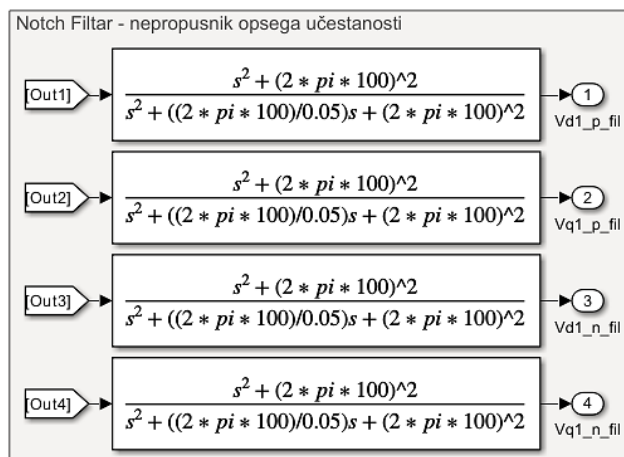


Slika 4.3. Sekundarna kontrola za kompenzaciju nesimetrije napona

Korekcionni signali za naponsku regulaciju svakog pretvarača dobijaju se od strane PID regulatora nakon Inverzne Parkove transformacije čime se prilagođavaju PR regulatorima u $\alpha\beta$ domenu.

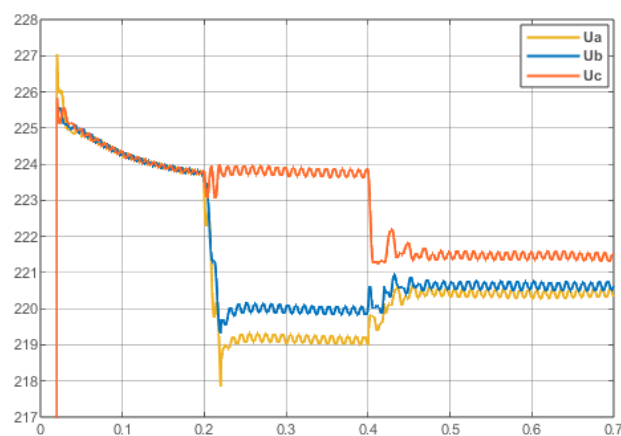


Slika 4.4. Pokretni usrednjavajući filter

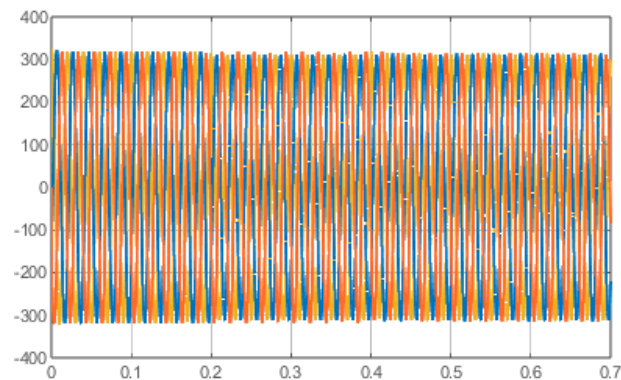


Slika 4.5. Filter nepropusnik opsega učestanosti

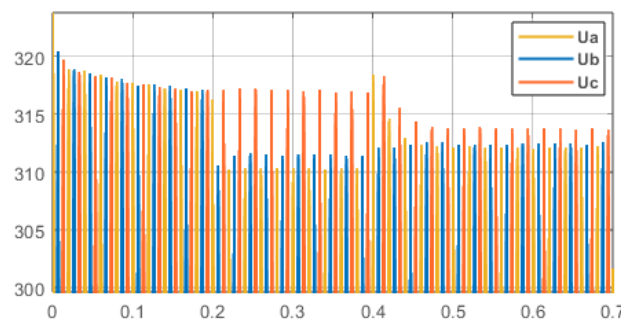
Sa slikama 4.6., 4.7., 4.8. se vidi uspješno smanjenje nesimetrije napona ispod graničnih vrednosti koje propisuju međunarodni standardi poput [5-7]. U trenutku $t = 0.2s$ je priključen neuravnotežen potrošač, a u $t = 0.4s$ je uključena sekundarna kontrola. Dodatnim podešavanjem parametara virtualne impedanse i parametara PID regulatora sekundarne kontrole, može se uticati na odzive napona. U slučaju gde su virtualne impedanse istih parametara, zabeležena je bolja kompenzacija nesimetrije, po ceni malo lošije raspodele proizvodnje aktivne i reaktivne snage (tabela 5.1.). Kada se pogledaju odzivi napona na izlazu pretvarača, zaključuje se da ova metoda kompenzacije nesimetrije, blago uvećava nesimetriju napona na izlazima pretvarača (slika 5.3.) uz bolju kompenzaciju nesimetrije na zajedničkoj sabirnici.



Slika 4.5. Efektivna vrednost napona zajedničke sabirnice pre i posle delovanja kompenzacije nesimetrije napona



Slika 4.6. Napon na zajedničkoj sabirnici pre i posle dejstva sekundarne kontrole



Slika 4.7. Napon na zajedničkoj sabirnici pre i posle dejstva sekundarne kontrole, uveličano

Koliko je bitna dobra parametrizacija virtuelne impedanse, govori i činjenica da je nesimetrija napona na izlazu pretvarača koji je električno udaljeniji od neuravnoteženog potrošača, više umanjena od nesimetrije napona prvog pretvarača.

Veličina	Iste impedanse	Različite impedanse
Nesimetrija V_{abcRMS} - pre	1.25 %	1.81 %
Nesimetrija V_{abcRMS} - posle	0.23 %	0.36 %
Nesimetrija $V_{oabc1RMS}$ - pre	1.21 %	1.75 %
Nesimetrija $V_{oabc1RMS}$ - posle	1.31 %	1.89 %
Nesimetrija $V_{oabc2RMS}$ - pre	1.47 %	2.06 %
Nesimetrija $V_{oabc2RMS}$ - posle	0.69 %	0.72 %

Tabela 4.1. Napon zajedničke sabirnice, pretvarača i nesimetrija napona pre i posle sekundarne kontrole

5. ZAKLJUČAK

Ovaj rad je za cilj imao predstavu hijerarhijske arhitekture upravljanja sa primarnom kontrolom i centralizovanom sekundarnom kontrolom za kompenzaciju nesimetrije napona zajedničke sabirnice sadejstvom dva pretvarača, u ostvrskoj mikromreži. Pripremanje simulacionog modela mreže za sekundarnu kontrolu započeto je sa jednim pretvaračem i projektovanjem primarne kontrole koja se sastoji od *DSOGI-PLL* sklopa fazno zaključane petlje sa ulogom precizne i pouzdane sinhronizacije pretvarača i učestanosti mreže, *LC* filtra za poboljšanje kvaliteta napona i struje na izlazu pretvarača, naponskih i strujnih regulacionih petlja sa *PR* regulatorima i blokom za filtriranje direktne i poprečne komponente napona direktnog i inverznog redosleda. Dalje proširenje modela postignuto je sa *droop* regulatorima i virtuelnim impedansama i na kraju dodavanjem sekundarne kontrole za kompenzaciju nesimetrije napona. Rezultati simulacije pokazuju uspešno suzbijanje nesimetrije napona zajedničke sabirnice ispod graničnih vrednosti koje preporučuju međunaradni standardi i propisi [5,6] uz povećanje nesimetrije napona na izlazima pretvarača. Usavršavanje celokupne kontrole potencijalno bi se usmerilo na rešavanje ostalih problema kvaliteta električne energije poput izobličenja napona i struje od strane viših harmonika, međuharmonika, naponskih skokova, ulegnuća, šuma, korišćenjem naprednih algoritama *droop* regulacije i virtuelne impedanse [3]-[7].

6. LITERATURA

- [1] Guchhait, R., Sarkar, B., *Increasing Growth of Renewable Energy: A State of Art*. Energies 2023, 16, 2665.
- [2] Ghosh, A., Zare, F., *Control of Power Electronic Converters with Microgrid Applications*. Hoboken, New Jersey: Wiley, 2023.
- [3] Lyu Z, Wei Q, Zhang Y, Zhao J, Manla E. *Adaptive Virtual Impedance Droop Control Based on*

Consensus Control of Reactive Current. Energies. 2018; 11(7):1801.

- [4] Savaghebi, M., Alireza, J., Juan Carlos, V., Josep M., G., *Secondary Control Scheme for Voltage Unbalance Compensation in an Islanded Droop-Controlled Microgrid*, IEEE Transactions on Smart Grid, vol 3, no. 2, Jun. 2012.
- [5] ANSI C84.1-2020, *American National Standard for Electric Power Systems and Equipment—Voltage Ratings (60 Hz)*
- [6] IEC 60034-26, *Rotating electrical machines - Part 26: Effects of unbalanced voltages on the performance of three-phase cage induction motors*
- [7] J. He and Y. W. Li, *An enhanced microgrid load demand sharing strategy*, IEEE Trans. Power Electron., vol. 27, pp. 3984–3995, Sept. 2012.

Kratka biografija:



Aleksandar Džever rođen je u Subotici 1996. godine. Osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu završio je 2022. godine na studijskom programu energetika, elektronika i telekomunikacije. Iste godine je upisao master akademske studije na studijskom programu energetska elektronika i električne mašine.



dr Marko Vekić je vanredni profesor na Fakultetu tehničkih nauka u Novom Sadu, na Katedri za Energetsku elektroniku i pretvarače. Diplomirao je 2005. godine na Fakultetu tehničkih nauka, smer energetska elektronika i električne mašine. Doktorske studije završio je 2014. godine na Fakultetu tehničkih nauka u Novom Sadu. Oblasti interesovanja su mu energetska elektronika u prenosnim i distributivnim mrežama, mikromreže i kvalitet električne energije.

ANALIZA MEHANIZAMA MREŽNE BEZBEDNOSTI NA AWS ANALYSIS OF NETWORK SECURITY MECHANISMS ON AWS

Nebojša Gordić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *Izrada sistema sastavljenog od više aplikacija/servisa sa akcentom na postavljanje istih u AWS cloud i njihovu zaštitu firewall-om.*

Ključne reči: *Firewall, AWS, Bezbednost, Cloud, Amazon Web Service, Bezbednosne grupe*

Abstract – *Creation of a system composed of several applications/services with an emphasis on placing them in the AWS cloud and protecting them with a firewall.*

Keywords: *Firewall, AWS, Security, Cloud, Amazon Web Service, Security Groups*

1. UVOD

U savremenom digitalnom dobu, internet komunikacija je ključna za funkcionisanje poslovnih sistema i aplikacija, omogućena putem internet protokola kao što su TCP/IP. HTTP i HTTPS su osnovni za pregledanje i transfere web sadržaja. Web aplikacije postaju suštinski alati za pružanje usluga globalno, omogućavajući jednostavnu interakciju između korisnika i kompleksnih sistema [1].

Sa porastom upotrebe web aplikacija dolaze i izazovi vezani za bezbijednost i zaštitu osjetljivih podataka. Mrežna bezbijednost, uključujući firewall rješenja, postaje kritična za zaštitu podataka. Firewall djeluje kao prva linija odbrane, kontrolišući mrežni saobraćaj putem bezbijednosnih pravila [2]. Tehnologije firewall-a su se razvile i postale su ključne u cloud okruženjima gdje je potrebna fleksibilna i skalabilna zaštita. Ispravna konfiguracija firewall-a omogućava zaštitu i optimizaciju mrežnog saobraćaja.

Rad se fokusira na analizu konfiguracije i poređenje firewall mogućnosti kod cloud provajdera. Cloud tehnologija je transformisala globalno poslovanje pružajući skalabilnost, fleksibilnost i dostupnost. Sa sve više aplikacija i podataka u cloudu, bezbijednost zauzima centralno mjesto, zahtijevajući inovativne pristupe za zaštitu podataka.

Istraživanje uključuje razvoj minimalne web aplikacije za ilustraciju praktične primjene rješenja. Uključuje analizu sajber bezbijednosti na AWS Cloud-u, razmatra prijetnje poput neovlašćenog upada i DDoS napada, te predlaže firewalle kao rješenje.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Vuković, docent.

Cilj je pružiti sveobuhvatni uvid u upravljanje bezbijednosnim izazovima cloud provajdera, identifikujući prednosti i izazove u primjeni firewall-a, kako bi razvojni timovi mogli unaprijediti bezbijednost svojih cloud instalacija putem efikasnih strateških smjernica.

2. Izazovi sajber bezbijednosti AWS Cloud i u drugi orkuženjima

U savremenom digitalnom okruženju, bezbijednost web aplikacija i infrastrukture je od kritičnog značaja, posebno kod cloud usluga kao što je Amazon Web Services (AWS). Cloud platforme omogućavaju organizacijama postavljanje aplikacija na skalabilnu infrastrukturu, ali donose izazove u osiguravanju zaštite od sajber prijetnji. Web aplikacije u cloudu često su dostupne preko interneta, čineći ih metama za bezbijednosne prijetnje poput DDoS napada, neovlašćenog pristupa, krađe podataka i ubrizgavanja zlonamjernog koda.

Bezbijednosni problemi mogu nastati zbog:

1. Neovlašćenih upada, gdje nedozvoljeni korisnici pokušavaju pristupiti osjetljivim informacijama.
2. DDoS napada, koji preopterećuju sistem i čine ga nedostupnim legitimnim korisnicima.
3. Presretanja i manipulacije podataka, ugrožavajući integritet sistema.
4. Neadekvatnih konfiguracija bezbijednosnih politika.

Rješenje ovih izazova je implementacija robusnih firewall rješenja. AWS kao jedan od vodećih pružalaca cloud usluga [3], uključuje napredne mjere sigurnosti kao što je AWS WAF (Web Application Firewall), koji omogućava definisanje pravila za blokiranje ili dozvoljavanje saobraćaja.

Firewall rješenja pomažu u kontroli i upravljanju saobraćajem prema dinamičkim pravilima, zaštiti od ranjivosti i održavanju integriteta, povjerljivosti i dostupnosti podataka.

3. UPOTREBLJENE TEHNOLOGIJE I ALATI

3.1. Frontend Tehnologija: React

React je popularna biblioteka za izgradnju brzih i interaktivnih korisničkih interfejsa. Kada se koristi sa TypeScript-om, omogućava tipizirani JavaScript, pružajući prednosti kao što je garantovanje tipova, što smanjuje broj grešaka. TSX sintaksa poboljšava rad u timovima, a podrška za TypeScript u IDE-u povećava produktivnost. Virtualni DOM u React-u omogućava brze i efikasne promjene u korisničkom interfejsu.

3.2. Backend Tehnologija: Node.js i Express

Node.js je platforma koja omogućava izvršavanje JavaScript-a na serveru, sa asinhronim I/O što je čini efikasnom za rukovanje brojnim zahtjevima. Express je lagan okvir za upravljanje HTTP zahtjevima, savršen za red aplikacija u realnom vremenu.

3.3. Baza podataka: PostgreSQL i Sequelize

PostgreSQL je odabran zbog svojih naprednih mogućnosti. Sequelize, kao ORM alat, olakšava rad s PostgreSQL kroz objektnu-orijentaciju u JavaScript-u.

3.4. Integracija sa udaljenim servisima

Axios se koristi za slanje HTTP zahtjeva, omogućavajući laku komunikaciju sa eksternim API-jevima poput Trello-a.

3.5. Slanje e-pošte: Nodemailer

Nodemailer, kao jedna od popularnijih biblioteka za slanje e-pošte [4], je korišten za integraciju funkcionalnosti slanja e-pošte, omogućavajući jednostavno slanje različitih formata poruka i dodavanje priloga.

3.6. AWS Servisi

Za hosting aplikacije izabrani su AWS EC2 i AWS Amplify servisi. EC2 pruža skalabilnost u oblaku, dok Amplify pojednostavljuje razvoj i deployment frontenda, omogućavajući brzu integraciju AWS usluga i sigurno hostovanje [5].

4. IMPLEMENTACIJA RADA SA PODACIMA IZ Trello Servisa

4.1. Opis arhitekture aplikacije

Arhitektura sistema zamišljena je kao modularni mikroservisni sistem koji uključuje napredni frontend interfejs i fleksibilan backend, omogućavajući integraciju sa eksternim platformama kao što je Trello, uz podršku infrastrukture hostovane na AWS-u.

Frontend (React + TypeScript):

React se koristi za kreiranje dinamičnog i intuitivnog korisničkog interfejsa sa jakom tipskom sigurnošću, omogućenom kroz TypeScript. Korisnici mogu upravljati zadacima pomoću interaktivnih komponenti kao što su boards, lists i cards.

Komponente poput AuthService, BoardService i OrganizationService olakšavaju komunikaciju s backendom putem REST API-ja.

Backend (Node.js + Express):

Node.js i Express pružaju strukturu za obradu HTTP(S) zahtjeva koristeći kontrolere (poput AuthController, BoardController) za autentifikaciju i upravljanje podacima. JWT tokeni i AuthMiddleware brinu se za sigurnu autentifikaciju korisnika.

ErrorHandlerMiddleware omogućava centralizovanu obradu grešaka.

Baza podataka (PostgreSQL + Sequelize):

PostgreSQL čuva korisničke podatke i zadatke, dok Sequelize ORM omogućava manipulaciju podacima kroz JavaScript API pozive.

Integracija sa eksternim servisima:

TrelloIntegrationService omogućava sinhronizaciju i upravljanje Trello boardovima u realnom vremenu. Mail Sender servis koristi Nodemailer za automatizaciju slanja e-pošte korisnicima.

AWS Hosting i Skaliranje:

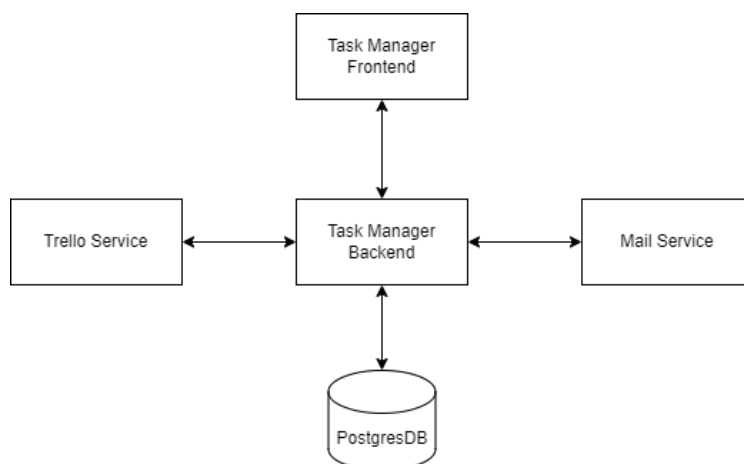
Infrastruktura i usluge:

1. EC2 Instance: Backend aplikacija i baza podataka PostgreSQL hostovani su na EC2 instanci, omogućavajući kontrolu i konfiguraciju servera.
2. AWS Amplify: Frontend aplikacija je hostovana na AWS Amplify-u, što omogućava lako ažuriranje i skalabilnost.

Bezbjednost i upravljanje pristupima:

1. AWS IAM omogućava precizno upravljanje pristupom i definisanje dozvola.
2. AWS Security Groups i Network ACLs djeluju kao virtualni firewall-ovi za kontrolu mrežnog saobraćaja i implementaciju sigurnosnih politika.

Ova struktura obezbjeđuje bezbjedan i skalabilan rad aplikacije, optimizovano upravljanje resursima i prilagođavanje zahtjevima korisnika.



Slika 1. Arhitektura aplikacije

4.2. Opis rada aplikacije

Frontend aplikacija nudi niz mogućnosti.

Sign in

Email *

nebojsagordic@gmail.com

Password *

.....

☐ Remember me

1

SIGN IN

[Forgot password?](#)

2

[Don't have an account? Sign Up](#)

Slika 2. Forma prijave korisnika

Na slici 2 prikazana je početna strana aplikacije sa dugmetom za prijavu u elipsi 1 ili registraciju u elipsi 2.

Registration

First Name *

Nebojsa

Last Name *

Gordic

Email *

nebojsagordic@gmail.com

User Name *

nebojsa

Password *

Field is required

Confirm Password *

Field is required

1

REGISTER

2

[Already have an account? Sign In](#)

Slika 3. Forma za registraciju korisnika

Na slici 3 se može uočiti forma za registraciju novih korisnika koja se potvrđuje klikom na dugme u elipsi 1 ili prelazak na prijavu klikom na dugme u elipsi 2.

Task Manager

Workspaces

Boards

Name	Actions
test	Edit Delete
test2	Edit Delete

Create New Task Board

Board name

Organization

1

REGISTER

Slika 4. Radna površina

Nakon prijave korisniku se prikazuje radna površina gdje su izlistani board-ovi koji u sebi sadrže card-ove. Pored toga moguće je pozvati akciju brisanja ili izmjene

izlistanih entiteta. Pored toga može se uočiti forma uz pomoć koje je moguće kreirati novi board.

Edit Board

Board name

test

New List name

ADD +

To Do

Doing

Done

Card Name

ADD CARD +

Card1

Card3

Card2

Card Name

ADD CARD +

1

2

CANCEL

SAVE

Slika 5. Dialog za pregled izmjenu board-a

Na slici 5 je prikaz otvorenog board-a. U dialogu je moguće preimenovati board, dodavati nove liste za card-ove, brisati ih ili im mijenjati imena. Takođe je omogućeno dodavanje novih card-ova, preimenovanje postojećih ili njihovo prebacivanje iz jedne u drugu listu.

Dugme u elipsi 1 omogućava odustajanje od napravljenih izmjena, dok dugme u elipsi 2 omogućava njihovo čuvanje. Sve napravljene izmjene se prenose na integrisani Trello servis.

5. AWS I BEZBIJEDNOST UPOTREBOM FIREWALL-A

5.1. AWS i njegove osnove

Amazon Web Services (AWS) je sveobuhvatna platforma za cloud computing koju pruža Amazon. AWS nudi širok spektar usluga za računarstvo, skladištenje, baze podataka, analitiku, razvoj aplikacija, mašinsko učenje i još mnogo toga. Omogućava kompanijama i razvojnim timovima brzu skalabilnost i integraciju resursa bez potrebe za održavanjem fizičke infrastrukture.

Ključne komponente AWS-a uključuju:

- EC2 (Elastic Compute Cloud),
- S3 (Simple Storage Service),
- RDS (Relational Database Service),
- Lambda,
- VPC (Virtual Private Cloud),
- DynamoDB,
- IAM (Identity and Access Management).

AWS omogućava organizacijama da povećaju IT infrastrukturu smanjujući troškove i poboljšavajući agilnost, pružajući pouzdanu i stabilnu platformu za razvoj i skaliranje aplikacija.

5.2. Virtual Private Cloud

Virtual Private Cloud (VPC) je AWS usluga koja omogućava kreiranje izolovanog dijela AWS oblaka za pokretanje AWS resursa, poput EC2 instanci. VPC pruža kontrolu nad virtualnim mrežnim okruženjem [6].

Ključne karakteristike VPC-a uključuju:

Izolovani mrežni prostor, konfigurisanje mreže, kontrola pristupa i integracija s lokalnom mrežom.

VPC je osnova za arhitekturu u AWS oblaku, omogućavajući prilagođavanje mrežnog okruženja na osnovu specifikacijskih i bezbjednosnih zahtjeva aplikacije.

5.3. Bezbjednosne grupe

Bezbjednosne grupe u AWS-u služe kao firewall za kontrolu saobraćaja do AWS resursa, kao što su EC2 instance. Ove grupe regulišu koji saobraćaj može pristupiti i koji može napustiti ove resurse. Kada se bezbjednosna grupa poveže sa EC2 instancom, ona upravlja ulaznim i izlaznim saobraćajem za tu instancu. VPC takođe posjeduje podrazumijevanu bezbjednosnu grupu [7].

Bezbjednosna grupa se može dodijeliti samo resursima unutar istog VPC-a, a jedan resurs može imati više grupa. Grupa je stateful, što znači da ako instanca inicira zahtev, odgovor može proći bez obzira na ulazna pravila.

Bezbjednosne grupe ne ograničavaju saobraćaj ka i od Amazon DNS-a, Amazon DHCP-a, EC2 i ECS metadata, ovjera Windows licenci i servisa za sinhronizaciju vremena. Postoje ograničenja u broju grupa unutar VPC-a, broju pravila po grupi i broju grupa na mrežnom interfejsu. Preporučuje se da samo određeni IAM korisnici mogu mijenjati grupe. Treba ograničiti pristup preko specifičnih TCP portova i izbjegavati korištenje velikog broja portova u grupi radi smanjenja rizika od napada.

5.4. Access Controll Lists

Mrežna kontrola pristupa (ACL) omogućava ili odbija određeni ulazni ili izlazni saobraćaj na nivou podsistema. U VPC-u se može koristiti podrazumijevana mrežna ACL ili se može kreirati prilagođena mrežna ACL koja sadrži pravila slična onima u bezbjednosnim grupama.

Jedna ACL se može povezati s više podsistema, ali jedan podsistem može imati samo jednu ACL. Sve ACL liste sastoje se od numerisanih pravila od 1 do 32766, koja se primjenjuju redom. Nova pravila je preporučljivo numerisati s velikim razmacima. Mrežne ACL nisu stateful, što znači da ne pamte prethodne zahtjeve – ako je ulazni saobraćaj dozvoljen, izlazni nije dozvoljen automatski. Mrežne ACL ne mogu blokirati DNS zahtjeve ka ili od Route 53 Resolver-a, niti saobraćaj ka IMDS. Takođe, ne filtriraju saobraćaj od: Amazon EC2 metadata, DHCP, Amazon ECS task metadata, aktivacija Windows licenci, Amazon servisa za sinhronizaciju vremena, i rezervisanih IP adresa podrazumijevanog VPC rutera [8].

5.5. AWS Network Firewall

AWS Network Firewall (ANF) korisnicima nudi deep packet inspection (DPI), mogućnost detekcije aplikativnih protokola, filtriranje domena i sistem za sprječavanje upada (IPS). ANF pruža stateful i stateless mehanizme za upravljanje pravilima saobraćaja unutar VPC-a, omogućavajući inspekciju saobraćaja u svim pravcima sa podrškom za hiljade pravila. Korisnici mogu usmjeriti

dolazni saobraćaj ka ANF krajnjoj tački, sa ANF smještenim u namjenskoj podmreži unutar VPC-a. ANF koristi AWS Gateway Load Balancer za raspoređivanje opterećenja i ulazno rutiranje VPC-a radi sveobuhvatne kontrole saobraćaja.

Podržani modeli primjene uključuju: Distribuirani model, Centralizovani model i Kombinovani model, a modeli se razlikuju od raspoređivanja u VPC [9].

6. ZAKLJUČAK

Tokom izrade ovog rada, istražena su i implementirana tehnološka rješenja za efikasno i sigurno upravljanje digitalnim resursima i aplikacijama. Razvijena aplikacija za upravljanje zadacima koristi AWS infrastrukturu za skalabilnost i pouzdanost, omogućavajući integraciju s eksternim servisima kao što je Trello.

Poseban akcenat stavljen je na bezbjednosne aspekte. Kao virtualni firewall-i, Security Groups omogućile su precizno definisanje ulaznih i izlaznih pravila, kritično za siguran i pouzdan prenos podataka. Ovaj bezbjednosni sloj kontroliše pristup EC2 instancama, minimizirajući rizike od neovlašćenog pristupa i ugrožavanja podataka.

Konfiguracija Security Groups bila je fokusirana na dozvoljavanje samo neophodnih protokola i portova, održavajući visoku bezbjednost bez ugrožavanja funkcionalnosti aplikacije. Tako je omogućen siguran prenos podataka između komponenti aplikacije i eksternih servisa. Dodatno, korištenje Security Groups omogućilo je primjenu promjena u bezbjednosnim pravilima u realnom vremenu bez prekida rada sistema. Rad demonstrira snagu AWS inovacija u razvoju aplikacija, ističući skalabilnost i integraciju kao čvrstu osnovu za razvoj sigurnih i robusnih aplikacija u oblaku. Očekuje se da će buduća istraživanja nastaviti unapređivanje bezbjednosnih i funkcionalnih aspekata razvoja u cloud computingu, doprinoseći boljoj zaštiti podataka i digitalnih resursa.

7. LITERATURA

- [1] Walter Goralski, *The illustrated network: how TCP/IP works in modern network*
- [2] Macfarlane, Richard; Buchanan, William, Ekonomou, Elias; Uthmani, Omari; Fan, Lu; Lo, Owen (2012). *Formal security policy implementations in network firewalls*
- [3] Gartner, *Gartner Magic Quadrant for Cloud Infrastructure and Platform Services*
- [4] NPM, *Nodemailer Download Statistics*
- [5] AWS Documentation, *AWS services by Category*
- [6] AWS Documentation, *Amazon VPC*
- [7] AWS Documentation, *Security Groups*
- [8] AWS Documentation, *Network ACL*
- [9] AWS Documentation, *AWS Network Firewall*

Kratka biografija:



Nebojša Gordić rođen je u Sokocu, BiH 1999. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Izazovi u vizuelizaciji pozicija i stanja malih elektrana odbranio je 2022.god. kontakt: nebojsagordic@gmail.com

JEZIK ZA SPECIFIKACIJU PRODAJNIH AKCIJA**A LANGUAGE FOR THE SPECIFICATION OF SALES PROMOTIONS**Nataša Rajtarov, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO**

Kratak sadržaj – U ovom radu će biti predstavljen jezik za dinamičko kreiranje prodajnih akcija, tokom rada aplikacije, ali i kreiranje same veb aplikacije. Fokus rada je na prikazu mogućnosti i prednosti ovog jezika u odnosu na postojeća rešenja poput Vtex-a, Drools-a i JHipster-a. Cilj je da se pokaže kako kreirani jezik omogućava veću fleksibilnost i smanjuje ograničenja za prodavce prilikom kreiranja akcija, za razliku od postojećih rešenja koja se oslanjaju na šablone.

Ključne reči: *textX, python, jezik specifičan za domen, prodajne akcije*

Abstract – This paper presents a language for the dynamic creation of sales promotions during application runtime, as well as for the creation of the web application itself. The focus of the paper is on showcasing the capabilities and advantages of this language compared to existing solutions like Vtex, Drools, and JHipster. The goal is to demonstrate how the developed language provides greater flexibility and reduces limitations for sellers when creating promotions, unlike existing solutions that rely on templates.

Keywords: *textX, python, domain specific language, prodajne akcije*

1. UVOD

Razvojem tehnologije, mnoge aktivnosti svakodnevnog života, poput trgovine, migrirale su na veb platforme. Prelazak na veb, omogućio je trgovcima da širi broj ljudi može da dobije uvid u njihovu delatnost. Pored mnogih prednosti, veb platforme su donele i ograničenja, kao što je ograničenje u raznolikosti prodajnih akcija. Potreba za rešenjem ovog problema se javlja kada prodavci žele da kreiraju jedinstvene akcije koje nisu unapred definisane ili čije ponašanje nije opisano u potpunosti kodom unutar aplikacije.

Zadatak ovog rada je implementacija jezika koji omogućava dinamičko kreiranje prodajnih akcija tokom rada aplikacije. Jezik se sastoji iz dva ključna dela: prvi deo koji omogućava kreiranje veb aplikacije za prodaju, dok drugi deo omogućava kreiranje akcija u realnom vremenu. Pored toga, rad analizira postojeća rešenja kao što su Drools, VTex koji omogućavaju kreiranje prodajnih akcija uz pomoć šablona.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Igor Dejanović.

Osnovni motiv za razvijanje jezika koji omogućava kreiranje prodajnih akcija u toku rada aplikacije jeste problem potrebe za šablonima koji ograničavaju prodavca prilikom kreiranja prodajnih akcija. Razlog podele jezika na dva dela, jeste omogućavanje korisnicima da efikasno kreiraju aplikaciju koja će im omogućiti trgovinu, kao i obezbediti platformu na kojoj mogu uz jezik opisan u ovom radu da kreiraju akcije.

2. JEZICI SPECIFIČNI ZA DOMEN

Kako bi računar izvršavao zahteve od strane čoveka, razvijen je jezik koji omogućava međusobnu komunikaciju. Postoje jezici opšte namene (JON) i jezici specifični za domen (JSD).

Jezici opšte namene, kao što su Python, Java, i JavaScript, omogućavaju kreiranje struktura i opisivanje ponašanja sistema u različitim aplikacijama i industrijama. Programski jezici opšte namene dizajnirani su da budu dovoljno fleksibilni i ekspresivni da mogu rešavati probleme nezavisno od domena kojem pripadaju. Međutim, njihova složenost može predstavljati izazov za korisnike koji nisu programeri, jer su dizajnirani da rešavaju širok spektar problema, što može otežati njihovu primenu u specifičnim domenima.

Za razliku od jezika opšte namene, jezici specifični za domen su usko orijentisani na jedan domen, što znači da su konstruisani tako da ispunjavaju zahteve samo određenog spektra zadataka, ali to rade uz povećanu produktivnost. Takav jezik bazira se na postojećem jeziku eksperata što olakšava ekspertima samo savladavanje tog jezika. Jezici specifični za domen su ograničene ekspresivnosti i usmereni su na određenu oblast primene, kao što objašnjava Martin Fowler [1].

Istraživanja kao što su [2, 3] pokazuju da korišćenje jezika specifičnih za domen povećava produktivnost kod njenih korisnika. „Sa odgovarajućim JSD-om, možete razviti kompletne aplikativne programe za domen brže i efikasnije nego sa jezikom opšte namene.“ [2] Ovakvi jezici omogućavaju brži razvoj aplikacija jer smanjuju količinu koda potrebnu za izražavanje određenih akcija u poređenju sa jezicima opšte namene. Pored toga, eksperti više nisu u obavezi da uče jezik koji nije u skladu sa njihovim domenom, čime se unapređuje efikasnost i tačnost. Uži domen i jezik prilagođen njemu, smanjuje nivo slobode koju jezik može da ostvari, ali samim tim i smanjuje verovatnoću grešaka, kao i ponavljanje koda. JSD-ovi su lakši za razumevanje nego JON jezici i nude mogućnost analize, verifikacije, optimizacije, paralelizacije i transformacije [4].

2. OSNOVNI KONCEPTI JEZIKA

Proces kreiranja novog jezika specifičnog za domen obuhvata 4 koraka:

- Definisanje domena – potrebno je tačno odrediti domen koji će obuhvatati jezik.
- Dizajn jezika i njegove semantike – U okviru ove tačke potrebo je razmotriti koje tačno funkcionalnosti treba da obuhvati dati jezik, šta će biti izlaz/produkt njegovog rada, ko će ga koristiti i na koji način da mu se prilagodi takav jezik. Modeluju se ograničenja koja će morati da podrži.
- Kreiranje alata koji će obrađivati jezik – Potrebno je kreirati alat koji će omogućiti prevod i korišćenje jezika.
- Korišćenje jezika.

Da bismo omogućili efikasno modelovanje, kako jezika specifičnog za domen tako i jezika opšte namene, potrebno je da koristimo 2 nivoa sintakse, a to su apstraktna i konkretna sintaksa. Apstraktna sintaksa se odnosi na strukturu jezika, nezavisno od specifičnih detalja konkretne sintakse. Ona opisuje šta se izražava u jeziku, a ne kako se izražava. Jedna apstraktna sintaksa može imati više svojih konkretnih sintaksi. Apstraktno sintakšno stablo je glavna struktura koja omogućava predstavu apstraktnih sintaksi. konkretna sintaksa omogućava ljudima da pišu i čitaju dati jezik. Ona obuhvata konkretne operatore, ključne reči, simbole, ili može imati i grafičku sintaksu. Ona određuje pravila kako se apstraktni pojmovi izražavaju kroz nju. Jezik se opisuje gramatikom jezika. Gramatika jezika predstavlja formalnu definiciju konkretne sintakse.

3. JEZIK ZA KREIRANJE E-KOMERC SAJTOVA I DEFINISANJE PRODAJNIH AKCIJA

Budući da kompanije mogu da imaju različite domene trgovine, neće svima odgovarati isti tip akcije. Da bi se omogućilo fleksibilno kreiranje akcija, stvoren je jezik koji je opisan u ovom radu. Sam jezik se sastoji iz dva dela:

- Deo za kreiranje prodajnih akcija
- Deo za kreiranje web sajtova koji podržavaju kreiranje akcija pomoću datog jezika

Za implementaciju jezika korišten je textX metajezik [5] koji omogućava istovremeno definisanje i konkretne i apstraktnih sintakse upotrebom jedne gramatike. Izlaz iz celog procesa je generisan kod koji je spreman da obavlja svoju namenu. Faze kroz koje prolazi program/model su:

1. Leksička analiza
2. Sintakсна analiza
3. Semantička analiza
4. Generisanje koda

Leksička analiza podrazumeva obradu ulaznog koda, tako da se razbije na lekseme, manje jedinice koda. Ona identifikuje ključne reči, identifikatore, literale i druge simbole. Izlaz su lekseme koje se u sintakсноj analizi obrađuju. Proverava se sintaksa jezika i lekseme se organizuju u sintakšno stablo. U okviru semantičke analize obrađuje se i proverava sintakšno stablo da li je

semantički ispravno. U ovoj fazi se proveravaju tipovi podataka, provere opsega i ostale provere vezane za sintaksu. Poslednja faza omogućava generisanje koda.

Prevođenje koda i generisanje u ovom projektu, opisano je na programskom jeziku Python. Međutim, za generisanje koda veb aplikacije, korištena je Jinja, budući da omogućava kreiranje koda kroz šablone. Generisana aplikacija se sastoji iz dva sloja: serverski i klijentski sloj. Za serverski sloj je korišten Springboot sa javom. Za komunikaciju sa bazom je korišten Nibernet. Klijentski sloj aplikacije je pravljen u React-u. Komunikacija između serverskog i klijentskog sloja je obavljana pomoću REST-a.

3.1. TextX

textX je alat razvijen u Python-u sa glavnim ciljem brzog razvoja jezika specifičnih za domen [5]. Zasnovan je na Arpeggio PEG parseru, što mu omogućava neograničeno gledanje unapred sa linearnim vremenima parsiranja i rad u stilu interpretera. textX nasleđuje prirodu Arpeggio parsera, ali i gramatiku višeg nivoa [5]. Nastao je kao implementacija Xtext meta-jezika u Python-u, ali se vremenom razvio i danas sadrži neke karakteristike drugačije od Xtext jezika. Osnovni radni proces i arhitektura textX alata uključuju sledeće korake:

1. Parsiranje gramatike jezika i dinamičko stvaranje meta-modela i Arpeggio parsera.
2. Parsiranje programa/modela u datom jeziku i kreiranje grafa Python objekata, gde je svaki objekat instanca određene klase iz meta-modela jezika.
3. Model može kasnije biti korišten za interpretaciju i/ili generisanje izvornog koda.

textX gramatika se sastoji od skupa pravila koja opisuju koncepte iz ciljnog jezika (tj. meta-klase) i sintakčne strukture konceptata predstavljenih korišćenjem tekstualne konkretne sintakse. Sva pravila rezultiraju kreiranjem python klase u vreme izvršavanja. Kreira se graf python klasa, koje će biti korištene za instanciranje datog koncepta tokom parsiranja modela. Model se kasnije može koristiti za interpretiranje ili generisanje koda.

3.2. Gramatika za modelovanje veb aplikacija

Gramatika definiše pravila koja su dovoljna da se opiše cela aplikacija. Kako bi se kreirala veb aplikacija sa osnovnim CRUD operacijama(create-kreiraj, read-čitaj, update-obnovi, delete-obriši), dovoljno je uneti podatke o domenskim klasama na osnovu kojih bi se definisali ostali slojevi aplikacije, repozitorijum, servis, kontroler, kao i klijentska strana. Zbog toga ovaj deo gramatike omogućava definisanje klasa, veza između njih, prostih atributa i enumeracija. Kako bi omogućili kreiranje veb prodavnice, omogućeno je opisivanje određenih klasa anotacijama koje omogućavaju proširenje tih klasa kroz dodate attribute i metode. Anotacije su:

- Bill - Anotacija će označiti klasu, čiji objekat se kreira prilikom potvrde kupovine. Izvršiće se obračun kupljenih stvari, pronaći će se odgovarajuća akcija i primeniti na cenu. Anotacija dobija nazive funkcija koje će se pozivati prilikom primene akcija.

- Bying - Anotacija će označiti klasu, čiji objekti će se naći u računu i čija cena će se obračunati prilikom kupovine. Anotacija može dobiti dve vrednosti: one ili more što omogućava kupovinu samo jednog ili više artikala.
- Item - Anotacija će označiti klasu produkta. Svaki produkt imaće listu objekata koji sadrže cenu, tj. onih objekata koji su pod anotacijom Bying.
- Basket – Anotacija će označiti klasu koja čuva objekte koji će se naći u korpi u toku kupovine. Kada se kreira račun, korpa će se isprazniti.
- Action – Anotacija će omogućiti oznaku klase koja će biti akcija. Dobiće podatke od kad do kad

Proizvodna akcija se opisuje kao niz instrukcija koje dovode do promene cene. Da bi se obavila akcija, moguće je posmatrati listu proizvoda koji su trenutno na akciji, proizvode koje potrošač trenutno kupuje i samog potrošača. Proizvodi koji su trenutno na akciji mogu se naći u klasi koju je kreator aplikacije označio sa anotacijom Action. Jedna akcija predstavlja funkciju od nula ili više iskaza. Kroz ovaj jezik moguće je kreirati promenljive, kreirati jednolinijske uslove, if-else uslove, nove objekte, ali ključni deo jezika su „kratke petlje“ koje omogućavaju iteriranje kroz liste elemenata uz različite uslove. Na primer, „kratka petlja“ max će naći najveći element u listi bez da se eksplicitno piše standardna for petlja sa uslovima kao u JON jezicima. Gramatika dela

```

76 ChaneId: ID ('.' ID)*;
77
78 Sum:      'sumOf'      PossibleLists      (('as:' (ChaneId) (UsingFormula)? WherePart)?);
79 Average:  'averageOf' PossibleLists      ('as:' ( ChaneId ) (UsingFormula)? WherePart)?;
80 Product:  'productOf' PossibleLists      ('as:' ( ChaneId ) (UsingFormula)? WherePart)?;
81
82 SelectTop: 'selectTop' INT 'of' PossibleLists 'as:' ID      (WherePart)?;
83 SelectIn:  'selectIn'  PossibleLists      'as:' ID      WherePart;
84 Count:     'countFrom' PossibleLists      ('as:' ID)?      (WherePart)?;
85 All:       'allOf'     PossibleLists      ('as:' ID)?      WherePart ;
86 Any:       'anyOf'     PossibleLists      ('as:' ID)?      WherePart;
87 NoOne:     'noneOf'    PossibleLists      ('as:' ID)?      WherePart;
88 Min:       'min'       PossibleLists      ('as:' (ChaneId))? (WherePart)?;
89 Max:       'max'       PossibleLists      ('as:' (ChaneId))? (WherePart)?;
90 Size:      'sizeOf'    PossibleLists ;
91 Add:       'add'       (ChaneId)          'in:' PossibleLists;
92 ForEach:   'for'      ( ChaneId)          'as:' ( ChaneId) (ForOperation)? ShortFunctions;
93
94 ForOperation: ('sumFrom'|'averageFrom'|'productFrom' );
95 PossibleLists: ( ChaneId | SelectIn);
96 WherePart: 'where' (Conditions | All | Any | NoOne);
97 UsingFormula: 'formula' Operations;
98 ShortFunctions: ( Size| Product | Count | All | Any | NoOne | Min | Max | Add |SelectTop| Sum |Average | SelectIn | ForEach);
99 value: (INT | STRING | FLOAT | BOOL );

```

Slika 1 "kratke petlje" gramatike opisane u textX-u

važi akcija i kod koji je menadžer/prodavac uneo kao i atribut prevedeni kod.

jezika za kreiranje akcija, napisana u textX nalazi se na slici Slika 1.

4. IZLAZNA APLIKACIJA

Izlazna veb-shop aplikacija sadrži tri uloge: administratora, prodavca i kupca. Administrator može da vrši CRUD operacije nad svim entitetima, prodavac može da rukuje sa objektima vezanim za proizvode, da kreira cene i akcije, dok kupac može da naručuje proizvode, pregleda korpu, svoje podatke i sl.

Proces kreiranja akcije se sastoji od unosa podataka o akciji, kao što su datumi od kad do kad traje i slično, ali i koda akcije. Kod akcije napisan u jeziku za kreiranje akcija, šalje se na python serveru koji vrši prevod akcije. Prevedeni kod se vraća na serverski sloj aplikacije. Gde se čuvaju podaci o akciji, kao i sam prevedeni kod. Korisnik se obaveštava o uspešnosti kreiranja akcije.

Sam proces naručivanja obuhvata odabir proizvoda i potvrdu kupovine od strane kupca. Podaci se šalju na serversku stranu gde se vrši obračun cena proizvoda i obračunavaju se akcije. Prilikom obračuna akcije, kreira se fajl koji čuva klasu sa vraćenim kodom funkcije. Klasa se čita i pomoću refleksije se izvršava kod koji je preveden. Refleksija omogućava da program pregleda, poziva

metode ili modifikuje strukturu i ponašanje objekata u toku izvršavanja programa. Budući da se šalju reference na objekte prevedenij funkciji, objekat račun se vraća sa novom cenom, nastalom primenom akcije. Podaci se čuvaju u bazi i šalje se obaveštenje korisniku.

5. PREGLED STANJA U OBLASTI

5.1. Drools

Drools se opisuje kao sistem za upravljanjem poslovnim pravilima, koji omogućava ulančavanje unapred i unazad (forward-chaining i backward-chaining) [22]. Koristi DRL (Drools Rule Language) za definisanje pravila koja omogućavaju rezonovanje na osnovu činjenica. DRL jezik može opisati pravila za bilo koji domen primene, ne ograničavajući se samo na trgovinu. Pored kreiranja pravila, Drools omogućava i kreiranje šablona, koji služe kao šabloni za unos pravila kroz aplikaciju, bez potrebe za učenjem samog Drools jezika. Međutim, ovaj pristup može biti kompleksan ako je potrebno definisati više šablona za različite tipove akcija. Drools je najbolje rešenje kada korisnik već zna DRL jezik ili kada postoji

ograničen broj akcija koje se mogu definisati šablonima. Problemi nastaju kada korisnik mora da uči DRL jezik ili želi da definiše širok spektar akcija.

5.2. Vtex

Vtex je onlajn platforma koja omogućava upravljanje online trgovinom. Koristi cloud tehnologiju, što omogućava skalabilnost, tj. da prodavnice mogu da porastu i još uvek sačuvati svoje performanse. Omogućava povezivanje sa drugim platformama kao što su Amazon i eBay. Pored svega toga, Vtex omogućava kreiranje promocija i akcija. Postoji više vrsta akcija koje podržava, a neke od njih su: popusti na proizvode, „Kupite jedan, dobijete jedan besplatno.“, količinski popust, promo kodovi itd. Prilikom kreiranja, kupac bira jedan od tipova akcije gde mu se nudi šablon sa podacima u vezi akcije, koji popunjava i čuva akciju. Ono što razlikuje kreiranje akcija u ovoj aplikaciji od jezika koji je opisan u ovom radu, je to da postoji set podataka koji prodavci mogu da iskoriste za svoje akcije. Taj skup podataka koji se unosi jeste širok i omogućava kreiranje različitih prodajnih akcija, ali nema slobodu koju nudi jezik. Pomoću jezika je moguće opisati širi skup akcija, jer je moguće pristupiti većem broju podataka o samom korisniku, njegovoj korpi sa proizvodima.

5.3. Jhipster

JHipster je generator koda i podržava kreiranje veb aplikacija i mikroservisa gde za serversku stranu generiše kod u Spring okviru, a klijentsku u Angular-u i React-u. Podržava i MVC šablon na serverskoj strani tako da je izgenerisana aplikacija je podeljena na model, kontroler, dok je za izgled zadužena klijentska strana. Jhipster koristi JDL (JHipster Domain Language) za modelovanje entiteta i odnosa u aplikaciji. JDL jezikom je moguće opisati entitete koje korisnik želi da ima u svojoj aplikaciji i da definiše attribute i veze između entiteta. Dok JDL omogućava osnovno modelovanje entiteta, jezik opisan u ovom radu nudi anotacije koje pružaju detaljnije opise entiteta, čime se omogućava bolji opis aplikacija čiji je domen trgovina. JHipster i njegov JDL pružaju dobar okvir za brzo kreiranje veb aplikacija, ali novi jezik i njegove anotacije predstavljaju korak napred u prilagođavanju alata specifičnim potrebama trgovinske industrije.

6. ZAKLJUČAK

Rad obuhvata razvoj jezika za modelovanje aplikacija za online trgovinu, koji omogućava kreiranje prodajnih akcija. Ovaj jezik je dizajniran kao jezik specifičan za domen kreiranja e-komerc sajtova. Rad opisuje proces kreiranja jezika, uključujući definisanje domena, modelovanje jezika, razvoj alata za podršku jezika, te generisanje aplikacija i akcija korišćenjem jezika. Korišćeni su različiti alati i tehnologije kao što su Python, textX, Java, Spring Boot i React, za realizaciju sintakse, semantike, i generisanje koda. Jezik omogućava modelovanje klasa i relacija, dodavanje specifičnih anotacija za proširenje funkcionalnosti, i generisanje web aplikacija koje sadrže serverski i klijentski sloj. Jezik omogućava kreiranje različitih prodajnih akcija, sa većim stepenom fleksibilnosti u odnosu na postojeća rešenja poput Drools, Vtex, i Shopify. Dalji razvoj jezika usmeren je ka praćenju novih trendova u online trgovini i unapređenju tehnologija korišćenih u razvoju.

7. LITERATURA

- [1] Fowler, Martin. „*Domain-specific languages*“ Pearson Education, 2010.
- [2] Hudak, Paul. „*Domain-specific languages.*“ Handbook of programming languages 3.39-60 (1997): 21.
- [3] Sujeeth, Arvind Krishna. „*Productivity and Performance with Embedded Domain-Specific Languages.*“ Stanford University, 2014.
- [4] M. Mernik, J. H. „*When and how to develop domain-specific.*“ (2005).
- [5] Dejanović, I. Vadera, R., Milosavljević, G., Vuković, Ž. „*Textx: a python tool for domain-specific languages implementation. Knowledge-based systems.*“ (2017).

Kratka biografija:



Nataša Rajtarov rođena je u Zrenjaninu 2000. god. Radi kao saradnik u nastavi na Fakultetu tehničkih nauka. Master rad na Fakultetu tehničkih nauka iz oblasti Softverskog inženjerstva o informacionih tehnologija odbranila je 2024.god.
kontakt:
rajtarovnatasa@gmail.com

FANTAZIJSKE IGRE BAZIRANE NA VELIKIM JEZIČKIM MODELIMA**FANTASY GAMES BASED ON LARGE LANGUAGE MODELS**

Aleksa Simić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO

Kratak sadržaj – Ovaj rad istražuje evoluciju interaktivne fikcije i njenu primenu u savremenim tehnologijama, sa fokusom na upotrebu velikih jezičkih modela za kreiranje igara i drugih sadržaja. Rad obuhvata analizu istorijskog razvoja, postojećih alata kao što su TADS i Inform, i integraciju naprednih modela putem OpenAI API-ja. Razvijeni meta-model i DSL omogućavaju generisanje igara. Rezultati pokazuju potencijal ovih tehnologija da revolucionizuju interaktivnu fikciju pružajući personalizovana i imerzivna iskustva.

Ključne reči: Interaktivna fikcija, GPT, DALL-E, OpenAI API, generisanje sadržaja, meta-model, DSL, imerzivna iskustva

Abstract – This thesis explores the evolution of interactive fiction and its application in contemporary technologies, focusing on the use of large language models for creating games and other content. The work includes an analysis of the historical development, existing tools such as TADS and Inform, and the integration of advanced models through the OpenAI API. The developed meta-model and DSL enable the generation of games. The results demonstrate the potential of these technologies to revolutionize interactive fiction by providing personalized and immersive experiences.

Keywords: Interactive fiction, GPT, DALL-E, OpenAI API, content generation, meta-model, DSL, immersive experiences

1. UVOD

Interaktivna fikcija predstavlja softver koji simulira okruženje u kojem igrači upravljaju likovima putem tekstualnih komandi. Ovaj rad istražuje istorijski razvoj interaktivne fikcije, kao i savremene pristupe koji uključuju velike jezičke modele za kreiranje igara i drugih sadržaja. Cilj rada je kreiranje alata koji koristi meta-model interaktivne fikcije i integraciju sa OpenAI API-jem.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Igor Dejanović, red. prof.

2. KRATAK PREGLED ISTORIJE INTERAKTIVNE FIKCIJE

Istorija interaktivne fikcije započinje u 1960-im godinama sa pojavom prvih kompjuterskih igara kao što je "Adventure". Tokom 1980-ih i 1990-ih, ovaj žanr je evoluirao sa pojavom grafičkih elemenata i komercijalnih uspeha igara poput "Wonderland" i "The Hobbit". Nakon 1990-ih, interaktivna fikcija se transformisala u umetnički oblik, doživljavajući nove inovacije kroz zajednicu entuzijasta [1].

3. POSTOJEĆA REŠENJA

Najpoznatiji alati za kreiranje interaktivne fikcije uključuju TADS, koji omogućava razvoj tekstualnih avantura i Inform, koji pruža alate za kreiranje interaktivnih priča. TADS 3 omogućava kreiranje kompleksnih interakcija sa NPC likovima i okruženjem, dok Inform 7 nudi intuitivno okruženje za razvoj igara zasnovano na prirodnom jeziku [2, 3].

4. KORIŠĆENI ALATI

Za razvoj aplikacije korišćeni su različiti alati, uključujući textX za definisanje meta-modela interaktivne fikcije, Tkinter za grafički korisnički interfejs, i OpenAI API za integraciju sa modelima GPT i DALL-E.

4.1. TextX

Meta-jezik korišćen za definisanje osnovnog jezika ovog master rada je textX. Inspirisan je Xtext-om. TextX se koristi za kreiranje jezika specifičnih za domen u Python programskom jeziku. Ovaj meta-jezik olakšava izgradnju novog domen-specifičnog jezika, dodatne podrške ili proširenje postojećih jezika. Meta-model se automatski gradi u textX-u kao skup Python klase, a uz njega se kreira i parser za novokreirani jezik koji parsira izraze i automatski izgrađuje model kao graf Python objekata koji odgovara meta-modelu [4].

4.2. Tkinter

Python ima mnogo GUI okvira (engl. frameworks), ali Tkinter je jedini okvir koji je direktno uključen u standardnu Python biblioteku. Naziv Tkinter je skraćenica od Tk Interface-a, gde Tk predstavlja višepatformski alat namenjen za kreiranje GUI-a. Neki od osnovnih elemenata Tkinter GUI-a su: prozor (engl. Window), prozor najvišeg nivoa (engl. Top-level window), widget, frame [5].

4.3. OpenAI API

OpenAI API je interfejs za programsko povezivanje koji omogućava pristup moćnim modelima mašinskog učenja

1693

7. GENERISANJE SADRŽAJA

U ovom radu, korišćenje GPT i DALL•E modela pojavljuje se u dva segmenta: za kreiranje gotovih igara interaktivne fikcije koje imaju kraj i za mod beskonačnog igranja, gde se generišu novi objekti, regije, predmeti, neprijatelji, oružja i oklopi.

7.1. Generisanje konačnih igara interaktivne fikcije

Generisanje igara sa konačnim trajanjem koristi se kada želimo da igra ima jasan početak, sredinu i kraj. Ovo je obično slučaj kod priča sa fiksiranim zapletom, gde igrač na kraju dospeva do određenog cilja ili misije.

Ovaj proces se deli na nekoliko koraka:

- Učitavanje gramatike i primera igre: Prvo se učitava gramatika iz textX DSL datoteke i primer igre, što služi kao vodič GPT modelu za kreiranje nove igre.
- Generisanje igre: Koristeći OpenAI GPT model, na osnovu unapred definisanih pravila i primera, GPT generiše kod za novu igru koja ima kraj. Posebna pažnja posvećena je dodatku završne regije kao kraja igre.
- Rezultat: Funkcija vraća generisani kod igre, koji može biti korišćen za igranje.

Proces generisanja slika za konačne igre se takođe može podeliti na nekoliko koraka:

- Učitavanje igre: Prvo se učitava igra, odnosno sve njene regije i opisi tih regija.
- Generisanje slike: Koristeći OpenAI DALL•E model, na osnovu prompta koji je zapravo originalni opis regije koji korisnik može dodatno menjati, DALL•E generiše četiri slike za trenutnu regiju. Korisnik bira jednu od četiri. Ovaj proces traje dok god postoji makar jedna regija koja nema sliku ili dok korisnik ne prekine proces kreiranja slika
- Rezultat: Grupa slika koja se koristi prilikom igranja.

7.2. Mod beskonačnog igranja

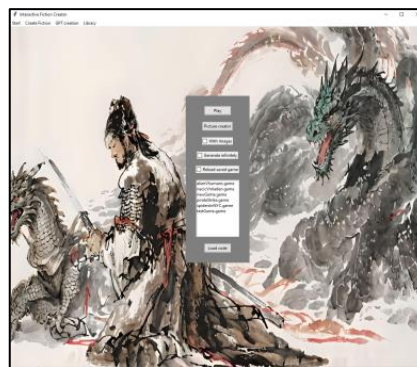
Beskonačni mod igranja omogućava igraču da istražuje nove regije, suočava se sa novim izazovima, neprijateljima i pronalazi nove predmete, oružja i oklope, bez krajnjeg cilja ili završne tačke. Ovaj mod je kreiran da uvek nudi nešto novo, omogućavajući igraču neograničeno istraživanje.

U ovom modu beskonačnog igranja, koristi se GPT model za generisanje novih delova igre, kao što su regije, neprijatelji, predmeti, oružje i oklop. Takođe, ako je izabran mod igranja sa slikama u ovom beskonačnom modu se koristi i DALL•E model koji ih generiše prilikom igračevog prvog ulaska u novootkrivene regije. Regije imaju šansu da budu povezane sa već postojećim regijama, a igra se nastavlja beskonačno jer se novi elementi generišu nakon što igrač dođe do kraja poznatog sveta. Svi elementi igre, uključujući nazive i opise, prilagođeni su koristeći prethodno generisane podatke i nasumično određene vrednosti, što stvara jedinstveno i dinamično iskustvo. Ovi procesi uključuju i ograničenja kako bi se obezbedila konzistentnost i kvalitet igre, kao što su karakterni limiti za opise.

8. GRAFIČKI INTERFEJS

Grafički korisnički interfejs aplikacije implementiran je pomoću ugrađene Python biblioteke Tkinter. Interfejs je podeljen u pet celina, od kojih svaka ima specifičnu funkciju u okviru aplikacije:

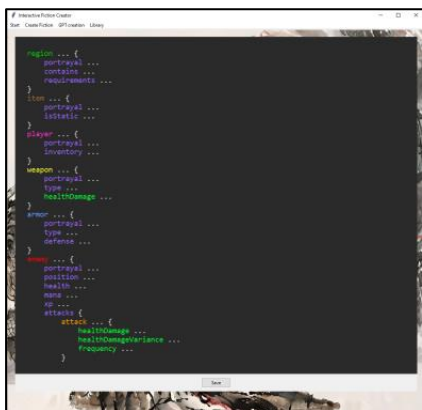
- Osnovni prozor aplikacije: Ovaj prozor sadrži navigacionu traku koja služi kao centralna tačka za upravljanje aplikacijom. Tačnije, osnovni prozor ove aplikacije predstavlja prozor pomoću kojeg je moguće doći do svih ostalih (slika 2).
- Code Editor: Ovaj deo interfejsa omogućava korisnicima da kreiraju igre interaktivne fikcije putem uređivanja koda. Code Editor pruža alate za pisanje i uređivanje koda, omogućavajući korisnicima da direktno upravljaju logikom i sadržajem igara (slika 3).
- Deo za kreiranje slika regiona: Ova sekcija interfejsa omogućava korisnicima da kreiraju vizuelne prikaze regiona u igri. Korisnici mogu dodavati i uređivati slike koje predstavljaju različite delove sveta igre, čime se vizuelno obogaćuje iskustvo igranja (slika 4).
- Deo za generisanje igara pomoću prompta: Ovaj deo interfejsa omogućava korisnicima da generišu igre interaktivne fikcije unosenjem njihovih promptova. Generisanje igara se vrši uz pomoć GPT modela, koji kreira scenarije i sadržaje igre na osnovu unetih promptova (slika 5).
- Deo za igranje igara interaktivne fikcije: Ovaj deo interfejsa služi za igranje kreiranih igara. Korisnici mogu testirati, odnosno igrati igre interaktivne fikcije koje su sami kreirali ili koje su generisane putem prompta, direktno unutar aplikacije (slika 6).



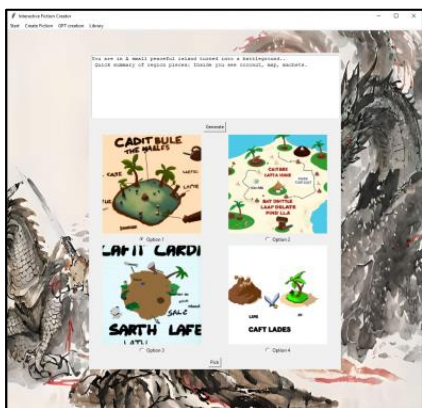
Slika 2. – Osnovni prozor

9. ZAKLJUČAK

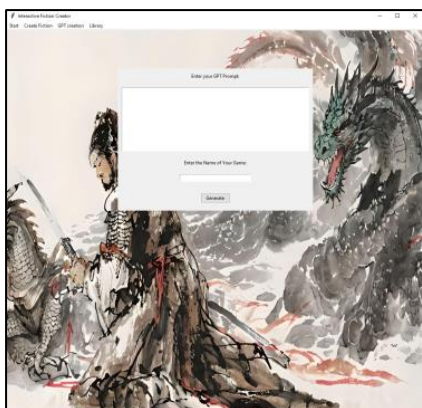
Ovaj rad naglašava važnost interaktivne fikcije kao softvera koji omogućava igračima da koriste tekstualne komande za upravljanje likovima i istraživanje virtuelnih svetova. Iako je njen značaj smanjen u poređenju sa ranijim periodima, interaktivna fikcija nastavlja da se razvija i privlači specifičnu publiku. U radu su prikazani alati i jezici korišćeni za razvoj interaktivne fikcije, uključujući textX za kreiranje prilagođenog jezika, Tkinter za grafički interfejs i OpenAI API za generisanje opisa i slika. Naglašena je važnost meta-modela i interpretera. Budući rad može uključivati poboljšanja u modeliranju sveta, proširenje mogućnosti generisanja sadržaja i istraživanje novih primena ovih tehnologija.



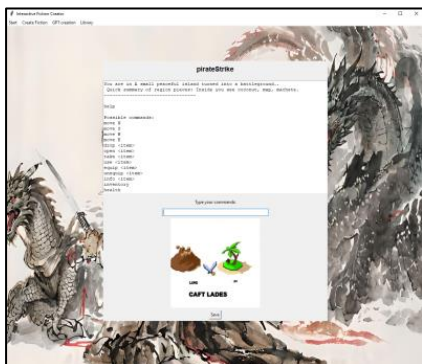
Slika 3. – Code Editor



Slika 4. – Prozor za kreiranje slike



Slika 5. – Prozor za GPT generisanje igre



Slika 6. –Prozor za igranje

10. LITERATURA

- [1] Montfort N. Riddle machines: The history and nature of interactive fiction. A companion to digital literary studies. 2013 May 17:267-82.
- [2] https://www.ifwiki.org/index.php/TADS_3 (pristupljeno u avgustu 2024.)
- [3] <https://inform-fiction.org/> (pristupljeno u avgustu 2024.)
- [4] Dejanović I, Vadera R, Milosavljević G, Vuković Ž. Textx: a python tool for domain-specific languages implementation. Knowledge-based systems. 2017 Jan 1;115:1-4.
- [5] <https://docs.python.org/3/library/tkinter.html> (pristupljeno u avgustu 2024.)
- [6] <https://openai.com/index/openai-api/> (pristupljeno u avgustu 2024.)
- [7] Wu T, He S, Liu J, Sun S, Liu K, Han QL, Tang Y. A brief overview of ChatGPT: The history, status quo and potential future development. IEEE/CAA Journal of Automatica Sinica. 2023 May 1;10(5):1122-36.
- [8] <https://openai.com/index/dall-e/> (pristupljeno u avgustu 2024.)
- [9] <https://www.graphviz.org/> (pristupljeno u avgustu 2024.)

Kratka biografija:



Aleksa Simić rođen je u Šapcu 2000. godine. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu, na smeru Softversko inženjerstvo i informacione tehnologije 2023. godine. Master rad na Fakultetu tehničkih nauka u Novom Sadu iz oblasti Elektrotehničko i računarsko inženjerstvo – Fantazijske igre bazirane na velikim jezičkim modelima, odbranio je 2024.godine.

ANALIZA ALGORITAMA ROJA ČESTICA ZA PROBLEM RUTIRANJA ELEKTRIČNIH VOZILA

ANALYSIS OF PARTICLE SWARM ALGORITHMS FOR THE ELECTRIC VEHICLE ROUTING PROBLEM

Nevena Radešić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO

Kratak sadržaj – Ovaj rad se bavi primjenom algoritma optimizacije rojem čestica (PSO) i njegovog unaprijeđenog oblika, generaliziranog algoritma optimizacije rojem čestica (GPSO), u rješavanju problema rutiranja električnih vozila (EVRP). Glavni cilj je poređenje performansi PSO i GPSO nad različitim reprezentacijama čestica, konkretno: vektora prioriteta i matrice susjedstva. Eksperimentalni rezultati pokazuju da matrična reprezentacija značajno nadmašuje vektorsku, i da GPSO algoritam postiže bolje performance od osnovnog PSO algoritma. Pored ovoga, u radu su objašnjena dva algoritma zasnovana na Belman-Fordovom algoritmu koji služe za podjelu kompletne rute na više manjih i određivanje optimalnih pozicija stanica za punjenje.

Ključne reči: EVRP, PSO, GPSO, vektor prioriteta, matrica susjedstva, Bellman-Ford algoritam

Abstract – This paper addresses the application of Particle Swarm Optimization (PSO) and its enhanced version, Generalized Particle Swarm Optimization (GPSO), in solving electric vehicle routing problem (EVRP). The main objective is to compare the performance of PSO and GPSO using different particle representations, specifically: priority vector and adjacency matrix. Experimental results show that the matrix representation significantly outperforms the vector representation, and the GPSO achieves better performance than the basic PSO. Additionally, the paper provides explanations of two algorithms based on the Bellman-Ford algorithm, which serve to divide the complete route into smaller ones and determine optimal positions for charging stations.

Keywords: EVRP, PSO, GPSO, Priority vector, Adjacency matrix, Bellman-Ford algorithm

1. UVOD

EVRP je složenija varijanta klasičnog problema rutiranja vozila (VRP). VRP se bavi optimizacijom ruta za vozila koja treba da opsluže skup geografski raširenih klijenata [1] sa poznatim zahtjevima, sa ciljem minimizacije ukupnih troškova prevoza.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Zoran Jeličić, red. prof.

EVRP proširuje ovaj koncept, uzimajući u obzir ograničenja dometa električnih vozila i potrebu za stanicama za punjenje, što značajno povećava kompleksnost problema.

2. DEFINICIJA PROBLEMA

Problem koji je rješavan se može predstaviti pomoću neusmjerenog grafa

$$G = (V, A)$$

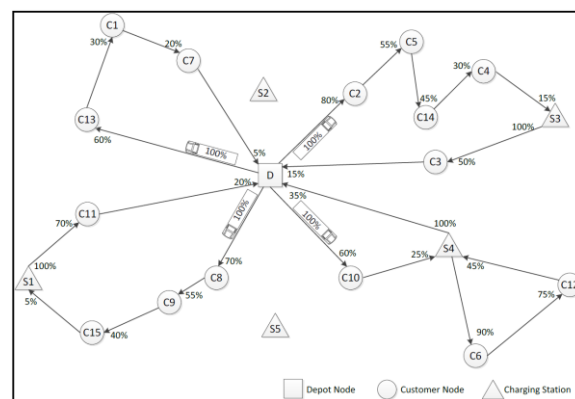
Gdje je V skup čvorova, a A skup grana. Čvorovi su klijenti, stanice za punjenje i skladište, a grane su veze između njih.

Skladište je jedinstveno i određeno je koordinatama u 2D prostoru. Svakog klijenta karakterišu potražnja i koodinate, dok su stanice određene koordinatama, tipom punjača i prosječnim vremenom čekanja.

Cilj je odrediti skup ruta za koje važe sledeća pravila:

- Svaka ruta počinje i završava se na skladištu.
- Svaki klijent mora biti uslužen tačno jednim električnim vozilom.
- Električna vozila mogu posjetiti stanicu za punjenje baterije između bilo koja dva klijenta.
- Nivo baterije električnog vozila mora uvijek biti između 20 i 80% kapaciteta baterije.
- Ukupno vrijeme putovanja je minimalno.

Primjer jednog rešenja za EVRP prikazan je na Slika 1 [2]:



Slika 1 – primjer rute za EVRP

3. ALGORITAM OPTIMIZACIJE ROJEM ČESTICA

Algoritam optimizacije rojem čestica (PSO) je stohastička optimizaciona metoda inspirisana socijalnim ponašanjem jata ptica ili riba.

Čestice predstavljaju skup potencijalnih rešenja i kreću se kroz prostor pretrage. Kretanje svake čestice određeno je njenom trenutnom pozicijom, prethodnom brzinom (komponenta inercije), ličnim najboljim iskustvom (kognitivna komponenta) i globalnim najboljim iskustvom cijelog roja (socijalna komponenta). Formalno zapisano, čestice se kreću na sledeći način:

$$v_i(t+1) = wv_i(t) + c_1r_1[\hat{x}_i(t) - x_i(t)] + c_2r_2[g(t) - x_i(t)] \quad (1)$$

gdje je:

$v_i(t)$ – brzina i -te čestice u t -toj iteraciji

$x_i(t)$ – položaj i -te čestice u t -toj iteraciji

$\hat{x}_i(t)$ – najbolja pozicija koju je i -ta čestica dostigla u svojoj istoriji

$g(t)$ – najbolja pozicija koju je dostigao cijeli roj

w , c_1 i c_2 – parametri koji kontrolišu odnos između tri komponente kretanja čestica

r_1 i r_2 – slučajni brojevi iz intervala $[0, 1]$.

4. GENERALIZOVANI ALGORITAM OPTIMIZACIJE ROJEM ČESTICA

Generalizovani algoritam optimizacije rojem čestica (GPSO) predstavlja unapređenje klasičnog PSO algoritma.

GPSO se temelji na reprezentaciji kretanja čestica kao stohastičkog, diskretnog, linearnog sistema drugog reda:

$$x[k+1] + a_1 * x[k-1] + a_0 * x[k] = b_p * p[k] + b_g * g[k] \quad (2)$$

Ova formulacija omogućava uvođenje tri nova parametra: ρ , ζ i c , koji zamjenjuju tradicionalne PSO parametre (w , c_p , c_a):

- ρ kontroliše stabilnost i brzinu konvergencije. Vrijednosti bliže 1 podstiču širu eksploraciju, dok niže vrijednosti ubrzavaju konvergenciju.
- ζ određuje kako čestice osciluju oko atraktora (lične i globalno najbolje pozicije). Za vrijednosti bliske 1, čestice se kreću bez mnogo oscilacija, dok vrijednosti bliske -1 uzrokuju drastične oscilacije.
- c je težinski faktor koji služi da zamijeni kognitivni i socijalni koeficijent. Veće vrijednosti favorizuju ličnu, a manje globalnu najbolju poziciju.

Imajući u vidu semantiku parametara, jasno je da bi bilo pogodno i poželjno njihove vrijednosti mijenjati u toku izvršavanja. Na primjer, ρ se može linearno smanjivati od 0.95 do 0.6 [3]. ζ se može nasumično birati iz intervala $[-0.6, 0.4]$ [3], a c se može smanjivati od 0.8 do 0.2 [3].

Glavna prednost GPSO-a u odnosu na klasični PSO je jasnija interpretacija parametara u smislu dinamike čestica, što omogućava finiju kontrolu nad procesom optimizacije. Empirijske studije pokazuju da GPSO često nadmašuje klasični PSO i genetske algoritme na mnogim standardnim testovima optimizacije.

5. PREDLOŽENO REŠENJE

5.1. Dva pristupa rešavanju problema

Za rješavanje EVRP grubo gledano, postoje 2 grupe pristupa [4]:

- 1) *Cluster-first route-second* – najprije se klijenti grupišu u klastere koje opslužuje jedno vozilo, a onda se za svaki klaster traži optimalna ruta.
- 2) *Route-first cluster-second* – najprije se kreira jedna džinovska ruta (engl. giant route), a zatim se ona dijeli na više kraćih ruta koje zadovoljavaju ograničenja kapaciteta.

U ovom radu je implementiran *Route-first cluster-second* pristup.

5.2. Reprezentacija čestica

S obzirom na to da se u radu koristi *Route-first cluster-second* pristup, jasno je da kôd čestice treba da sadrži samo informaciju o redoslijedu klijenata. Za to su implementirala dva pristupa: vektor prioriteta i matrica susjedstva.

5.2.1. Vektor prioriteta

U ovoj reprezentaciji svakom klijentu se dodjeljuje prioritet. Veći prioritet znači da klijent treba da bude uslužen ranije. Na primjer, ako imamo 5 klijenata označenih brojevima od 1 do 5 i reprezentacija jedne čestice je:

[0.2, 0.5, 0.1, 0.3, 0.4],

onda taj vektor reprezentuje sledeći redoslijed klijenata:

[2, 5, 4, 1, 3].

Prednosti ovog pristupa su jednostavnost za implementaciju i razumijevanje, a glavni nedostatak je što se osim prioriteta, ne čuva nikakve druge informacije o ruti ili klijentima.

5.2.2. Matrica susjedstva

Pomenuti nedostatak ispravlja matrična reprezentacija. Svako polje (i, j) matrice označava vjerovatnoću/prioritet da se nakon klijenta i u ruti pojavi klijent j . Na primjer, Tabela 1 kodira česticu:

[Складиште, клијент 3, клијент 2, клијент 1]

Tabela 1 – primjer koda jedne čestice

(i, j)	skladište	klijent 1	klijent 2	klijent 3
skladište	0.0	0.1	0.2	0.7
klijent 1	0.3	0.0	0.5	0.2
klijent 2	0.4	0.3	0.0	0.3
klijent 3	0.6	0.1	0.3	0.0

5.3. Generisanje inicijalne populacije

Nakon što je određen način kodiranja čestica, za primjenu PSO algoritma potrebno je generisati inicijalnu populaciju čestica.

Inicijalna populacija za vektorsku reprezentaciju generisana je nasumično. Svaki element čestice je kreiran

kao nasumičan broj iz intervala (0, 1). Nakon toga su elementi vektora skalirani tako da je suma elemenata jednaka 1.

Za matricnu reprezentaciju nasumično generisanje nije dalo zadovoljavajuće rezultate, najvjerojatnije zbog prevelike dimenzionalnosti problema (n^2). Umjesto toga, odlučeno je da se iskoristi semantika matrice i dostupni podaci. Odnosno, da se svakom polju (i, j) dodijeli recipročna vrijednosti rastojanja između klijenata i i j . Na ovako kreiranu matricu dodat je šum, da bi čestice bile što raznovrsnije. Konačno, vrijednosti u matrici su skalirane da bi se vrijednosti dovele u isti opseg.

5.4. Mutacije

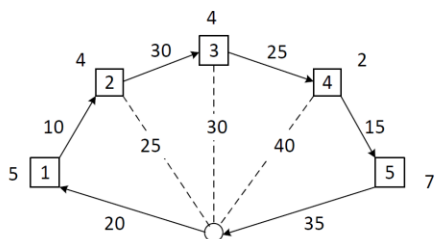
Mutacije predstavljaju slučajne izmjene u individualnim rešenjima populacije. Svrha mutacija je da prošire raznovrsnost postojećih rešenja i izbjegnu upadanje u lokalni optimum.

Za vektorsku reprezentaciju je implemenirana mutaciju zamjene (*swap*). Ova mutacija nasumično bira dvije pozicije u vektoru i mijenja njihove vrijednosti

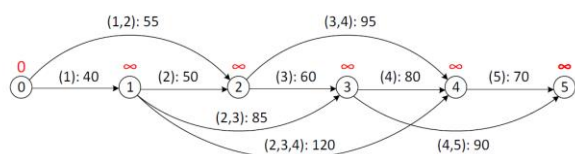
Za matricnu reprezentaciju smo primijenili mutaciju dodavanja šuma. Ovaj pristup dodaje mali, nasumični šum na elemente matrice, nakon čega se vrši renormalizacija.

5.5. Algoritam podjele

Korištenjem prethodno opisanih reprezentacija i pravila dobijena je ruta koja posjećuje sve klijente tačno jednom. Tako dobijenu rutu je potrebno podijeliti na više manjih. Problem takve podjele se može posmatrati kao problem najkraćeg puta u grafu ako se izvrše potrebne transformacije prikazane na Slika 2 i Slika 3 [4].



Slika 2 – prvi korak transformacije grafa



Slika 3 – drugi korak transformacije grafa

Ovako transformisan graf pogodan je za primjenu Belman-Fordovog algoritma koji pronalazi optimalnu podjelu.

5.6. Algoritam labeliranja

Posljednji korak u formiranju ruta je određivanje položaja stanica za punjenje. Za to je korišten algoritam labeliranja. Njegova ideja se zasniva na konceptu labele. Svaka labela čuva podatke o čvoru na koji se odnosi, nivou baterije, pređenom putu i prethodnoj labeli. Ova struktura omogućava algoritmu da "pamti" nekoliko

potencijalnih putanja do svakog čvora, uzimajući u obzir različite scenarije punjenja. Proces optimizacije se odvija kroz ekspanziju labela. Za svaki čvor, algoritam razmatra dvije mogućnosti: direktno kretanje do sljedećeg čvora ili odlazak do stanice za punjenje prije nastavka rute. Nova labela se dodaje na čvor samo ako ne postoji druga labela koja ju dominira (odnosi se na isti čvor, ali ima veći nivo baterije).

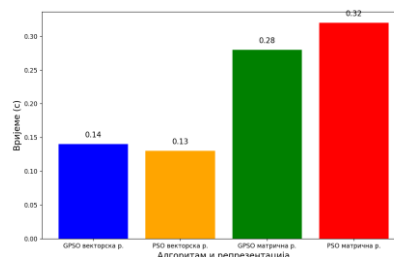
6. EKSPERIMENTALNI REZULTATI

Evaluacija predloženih metoda vršena je uz pomoć Šnajderovog skupa podataka [4]. On se sastoji iz dvije grupe instanci:

- Manje instance – do 15 klijenata;
- Veće instance – preko 100 klijenata

Prije testiranja bilo je potrebno podesiti vrijednosti parametara. To je urađeno nad jednim dijelom Šnajderovog skupa koji je izdvojen kao validacioni. Finalna testiranja vršena su nad drugim, testnim skupom.

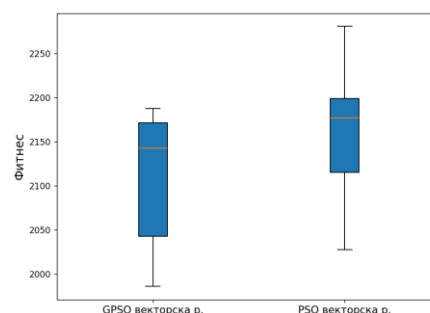
Nad manjim instancama sve metode su pronalazile optimalne rute. Razlika je bila samo u vremenu izvršavanja. Na Grafik 1 prikazano je prosječno vrijeme za 700 izvršavanja svakog pristupa.



Grafik 1 – vrijeme izvršavanja za manje setove

Očekivano, matricne reprezentacije su zbog svoje dimenzionalnosti imale duže vrijeme izvršavanja. PSO i GPSO algoritam imali su približno jednake performanse.

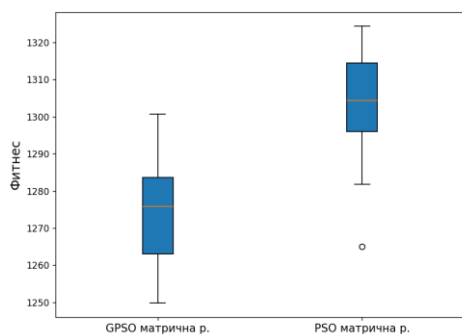
Stvarne razlike u performansama jasnije se vide na velikim instancama. Grafik 2 prikazuje boxplot vrijednost fitnes funkcije za PSO i GPSO nad vektorskom reprezentacijom čestica.



Grafik 2 – boxplot za vektorsku reprezentaciju

Možemo primijetiti da GPSO ima manji interkvartilni raspon u odnosu na PSO, što ukazuje na veću konzistentnost. Manja medijalna vrijednost ukazuje na bolju sposobnost pretrage GPSO algoritma.

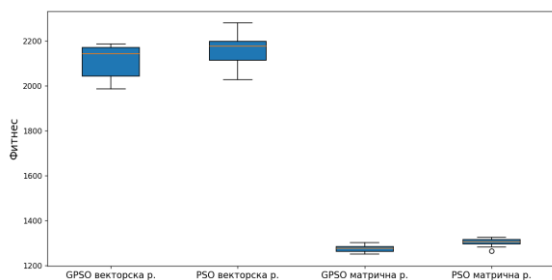
Grafik 3 prikazuje sličan grafikon, ali za matricnu reprezentaciju.



Grafik 3 – boxplot za matričnu reprezentaciju

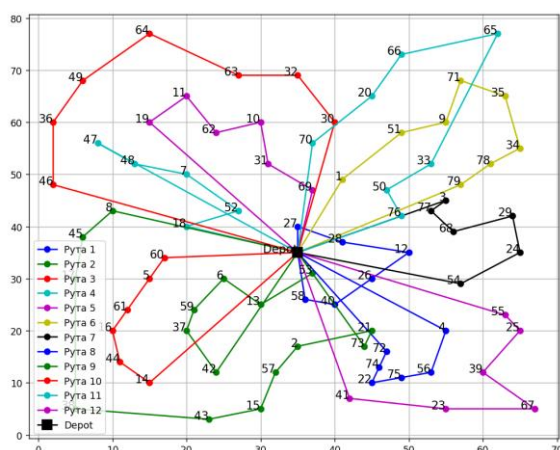
Ovdje vidimo da je razlika između algoritama još izraženija nego kod vektorske reprezentacije. PSO algoritam pokazuje veći raspon vrednosti, ima veću vrijednost medijane i outlier-e.

Ako oba boxplota smjestimo na isti grafik (Grafik 4), možemo primijetiti da matrična reprezentacija generalno daje niže vrijednosti fitness funkcije. Ovo jasno pokazuje da je matrična reprezentacija pogodnija za rješavanje EVRP.

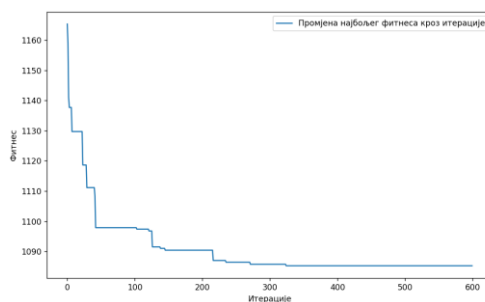


Grafik 4 – boxplotovi za sve pristupe

Slika 4 vizualizuje jedan primjer rute koju je pronašao GPSO algoritam sa matričnom reprezentacijom, dok Slika 5 prikazuje promjenu najboljeg fitnessa kroz iteracije. Iako nije pronađeno globalno najbolje rješenje, možemo reći da ovaj algoritam ima potencijal za rješavanje EVRP s obzirom na kompleksnost problema i skupa podataka.



Slika 4 – primjer rute



Slika 5 – promjena najboljeg fitnessa kroz iteracije

7. ZAKLJUČAK

U ovom radu fokus je bio na komparativnoj analizi PSO algoritma i njegove unapređene verzije GPSO nad dve reprezentacije čestica - vektora prioriteta i matrice susjedstva. Rezultati nedvosmisleno pokazuju da matrična reprezentacija čestica nadmašuje vektorsku u kvalitetu dobijenih rešenja. Pored toga, eksperimenti su pokazali da GPSO algoritam postiže bolje performanse od osnovnog PSO algoritma.

8. LITERATURA

- [1] J. Ochelska-Mierzejewska, A. Poniszewska-Marańda, and W. Marańda, "Selected genetic algorithms for vehicle routing problem solving," *Electronics*, vol. 10, no. 24, p. 3147, Dec. 2021.
- [2] I. Kucukoglu, R. Dewil, and D. Cattrysse, "The electric vehicle routing problem and its variations: A literature review," *Computers & Industrial Engineering*, vol. 161, p. 107650, Nov. 2021.
- [3] M. R. Rapaic, Z. Kanovic, Z. D. Jelcic, and D. Petrovacki, "Generalized PSO algorithm—An application to Lorenz system identification by means of neural-networks," in *Proc. 9th Symposium on Neural Network Applications in Electrical Engineering*, Sep. 2008, pp. 31-35.
- [4] C. Prins, N. Labadi, and M. Reghioi, "Tour splitting algorithms for vehicle routing problems," *International Journal of Production Research*, vol. 47, no. 2, pp. 507-535, Jan. 2009.
- [5] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle-routing problem with time windows and recharging stations," *Transportation Science*, vol. 48, no. 4, pp. 500-520, Nov. 2014.

Kratka biografija:



Nevena Radešić je rođena u Doboju 2000. godine. Diplomirala je 2023. godine na Fakultetu tehničkih nauka u Novom Sadu, na smjeru Softversko inženjerstvo i informacione tehnologije. Master rad iz oblasti Elektrotehničkog i računarskog inženjerstva, pod naslovom "Analiza algoritama roja čestica za problem rutiranja električnih vozila", odbranila je 2024. godine.

BEZBJEDNOSNA ANALIZA CAN PROTOKOLA KORIŠĆENOG U AUTOMOBILIMA**SECURITY ANALYSIS OF THE CAN PROTOCOL USED IN CARS**Dragana Filipović, Nikola Luburić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO**

Kratak sadržaj – U ovom radu je analizirana bezbjednost vozila sa posebnim fokusom na CAN protokol i njegove komponente. Identifikovani su potencijalni napadači i detaljno su opisani njihovi ciljevi. Opisana je arhitektura automobila, s naglaskom na elektronske kontrolne jedinice i CAN protokol kao komunikacionu mrežu. Kao rezultat ovog istraživanja, razvijen je model prijetnji koji obuhvata identifikovane i analizirane prijetnje i napade na CAN protokol, pri čemu je za kategorizaciju prijetnji korišćen STRIDE okvir. Takođe su identifikovani mitigacioni mehanizmi, koji su klasifikovani u reaktivne i preventivne, te su ovi mehanizmi detaljno opisani..

Ključne reči: bezbjednost automobila, CAN protokol, model prijetnji

Abstract – This paper analyzes vehicle security with a particular focus on the CAN protocol and its components. Potential attackers have been identified, and their objectives are thoroughly described. The vehicle's architecture is outlined, with an emphasis on electronic control units, the communication network, and the CAN protocol. As a result of this research, a threat model has been developed that encompasses the identified and analyzed threats and attacks on the CAN protocol, with the STRIDE framework used for threat categorization. Mitigation mechanisms have also been identified, classified into reactive and preventive measures, and these mechanisms are described in detail.

Keywords: automotive security, CAN protocol, threat model

1. UVOD

Jedna od značajnih oblasti računarske bezbjednosti jeste bezbjednost automobila. Sa razvojem modernih automobila koji imaju integrisane napredne tehnologije, oni postaju sve ranjiviji na sajber prijetnje. CAN protokol [1][2][3] kao centralna komponenta u komunikaciji između različitih elektronskih modula je dizajniran osamdesetih godina kada bezbjednost nije bila prioritet. Zbog manjka bezbjednosnih mehanizama CAN protokol je ranjiv na različite oblike sajber napada preko kojih napadači mogu da dobiju kontrolu nad mnogim kritičnim funkcijama vozila. Na taj način ugrožavaju ne samo putnike, nego i druge učesnike u saobraćaju.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Nikola Luburić.

Bezbjednost automobila je izuzetno složen zadatak zbog same kompleksnosti sistema i rastućeg broja potencijalnih ranjivosti. Cilj ovog istraživanja je da se identifikuju i analiziraju potencijalne prijetnje sa fokusom na CAN protokol, da se identifikuju napadi i predlože mitigacije za unapređenje sigurnosti vozila i sprečavanje napada. Naredno poglavlje govori o identifikaciji glavnih napadača i njihovih ciljeva. Treće poglavlje opisuje arhitekturu automobila dok četvrto opisuje model prijetnji sa pregledom napada i mitigacija. U petom poglavlju je data diskusija sa ograničenjima istraživanja. Posljednje poglavlje je zaključak koji sadrži sažetak istraživanja i moguća buduća unaprijeđenja.

2. PRIJETNJE VISOKOG NIVOA

Prijetnje visokog nivoa predstavljaju događaje koji omogućavaju napadačima da zadovolje svoje ciljeve ili da steknu prednost nad svojim metama. Ovo poglavlje će obuhvatiti analizu potencijalnih napadača i njihove motive, kao i identifikaciju prijetnji visokog nivoa koje ovi napadači mogu iskoristiti za postizanje svojih ciljeva.

2.1 Glavni napadači

Glavni napadači uključuju maliciozne korisnike, kriminalne grupe i insajdere kao što je identifikovano u radu [4]. Pored tradicionalnih napadača, bitno je pomenuti i naučnike i istraživače. Oni se nazivaju etički hakeri i traže ranjivosti u sistemu i načine da se poboljša bezbjednost u istim.

2.2 Identifikacija prijetnji i ciljevi napadača

Zbog sve veće povezanosti vozila sa softverom i mrežnim komunikacijama, vozila su postala lakše mete za sajber napade jer se inicijalni skup načina da se ispune ciljevi napadača (skup prijetnji) proširio. Neke od prijetnji visokog nivoa identifikovanih u radovima [5] i [6] su:

- Preuzimanje kontrole nad vozilom
- Krađa osjetljivih podataka
- Otključavanje vozila
- Krađa vozila
- Praćenje vozila
- Otkupnina
- Onesposobljavanje ili narušavanje funkcionalnosti sistema

Ove prijetnje omogućavaju napadačima dobitak profita, zadovoljenje ličnih izazova, sabotazu konkurencije, izazivanje straha i panike ili zloupotrebu radi kriminalnih aktivnosti.

3. ARHITEKTURA VOZILA

U ovom poglavlju je objašnjena složena arhitektura modernih vozila koja je sastavljena iz velikog broja elektronskih kontrolnih jedinica (*ECU*), softverskih komponenti i mehaničkih dijelova. Poseban fokus je stavljen na *CAN* protokol, pomoću kojeg se odvija komunikacija između elektronskih modula.

3.1 Elektronske kontrolne jedinice

U modernim vozilima se nalazi između 70 i 100 elektronskih kontrolnih jedinica koje razmjenjuju preko 2500 signala kako je u radu [7] navedeno.

Neki od ključnih *ECU* modula su [8]:

- *ECM* – kontroliše rad motora.
- *EBCM* – kontroliše rad kočnica.
- *TCM* – kontroliše rad mjenjača i prenos snage.
- *BCM* – kontroliše razne funkcionalnosti vozila kao što je upravljanje svjetlima, brisačima, zaključavanjem vrata, itd.
- *Telematics* – komunicira sa spoljašnjim mrežama kao što su telekomunikacione mreže i GPS, omogućavajući daljinsko upravljanje i praćenje vozila.
- *RCDLR* – omogućava daljinsko otključavanje i zaključavanje vrata.
- *HVAC* – kontroliše sistem za grijanje, ventilaciju i klimatizaciju u vozilu.
- *SDM* – upravlja vazдушnim jastucima i zatezačima pojasa.
- *IPC/DIC* – ispisuje informacije vozaču o brzini, potrošnji goriva i druge relevantne informacije.
- *Radio* – pored kontrole radija, kontroliše sve dodatne zvukova u vozilu.
- *TDM* – onemogućava krađu vozila bez odgovarajućeg ključa.

3.2 CAN protokol

CAN komunikacioni sistem je serijski dvožični multimaster sistem, dizajniran od strane Roberta Boša osamdesetih godina. Danas je najčešće korišćen komunikacioni sistem u automobilima i propisuje ga *ISO 11898* [1][2][3]. Glavne osobine *CAN* protokola su:

- Multimaster sistem – svaki uređaj koji je na mreži može da pošalje poruku. Ukoliko više uređaja istovremeno pokuša da pošalje poruku, dolazi do kolizije koja se rješava pomoću arbitraže.
- Broadcast priroda – Poruke se šalju svim čvorovima na mreži, a svaki čvor može da primi poruku.
- Prioritet poruka – Poruke sa nižim vrijednostima identifikatora imaju viši prioritet, i oni imaju prvenstvo kada se radi o slanju poruka.
- Zahtjev za podacima – Čvorovi mogu da zahtijevaju podatke od drugih, što smanjuje opterećenje mreže.
- Umetanje bitova – Ako postoji više od pet bitova istog polariteta, umeće se bit suprotnog polariteta kako bi se ispoštovao standard.

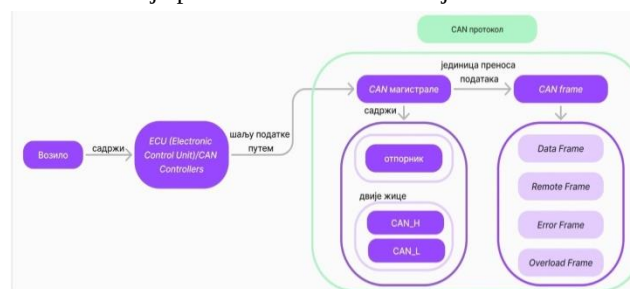
- Detekcija greške – Čvorovi mogu da detektuju greške i obavijeste ostale čvorove na mreži da je do greške došlo.

Ako posmatramo *CAN* protokol sa fizičkog sloja, možemo reći da se sastoji iz kabla sa upređenim paricama i da se završava sa otpornicima na oba kraja. Sadrži dvije žice *CAN_L* i *CAN_H*. Binarni signali koje se transmituju se mapiraju tako da se logička jedinica mapira na visoki napon (dominantni bit), a logička nula na niski napon (recesivni bit).

Postoje dva tipa *CAN* protokola u zavisnosti od brzine prenosa podataka: High-speed (*ISO 11898-2*) i Low-speed (*ISO 11898-3*). Različite elektronske jedinice koriste različite tipove *CAN* protokola u zavisnosti od potrebne brzine prenosa podataka, tj. od kritičnosti sistema.

U *CAN* komunikaciji postoje četiri tipa poruka: poruka sa podacima (engl. *Data frame*), poruka sa zahtjevom za podacima (engl. *Remote frame*), poruka o grešci (engl. *Error frame*) i poruka o zauzetosti (engl. *Overload frame*). Ove poruke čvorovi na mreži šalju preko *CAN* magistrale.

Na Sliku 3.1 je prikazan tok komunikacije u vozilu.



Slika 3.1 Tok komunikacije u vozilu

4. MODEL PRIJETNJI

Modelovanje prijetnji predstavlja proces identifikacije, analize i prioritizacije potencijalnih bezbjednosnih prijetnji koje mogu ugroziti integritet, povjerljivost i dostupnost sistema. Prvi korak u modelovanju prijetnji je identifikacija površine napada, odnosno definisanje ulaznih tačaka u sistem.

U vozilu postoji komunikacija putem *CAN* protokola između *ECU*, ali i komunikacija sa spoljašnjim elementima putem mobilne mreže, *GPS*-a, *Bluetooth*-a, *Wi-Fi*-a i *OBD-II* porta. Ove ulazne tačke su potencijalne mete napada kroz koje napadači mogu da pristupe sistemu i izvrše maliciozne aktivnosti.

Napadi mogu da budu fizički kroz direktan pristup vozilu i povezivanje na *CAN* mrežu što je omogućeno preko *OBD-II* porta, ili bežično, putem *Bluetooth*-a, *Wi-Fi* mreže ili mobilnih aplikacija, što omogućava udaljeni pristup *CAN* mreži [8][9][10].

4.1 Identifikacija prijetnji

U drugom poglavlju su identifikovane prijetnje visokog nivoa koje predstavljaju ozbiljne rizike po bezbjednost vozila i putnika u istom. Kako bi se izvršila identifikacija i kategorizacija prijetnji na nižem nivou, korišćen je *STRIDE* model [11], koji klasifikuje prijetnje u šest glavnih kategorija:

1. *Spoofing* – napadač se lažno predstavlja da je drugi uređaj ili komponenta.
2. *Tampering* – izmjena podataka u porukama, manipulacija podacima.
3. *Repudiation* – napadač izvodi akcije koje ne mogu da odaju njegov identitet.
4. *Information disclosure* – napadač ima pristup podacima kojime ne bi trebao da ima pristup.
5. *Denial of Service* – napadač narušava dostupnost sistema.
6. *Elevation of Privilege* – napadač izvodi akcije koje ne zahtjevaju autorizaciju ili je zaobilazi.

Tabela 4.1 Mapiranje prijetnji na STRIDE kategorije

Prijetnje visokog nivoa	Prijetnje niskog nivoa	STRIDE kategorije
Preuzimanje kontrole nad vozilom	Umetanje lažnih CAN poruka	<i>Spoofing</i>
Onesposobljavanje sistema	Zaustavljanje CAN mreže	<i>Denial of Service (DoS)</i>
Krađa podataka	Prisluškivanje CAN komunikacije	<i>Information Disclosure</i>
Otključavanje vozila	Manipulacija signalima	<i>Tampering</i>
Krađa vozila	Preuzimanje kontrole nad sistemom pokretanja	<i>Spoofing</i> <i>Tampering</i> <i>Elevation of Privilege</i>
Praćenje vozila	Prisluškivanje poruka sa lokacijom	<i>Information disclosure</i>
Otkupnina	Preuzimanje kontrole nad kritičnim sistemima	<i>Spoofing</i> <i>DoS</i>

4.2 Identifikacija napada

Na osnovu radova [8][12][13][14][15] identifikovani su napadi poput *Spoofing*, *Sniffing*, *Replay*, *DoS*, *Fuzzing*, *Reverse Engineering*, *Message Injection* i *Bus-off*.

Karakteristike napada:

- *Spoofing* – lažno predstavljanje
- *Sniffing* – prisluškivanje poruka na mreži
- *Replay* – slanje legitimnih poruka
- *DoS* – slanje poruka visokog prioriteta
- *Fuzzing* – slanje nasumičnih poruka
- *Reverse engineering* – otkrivanje funkcije poruka
- *Message injection* – slanje izmjenjenih poruka
- *Bus-off* – uklanjanje čvora sa mreže

4.3 Ranjivosti CAN protokola

CAN protokol nije razvijan sa bezbjednošću na umu zbog čega je navedene prijetnje i napade moguće realizovati. Ranjivosti koje posjeduje su sljedeće:

- Broadcast priroda – Svaka poruka je vidljiva svim čvorovima na mreži čime se narušava povjerljivost.
- Nedostatak autentifikacije – CAN poruke nemaju zaštitu od lažnih poruka.
- Slabe kontrole pristupa – Nedostaje autorizacija.
- Arbitraža – DoS napadi mogu prekinuti komunikaciju čvorova zbog prioritetskog slanja poruka.
- Nedostatak neporecivosti – Nemoguće je dokazati ko je poslao ili primio poruku.
- Loša segmentacija mreže – Kompromitacijom jednog modula se ugrožava cijeli sistem.

4.4 Bezbjednosni mehanizmi

Bezbjednosni mehanizmi se dijele na preventivne i reaktivne [16], pri čemu preventivni uključuju tehnike poput enkripcije, *Message Authentication Code*, tehnike upravljanje ključevima [15][17] i challenge-reposne metode [18]. Ovi mehanizmi pomažu u zaštiti sistema tako što sprečavaju pojavu napada poput *Sniffing*, *Spoofing*, *Replay*.

Reaktivni mehanizmi su dizajnirani da otkriju i odgovore na napade u toku ili nakon što su se desili. Ovi sistem uključuju *Intrusion Detection Systems*, forenzičku analizu i honeypot sisteme [16]. IDS sistemi se mogu zasnivati na *host-u* ili mreži [19][20], a metode detekcije mogu biti bazirane na potpisima ili anomalijama [17].

Međutim, uvođenje ovih mehanizama se suočava sa izazovima kao što su ograničenja hardvera, potreba za reakcijom u realnom vremenu i kompatibilnost sa postojećim tehnologijama.

5. DISKUSIJA

Preko fizičkog pristupa (OBD-II port) i bežične mreže (*Bluetooth*), omogućen je pristup CAN magistrali i analizirani su svi gore pomenuti napadi koji su omogućili preuzimanje kontrole nad vozilom, otključavanje i krađu vozila. Preporučene sugestije vozaču su držanje ključeva što dalje od vozila, redovno ažuriranje softvera na vozilu, i pažljivo parkiranje vozila kako bi se smanjila izloženost napadima.

Istraživanje je bilo ograničeno na CAN protokol, s fokusom na teorijske prijetnje bez testiranja u realnim uslovima. Pored toga, istraživanje je obuhvatilo samo poznate prijetnje ostavljajući mogućnost da određeni napadi ostanu neotkriveni. Takođe, proizvođači vozila često kriju implementacione detalje, čime se analiza mitigacionih mehanizama otežava.

6. ZAKLJUČAK

Ovo istraživanje analizira bezbjednosne izazove u automobilske industriji, s naglaskom na CAN protokol, koji je ključan za komunikaciju u modernim vozilima, ali nema adekvatne bezbjednosne mehanizme. Korišćenje STRIDE modela omogućilo je klasifikaciju prijetnji koje mogu ugroziti integritet, povjerljivost i dostupnost sistema. Naglašena je važnost preventivnih i reaktivnih

mjera kao što su enkripcija, *Message Authentication Code*, i sistemi za detekciju upada. Istraživanje je bilo otežano zbog ograničenog pristupa informacijama o zaštitnim mjerama koje proizvođači vozila interno primjenjuju. Zbog brzog razvoja tehnologija poput autonomnih i povezanih vozila, potreba za naprednijim bezbjednosnim mjerama postaje sve veća. Buduća istraživanja će se usmjeriti na razvoj novih metoda zaštite vozila od nepoznatih prijetnji, uz kontinuirano poboljšanje sigurnosti u automobilske industriji.

7. LITERATURA

- [1] Lawrenz, W. (2013b). *Can system engineering: From theory to practical applications* (Corr. Edition ed). New York: Springer.
- [2] Ibrahim, D. (2016). *Controller Area Network projects with arm and Arduino* Dogan Ibrahim. London: Elektor International Media BV.
- [3] Di Natale, M. (2012). *Understanding and using the Controller Area Network Communication Protocol Theory and Practice*. New York, NY: Springer New York.
- [4] McAfee. (2017). *Automotive Security Best Practices: Recommendations for security and privacy in the era of the next-generation car*. <https://motordna.io/static/stickerlook/images/wp-automotive-security.pdf> [datum pristupa júl 2024]
- [5] Smith, C. (2016). *The car hacker's Handbook: A guide for the penetration tester*. San Francisco, CA: No Starch Press.
- [6] Studnia, I., Nicomette, V., Alata, E., Deswarte, Y., Kaâniche, M., & Laarouchi, Y. (2013, June). Survey on security threats and protection mechanisms in embedded automotive networks. In *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)* (pp. 1-12). IEEE.
- [7] Charette, R. N. (2009). This car runs on code. *IEEE spectrum*, 46(3), 3.
- [8] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., ... & Savage, S. (2010, May). Experimental security analysis of a modern automobile. In *2010 IEEE symposium on security and privacy* (pp. 447-462). IEEE.
- [9] Bozdal, M., Samie, M., & Jennions, I. (2018, August). A survey on can bus protocol: Attacks, challenges, and potential solutions. In *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)* (pp. 201-205). IEEE.
- [10] Carsten, P., Andel, T. R., Yampolskiy, M., & McDonald, J. T. (2015, April). In-vehicle networks: Attacks, vulnerabilities, and proposed solutions. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference* (pp. 1-8).
- [11] M. Howard and S. Lipner, *The Security Development Lifecycle*. Redmond, WA, USA: Microsoft Press, 2006
- [12] Luo, F., Jiang, Y., Wang, J., Li, Z., & Zhang, X. (2023). A Framework for Cybersecurity Requirements Management in the Automotive Domain. *Sensors*, 23(10), 4979.
- [13] Aliwa, E., Rana, O., Perera, C., & Burnap, P. (2021). Cyberattacks and countermeasures for in-vehicle networks. *ACM computing surveys (CSUR)*, 54(1), 1-37.
- [14] Rathore, R. S., Hewage, C., Kaiwartya, O., & Lloret, J. (2022). In-vehicle communication cyber security: challenges and solutions. *Sensors*, 22(17), 6679.
- [15] Hossain, M. D., Inoue, H., Ochiai, H., Fall, D., & Kadobayashi, Y. (2020). LSTM-based intrusion detection system for in-vehicle can bus communications. *Ieee Access*, 8, 185489-185502.
- [16] Gmiden, M., Gmiden, M. H., & Trabelsi, H. (2019, March). Cryptographic and Intrusion Detection System for automotive CAN bus: Survey and contributions. In *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)* (pp. 158-163). IEEE
- [17] Woo, S., Jo, H. J., & Lee, D. H. (2014). A practical wireless attack on the connected car and security protocol for in-vehicle CAN. *IEEE Transactions on intelligent transportation systems*, 16(2), 993-1006.
- [18] Hoppe, T., Kiltz, S., & Dittmann, J. (2008). Security threats to automotive CAN networks—practical examples and selected short-term countermeasures. In *Computer Safety, Reliability, and Security: 27th International Conference, SAFECOMP 2008 Newcastle upon Tyne, UK, September 22-25, 2008 Proceedings 27* (pp. 235-248). Springer Berlin Heidelberg.
- [19] Young, C., Zambreno, J., Olufowobi, H., & Bloom, G. (2019). Survey of automotive controller area network intrusion detection systems. *IEEE Design & Test*, 36(6), 48-55.
- [20] F. M. Tabrizi and K. Pattabiraman, "Flexible intrusion detection systems for memory-constrained embedded systems," in *Proc. of Dependable Computing Conference (EDCC)*. IEEE, 2015, pp. 1–12.

Kratka biografija:



Dragana Filipović je rođena 24. decembra 1998. godine u Doboju. U Modriči upisuje opštu gimnaziju „Jovan Cvijić“. Kao nosilac diplome „Vuk Karadžić“ 2018. Godine upisuje Fakultet tehničkih nauka, smjer Softversko inženjerstvo i informacione tehnologije. Po završetku studija u roku, upisuje master akademske studije koje završava 2024. godine.

ZAGONETNA AVANTURA U VIRTUELNOJ REALNOSTI**PUZZLE ADVENTURE IN VIRTUAL REALITY**Danilo Bulatović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Ovaj rad opisuje razvoj igre *Dungeon Quest VR*, zagonetne avanture u virtualnoj realnosti. Igra je razvijana za VR uređaje *Meta Quest 2* i *Meta Quest 3*. Rad detaljno opisuje priču, likove, gejملهj, mehanizme igre, dizajn nivoa, audio dizajn i korisnički interfejs.

Ključne reči: Razvoj igre, Virtualna realnost, *Meta Quest*, *Dungeon Quest VR*, Zagonetna avantura

Abstract – This paper describes the development of *Dungeon Quest VR*, a puzzle adventure in virtual reality. The game was developed for the *Meta Quest 2* and *Meta Quest 3* VR devices. The paper provides a detailed description of the story, characters, gameplay, game mechanics, level design, audio design and user interface.

Keywords: Game Development, Virtual Reality, *Meta Quest*, *Dungeon Quest VR*, Puzzle adventure

1. UVOD

Virtualna realnost omogućava korisnicima da urone u potpuno digitalne svetove, simulirajući prisustvo i interakciju u virtualnom okruženju. Zahvaljujući naretku u hardveru i softveru, virtualna realnost je postala pristupačna i široko rasprostranjena, pružajući ne samo zabavu već i mogućnosti za edukaciju, obuku, terapiju i mnoge druge primene. Moć tehnologije virtualne realnosti u stvaranju imerzivnih iskustava bila je ključna inspiracija za razvoj igre *Dungeon Quest VR* koja je predstavljena u ovom radu. U radu će biti predstavljen sveobuhvatan pregled igre, što uključuje priču igre, likove u igri, gejملهj, mehanizme igre, kao i dizajn nivoa, zajedno sa audio efektima i tokom interakcije između igrača i igre.

Dungeon Quest VR je 3D avanturistička igra za *Meta Quest* [1, 2] VR platformu. Igra je smeštena u tamnici punoj zagonetki i izazova, gde igrač preuzima ulogu mladog junaka koji zajedno sa prijateljem vitezom istražuje tamnicu i rešava izazove i zagonetke u njoj u potrazi za Večnim draguljem.

Igra je namenjena avanturistima koji vole „escape room“ igre, nudeći zanimljive izazove i zagonetke. Interaktivno okruženje podstiče radoznalost i kreativnost, omogućavajući igračima da istražuju i manipulišu objektima. Zvučni efekti i muzika dodatno obogaćuju imerzivno iskustvo, čineći igru privlačnom za ciljnu grupu.

NAPOMENA:

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

2. META QUEST 2 I META QUEST 3 UREĐAJI

Virtualna realnost se brzo razvija kao ključna tehnologija koja omogućava korisnicima da istražuju digitalne svetove na potpuno nov način. *Meta Quest* serija, koju razvija kompanija *Meta*, ističe se kao lider na tržištu VR uređaja zahvaljujući svojoj pristupačnosti i inovativnim mogućnostima.

Meta Quest 2 [1], lansiran u oktobru 2020. godine, brzo je postao jedan od najpopularnijih VR uređaja zbog svoje samostalnosti, pristupačne cene, i bogatog ekosistema aplikacija. Naslednik ovog uređaja, *Meta Quest 3* [2], predstavljen je u oktobru 2023. godine, donoseći brojne tehnološke inovacije i unapređenja.

U ovom poglavlju istražujemo ključne karakteristike i razlike između *Meta Quest 2* i *Meta Quest 3* uređaja.

2.1. Meta Quest 2

Meta Quest 2 [1] je uređaj za virtualnu realnost razvijen od strane *Reality Labs*, ranije poznate kao *Oculus VR*, koja je u vlasništvu kompanije *Facebook*, sada poznate kao *Meta*. Predstavlja jedno od najpopularnijih i najpristupačnijih rešenja za virtualnu realnost na tržištu. Lansiran u oktobru 2020. godine, *Quest 2* nasleđuje originalni *Oculus Quest* i donosi nadogradnje hardvera, softvera i korisničkog interfejsa. Njegova jednostavnost i svestranost korišćenja učinili su ga primamljivim izborom kako za nove korisnike VR-a, tako i za iskusne entuzijaste. Uređaj je prikazan na slici 1.



Slika 1. Prikaz *Quest 2* uređaja i kontrolera [3, 4]

Quest 2 je samostalan VR uređaj, što znači da ne zahteva povezivanje sa računarom ili konzolom za rad. Svi potrebni hardverski i softverski resursi su integrisani unutar uređaja. Ova karakteristika ga čini izuzetno pristupačnim i prenosivim rešenjem za VR.

Pored toga, *Quest 2* nudi opciju povezivanja sa računarom što omogućava korisnicima da iskoriste prednosti snažnijeg hardvera računara za pokretanje zahtevnijih VR aplikacija i igara.

Jedna od ključnih karakteristika *Meta Quest 2* je njegova sposobnost unutrašnjeg praćenja. Ugrađene kamere na uređaju omogućavaju precizno praćenje pokreta korisnika i kontrolera u stvarnom vremenu, bez potrebe za

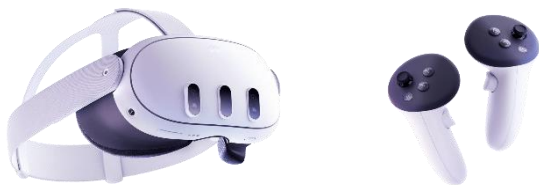
eksternim senzorima. Ovo pruža veću slobodu kretanja i pojednostavljuje postavljanje uređaja.

Quest 2 dolazi s dva *Touch* kontrolera, koji su dizajnirani da budu kompaktni i intuitivni. Oni koriste senzore pokreta i dugmad osjetljiva na dodir kako bi omogućili precizno upravljanje u VR okruženju. Kontroleri su prikazani na slici 1.

Meta Quest 2 je postao sinonim za pristupačnu i kvalitetnu virtuelnu realnost. Njegova kombinacija moćnog hardvera, jednostavnosti korišćenja i bogatog ekosistema čini ga idealnim izborom za širok spektar korisnika.

2.2. Meta Quest 3

Meta Quest 3 [2] je najnoviji uređaj za virtuelnu realnost iz kompanije *Meta*, naslednik popularnog *Meta Quest 2*. Predstavljen je u oktobru 2023. godine, sa ciljem da unapredi iskustvo VR-a i prošire realnosti, tj. AR, kombinacijom vrhunskog hardvera, naprednih mogućnosti praćenja i poboljšane optike. *Quest 3* pruža još veći nivo imerzije i performansi, ciljajući kako na obične korisnike, tako i na entuzijaste. Uređaj je prikazan na slici 3.



Slika 2. Prikaz *Quest 3* uređaja i kontrolera [2]

Kao i prethodnici, *Meta Quest 3* je samostalan uređaj, što znači da za svoj rad ne zahteva povezivanje sa spoljnim računarom ili konzolom. Međutim, može se koristiti s računarom omogućavajući korisnicima da iskoriste prednosti snažnijeg hardvera za zahtevnije igre i aplikacije.

Jedna od najvećih inovacija koju *Meta Quest 3* donosi u odnosu na svog prethodnika je integracija proširene realnosti, tj. AR, koja omogućava korisnicima da kombinuju digitalne elemente s fizičkim svetom. Ovo otvara potpuno nove mogućnosti za korisničko iskustvo i proširuje potencijalne aplikacije uređaja.

Quest 3 omogućava razvoj naprednih AR aplikacija koje mogu koristiti prostor oko korisnika na nove i inovativne načine. Od edukativnih aplikacija koje koriste proširenu realnost za prikazivanje interaktivnih modela do igara koje se mešaju sa fizičkim okruženjem korisnika, mogućnosti su široke i raznovrsne.

Novi *Touch Plus* kontroleri dolaze s haptičkim povratnim informacijama i poboljšanim trajanjem baterije. Kontroleri sada sadrže poboljšane senzore za praćenje pritiska i pokreta, omogućavajući korisnicima veću preciznost i kontrolu. Kontroleri su prikazani na slici 2.

Meta Quest 3 predstavlja veliki korak napred u svetu virtuelne realnosti, nudeći kombinaciju vrhunskog hardvera i naprednih mogućnosti. Kako tehnologija virtuelne realnosti nastavlja da evoluira, *Meta Quest 3* je postavljen da ostane ključni igrač u oblikovanju budućnosti ove tehnologije.

3. PRIČA IGRE

Priča igre prati Edgara, mladića iz srednjovekovnog sela, koji živi sa ocem Alfredom, bivšim vitezom koji se povukao iz ratovanja nakon što je izgubio ruku. Iako Edgar mašta o avanturama, njegov otac ga odvraća od toga, strahujući za njegovu sigurnost. Kada se u kraljevstvu pojavi mračni portal, a sile tame počnu da prete selu, dolazi Garlan, stari prijatelj Alfreda, u potrazi za Večnim draguljem koji može zatvoriti portal. Iako Alfred odbija da krene, Edgar se odlučuje da pođe sa Garlanom, uz očevo nevoljno odobrenje.

Na putovanju, Garlan postaje Edgarov mentor, dok zajedno prolaze kroz opasnosti začarane tamnice. U hodnicima prepunim zagonetki i zamki, njihova hrabrost i domišljatost su na testu. Konačno, pronalaze Večni dragulj u mračnoj odaji, ali njihov zadatak nije završen, jer se moraju vratiti kući, progonjeni mračnim silama koje žele da povrate dragulj i zadrže portal otvorenim.

Kroz niz opasnosti i bitaka, Edgar i Garlan uspevaju da sačuvaju dragulj i, po povratku u kraljevstvo, koriste njegovu moć da zatvore portal, žrtvujući dragulj za mir u kraljevstvu. Vraćaju se u selo, gde ih Alfred, sada ponosan na sina, radosno dočekuje.

5. LIKOVI

U ovom poglavlju će biti detaljno opisani likovi u igri, počevši od protagoniste igre, nakon koga će biti opisan mentor.

5.1. Protagonista

Edgar je protagonista igre. Živi na selu sa svojim ocem u kraljevstvu koje je trenutno napadnuto od strane mračnih stvorenja koja su došla iz mračnog portala. On je mladić u ranim dvadesetim godinama. Ima kratku crnu kosu, braon oči i visok je. Nosi srednjovekovnu seosku odeću. Pametan je i bistar, što mu pomaže prilikom rešavanja zagonetki u tamnici.

Pošto je u pitanju igra za uređaje virtuelne realnosti, igrač je u ulozi protagoniste, tako da njegov model nikada nije vidljiv u igri, samo njegove, odnosno igračeve šake.

5.2. Mentor

Garlan je mentor protagoniste. On je vitez koji je bio zajedno u ratu sa Edgarovim ocem. Nakon otvaranja mračnog portala u kraljevstvu, ima misiju da pronađe Večni dragulj i zatvori portal. U ovoj misiji mu pomaže protagonista. Garlan je vitez koji ima dugu narandžastu kosu, braon oči i visok je. Nosi zlatni viteški oklop. U igri, on daje savete protagonisti i prati ga. Izgled modela mentora je prikazan na slici 3.



Slika 3. Izgled modela mentora

6. GEJMPLEJ

U igri *Dungeon Quest VR*, igrač se kreće u 3D prostoru, istražuje tamnicu i rešava izazove i zagonetke u njoj. Pored samog kretanja, igrač može da interaguje sa raznim predmetima u tamnici kako bi rešio izazove.

Igrač može da uzme predmete u ruke, povlači poluge rukama, kao i otvara kovčeg kako bi pronašao određene predmete u njima. Pored toga, igrač može da uzima koplja i stavi ih na štitove kako bi rešio neke izazove, npr. otključao vrata do druge prostorije. Takođe, igrač može rešavati klizne puzzle kako bi rešio određene izazove. Igrač takođe može da uzme napitke i sipa ih u prazan kazan kako bi ga napunio. Nakon što je napunio kazan, može da uzme praznu posudu i da je iskoristi kako bi sadržaj kazana sipao u nju. Time igrač dobija punu posudu. Punu posudu može staviti na prazan tas vage kako bi je balansirao i tako rešio određeni izazov.

Pored svih ovih radnji, igrač u svakom trenutku može pitati svoga mentora za savet kako rešiti određeni izazov ili zagonetku. Cilj igre je da igrač pronađe Večni dragulj. Da bi stigao do cilja, igrač mora da istražuje tamnicu i rešava izazove i zagonetke u njoj. Rešavajući izazove, igrač napreduje do sledećeg nivoa, dok na kraju ne dođe do Večnog dragulja. Kada konačno pronađe Večni dragulj, igrač pobeđuje igru.

7. MEHANIZMI

Mehanizmi igre *Dungeon Quest VR* se sastoje od *core* i *satellite* mehanizama. Lokomotivni sistem i sistem interakcije sa objektima predstavljaju *core* mehanizme, dok sistem interakcije sa mentorom predstavlja *satellite* mehanizam.

U nastavku ovog poglavlja će detaljno biti opisani *core* i *satellite* mehanizmi u igri.

7.1. Core mehanizmi

Core mehanizme predstavlja lokomotivni sistem i sistem interakcije sa objektima. Lokomotivni sistem se sastoji iz stajanja i hodanja. Igrač počinje igru u stanju stajanja. Kako bi prešao u stanje hodanja, koristi *thumbstick* na levom kontroleru. Pravac pomeranja *thumbstick*-a predstavlja pravac kretanja igrača. Kada prestane da hoda, odnosno pusti levi *thumbstick*, igrač se vraća u stanje hodanja.

Sistem interakcije sa objektima se sastoji iz uzimanja i puštanja objekata. Kako bi igrač uzeo objekat, tj. predmet, u ruke, potrebno je da pritisne dugme *grip* na levom ili desnom kontroleru, u zavisnosti kojom rukom želi da uzme objekat. Kako bi stvarno uzeo objekat u ruke, ruka kojom želi da uzme objekat mora biti blizu predmeta koji želi uzeti. Ako ruka nije blizu predmeta, ništa se neće dogoditi pritiskom dugmeta. Objekat će ostati u igračevim rukama sve dok je dugme *grip* pritisnuto. Kada se to dugme pusti, predmet koji se nalazi u toj ruci će biti ispušten iz ruke.

Na slici 4 su prikazani mehanizmi i interakcije u igri, a to su uzimanje predmeta u ruke, otvaranje kovčega, povlačenje poluga, otključavanje vrata, punjenje kazana napicima i punjenje posude, interakcija sa kopljima i štitovima, kao i rešavanje kliznih puzzle.



Slika 4. Različite vrste interakcija u igri

7.2. Satellite mehanizmi

U igri postoji i mehanizam traženja saveta od mentora. Ako igrač ima poteškoća prilikom rešavanja nekog izazova ili zagonetke, može tražiti savet od mentora. Kako bi to uradio, igrač treba da usmeri ruku prema mentoru i pritisne *grip* dugme na kontroleru koji odgovara usmerenoj ruci. Nakon toga će mentor igraču dati savet. Tok interakcije igrača sa mentorom je prikazan na slici 5.



Slika 5. Tok interakcije igrača sa mentorom

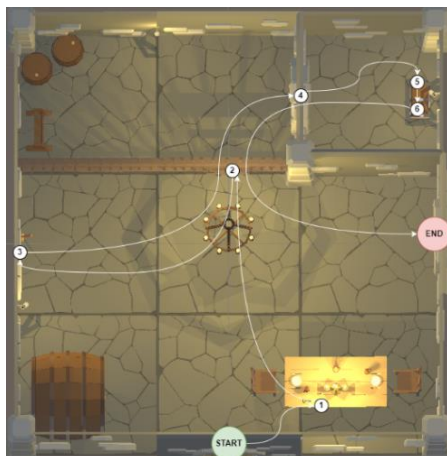
8. DIZAJN NIVOVA

Svaki nivo u igri predstavlja deo tamnice i sastoji se od jedne ili više soba. Počevši od prve prostorije, igrač istražuje tamnicu, otključava vrata do drugih soba i nivoa, kako bi na kraju došao do poslednje sobe u kojoj se nalazi Večni dragulj.

U svakom nivou, igrač se suočava sa raznim izazovima i zagonetkama, koje mora rešiti kako bi otključao naredni nivo. Ti izazovi su pronalaženje predmeta, povlačenje poluga, otvaranje kovčega, rešavanje puzzle itd.

Nivoi su dizajnirani tako da svaki naredni bude izazovniji od prethodnog. Početni nivo uči igrača osnovnim mehanizmima koje će igrač koristiti u narednim nivoima. Nakon nivoa gde igrač savlada osnovne mehanizme, slede nivoi gde se pojavljuju novi mehanizmi koje igrač treba

da nauči. Nakon što igrač nauči nove mehanizme, napreduje do nivoa gde se ti novi mehanizmi koriste zajedno sa starim, već naučenim i savladanim, mehanizmima. Na slici 6 je prikazan pogled odozgo na jedan od nivoa u igri, zajedno sa oznakom redosleda i toka rešavanja nivoa.



Slika 6. Prikaz jednog od nivoa odozgo

9. AUDIO DIZAJN

Zvučni efekti su jako važni u igrama, jer doprinose imerzivnosti igre i stvaranju dinamičnog i interaktivnog okruženja. Igra *Dungeon Quest VR* sadrži mnoge zvučne efekte, kako bi se pružilo bolje iskustvo igračima.

Uzimanje predmeta u ruke je dopunjeno zvučnim efektom koji pored imerzivnosti, služi i kao fideb igraču da je predmet uspešno uzet u ruke. Takođe, povlačenje poluga i otvaranje kovčega ima svoje zvučne efekte, kao i otključavanje i otvaranje vrata. Pored toga, uzimanje i postavljanje koplja na štit ima svoje zvučne efekte, kao i sipanje napitaka u kazan i pražnjenje kazana praznom posudom. Stavljanje pune posude na tas vage takođe ima propratni zvučni efekat. Mentor ima zvučni efekat prilikom kretanja, tj. zvuk koraka. Na kraju, uzimanje Večnog dragulja i završetak igre imaju zvučne efekte.

Osim zvučnih efekata, dodatnoj imerzivnosti i boljem doživljaju tokom igranja doprinosi propratna pozadinska muzika. Ova muzika ima srednjovekovni zvuk i usklađena je sa atmosferom igre i temom istraživanja tamnice.

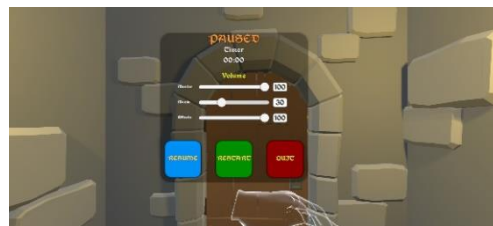
10. KORISNIČKI INTERFEJS

Ovo poglavlje opisuje interakciju igrača sa igrom putem korisničkog interfejsa. Na početku igre, igraču se prikazuje početni meni sa tri dugmeta, a to su „Play“ za početak igre, „Quit“, za izlazak i „Options“ za podešavanje. Izgled ovog menija je prikazan na slici 7.



Slika 7. Prikaz početnog menija

Na zglobu leve ruke igrač ima dugme „Menu“, koje pauzira igru i omogućava pristup meniju za zvuk, nastavak igre, ponovno igranje ili izlazak. Izgled ovom menija je prikazan na slici 8.



Slika 8. Prikaz menija na zglobu ruke

U igri, igrač može tražiti savet od mentora, prilikom čega se prikazuje dijalog sa savetom, a na kraju igre se prikazuje završni meni sa vremenom igranja i opcijama za ponovno igranje ili izlazak iz igre.

11. ZAKLJUČAK

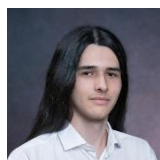
Ovaj rad opisuje razvoj igre *Dungeon Quest VR*, zagonetne avanture u virtuelnoj realnosti namenjene mladim adolescentima i odraslima koji uživaju u avanturama i „escape room“ igrama. Igra je razvijana za VR uređaje *Meta Quest 2* i njegovog naslednika *Meta Quest 3*, koji donosi tehnološka unapređenja. Razvoj je sproveden koristeći *Unity* [5], popularni alat za razvoj igara, i *OpenXR* [6], standardni API koji omogućava kompatibilnost s različitim VR uređajima.

Rad detaljno opisuje priču, likove, gejملهj, mehanizme igre, dizajn nivoa i audio. Posebna pažnja posvećena je dizajniranju zagonetki i izazova, balansiranju nivoa, kao i učenju igrača novih mehanizama tokom igre. Iako je igra završena, rad prepoznaje mogućnosti za dalji razvoj, uključujući dodavanje novih nivoa, likova i mehanizama, što bi doprinelo bogatijem iskustvu igranja.

12. LITERATURA

- [1] meta.com/quest/products/quest-2, Zvanična stranica *Meta Quest 2* uređaja. (pristupljeno u avgustu 2024.)
- [2] meta.com/quest/quest-3, Zvanična stranica *Meta Quest 3* uređaja. (pristupljeno u avgustu 2024.)
- [3] about.fb.com/news/2020/09/introducing-oculus-quest-2-the-next-generation-of-all-in-one-vr, Predstavljanje *Oculus Quest 2* uređaja. (pristupljeno u avgustu 2024.)
- [4] meta.com/quest/accessories/quest-2-controllers, Zvanična stranica *Meta Quest 2* kontrolera. (pristupljeno u avgustu 2024.)
- [5] docs.unity.com, Dokumentacija *Unity* softvera. (pristupljeno u avgustu 2024.)
- [6] docs.unity3d.com/Packages/com.unity.xr.openxr@1.1.2, Dokumentacija *OpenXR* plugina za *Unity*. (pristupljeno u avgustu 2024.)

Kratka biografija:



Danilo Bulatović rođen je u Brčkom 2000. godine. Diplomirao je na Fakultetu tehničkih nauka 2023. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva odbranio je 2024. godine. kontakt: danilo.bulatovic@uns.ac.rs

IZAZOVI U KREIRANJU 2D HEKANDSLEŠ VIDEO IGRE CHALLENGES IN CREATING A 2D HACK AND SLASH VIDEO GAME

Jovan Svorcan, *Fakultet tehničkih nauka, Novi Sad*

Oblast – SOFTVERSKO INŽENJERSTVO I INFORMACIONE TEHNOLOGIJE

Kratak sadržaj – U ovom radu je opisan tok razvoja računarske igre. Opisana je radnja unutar igre kao i osnovna pravila i kako je osmišljen dizajn nivoa sa primerima. Zatim se priča o korisničkim interfejsima i modelovanju glavnog lika. Dalje se priča o elementima multimedije u igri, uključujući animacije, zvuk i boje. Na kraju je ukratko opisan i pozadinski kod koji je korišćen u igri.

Ključne reči: Računarska igra, modeli, kamera, zvučni efekti

Abstract – This paper describes the process of developing a computer game. The plot within the game is described along with the basic rules and the plan for level designs with examples. Next are the user interfaces and the modeling of the main character. After that the elements of multimedia in the game is talked about, including animation, sound and color. Finally, the underlying code, that is used in the game is described.

Keywords: Computer game, models, camera, sound effects

1. UVOD

Zadatak ovog rada je bila izrada video igre Undead Savior, namenjena deci sa 13-17 godina, u kojoj će se susretati sa mnogih izazovima za savladati u obliku neprijatelja protiv kojih će morati da se bore. Ovakva tema je izabrana za rad jer je savladavanje izazova bitan deo odrastanja, a borbe u ovoj video igri bi trebalo da ciljnu publiku nateraju na razmišljanje, a paralelno i da im daje osećaj uspeha kroz uzbudljiv i napet gameplay.

U radu je opisan razvoj igre Undead Savior. Glavni junak je bezimeni junak probuđen iz mrtvih. Razlog za ovaj izbor je da se igračima jasno ustanovi da je ovde mračnija atmosfera nego u video igrama na koje su verovatno prethodno navikli gde bi uglavnom igrali kao neka životinja koja priča ili kao junak iz nekog crtanog filma. Isti razlog je uticao i na izbor boja za model protagoniste, o čemu će više reći biti kasnije.

Tokom izrade bilo koje računarske igre, postoji nekoliko faza. Postoji deo koji je zadužen za modelovanje igre, gde se određuje kako će scena izgledati, kako će izgledati karakteri i ostali objekti koji se pojavljuju u igri. Takođe, postoje delovi razvoja gde se opisuju načini interakcije korisnika sa samom igrom, određuje se šta bi se u kom trenutku igre moglo desiti, i delovi gde se određuje kako će se neka povratna informacija prikazati korisniku. Ove faze ne mogu da funkcionišu jedna bez druge jer same za sebe ne čine računarsku igru. Neophodno je imati sve njih zajedno kako bi oni zajedničkim delovanjem dali korisniku jednu veliku celinu.

U ovom radu biće detaljno opisana svaka faza izrade Undead Savior računarske igre. Prvo će biti opisana celokupna igra i navešće se video igre koje su služile kao glavne inspiracije. U drugom delu opisać se pozadinska priča, tok radnje tokom cele video igre i navešće se kratki opisi svih glavnih likova i njihovih uloga. U trećem poglavlju će se detaljno opisati kako teče sam gameplay igre i koje sve slobode i ograničenja pokreta će igrač da ima u zavisnosti od svoje trenutne situacije. Takođe će biti o reči o tome kakav je zamišljen dizajn nivoa. U četvrtom se priča o svim korisničkim interfejsima tokom igre. U petom poglavlju ovog rada biće opisan tok modelovanja glavnog lika. Takođe će se ukratko opisati prebacivanje tog i drugih modela na Unity. U šestom poglavlju se priča o raznim oblicima upotrebljenih elemenata multimedije: boje, audio i animacije. Poseban naglasak će biti na animacijama glavnog lika. Na samom kraju rada čitalac će imati uvid u literaturu koja se tokom izrade ovog rada koristila.

2. SIŽE RAČUNARSKE IGRE UNDEAD SAVIOR

Video igra Undead Savior je fantaziona akciona hack and slash avantura u trećem licu sa rejtingom T (Teen) jer uključuje mračniju atmosferu, stvorenja, kao i nasilja (ne preterano krvavog, ali taman dovoljno da nije preporučljivo maloj deci da je igraju).

Undead Savior je imao više inspiracija za svoj razvoj i to tako što je kombinovao neki deo od svake od navedenih video igra: Dark Souls, God of War, Resident Evil 4 i Bioshock.

3. OPIS RADNJE

Undead Savior pruža igračima priču u kojoj oni ulaze u lika koji se na samom početku izdiže iz mrtvih bez ikakvog sećanja na svoj prethodni život i bez ikakvog znanja o svetu u kojem se nalazi. Isto tako, igraču ništa skoro nije rečeno na samom početku video igre jer baš zato on može da sam da je otkriva u toku igranja i to ne

NAPOMENA:

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

samo u toku glavnih nivoa, već će se dosta delova ukupne priče nalaziti u sporednim prostorija igre koje igrač neće ni morati da istražuje, kao što je predloženo [6]. Na taj način bi trebalo da je igrač motivisan da hoće da pređe svaki deo igre kako bi mogao da dobije punu sliku, a ujedno će biti upućen u priču samo onoliko koliko i želi da bude.

Likovi su svi konstruisani na Jungovim arhitipovima gde svaki lik ima svoju ulogu u priči i ne nijedan lik ne bi trebalo da je suvišan. Poenta ovih uloga jeste da se osigura da igrač ni sa kojim likom neće imati isti oblik interakcija, odnosno svaki bi trebalo da mu pruža drugačiji oblik pomoći ili drugačiji izazov za preći. Neki od likova preuzimaju i po dve uloge u priči, mada većina spada jednu od glavnih 8 kategorija. Pored razlika između ovih likova postojaće i više sličnosti između njih kako bi priča bila kompleksnija gde bi se igrač stalno mogao pitati kako su ovi likovi mogli da završe drugačije da su pošli bili drugim putevima.

4. GAMEPLAY

Undead Savior pruža gameplay sličan više klasičnim 3d akcionim avanturama nego 2d i to zato što baš zato što kada ciljna publika počne da igra video igre sa ograničenim rejtingom, to vrlo budu igre koje su 2d. Glavni cilj ove igre je proba da obezbedi igračima slično iskustvo i pokaže da je to moguće u 2d okruženju. Igrač se u toku istraživanja mape stalno kreće istom brzinom što je malo brži tempo i to uglavnom ide samo napred-nazad, odnosno levo-desno, pritiskom na dva tastera ili dugmeta (pošto video igra verujem da bi mogla lako da se portuje i na konzole). Usput postoji i taster koji pritiskom izazove lika da skoči u vazduh, što uglavnom služi za istraživanje i ne može se koristiti tokom borbe.

Po uzoru na [7] igrač tokom borbe ima iste opcije napada kao i protivnici i svoj izbor može stalno da menja tokom igre. Protagnosta ima više tela, odnosno više verzija svog dizajna, od kojih svaka pruža igraču drugi tip napada. Svako telo se bira drugim tasterom i svako telo ima jedan taster za napad i jedan za odbranu, a odbrana se vrši u obliku blokiranja, odnosno smanjivanja štete koju bi inače izazvala napadi protivnika i to igrač može da koristi samo dok je nepokretan. Igrač tokom borbe ima iste opcije napada kao i protivnici i svoj izbor može stalno da menja tokom igre. Protagnosta ima više tela, odnosno više verzija svog dizajna, od kojih svaka pruža igraču drugi tip napada. Svako telo se bira drugim tasterom i svako telo ima jedan taster za napad i jedan za odbranu, a odbrana se vrši u obliku blokiranja, odnosno smanjivanja štete koju bi inače izazvala napadi protivnika i to igrač može da koristi samo dok je nepokretan.

4.1. Dizajn nivoa

Nivoi u igri Undead Savior su naizgled vrlo jednostavni: kreći se ka desno dok ne stigneš do kraja nivoa gde onda prelaziš na naredni i usput pobeđi sve protivnike koji ti stoje na putu. Ništa ne sprečava igrača da igra na ovaj način jer bi igrač uvek trebalo da ima što veću slobodu da igra kako njemu/njoj najviše odgovara i tako oblikuje svoje iskustvo. Međutim, ovo je gde skok igra ulogu u dizajnu nivoa: na svakom nivou se nalaze platforme po kojima igrač može da se kreće i to iznad prizemlja. To

može da bude ne samo jedna takva platforma, već na više visinskih nivoa i to još da idu ka desno i ka levo. Igrač se ohrabruje da prolazi kroz sve delove ove recimo skrivene delove mape jer će ovako obično nailaziti na objekte ili likove koji će mu pružati dodatnu pomoć i olakšati prelaženje nivoa, koji bi mu bez dodatnog istraživanja postajali biti možda i preteški što bi više napredovao.

4.1.1. Ostali tipovi NPC likova

Pored običnih protivnika, među drugim NPC likovima na koje igrač može da naiđe je mentor. On stoji nepokretan na određenom delu u većini nivoa i kada igrač interaguje sa njim, kroz korisnički interfejs se prenosi dijalog, koji se uvek sastoji iz više rečenica. Srha mu je prvensteno da pripoveda priču igraču i usput da mu daje savete i upozorenja šta mu sve sledi.

Osim mentora, od prijateljskih NPC likovima igrač može još da naiđe i na pomenutu misterioznu vešticu koja ima ulogu merchant-a, odnosno in-game prodavačice. Preko nje igrač može da kupuje nadogradnje za protagonistu poput većeg života ili veće jačine nekog od svojih oružja/moći.

Poslednji tip NPC likova na koje će igrač nailaziti su bosevi. Ovo su posebni tipovi neprijatelja koji se nalaze krajem nekih nivoa i služe kao glavni izazovi za igrače jer oni, pored raznih dodatnih promena u dizajnu njih i samog okruženja tokom borbe, zahtevaju od igrača da kombinuje sve što je do tada naučio kako bi mogao da ih pobeđi.

5. KORISNIČKI INTERFEJSI

Korisnički interfejs video igre je sistem vizuelnih komponenti koji omogućava igračima da komuniciraju sa pričom igre (narativom) i probiju u prostor igre (četvrti zid). Undead Savior ih koristi u obe svrhe i zato će biti odvojeno opisani interfejs vidljiv kada traje sama igra i interfejsi vidljivi dok je igra pauzirana, odnosno meniji.

5.1. In-game korisnički interfejs

Korisnički interfejs tokom igranja je poprilično jednostavan jer gleda da ne prikazuje igraču previše stvari odjednom i time ga zamara, već samo one delove koji su mu bitni. Tako je predloženo u [9].

Najveći deo ekrana zauzima health bar u gornjem levom uglu i to mu pokazuje igraču koliko je on blizu ili daleko od smrti. Sastoji se od dve boje: jedna za pozadinu i jedna za popunjenost, odnosno koliki procenat života je ostao protagonistu. Pozadina je uvek crne boje jer predstavlja prazninu, a druga boja varira od trenutne dužine života protagoniste. Kada je pun health bar, on je sav svetlo zelen, dok je pri smrti ta boja crvena. Svaki primljen napad od strane protivnika izaziva da se ta boja promeni da bude bliža crvenoj nego zelenoj boji.

Desni deo ekrana zauzima broj duša kojih igrač trenutno ima na sebi i pored toga ikonica objekta koje igraču daju veći broj duša kada ih pokupi. Sama cifra može da bude bele boje, što je difoltna varijanta jer se tada broj predstavlja samo navedeno, ili crvene, što je slučaj kada igrač dozvoli da može da daje delove svog života kao duše kada uzima nadogradnje za protagonistu. U drugom slučaju je cifra zbir navedenog broja kao trenutni život

protagoniste, odnosno menjaće se prilikom svake štete navedene od strane neprijatelja.

Na sredini gornjeg ekrana se vidi tekst koji govori igraču šta će se desiti kada pritisne dugme ili taster koji služi za interakciju sa okruženjem i to samo kada protagonista dodiruje dati objekat ili karaktera. Ništa se neće desiti ako igrač bude to pritisnuo, a ne dodiruje ih jer bi to u suprotnom moglo da izazove više problema. Funkcije koji bi se ovde mogle odigrati su: pokupljenje objekata, otvaranje dijaloga sa drugim likovima, snimanje igre i prelazak na naredni nivo.

Tokom borbe sa bosevima se na dnu ekrana nalazi naziv njihov health bar i njihovo ime, kao na Slici 1 jer je tako predloženo u [8]. Ime služi na naglasi njihov značaj, a health bar im funkcioniše isto kao sa igračevim, osim što je ovde popunjenost sve vreme crvene boje.



Slika 1: Protagonista sa jednim od boseva

5.2. Meniji

Glavni meni služi kako bi igrač podesio sebi idealno okruženje za igranje, uključujući kvalitet grafike i da li će igra da zauzima čitav ekran ili ne (naravno samo ukoliko je kompjuter platforma na kojoj se igra).

Meni za pauze tokom igre ima još jednostavniji dizajn i to prvenstveno zato što on ima manje funkcionalnosti. Pored pauziranja toka igre, jedina od drugih funkcionalnosti je da se igrač vrati na početni meni, što naravno podrazumeva da neće biti sačuvan progres koji je igrač napravio od trenutka kada je poslednji put snimio igru. Poslednja funkcionalnost je da prosto izađe iz same igre, jer bi bilo suvišno kada bi igrač morao da prođe kroz dva interfejsa kako bi mogao da prekine igru.

6. MODELOVANJE GLAVNOG LIKA

Protagonista nema građu tela kakvu glavni junaci uglavnom imaju u akcionim video igrama, već na prvi pogled bi možda čak i delovao kao jedan od običnih protivnika sa kojima bi se igrač inače sreo na nekom početnom nivou. Ovo je zato što ti drugi protagonisti često u kontekstu radnje nemaju opravdan razlog zašto se muče protiv daleko slabijih protivnika nego onih sa kojima se susreću u kasnijim delovima igre, a promene se uopšte ne odražavaju na njihovim izgledima koji ostaju nepromenjeni. Protagonista ovde izgleda kao neko ko ne bi mogao da pobedi sve izazove na koje će naići tokom igre i zato kada to uspe da postigne, igrač ima više osećaj postignuća. Prikazan je na Slici 2.

Prateći [12] odeća protagnosite je sive boje, pošto on ipak jeste deo ovog mračnog sveta a i ima mračniju stranu u sebi koju treba da koristi kako bih mogao da izvrši svoj zadatak. Njegov šal je crne boje i namerno je to manji deo odeće jer se time ipak ističe da je ta mračna strana mali deo njega i da može on da je na kraju savlada i ne padne u iskušenje. Boja kože mu je plava jer ona predstavlja nadu.



Slika 2: Jedan od modela protagoniste

7. UPOTREBA MULTIMEDIJE

Glavni cilj upotrebe multimedije u video igrama je povećanje igračeve imersivnosti. Jedan od načina na koji Undead Savior to omogućuje je kroz razne zvukove koji mogu da se čuju kao što su muzika u pozadini (u toku ili van borbe) ili zvučni efekti za određene poteze napada ili odbrane od strane protagoniste ili nekog od neprijatelja. Osim zvukova upotrebljene su animacije pokreta likova kako bi se vizuelno prikazali likovi i da se istaknu njihove najbitnije karakteristike. Pored pokreta, u istu svrhu su bile upotrebljene i boje likova, kao i samog okruženja. Pored njegovih nepromenljivih boja, protagonista ima više tetovaža i oznaka na svome šalu čije boje se menjaju u zavisnosti od tela (oružja ili moći) sa kojim igrač trenutno igra. Ove boje su jačeg intenziteta kako bi mogle da se istaknu u odnosu na plavu kožu protagoniste.

7.1. Audio

Po uzoru na [10] se pušta neka muzika sve vreme i to ukupno ima tri tipa muzike: u toku borbe, van borbe, dok se nalazi u početnom meniju, i tokom borbe sa bosem. Na taj način se obezbeđuje da igrač sve vreme može preko muzike da se više investira u proces igranja. U toku borbe se pušta brža i glasnija muzika koja služi da prikaže opasnost koju neki protivnik (ili protivnici) predstavlja igraču dok ih on ne pobedi kao i veću brzinu njegovih napada na protivnika. Van borbe se čuje muzika koja je opuštajuća, ali takođe i melanholična, jer se tako najbolje prenosi igraču da nije trenutno u opasnosti, ali ni da nikada ne može da računa na to da je on 100% bezbedan čime da trebalo da doživljava veću imersivnost u toku igranja.

Dok pozadinska muzika služi da prenese igraču atmosferu u kojoj se trenutno nalazi, zvukovi koji proizvode likovi sa svojim akcijama (obično napad ili odbrana) služe da dodaju veću težinu tim akcijama čime će se one osećati bitnijim. Ovde se pojave zvuci povodom narednih akcija: napad igrača, uspešno ili neuspešno odbranjen napad od strane protivnika, smrt nekog od likova video igre ili

pokupljen objekat. Dakle, većina zvukova za akcije likova su zvukovi koje proizvodi protagonista ili efekti koje protivnici vrše na njemu, jer je protagonista, u kontekstu priče, ratnik vraćen iz mrtvih sa zadatkom da postane novi vladar zemlje što može da postigne samo nakon što samostalno uspe da ubije ogroman broj protivnika.

7.2. Animacije likova

Likovi u igri Undead Savior imaju 5 tipova animacija: mirovanje, kretanje, napad, odbrana i smrt. U suštini, ni u jednom trenutku neće postojati lik nad kojim se ne izvršava nikakva animacija. U suprotnom bi bili viđeni kao neživi objekti i time bi imersivnost bila uveliko smanjena za igrača, na šta upozorava [11]. Animacije u toku mirovanja su sitne, u trenucima možda čak i neprimetne, ali to je zato što animacije ni ne treba da budu prevelike ako neki lik samo treba da stoji u mestu. U tim slučajevima likovi prave samo sitne pokrete kako igrač ne bi imao osećaj da su samo objekti, ali ujedno ni da se ne kreću, jer te animacije trebaju da imaju veći i očigledniji pokreti kako bih bolje privukli pažnju igrača.

8. POZADINSKI KOD

Pored svih elemenata multimedije, radnje i pravila kako video igra funkcioniše ključan deo u pravljenju ove igre je bio i pisanje koda, jer je njegova glavna svrha da primi sva naređenja od korisnika i prenese ih u samoj igri, bilo to pokretanje protagoniste ili nameštanje bilo kakve primene u podešavanjima igre. Pored toga je kod bitan za sve elemente igre koji se menjaju dinamično, poput boje osvetljenja pozadine, ponašanje protivnika u zavisnosti od daljine protagoniste ili trenutne veličine njihovog života (u slučaju boseva), eliminisanje objekata nakon što ih igrač pokupi i naravno snimanje trenutnog progressa igrača.

9. ZAKLJUČAK

Glavno pitanje postavljeno pre početka razvoja igre Undead Savior je bilo: Da li je moguće da 2D video igra ima dobar gameplay uz minimalnu upotrebu skoka. Skakanje je vrlo često glavna akcija u 2D video igrama i ovde se gledalo da se pita da li je zato što tako mora ili prosto niko još nije pokušao da uradi suprotno.

Kao zamena za skakanje se javlja promena atmosfere kao dobar način da se igračovo iskustvo oseća jednako uzbudljivo i dinamično kao i u drugim 2D igrama. Zato se ujedno koristi veliki broj skripti: kako bi se stalno mogli menjati elementi prikazani na ekranu, uključujući likove, objekte, boju pozadine, itd.

Borbe su područje gde se najviše ispoljava mračnija atmosfera igre, jer bi trebalo da bude vrlo retko da se desi jedna gde igrač uspe da izbegne svaki udarac protivnika. Iz istog razloga bliokiranje napada samo ublažava njihov uticaj ali ga ne poništava. Svaki susret bi trebalo da izaziva napetost kod igrača, doprinoseći time imersivnosti igre.

Pošto je ciljna publika ta koja polako ulazi u zrelost, gleda se da što više razvija njihove sposobnosti, bilo da to bili brzi refleksi i smišljanje strategije tokom borbi ili donošenje odluka o potrošnji novca u igri. Priča je isto tako jedna koja neće igračima biti sva otvoreno rečena,

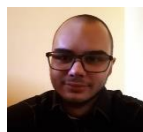
već se očekuje da se u njima probudi zainteresovanost čime će je sami istraživati u samoj igri.

Undead Savior pokušava da koristi razne elemente multimedije, scenografije kao i na takav način da pozajmljuje ideje od drugih naslova sa ciljem da koristi proverene metode radi veće garancije uspeha, ali istovremeno gleda da uvede dovoljno novine kako bi i dalje mogla da se istakne u odnosu na svoje inspiracije. Video igra ni u kojem slučaju nije gotova, ali svakako smatram da ostavlja dovoljno prostora za poboljšanje i da će jednog dana moći da bude uporediva sa drugim video igrama sličnih žanrova.

10. LITERATURA

- [1] <https://unity.com/>
- [2] <https://assetstore.unity.com>
- [3] <https://krita.org/en/>
- [4] <https://pixabay.com/sound-effects/search/storm/>
- [5] <https://audiotrimmer.com>
- [6] <https://stonesoup.com/post/video-games-and-the-power-of-storytelling/> (pristupljeno u martu 2024.)
- [7] <https://book.leveldesignbook.com/process/combat> (pristupljeno u maju 2024.)
- [8] <https://yukaichou.com/advanced-gamification/game-design-technique-14-mastering-the-boss-fight/> (pristupljeno u martu 2024.)
- [9] <https://www.sketch.com/blog/game-ui-design/> (pristupljeno u julu 2024.)
- [10] <https://www.readspeaker.com/blog/audio-in-video-games/> (pristupljeno u avgustu 2024.)
- [11] <https://www.torontofilmschool.ca/blog/what-is-video-game-animation-and-how-does-it-work/#:~:text=In%20simple%20terms%2C%20video%20game,by%20D%20and%203D%20artists.> (pristupljeno u avgustu 2024.)
- [12] <https://www.gamedeveloper.com/design/color-in-games-an-in-depth-look-at-one-of-game-design-s-most-useful-tools> (pristupljeno u aprilu 2024.)

Kratka biografija:



Jovan Svorcan rođen je 31. marta 1998. godine u Novom Sadu. Završio je Osnovnu školu „Ivan Milutinović“ u Subotici 2013. godine odličnim uspehom. Gimnaziju „Svetozar Marković“ u Subotici završava 2017. godine. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu 2021. godine, na smeru Softversko inženjerstvo i informacione tehnologije.

TOK SNAGE IZMEĐU KANALA IZOLOVANOG VIŠEKANALNOG PRETVARAČA **CHANNEL POWER EXCHANGE IN ISOLATED MULTI-PORT CONVERTERS**

Vukašin Bukvić, Ivan Todorović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Rad se bavi naprednim pretvaračima energetske elektronike, koji će predstavljati značajnu gradivnu jedinicu modernih elektroenergetskih sistema. Višekanalni jednosmerni pretvarači omogućavaju jednostavnu integraciju više izvora električne energije, skladišnih sistema i potrošača. Ovi pretvarači, realizovani sa galvanskom izolacijom, pružaju i druge značajne funkcionalnosti (povećana sigurnost i pouzdanost). U radu je prikazan detaljan analitički postupak dobijanja opšte formule za razmenu snage između bilo koja dva kanala pretvarača. Ovaj izraz je od velike važnosti kako za razumevanje rada pretvarača, tako i za razvoj upravljačkih struktura, koje upravljaju tokom snage u pretvaraču, a samim tim i tokom snage u sistemu kojem pretvarač pripada.

Ključne reči: Obnovljivi izvori energije, višekanalni pretvarači, tok snage

Abstract The paper deals with advanced power electronics converters, which will represent a significant building block of modern power systems. Multiport dc-dc converters enable the simple integration of multiple energy sources, storage systems, and consumers. These converters, implemented with galvanic isolation, also provide other important functionalities (increased safety and reliability). The paper presents a detailed analytical procedure for deriving a general formula for power exchange between any two channels of the converter. This expression is of great importance both for understanding the operation of the converter and for the development of control structures, which manage the power flow in the converter and consequently the power flow in the system to which the converter belongs.

Keywords: Renewable sources of energy, multiport converters, power flow

1. UVOD

Konvencionalne distributivne elektroenergetske mreže, koje napajaju električnom energijom veliki broj potrošača i služe kao glavna energetska infrastruktura, čime se u velikoj meri omogućava svakodnevni život ljudi, prilikom rada moraju da obezbede stabilnosti napajanja, pouzdanost i robusnost (na spoljašnje činioce u vidu vremenskih prilika ili prirodnih katastrofa) u radu [1].

NAPOMENA:

Ovaj rad proistekao je iz master rada, čiji mentor je bio dr Ivan Todorović, vanr. prof.

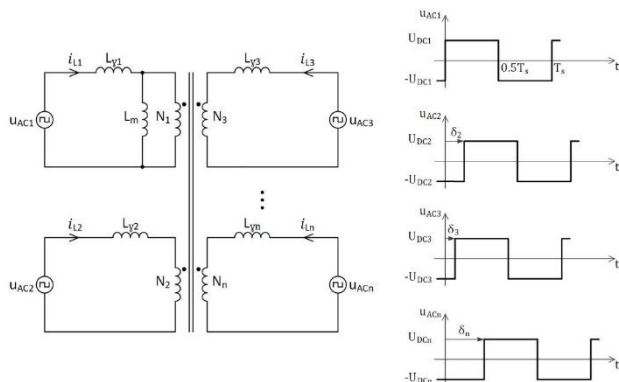
U svrhu smanjenja emisije štetnih gasova i energetske održivosti, u sve većoj meri koriste se obnovljivi izvori energije i energetska skladišta, te se ovi sistemi integrišu u već postojeće elektroenergetske mreže. Kako obnovljive izvore energije karakterišu brze i stalne promene u količini proizvodnje električne energije usled uticaja vremenskih prilika (količina vetra, vode, osvetljaja, itd.), javljaju se novi tehnički problemi balansiranja proizvodnje i potrošnje i regulacije napona na mestima na kojima su priključeni pomenuti izvori. Višak snage predstavlja velik problem sa kojim se konvencionalna mreža bori tako što smanjuje proizvodnju električne energije. Međutim, tada se ne koristi pun potencijal elektrane, u smislu moguće maksimalne proizvedene energije, te se u neku ruku ona može smatrati izgubljenom.

Sa razvojem obnovljivih izvora energije, došao je i razvoj novih konfiguracija pretvarača energetske elektronike. Većina novoistraženih izvora i energetske skladišta su po prirodi jednosmerni, što DC mreže malih ili velikih razmera postavlja na visoko mesto kandidata za buduću arhitekturu elektroenergetskih distributivnih mreža. Fleksibilnost i jednostavnost primene komponenata energetske elektronike, kao i korišćenje odgovarajućih postojećih naprednih zaštitnih tehnologija, predstavlja osnovne razloge koji favorizuju sve učestalije korišćenje modernih mreža [2].

Čitav niz pretvarača je predložen za integraciju obnovljivih izvora i skladišnih sistema u poslednjih nekoliko decenija. Jedan od najpoznatijih je dvostrani aktivni most (DAB – Dual Active Bridge). Ipak, u svetlu nedostataka koje poseduje (ograničenosti integracionih kapaciteta), u skorije vreme dolazi do razvoja višekanalnih pretvarača energetske elektronike. Veći broj kanala omogućuje raznovrsniji tok snaga između kanala pretvarača, odnosno integrisanih sistema koji su na kanale pretvarača povezani. Sa druge strane, upravljanje se višestruko usložnjava sa povećanjem broja kanala, u poređenju sa relativno jednostavnim upravljanjem kod dvostranog aktivnog mosta. Stoga je sledeće poglavlje posvećeno pravo izvođenju izraza na kojima se bazira upravljanje tokom snage između kanala razmatranog pretvarača.

2. ANALIZA TOKOVA SNAGA

U okviru rada, pristup analizi razmene snaga između kanala započet je pojednostavljenjima vezanim za električno kolo u potpunosti izolovanog višekanalnog pretvarača, a zatim nastavljen izvođenjem izraza za razmenjenu snagu između bilo koja dva kanala višekanalnog pretvarača. Polazeći od poznatih definicionih obrazaca koji opisuju prostiranje struje kroz prigušnicu



Slika 1. Zamenska predstava višekanalnog pretvarača i odzivi napona kanala.

usled nametnutog napona na njenim krajevima, a implementirajući odgovarajuće matematičke operacije, došlo se do konačnog oblika jednačine za proračun razmenjene snage između dva kanala. Rezultat sam po sebi nije od velikog značaja, međutim iskorišćen je za dobijanje uopštene formule kojom se računa snaga razmenjena između proizvoljna dva kanala, a koja će važiti za pretvarač sa arbitrarnim brojem kanala.

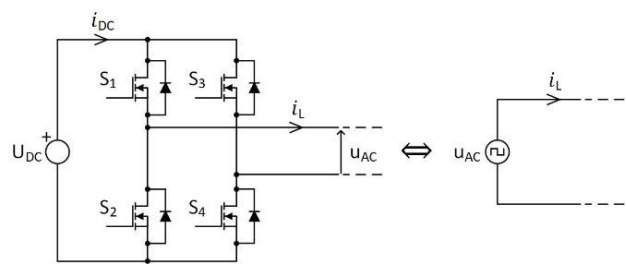
Niz jednačina i šema priloženi su kako bi se oformio što jednostavniji model izolovanog višekanalnog pretvarača. Rezultat jeste pretvarač koji se nalazi u otvorenoj sprezi, a koji upravljanjem faznih pomeraja između generisanih signala na krajevima sprežnih induktivnosti utiče na vrednost razmenjene snage. Daljim razvijanjem kontrolnih struktura moguće je ostvariti zatvorenu povratnu spregu po snazi, odnosno regulaciju snage putem mikrokontrolera, koji utiče na vrednosti faznih pomeraja. Upravljanje tokom snage u zatvorenoj petlji nije dalje razmatrano u radu.

Odabrana je topologija višekanalnog pretvarača prikazana na slici 1. Ostvarenje željenog toka snage između kanala postiže se posredstvom induktivnosti rasipanja namotaja visokofrekventnog transformatora kao medijuma za prenos. Svakom od kanala pripada tranzistorski pretvarač u vidu H-mosta, čiji je zadatak da proizvede impulse napona faktora ispune 50%, što se i postiže pravougaonom modulacijom, odnosno generisanjem upravljačkih signala za poluprovodničke elemente. Promenom faznog stava napona na izlazu iz pretvarača svakog od kanala, a samim tim i faznog pomeraja između talasnih oblika napona na izlazu iz pretvarača, utiče se na vrednost razmenjene snage između kanala.

2.1. Izvođenje pojednostavljene šeme za analizu tokova snaga

Ako se na svaki H-most primeni pravougaona modulacija, opravdano je iste ekvivalentirati sa idealnim naponskim izvorima signala pravougaonih talasnih oblika (slika 2). Pod pravougaonom modulacijom podrazumeva se upravljanje stanjima prekidača tako da su istovremeno aktivni ili isključeni dijagonalni poluprovodnički uređaji S_1 i S_4 , odnosno S_2 i S_3 i to polovinu prekidačkog perioda. Rezultat jeste generisanje naizmeničnih visokofrekventnih pravougaonih signala naponskih nivoa određenih od strane jednosmernog kola (U_{DC}), koji se dalje nameću krajevima namotaja transformatora.

Na slici 3 prikazan je pojednostavljen u potpunosti izolovan višekanalni pretvarač sa n kanala, gde su



Slika 2. Zamenska predstava H-mosta.

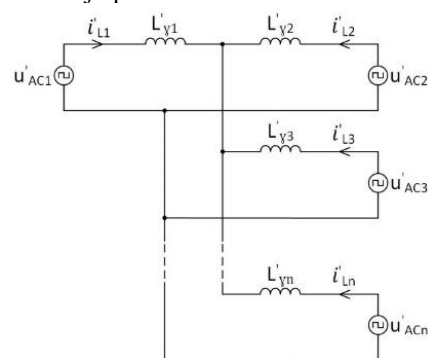
H-mostovi zamenjeni sa idealnim naponskim generatorima pravougaonih signala, a visokofrekventni višenamotajni transformator preko rasipnih induktivnosti namotaja $L_{Y1}, L_{Y2}, L_{Y3}, \dots, L_{Yn}$, broja navojaka namotaja $N_1, N_2, N_3, \dots, N_n$ i induktivnosti magnećenja L_m . Rezultat pravougaone modulacije jesu priloženi talasni oblici koji polovinu vremena prekidačkog perioda T_s iznose pozitivnu vrednost pripadajućih napona jednosmernog kola ili izvora kanala, dok drugu polovinu trenutna vrednost napona poprima istu apsolutnu vrednost sa promenjenim predznakom. Pored spomenutog, ovim načinom upravljanja stanjima uključenosti prekidača realizuje se i fazni pomeraj između signala generisanih na različitim kanalima čime se postiže željeni kontrolisani tok snage između kanala. Sa δ je označen fazni pomeraj svakog signala, a sa indeksom je označen kanal na koji se taj fazni pomeraj odnosi. Signal koji se generiše na prvom kanalu uzet je kao referentni. Fazni pomeraj bilo kog signala se posmatra u odnosu na uzlaznu ivicu referentnog. Prema tome, δ_1 nije obeležen na slici jer će uvek iznositi nula, dok su redom obeleženi $\delta_2, \delta_3, \dots, \delta_n$ kao fazni pomeraji generisanih talasnih oblika na drugom, trećem i tako redom do n -tog kanala respektivno. Fazni pomeraji se kreću u opsegu $[-\pi, \pi]$.

U nastavku se ekvivalentna šema pretvarača dodatno pojednostavljuje prebacivanjem veličina i parametara, pomoću odgovarajućih transformacijskih odnosa za svaki od kanala, na primarnu stranu transformatora, čime se eliminišu transformatori iz električnog kola (slika 4). Transformacijski odnosi za transformisanje veličina na primarnu stranu su sledeći:

$$n_2 = \frac{N_1}{N_2}, n_3 = \frac{N_1}{N_3}, \dots, n_n = \frac{N_1}{N_n}, \quad (1)$$

gde se sa n_2 transformišu veličine i parametri drugog kanala, sa n_3 trećeg, i tako redom sve do veličina i parametara n -tog kanala.

Na osnovu definisanih prenosnih odnosa (1) sada se može pristupiti svođenju parametara i veličina na sledeći način:



Slika 3 Uprošćena šema u izolovanog višekanalnog pretvarača, čiji su parametri i veličine svedene na primar transformatora.

$$\begin{aligned}
L'_{\gamma 2} &= n_2^2 L_{\gamma 2}, L'_{\gamma 3} = n_3^2 L_{\gamma 3}, \dots, L'_{\gamma n} = n_n^2 L_{\gamma n}, \\
u'_{AC2} &= n_2 u_{AC2}, u'_{AC3} = n_3 u_{AC3}, \dots, u'_{ACn} = n_n u_{ACn}, \\
i'_{L2} &= i_{L2}/n_2, i'_{L3} = i_{L3}/n_3, \dots, i'_{Ln} = i_{Ln}/n_n, \\
L'_{\gamma 1} &= L_{\gamma 1}, u'_{AC1} = u_{AC1}, i'_{L1} = i_{L1},
\end{aligned} \quad (2)$$

gde su $L'_{\gamma 2}, L'_{\gamma 3}, \dots, L'_{\gamma n}$ induktivnosti rasipanja, $u'_{AC2}, u'_{AC3}, \dots, u'_{ACn}$ naponi idealnih naponskih generatora pravougaonih talasnih oblika i $i'_{L2}, i'_{L3}, \dots, i'_{Ln}$ struje kroz namotaje transformatora svedene na primarnu stranu, dok veličine i parametri na naponskom nivou primara ostaju jednakih vrednosti.

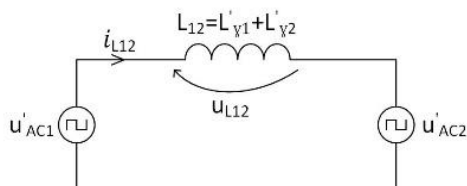
2.2 Razmena snage između dva kanala

Za izvođenje analitičkih izraza za prenos snage u višekanalnom pretvaraču, kao polazna tačka koristi se izraz za dvokanalni pretvarač. Ovakva konfiguracija se postiže tako što se svi kanali osim dva ostave u praznom hodu, čime se unutar višekanalnog izdvoji dvokanalni pretvarač. Bez obzira na to koja dva kanala su odabrana da čine novonastali dvokanalni pretvarač, u nastavku će veličine i parametri koji opisuju prvi biti označeni sa indeksom jedan, dok će oni vezani za drugi biti označeni sa indeksom dva. To znači da sada indeksi neće označavati redni broj originalnih kanala višekanalnog pretvarača, jer je moguće dvokanalni dobiti od bilo koje kombinacije kanala višekanalnog.

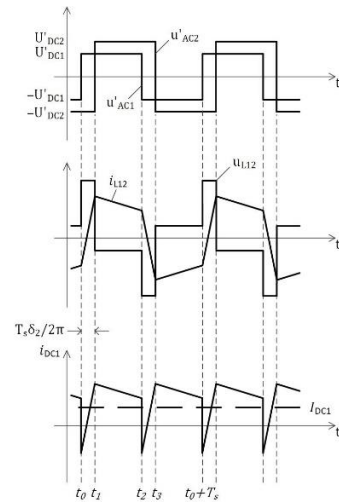
Uprošćeno električno kolo dvokanalnog pretvarača (slika 4), kod koga su sve veličine i parametri svedeni na naponski nivo jednog kanala (što je naznačeno apostrofima) pomoću jednačina (2), se sastoji od dva idealna naponska generatora pravougaonih signala spregnutih putem induktivnosti L_{12} koja predstavlja sumu rasipnih induktivnosti kanala višekanalnog pretvarača koji nisu ostavljeni u praznom hodu. Sada je jasno da sprežna sumarna induktivnost igra glavnu ulogu u prenosu aktivne snage između dva kanala.

Razlika napona $u_{L12} = u'_{AC1} - u'_{AC2}$ diktira talasni oblik struje induktivnosti i_{L12} . Kada pretvarač dostigne stabilno odnosno stacionarno stanje, tada je struja induktivnosti i_{L12} naizmenična veličina. Kontrolom prekidača mosnog pretvarača na primarnoj i sekundarnoj strani, ova naizmenična struja se prevodi u jednosmerne struje kanala (struja i_{DC} naznačena na slici 5).

Imajući na umu da pretvarač svakog kanala generiše naponski signal pravougaonog talasnog oblika sa fiksnim faktorom ispunje od 50%, uz uvažanje mogućih različitih naponskih nivoa u jednosmernom kolu svakog kanala i primenom drugog Kirhofovog zakona i definicionog obrasca za struju koja protekne kroz prigušnicu usled nametnutog napona na njenim krajevima, talasni oblik struje kroz rezultantnu sprežnu induktivnost između dva kanala ilustrovan je na slici 5. Prikazani odzivi važe u slučaju kada napon u'_{AC1} prednjači naponu u'_{AC2} za ugao δ_2 .



Slika 4. Uprošćena šema novoformiranog dvokanalnog pretvarača za izučavanje prenosa snage između dva kanala



Slika 5. Idealizovani talasni oblici struja i napona dvokanalnog pretvarača.

U nastavku se pristupa izvođenju analitičkog izraza struje prigušnice, odnosno dobijanju izraza za vrednosti koje struja poseduje u karakterističnim vremenima prekidanja kao i vremenske zavisnosti struje ovisno od odabranog intervala.

$$\begin{aligned}
u'_{AC1}(t) - u'_{AC2}(t) &= L_{12} \frac{di_{L12}(t)}{dt} \\
i_{L12}(t) &= \frac{u'_{AC1}(t) - u'_{AC2}(t)}{L_{12}}(t - t_i) + i_{L12}(t_i); t_i \leq t \leq t_j;
\end{aligned} \quad (3)$$

gde su t_i i t_j početno i krajnje vreme posmatranog perioda prekidanja, $i_{L12}(t_i)$ vrednost struje između dva kanala u početnom trenutku datog perioda, L_{12} sprežna induktivnost između dva kanala, $u'_{AC1}(t)$ i $u'_{AC2}(t)$ naponski signali na izlazu pretvarača za oba kanala.

○ Za $t_1 \leq t \leq t_2$:

$$i_{L12}(t) = \frac{U'_{DC1} - U'_{DC2}}{L_{12}}(t - t_1) + i_{L12}(t_1) \quad (4)$$

gde je $t_2 = t_0 + \frac{T_s}{2}$.

○ Za $t_2 \leq t \leq t_3$:

$$i_{L12}(t) = \frac{-U'_{DC1} - U'_{DC2}}{L_{12}}(t - t_2) + i_{L12}(t_2) \quad (5)$$

gde je $t_3 = t_2 + \frac{\delta_2 T_s}{2\pi}$.

○ Za $t_3 \leq t \leq t_0 + T_s$:

$$i_{L12}(t) = \frac{-U'_{DC1} + U'_{DC2}}{L_{12}}(t - t_3) + i_{L12}(t_3) \quad (6)$$

Zbog simetrije talasnog oblika $i_{L12}(t)$ sledi:

$$i_{L12}(t_2) = i_{L12}\left(t_0 + \frac{T_s}{2}\right) = -i_{L12}(t_0), \quad (7)$$

$$i_{L12}(t_3) = i_{L12}\left(t_1 + \frac{T_s}{2}\right) = -i_{L12}(t_1) \quad (8)$$

Na osnovu izraza (4), (7) i vremena prekidanja sledi:

$$i_{L12}(t_0) = -i_{L12}(t_2) = -\frac{(U'_{DC1} - U'_{DC2})(\pi - \delta_2)}{2\pi L_{12} f_{sw}} - i_{L12}(t_1) \quad (9)$$

gde je $f_{sw} = 1/T_s$ prekidačka frekvencija pretvarača.

Na osnovu relacija (5), (8) i vremena prekidanja sledi:

$$i_{L_{12}}(t_1) = -i_{L_{12}}(t_3) = \frac{U'_{DC1} + U'_{DC2}}{2\pi L_{12}f_{sw}} \delta_2 - i_{L_{12}}(t_2) \quad (10)$$

Sabiranjem jednačina (9) i (10) dobija se:

$$i_{L_{12}}(t_1) = -i_{L_{12}}(t_3) = \frac{(U'_{DC2} - U'_{DC1})\pi + 2U'_{DC1}\delta_2}{4\pi L_{12}f_{sw}} \quad (11)$$

Zamena (11) u (9) rezultuje u:

$$i_{L_{12}}(t_0) = -i_{L_{12}}(t_2) = -\frac{(U'_{DC1} - U'_{DC2})\pi + 2U'_{DC2}\delta_2}{4\pi L_{12}f_{sw}} \quad (12)$$

Sada se na osnovu dobijenih jednačina konačno može pristupiti izvođenju izraza za razmenjenu snagu između bilo koja dva kanala uz zanemarenje gubitaka pretvarača, počevši od definicionog integrala za pronalaženje srednje vrednosti koji se primenjuje na trenutnu vrednost snage:

$$P_{12} = \frac{1}{T_s} \int_0^{T_s} u_{DC1}(t) i_{DC1}(t) dt = \frac{1}{T_s} \int_0^{T_s} u_{DC2}(t) i_{DC2}(t) dt \quad (13)$$

gde je P_{12} srednja snaga koja se razmenjuje između prvog i drugog kanala, $u_{DC}(t)$ trenutna vrednost napona koji se ima na jednosmernoj strani pretvarača u datim granicama i $i_{DC}(t)$ trenutna vrednost struje na jednosmernoj strani.

$$\begin{aligned} P_{12} &= U'_{DC1} \frac{1}{T_s} \int_{t_0}^{t_0+T_s} i_{DC1}(t) dt \\ &= U'_{DC1} \frac{2}{T_s} \left(\int_{t_0}^{t_1} i_{L_{12}}(t) dt + \int_{t_1}^{t_2} i_{L_{12}}(t) dt \right) \\ &= U'_{DC1} \frac{2}{T_s} \frac{U'_{DC2}(2\pi\delta_2 - 2\delta_2^2)}{8\pi^2 L_{12}f_{sw}^2} \\ &= \frac{\delta_2(\pi - \delta_2)U'_{DC1}U'_{DC2}}{2\pi^2 L_{12}f_{sw}} \quad (14) \end{aligned}$$

U prethodnim razmatranjima, smatra se da je fazor napona kojeg generiše prvi kanal poklopljen sa referentnom osom, što rezultuje nultoj faznoj smaknutosti ($\delta_1 = 0$). U opštem slučaju postoji fazna smaknutost oba kanala u odnosu na referentni, te se stoga prirodno nameće nadogradnja izraza (14) na sledeći način:

$$P_{xy} = \frac{(\delta_y - \delta_x)(\pi - |\delta_y - \delta_x|)U'_{DCx}U'_{DCy}}{2\pi^2 L_{xy}f_{sw}} \quad (15)$$

gde su $x, y = 1, 2, 3, \dots, n$; indeksi kanala, a n ukupan broj kanala višekanalnog pretvarača, P_{xy} razmenjena snaga između dva određena kanala, δ_x, δ_y fazni pomeraji naponskih signala na izlazu iz pretvarača kanala x i y respektivno, U'_{DCx}, U'_{DCy} vrednosti napona jednosmernog izvora kanala x i y svedene na primar visokofrekventnog višenamotajnog transformatora, L_{xy} sprežna induktivnost između kanala x i y , f_{sw} frekvencija prekidanja poluprovodničkih elemenata H-mostova.

Jednačina (15) predstavlja konačan opšti izraz za proračun vrednosti razmenjene snage između bilo koja dva kanala u potpunosti izolovanog višekanalnog pretvarača. Na osnovu izraza se stiče uvid na to da isključivo vrednosti faznih pomeraja generisanih napona na izlazu iz pretvarača određuju smer toka snage između kanala, ali i količinu razmenjene snage.

Ovim izrazom se omogućava korišćenje kapaciteta izvora i skladišta električne energije, a spram potreba samog elektroenergetskog sistema na koji pretvarač može biti povezan (jedan kanal pretvarača je povezan na distributivnu mrežu, posredstvom još jednog pretvarača).

Dodatno, kako energija može da se razmenjuje između bilo kojih kanala, omogućen je i nezavisan rad, bez prisustva električne mreže.

3. ZAKLJUČAK

S obzirom na prednosti koje pružaju moderne električne mreže sa integrisanim obnovljivim izvorima i skladišnim sistemima, poželjno je omogućiti njihov stabilan i pouzdan rad. Ovo je moguće ostvariti kroz razvijanje novih topologija pretvarača energetske elektronike i konstruisanje novih sistema za pretvaranje energije. Topologija pretvarača koja iziskuje posebnu pažnju u ovom kontekstu jeste izolovani višekanalni jednosmerni pretvarač. Ovaj pretvarač, uz pravilno razvijenu kontrolu, doprinosi željenom toku snage između priključaka. U radu je pokazano da pravougaono modulirani H-mostovi mogu da generišu smaknute pravougaone talasne oblike na svojim izlazima, kojima se realizuje željena razmena snage. Konkretno, kontrolom faznog pomeraja između pravougaonih signala se utiče na vrednost razmenjene snage i smer toka snage. Izvedena je opšta jednačina za izračunavanje količine razmenjene snage između proizvoljnih kanala, u funkciji faznih pomeraja.

4. LITERATURA

- [1] Y. Kado, D. Shichijo, K. Wada, and K. Iwatsuki, 'Multiport power router and its impact on future smart grids', *Radio Science*, vol. 51, no. 7, pp. 1234–1246, Jul. 2016, doi: 10.1002/2016RS006041.
- [2] Y.-K. Tran and D. Dujic, 'A multiport isolated DC-DC converter', in 2016 IEEE Applied Power Electronics Conference and Exposition (APEC), Long Beach, CA: IEEE, Mar. 2016, pp. 156–162. doi: 10.1109/APEC.2016.7467867.
- [3] H. Wu, P. Xu, H. Hu, Z. Zhou, and Y. Xing, 'Multiport Converters Based on Integration of Full-Bridge and Bidirectional DC-DC Topologies for Renewable Generation Systems', *IEEE Trans. Ind. Electron.*, vol. 61, no. 2, pp. 856–869, Feb. 2014, doi: 10.1109/TIE.2013.2254096.
- [4] H. Tao, A. Kotsopoulos, J. L. Duarte, and M. A. M. Hendrix, 'Family of multiport bidirectional DC-DC converters', *IEE Proc., Electr. Power Appl.*, vol. 153, no. 3, p. 451, 2006, doi: 10.1049/ip-epa:20050362.
- [5] N. Swaminathan and Y. Cao, 'An Overview of High-Conversion High-Voltage DC-DC Converters for Electrified Aviation Power Distribution System', *IEEE Trans. Transp. Electrific.*, vol. 6, no. 4, pp. 1740–1754, Dec. 2020, doi: 10.1109/TTE.2020.3009152.
- [6] H. Tao, A. Kotsopoulos, J. L. Duarte, and M. A. M. Hendrix, 'A Soft-Switched Three-Port Bidirectional Converter for Fuel Cell and Supercapacitor Applications', in *IEEE 36th Conference on Power Electronics Specialists, 2005.*, Aachen, Germany: IEEE, 2005, pp. 2487–2493. doi: 10.1109/PESC.2005.1581982.

Kratka biografija:

Vukašin Bukvić rođen je u Novom Sadu 2000. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine, odbranio je 2024.god.

Ivan Todorović radi na Fakultetu tehničkih nauka od 2014. godine. Trenutno je u zvanju vanrednog profesora. Oblast istraživanja su mu pretvarači energetske elektronike.

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2023. godine učestvovali su sledeći recenzenti:

Aco Antić	Dunja Vrbaški	Marinko Maslarić	Nemanja Tasić
Aleksandar	Đorđe Vukelić	Marko Marković	Nenad Simeunović
Anđelković	Goran Jeftenić	Marko Todorov	Nikola Vojnović
Aleksandar Kovačević	Goran Savić	Marko Vekić	Platon Sovilj
Aleksandar	Goran Sladić	Maša Bukurov	Predrag Teodorović
Kupusinac	Goran Stojanović	Mijodrag Milošević	Radivoje Dinulović
Aleksandar Selakov	Goran Tepić	Milan Delić	Radomir Kojić
Aleksandra Radulović	Gordan Stojić	Milan Gavrić	Romana Bošković-
Aleksandra Pešterac	Gordana Ostojić	Milan Marinković	Živanović
Ana Nešić Tomašević	Gordana	Milan Mirković	Sandra Dedijer
Andraš Anderla	Milosavljević	Milan Rapajić	Saša Medić
Andrija Rašeta	Igor Dejanović	Milan Rackov	Slavica Mitrović
Atila Zelić	Igor Maraš	Milan Segedinac	Senka Bajić
Bojan Batinić	Igor Peško	Milan Trivunić	Slađana Milićević
Bojan Matić	Ilija Bašićević	Milan Vidaković	Slobodan Morača
Bojan Tepavčević	Iva Šiđanin	Milena Krklješ	Slobodan Šupić
Bojan Jovanović	Ivan Mezei	Milica Vračarić	Slobodan Tabaković
Boris Agarski	Ivan Prokić	Milica Miličić	Srđan Popov
Boris Stojić	Ivana Mihajlović	Milica Kisić	Srđan Vukmirović
Branko Milosavljević	Ivana Vasiljević	Miloš Simić	Stevan Gostojić
Damir Đaković	Ivana Katić	Miloš Šešlija	Stevan Milisavljević
Danijela Ćirić	Ivana Maraš	Milovan Lazarević	Stevan Stankovski
Danijela Gračanin	Ivana Miškeljin	Milja Simeunović	Suzana Draganić
Danijela Lalić	Jelena Atanacković	Miljana Prica	Svetlana Bačkalić
Darko Čapko	Jeličić	Miodrag Milutinov	Svetlana Nikoličić
Darko Stefanović	Jelena Borocki	Miodrag Žigić	Tamara Škorić
Dejan Ecet	Jelena Ivetić	Miroslav Dramićanin	Teodora Vučković
Dejan Reljić	Jelena Radonić	Miroslav Zarić	Vesna Stojaković
Dejan Movrin	Jelena Slivka	Mirko Raković	Višnja Žugić
Dejan Ubavin	Jelena Spajić	Miro Govedarica	Vladimir Đaković
Dejana Nedučin	Kalman Babković	Miroslav Kljajić	Vladimir Mučenski
Dragan Adamović	Lazar Kovačević	Miroslav Zarić	Vojin Ilić
Dragan Dinu	Lidija Krstanović	Mladen Tomić	Vuk Bogdanović
Dragan Ivanović	Ljiljana Popović	Mladen Radišić	Vuk Vranjkovic
Dragan Ivetić	Ljubica Duđak	Nataša Milosavljević	Zoran Brujić
Dragan Jovanović	Ljubo Nedović	Nebojša Brkljač	Zoran Čepić
Dragan Pejić	Ljubomir Budinski	Nebojša Radović	Zoran Jeličić
Dragan Ružić	Magdolna Pal	Nebojša Ralević	Željen Trpovski
Dragana	Maja Turk Sekulić	Neda Milić Keresteš	Željko Kanović
Konstantinović	Maja Petrović	Nemanja Kašiković	
Dragoljub Šević	Marija Silađi	Nemanja Sremčev	
Drago Žarković			