



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXXVIII

Број: 1/2023

Нови Сад

Едиција: „Техничке науке – Зборници“

Година: XXXVIII

Свеска: 1

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 Балоиш
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 цм. –(Едиција: Техничке науке – зборници)

Месечно

ISSN 0350-428X

COBISS.SR-ID 58627591

ПРЕДГОВОР

Поштовани читаоци,

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

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

Друштво у коме живимо базирано је на знању. Оно претпоставља реорганизацију наставног процеса и увођење читавог низа нових струка, као и квалитетну организацију научног рада. Значајне промене у структури високог образовања, везане за имплементацију Болоњске декларације, усвајање нове и активне улоге студената у процесу образовања и њихово све шире укључивање у стручне и истраживачке пројекте, као и покретање нових мастер и докторских студија, доносе потребу да ови, веома значајни и вредни резултати, постану доступни академској и широј јавности. Оживљавање „Зборника радова Факултета техничких наука“, као јединственог форума за презентацију научних и стручних достигнућа, пре свега студената, обезбеђује услове за доступност ових резултата.

Због тога је Наставно-научно веће ФТН-а одлучило да, од новембра 2008. год. у облику пилот пројекта, а од фебруара 2009. год. као сталну активност, уведе презентацију најважнијих резултата свих мастер радова студената ФТН-а у облику кратког рада у „Зборнику радова Факултета техничких наука“.

Поред студената мастер студија, часопис је отворен и за студенте докторских студија, као и за прилоге аутора са ФТН или ван ФТН-а.

Зборник излази у два облика – електронском на веб сајту ФТН-а (www.ftn.uns.ac.rs) и штампаном, који је пред вама. Обе верзије публикују се сваки месец, у оквиру промоције дипломираних мастера.

У овом броју штампани су радови студената мастер студија, сада већ мастера, који су радове бранили у периоду од 20.09.2022. до 27.10.2022. год., а који се промовишу 27.01.2023. год. То су оригинални прилози студената са главним резултатима њихових мастер радова.

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

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

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

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

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

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

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

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

„Високо место у друштву најбољих“

Уредништво

SADRŽAJ

STRANA

Radovi iz oblasti: Mašinstvo

1. Лука Куцурски, Никола Иланковић, АНАЛИЗА ЗАМОРНОГ ПОНАШАЊА ТРАНСПОРТНЕ ТРАКЕ МЕТОДОМ КОНАЧНИХ ЕЛЕМЕНАТА	1-4
2. Aron Pinter, Nikola Ilanković, ANALIZA NAPONSKOG STANJA KONSTRUKCIJE MODULARNOG TRAKASTOG TRANSPORTERA PRIMENOM METODE KONAČNIH ELEMENATA	5-8
3. Јован Косановић, ХИДРАУЛИЧКИ ПРОРАЧУН И ИЗБОР ОПРЕМЕ У РАСХЛАДНОЈ СТАНИЦИ ИНДУСТРИЈСКОГ ОБЈЕКТА ЗА ПОТРЕБЕ ПРОЦЕСНОГ ХЛАЂЕЊА ПРОИЗВОДНИХ МАШИНА	9-12
4. Мирјана Дабић, Сања Бојић, НИВЕЛИСАЊЕ ПРОЦЕСА МАШИНСКЕ ОБРАДЕ ПРИМЕНОМ HEIJUNKE	13-16
5. Ervin Molnar, ERGONOMSKE KARAKTERISTIKE VILJUŠKARA NA UTICAJ RUKOVAOCA I BEZBEDNOSNI USLOVI KORIŠĆENJA VILJUŠKARA	17-20
6. Nikola Grdić, Borislav Savković, ODREĐIVANJE PARAMETARA REZANJA PRI IZRADI TELA VENTILA VISOKOG PRITISKA	21-24
7. Danijel Đurković, RAZVOJ NANOFLUIDA VODA/ Al_2O_3 PRIMENOM RAČUNARSKE DINAMIKE FLUIDA	25-28

Radovi iz oblasti: Elektrotehnika i računarstvo

1. Boris Deak, Boris Dumnić, PRIMENA CAN I SPI KOMUNIKACIONIH PROTOKOLA PRILIKOM IZVRŠAVANJA INTEGRACIONIH TESTOVA APLIKACIJE INSTRUMENT TABLE	29-32
2. Luka Bosanac, EKSPLOATACIJA BATTERY ENERGY STORAGE-A U DERMS-U	33-36
3. Aleksandar Grahovac, PARALELIZACIJA DFA ALGORITMA ZA OBUKU DUBOKIH NEURONSKIH MREŽA	37-40
4. Tijana Terzić, Luka Strezoski, PRORAČUN NESIMETRIČNIH TOKOVA SNAGA DISTRIBUTIVNIH MREŽA SA DISTRIBUTIVNIM GENERATORIMA ZASNOVANIM NA UREĐAJIMA ENERGETSKE ELEKTRONIKE	41-44

	STRANA
5. Ana Rudić, RAZVOJ SISTEMA ZA VIDEO KONFERENCIJE	45-48
6. Вељко Максимовић, ИМПЛЕМЕНТАЦИЈА TASK QUEUE СЕРВИСА У ОКВИРУ ПЛАТФОРМЕ ЗА ИВИЧНО РАЧУНАРСТВО	49-52
7. Aleksandar Savić, AUTOMATIZOVANO TESTIRANJE KORISNOSTI VEB APLIKACIJA POMOĆU JEST OKVIRA	53-56
8. Marko Jovanović, PREDIKCIJA ŽANRA PESME NA OSNOVU TEKSTA PESME POMOĆU ALGORITAMA MAŠINSKOG UČENJA	57-60
9. Kristina Milić, UTICAJ DISTRIBUTIVNIH GENERATORA NA REGULACIJU NAPONA DISTRIBUTIVNIH MREŽA	61-64
10. Boris Šuličenko, ANALIZA BEZBEDNOSNIH NAPADA U NODEJS EKOSISTEMU	65-68
11. Milica Cvjetićanin, PRILOG KREIRANJU OPTIMALNOG PLANA SADNJE KUKURUZA UPOTREBOM METODE DINAMIČKOG PROGRAMIRANJA	69-72
12. Никола Мандић, МИКРОФРОНТЕНД АПЛИКАЦИЈЕ	73-76
13. Stefan Stojaković, TESTNO OKRUŽENJE ZA RUKOVANJE METEOROLOŠKIM PODACIMA POMOĆU SPARQL UPITA	77-80
14. Andrej Karavida, APLIKACIJA ZA PRORAČUN ISPLATIVOSTI INVESTICIJE UGRADNJE SOLARNIH PANELA	81-84
15. Biljana Vukelić, RAZVOJ APLIKACIJE ZA PRORAČUN ISPLATIVOSTI IZGRADNJE VETROGENERATORA U AZURE OKRUŽENJU	85-88
16. Tomislav Tumbas, PRIMENA METRIČKOG UČENJA U POLJOPRIVREDI	89-92
17. Vuk Stanojev, AUTOMATSKA REKONSTRUKCIJA DIJAKRITIČKIH ZNAKOVA U TEKSTOVIMA NA SRPSKOM JEZIKU PRIMENOM MAŠINSKOG UČENJA	93-96
18. Светлана Стојадинов, ПРЕДВИЂАЊЕ ТОКА КАРИЈЕРЕ ТЕНИСЕРА ПОМОЋУ ТЕХНИКА АНАЛИЗЕ ПОДАТАКА ...	97-100
19. Жељана Шиповац, ФАЗ ТЕСТИРАЊЕ ВЕБ АПЛИКАЦИЈА	101-104
20. Tamara Veličković, PRIMENA NFT TOKENA U NAUČNIM PUBLIKACIJAMA	105-108
21. Adedotun Olasunkanmi Ogundare, ISPITIVANJE ALGORITAMA MAŠINSKOG UČENJA ZA PREDVIĐANJE NAPADA NA SOFTVERSKI DEFINISANE MREŽE (INVESTIGATING MACHINE LEARNING ALGORITHMS PERFORMANCE FOR PREDICTING ATTACKS ON SOFTWARE DEFINED NETWORKS)	109-112
22. Dušan Ilić, LED ПАМЕТНА РАСВЕТА ЗА УЗГОЈ АКВАРИЈУМСКОГ БИЈА	113-116
23. Радош Аћимовић, СИСТЕМИ ЗА УПРАВЉАЊЕ СОФТВЕРСКИМ КОНТЕЈНЕРИМА	117-120

Radovi iz oblasti: Građevinarstvo

1. Slobodan Miletić, UPOREDNA ANALIZA MEĐUSPRATNIH SPREGNUTIH KONSTRUKCIJA: TIPOVI MOŽDANIKA I ROBUSTNOST	121-124
2. Maja Dragičević, Igor Čolev, PROJEKAT MOSTA DUŠANOVAČ-PRAHOVO NA STACIONAŽI KM 6+734.824 LEVO I KM 6+801.266 DEŠHO	125-128
3. Boris Zastranović, IDEJNO REŠENJE VODOSNABDEVANJA I ODVOĐENJA ATMOSFERSKIH VODA U NASELJU SEFKERIN	129-132
4. Marija Došlić, ANALITIČKA METODA ZA KONTROLU PROHODNOSTI I PRORAČUN ELEMENATA SITUACIONOG PLANA POVRŠINSKIH RASKRŠNIČA	133-136
5. Dejan Šipovac, PROCJENA STANJA I SANACIJA PROIZVODNE HALE U SKLOPU FABRIKE ŠEĆERA U CRVENKI SA ASPEKTA TRAJNOSTI I PROMENE NAMENE	137-140

Radovi iz oblasti: Saobraćaj

1. Ljubinka Vidović, PRIMJENA SISTEMA POSLOVNE INTELIGENCIJE U BEZBEDNOSTI SAOBRAĆAJA	141-144
2. Vučur Buraš, Goran Stojić, ANALIZA ŽELEZNICHKIH NESREĆA NA PUTNIM PRELAZIMA NA TERITORIJI AP VOJVOĐINE	145-148

Radovi iz oblasti: Mehatronika

1. Aleksandar Kičić, MAPIRANJE, LOKALIZACIJA I NAVIGACIJA PO NERAVNOM TERENU AUTONOMNO VOĐENOG VOZILA PROMENLJIVE GEOMETRIJE	149-152
2. Nikola Bajandić, PROJEKTOVANJE KOMPAKTNE TOPLOTNE PODSTANICE	153-156

**АНАЛИЗА ЗАМОРНОГ ПОНАШАЊА ТРАНСПОРТНЕ ТРАКЕ МЕТОДОМ
КОНАЧНИХ ЕЛЕМЕНАТА****ANALYSIS OF THE CONVEYOR BELT FATIGUE BEHAVIOR USING THE FINITE
ELEMENT METHOD**Лука Куцурски, Никола Иланковић, *Факултет техничких наука, Нови Сад***Област - МАШИНСТВО**

Кратак садржај – Циљ овог рада је да се истраже и поставе основе за анализу замора композитних материјала применом симулација. На самом почетку дате су теоријске основе из области композита и замора материјала. Након тога, помоћу софтверског пакета Autodesk Inventor Nastran, представљен је детаљан приказ дефинисања ламинарних композита као и поставка анализе замора у софтверу. На крају је обрађена заморна анализа транспортне траке, чиме се добила јасна слика предности и тренутних недостатака софтвера као и смернице за правце даљих истраживања.

Кључне речи: композити, заморно понашање, транспортна трака, метода коначних елемената

Abstract – The goal of this paper is to investigate and establish the basis for fatigue analysis of composite materials using simulation. At the very beginning of the paper, the theoretical foundations in the field of composite materials and fatigue behavior are given. After that, using the Autodesk Inventor Nastran software package, a detailed description of the creation of laminar composites as well as a fatigue analysis setup in the software is presented. At the end of last chapter, where fatigue behavior of the conveyor belt is analyzed, a clear picture of the advantages and current disadvantages of the softwer is obtained. Also the possible directions of further research.

Keywords: composite materials, fatigue behavior, conveyor belt, finite element method

1. УВОД

У данашње време промене се дешавају невероватном брзином, свакодневно се јављају новине које постају део свакодневнице. Практично је немогуће замислити посао инжењера без употребе рачунарске технике и симулација. У претходних неколико деценија дошло је до огромног развоја рачунарске технике, самим тим и рачунарских софтвера, који су омогућили употребу симулација и тиме у многоме олакшале инжењерски посао и донеле низ предности.

НАПОМЕНА:

Овај рад је проистекао из мастер рада чији ментор је био др Драган Живанић, ванр. проф.

Постоје многе области које још увек нису довољно истражене или још увек не постоји потпуно развијен софтвер који може да одговори на све захтеве које диктира убрзани технолошки развој. Једна од тих области јесте и замор материјала, поготово замор код композитних материјала, који по својој природи имају комплексну структуру.

2. КОМПОЗИТНИ МАТЕРИЈАЛИ

По дефиницији, композитни материјали или укратко композити су материјали добијени спајањем два или више материјала различитих својстава у микро и/или макроскопској размери, са циљем добијања материјала који најбоље показује својства својих компоненти или добијања таквих својстава каква не поседује нити једна компонента сама за себе. Сваки композит се састоји од матрице, која представља основни материјал одређених својстава, и ојачања које представља материјал чијим се додавањем постижу потребне комбинације својстава композита, слика 1. Композитни материјали се могу поделити на основу више критеријума нпр. на основу материјала матрице, на основу материјала ојачања, према својој структури и понашању тј. својствима у различитим правцима.

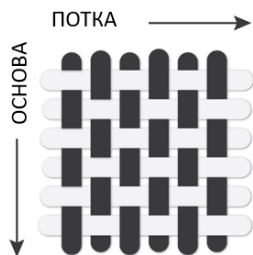


Слика 1. Градивни елементи композита

2.1. Текстилни композити – 2Д и 3Д тканине

Транспортна трака, као основни елемент тракастих транспортера, представља ламинарни композит, који је састављен од 2Д тканина, па је посебна пажња у раду посвећена текстилним композитима. 2Д тканине настају технолошким процесом који се назива ткање. Разноврсним укрштањем настају тканине различитих особина, изгледа и структура. 2Д тканине имају добре механичке особине у својој равни, високу специфичну крутост и чврстоћу. Међутим, у многим применама јављају се и оптерећења ван њихове равни где 2Д

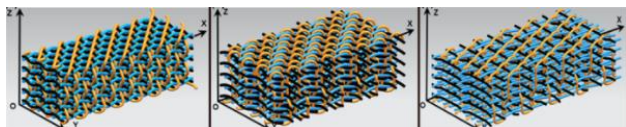
тканине нису право решење. Због јављања потребе за тканинама које имају добре карактеристике и подносе оптерећења ван своје равни, настају 3Д тканине које улазе у састав 3Д композита.



Слика 2. 2Д тканина

Разлика између 3Д и 2Д тканине је што 3Д тканине поред основе и потке садржи и везиво. У зависности од структуре тканине, нити потке, основе и везива пролазе, преко, дуж и кроз тканине у смеровима X, Y, Z. Последњих година 3Д тканине су технологија у повоју.

Поред добрих карактеристика на оптерећење ван равни, 3Д тканине имају низ предности. Због тога је 3Д ткање технологија у развоју која нуди разне предности у односу на 2Д тканине. Кључне предности су смањена маса, уклањање деламинације, смањен ризик од пукотина, краће време производње и смањење трошкова. Приказ неких од типова 3Д тканина дат је на слици 3.



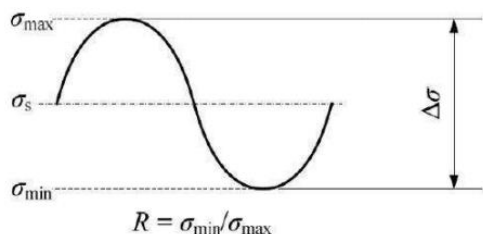
Слика 3. 3Д тканина

3. ЗАМОР МАТЕРИЈАЛА

Замор представља оштећење материјала изазавано периодично променљивим оптерећењима која по свом интензитету не прелазе напон текућа. Лом услед замора је последица еластичних и еласто-пластичних деформација које се, због нехомогености материјала, неравномерно распоређују по запремини тела.

Почетне прелине настају у микрзапреминама неповољно оријентисаним према спољашњем оптерећењу, пренапрегнутим зонама и локалним ослабљењима услед геометријских неправилности.

Сами циклуси тј. циклично оптерећење има неколико карактеристика које су од посебног значаја – максималне и минималне вредности оптерећења (напона), као и амплитуду. Разлика између максималне и минималне вредности назива се распон напона и обележава се са $\Delta\sigma$, слика 4.



Слика 4. Амплитуда оптерећења

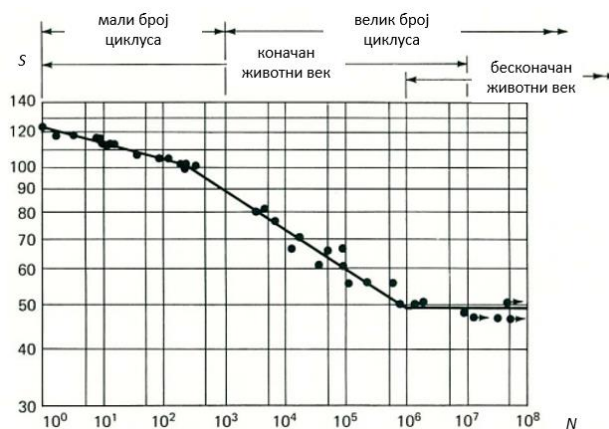
Однос ове две вредности, односно минималног и максималног напона назива се фактор асиметрије циклуса R , и од великог је значаја при нумеричкој симулацији конструкција оптерећених на замор.

Такође, величине које су карактеристичне код циклично оптерећења су средњи напон σ_m и амплитудни напон σ_a , који се израчунавају као:

$$\sigma_m = \frac{\sigma_{max} + \sigma_{min}}{2}, \quad \sigma_a = \frac{\sigma_{max} - \sigma_{min}}{2} \quad (1)$$

3.1. S-N дијаграм

Једна од основних метода за предвиђање животног века при замору јесте S-N дијаграм, слика 5. S представља опсег циклично оптерећења, док N представља број циклуса до отказа. Да би се формирала крива, потребно је да се низ узорака тестира до отказа у различитим опсезима оптерећења. Уобичајена је пракса да се крива учртава кроз средњу вредност у сваком опсегу напона.



Слика 5. Приказ S-N дијаграма

3.2. Гудманов дијаграм

Пре самог приказа дијаграма, слика 6, потребно је да се образложе величине које фигуришу у дијаграму, а то су:

- средњи напон;
- амплитудни напон;
- затезна чврстоћа – представља највећу вредност напона којим се материјал може оптеретити затезањем а да притом не дође до лома, означава се са R_m ;
- граница развлачења – представља напон изнад којег материјал почиње трајно да мења облик (пластична деформација), означава се са R_p ;
- трајна динамичка чврстоћа - представља напон испод којег материјал може издржати бесконачан број поновљених циклуса оптерећења, означава се са R_D .

На Гудмановом дијаграму може се видети да је на хоризонталној оси постављен средњи напон, а на вертикалној оси амплитудни напон. За било коју комбинацију ова два напона, која дају тачку која се налази у области испод Гудманове криве, каже се да материјал тј. узорак може да издржи такво оптерећење бесконачно много циклуса. За комбинације напона које дају тачку изнад Гудманове криве каже се да ти узорци, под таквим оптерећењем, могу да издрже коначан број циклуса.



Слика 6. Гудманов дијаграм

4. РАД СА ЛАМЕНАРНИМ КОМПОЗИТА У AUTODESK INVENTOR NASTRAN

Autodesk Inventor је софтверски пакет за пројектовање помоћу рачунара, служи за 3Д машински дизајн, симулацију, визуелизацију и документацију, развијен је од стране фирме *Autodesk*. *Inventor* омогућава интеграцију 2Д и 3Д података у једно окружење, стварајући виртуелну презентацију финалног производа која омогућава корисницима да провере форму, уклапање и функцију производа пре него што се сам производ изради. *Nastran*, као додатни модул који је интегрисан у *Inventor*, веома је моћан алат за анализу методом коначних елемената (е. *FEA*). Користи се за анализу статичких и динамичких напрезања, као и анализу преноса топлоте конструкција и механичких компоненти.

Поред анализе конвенционалних инжењерских материјала, *Nastran* је опремљен да симулира понашање најновијих напредних материјала, укључујући композите, пластику, хипереластичне, вискоеластичне и крте материјале као што је бетон. Први корак код дефинисања композита је одабир материјала, преко картице *Materials*. Софтвер нуди могућност одабира материјала из његове библиотеке, али пружа могућност кориснику да дефинише особине материјала по жељи.

Када се дефинишу материјали који ће сачињавати композит, *Laminate* алат омогућава да се дефинише сваки слој тако што ће се дефинисати материјал слоја један по један заједно са њиховим дебљинама, угловима постављања итд. Поред основних поставки, корисник је у могућности да дефинише и која теорија отказа ће се користити (*Hill*, *Hoff* и *Tsai*, *Max Stress*), интерламинарни смичући напон, коефицијент пригушења и многе друге параметре. Ако се у композиту јављају слојеви са истим или сличним особинама, веома је корисна алатка *Global Ply* преко које се веома брзо бирају претходно дефинисани слојеви.

Када су слојеви композита дефинисани, помоћу алатке *Idealizations*, жељеном 3Д моделу се додељују претходно дефинисане карактеристике композита. У алатци је потребно за тип елемента изабрати *Shell Elements*, затим обележити поље *Associated Geometry* и одабрати геометрије на којима се жели применити идеализација. Да би резултати анализе били што тачнији, на крају је неопходно дефинисати оријентацију материјала, помоћу алатке *Material Orientation*. Када су претходни кораци урађени може се приступити статичкој анализи композита.

5. АНАЛИЗА ЗАМОРА ТРАНСПОРТНЕ ТРАКЕ У AUTODESK INVENTOR NASTRAN

Најраспрострањеније транспортно средство непрекидног транспорта јесте тракасти транспортер. Разлог широке примене је тај што се са овим транспортерима постижу високи техничко-економски ефекти, потпуна аутоматизација свих фаза технолошког процеса, уз пуну радну сигурност и одговарајуће усклађености коришћења капацитета других уређаја у технолошком процесу [1]. Нажалост, у великом броју случајева долази до пуцања транспортне траке при нивоима оптерећења који су нижи од дозвољеног. То је јасан знак да велику улогу у експлоатационом веку транспортне траке игра заморно оптерећење. Због тога што замор још увек представља нову и недовољно истражену област у софтверском пакету *Autodesk Inventor Nastran*, у раду се анализа замора материјала формирала на основу експерименталних истраживања и резултата из докторске дисертације [2]. Транспортна трака се састоји од гуме, као матрице, и полиестер-полиамид влакана, као ојачивача. Како је транспортна трака састављена од горњег и доњег гуменог слоја и језгра, које се састоји од полиестер-амида, својства испитане траке су дата у табели 1. и табели 2.

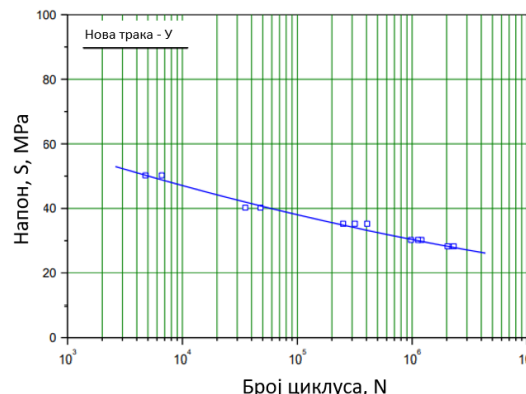
Табела 1. Физичко механичке карактеристике језгра транспортне траке

Особина материјала	Врста материјала	
	Р (полиамид)	Е (полиестер)
Затезна чврстоћа [N/mm ²]	990	1150
Издужење при кидању [%]	17	14
Модул еластичности [N/mm ²]	5,5	13,8
Густина материјала [g/cm ³]	1,14	1,38

Табела 2. Карактеристике транспортне траке

Дебљина транспортне траке [mm]	20
Дебљина горњег заштитног гуменог слоја [mm]	6
Дебљина доњег заштитног гуменог слоја [mm]	4
Број носећих слојева [mm]	4
Дебљина носећег слоја [mm]	2,5
Јачина на кидање [mm]	2000

Према [2], на основу утврђених модела стандардних метода испитивања композитног материјала, одређена су основна макромеханичка својства траке: затезна чврстоћа, модул еластичности, Поасонов коефицијент и деформација.



Слика 7. S-N дијаграм добијен експерименталним путем

Испитивања на затезање су урађена на епруветама композита гума-полиестер-полиамид и изведена су према стандарду ASTM D 3039, док су испитивања у условима динамичког деловања силе рађена по стандардима ASTM E 466, ASTM E 467 и ASTM E468. На основу резултата у условима динамичког оптерећења, формиран је дијаграм приказан на слици 7. Са дијаграма су очитане вредности напона, за одређене бројеве циклуса, и употребљене за добијање сила за испитивање затезне чврстоће и трајне динамичке чврстоће, за формирани модел епрувете у софтверу.

Због немогућности софтвера да покрене анализу замора када је модел креиран као композитни материјал, модел тј. епрувета за коју је урађена анализа је креирана као *Solid*. За дебљину епрувете је дефинисана вредност од 10 mm, што у ствари представља 4 слоја од по 2,5 mm. Та 4 слоја су заправо и једини носећи слојеви у транспортној траци. Анализа је замишљена да се верификује кроз пет итерација.

Табела 3. План испитивања

Бр. итерације	Сила [N]	Предвиђен број циклуса до отказа
1	48.000	1.000
2	40.000	5.000
3	32.000	50.000
4	28.000	300.000
5	24.000	1.000.000

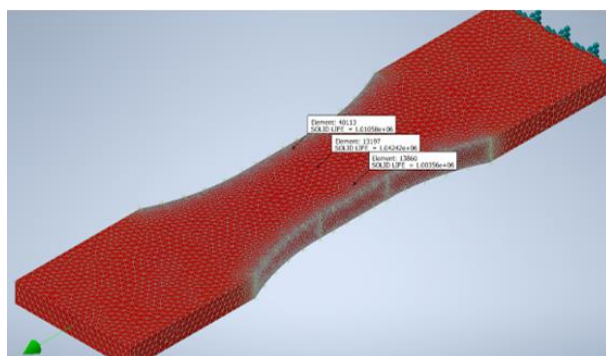
На почетку је у софтверу дефинисан материјал према подацима добијених из испитивања. Након тога, посебна пажња је посвећена дефинисању S-N криве у оквиру за дијалог *Fatigue*. Унесене су вредности нагиба криве, затезне чврстоће, трајне динамичке чврстоће. Након тога, да би се симулирао рад серво хидрауличке кидалице, на један крај епрувете је постављено укљештење, док је на другом крају постављено оптерећење које је фигурирало кроз анализу. За анализу замора неопходно је и дефинисати како ће се оптерећење мењати кроз време, у оквиру за дијалог *Load History Table Data*. Пре покретања анализе, за сваки од случајева оптерећења, потребно је дефинисати мрежу коначних елемената. Преко алатке *Mesh control* дефинише се локална мрежа. Дефинисана је величина од 0,5 mm. Затим је дефинисана глобална мрежа коначних елемената, преко алата *Generate Mesh*. Дефинисани су елементи величине 3 mm. Када су претходни кораци урађени, анализа је пуштена за пет случајева оптерећења, наведених у табели 3. Резултати анализе замора добијени након пет итерација истог су реда величине као и резултати добијени експерименталним путем.

Табела 4. Резултати рачунарског експеримента

Бр. итерације	Сила [N]	Број циклуса до отказа за посматране елементе		
		1	2	3
1	48.000	1.010	1.023	1.003
2	40.000	5.426	5.539	5.420
3	32.000	51.967	55.819	51.029
4	28.000	298.774	311.663	294.629
5	24.000	1.010.580	1.042.420	1.003.560

Овим је потврђена валидност резултата симулације, као и самих апроксимација које су уведене у поставку

анализе замора. Резултати су представљени у табели 4. Такође један од резултата симулације је приказан на слици 8.



Слика 8. Приказ резултата из пете итерације

6. ЗАКЉУЧАК

Експериментална истраживања везана за анализу замора су захтевна, изискују доста времена и финансијских средстава. Коришћењем симулација, као полазне тачке за анализу замора, елиминишу се већина ограничавајућих фактора присутних код експерименталних метода. Још једна од великих предности је могућност брзе измене модела, његове структуре или специфичних карактеристика у почетним фазама истраживања и након тога брзо добијање нових резултата. Све то доводи до тога да даља експериментална истраживања буду мањег обима, јефтинија и бржа. Поред свих претходно наведених могућих бенефита приликом коришћења софтвера за анализу замора, један од конкретних бенефита се јавља у области транспортних трака. Степен сигурности приликом прорачуна транспортне траке је тренутно велик да би се елиминисао ефекат замора. Добијањем реалног животног века транспортне траке, степен сигурности би могао драстично да се смањи, чиме би дошло до избора транспортне траке оптималних карактеристика, а тиме и до значајних уштеда.

7. ЛИТЕРАТУРА

- [1] Колоња Б., Кнежевић Д., „Транспорт у припреми минералних сировина“, 2000.
- [2] Пирић Е., „Утицај експлоатационих услова на понашање гумене траке тракастих транспортера“, 2015.

Кратка биографија:



Лука Куцурски рођен је у Сомбору, Србија 1998. год. Дипломирао је 2020. год. на Факултету техничких наука, смер Механизација и конструкционо машинство, на којем исте године уписује мастер академске студије.



Никола Иланковић рођен је у Суботици 1994. год. Завршио је мастер академске студије 2018. године на Факултету техничких наука и исте године уписао докторске студије машинства. Запослен је у звању асистент мастер на Факултету техничких наука у Новом Саду.

ANALIZA NAPONSKOG STANJA KONSTRUKCIJE MODULARNOG TRAKASTOG TRANSPORTERA PRIMENOM METODE KONAČNIH ELEMENATA

STRESS ANALYSIS OF THE MODULAR BELT CONVEYOR CONSTRUCTION USING FINITE ELEMENT METHOD

Aron Pinter, Nikola Ilanković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – U radu je izvršena analiza naponskog stanja konstrukcije modularnog trakastog transportera pomoću softvera Autodesk Inventor i dodatnog modula Nastran. Urađen je proračun osnovnih mašinskih elemenata i na osnovu dobijenih rezultata je primenjena optimizacija konstrukcije transportera.

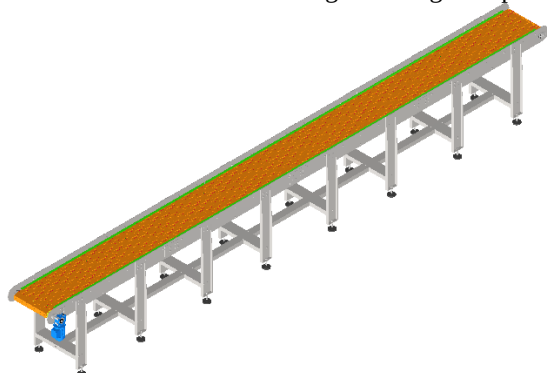
Ključne reči: analiza, metod konačnih elementa, trakasti transporter

Abstract – In this paper an stress analysis of the modular belt conveyor construction was performed using the software Autodesk Inventor and the additional modul Nastran. The calculation of the basic machine elements was done and based on the obtained results, optimization of the conveyor construction was applied.

Keywords: analysis, finite element metod, belt conveyor

1. UVOD

U radu je izvršena analiza naponskog stanja konstrukcije modularnog trakastog transportera, pomoću programa Autodesk Inventor i dodatnog modula Inventor Nastran. Rad sadrži proračun opterećenja noseće konstrukcije i proračun osnovnih mašinskih elemenata konstrukcije. Dat je proces idealizacije modela i priprema modela prednjeg, srednjeg i zadnjeg modula za analizu metodom konačnih elemenata. Prikazani su dobijeni rezultati simulacije, kao i moguće korekcije u odnosu na prvobitnu konstrukciju, na osnovu rezultata analize i proračuna. Na slici 1. je prikazan CAD model modularnog trakastog transportera.

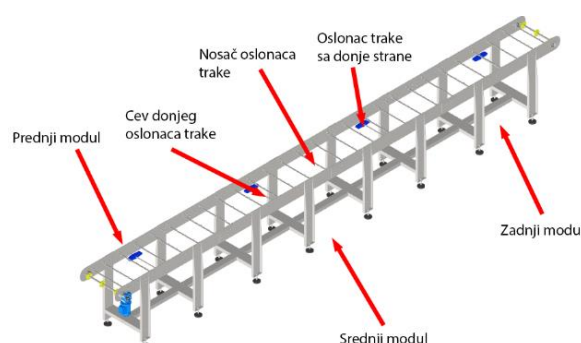


Slika 1. Modularni trakasti transporter

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Živanić, vanr. prof.

Za oslanjanje trake i materijala, osim same noseće konstrukcije, služi nosač oslonaca trake, oslonac trake sa donje strane i cev donjeg oslonaca trake. Na slici 2. je prikazan model transportera bez trake, U i L oslonci trake sa najvažnijim elementima koji su potrebni za analizu.



Slika 2. Elementi modularnog trakastog transportera za oslanjanje trake i materijala

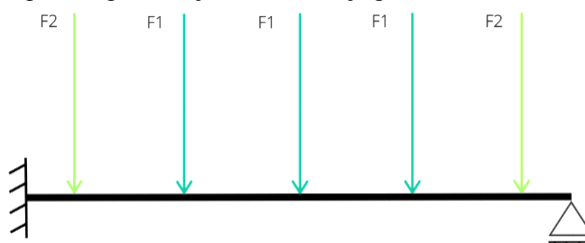
Materijal noseće konstrukcije je čelik S235, vijaka, podložki i navrtki je galvanizovan čelik a donjeg držača trake je PA6. Osobine upotrebljenih materijala pojedinih elemenata transportera su date u tabeli 1.

Tabela 1. Osobine materijala pojedinih elemenata transportera [1] [2] [3] [4] [5]

	S235	galv. čelik	PA6
gustina [kg/m ³]	7850	7850	1150
Jungov modul el.[MPa]	210	200	2400
Poasonov koeficijent	0,3	0,3	0,39
term. koef širenja [K ⁻¹]	12×10 ⁻⁶	12×10 ⁻⁶	101,5×10 ⁻⁶
napon tečenja [MPa]	235	207	66,5

2. PRORAČUN OPTEREĆENJA KONSTRUKCIJE

Raspored opterećenja kod nosača je prikazan na slici 3.



Slika 3. Raspored opterećenja nosača oslonaca trake

Sila koja deluje na sredini nosača oslonaca trake na vertikalne štapove određuje se pomoću sledeće formule:

$$F_1 = \frac{(3 \cdot m_{U0,29} + m_{T0,29} + m_P) \cdot 10}{3} = \frac{(3 \cdot 0,08 + 0,87 + 3) \cdot 10}{3} = 13,7 \text{ [N]}$$

gde su:

- $m_{U0,29}$ [kg] – masa U oslonaca trake na 0,29 m;
- $m_{T0,29}$ [kg] – masa trake na 0,29 m;
- m_P [kg] – masa paketa.

Izbor trake, L i U oslonaca trake i zupčanika sa potrebnim proračunima izvršeno je u [6]. Sila koja opterećuje vertikalne štapove nosača oslonaca trake se određuje pomoću formule:

$$F_2 = m_{L0,29} \cdot g = 0,1 \cdot 10 = 1 \text{ [N]}$$

gde su:

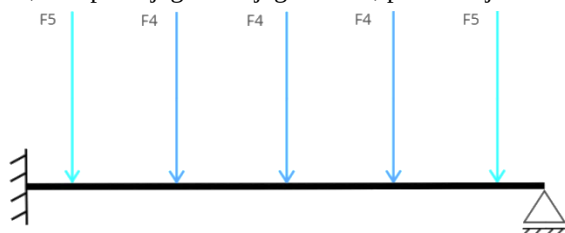
- $m_{L0,29}$ [kg] – masa L oslonaca trake na 0,29 m;
- g [m/s²] – gravitaciono ubrzanje Zemljine težine.

Sila koja deluje na donji oslonac trake određuje se pomoću sledeće formule:

$$F_3 = \frac{m_{T2} \cdot g}{2} = \frac{6,03 \cdot 10}{2} = 30,15 \text{ [N]}$$

gde je m_{T2} [kg] – masa trake na 2 m.

Raspored opterećenja kod spoljašnjih nosača oslonaca trake, kod prednjeg i zadnjeg modula, prikazan je na sl. 4.



Slika 4. Raspored opterećenja spoljašnjih nosača oslonaca trake

Sila koja deluje na sredini spoljašnjih nosača oslonaca trake se određuje pomoću sledeće formule:

$$F_4 = \frac{F_1 \cdot l_{0,2}}{l_{0,29}} = \frac{13,7 \cdot 0,2}{0,29} = 9,45 \text{ [N]}$$

gde su:

- F_1 [N] – sila koja deluje na nosače oslonaca trake koje se nalaze na međusobnom rastojanju 0,29 m;
- $l_{0,2}$ [m] – rastojanje između spoljašnjih nosača oslonaca trake i nosača pored njega;
- $l_{0,29}$ [m] – rastojanje između nosača oslonaca trake na ostalim delovima konstrukcije.

Sila koja deluje na krajeve spoljašnjih nosača oslonaca trake se određuje pomoću sledeće formule:

$$F_5 = \frac{F_2 \cdot l_{0,2}}{l_{0,29}} = \frac{1 \cdot 0,2}{0,29} = 0,69 \text{ [N]}$$

gde je F_2 [N] – sila koja deluje na nosače oslonaca trake koji se nalaze na međusobnom rastojanju 0,29 m.

3. PRORAČUN VRATILA, LEŽAJEVA I KLIHOVA

Prečnici vratila, na kojima su postavljeni zupčanici preko kojih se pogoni, odnosno savija traka, kod prednjeg i zadnjeg modula, su zbog geometrijskih ograničenja usvojeni da imaju standardnu vrednost 15 mm. Proračunima su dobijeni zadovoljavajući stepeni sigurnosti, tako da isti odgovaraju [6].

Na delu vratila sa prečnikom 25 mm, za vezu sa zupčanikom, usvojena je standardna dužina klina 80 mm, a kod dela vratila koji se nalazi kod motora standardna dužina od 90 mm. Stepenn sigurnosti je u dozvoljenim granicama, kod oba klina.

Pošto je broj obrtaja vratila veći od 10 min⁻¹, izvršen je dinamički proračun oba ležaja, kod oba vratila. Na osnovu dobijenih rezultata ustanovljeno je da ležajevi imaju veći radni vek, u odnosu na preporuku proizvođača, tako da zadovoljavaju uslov.

4. IDEALIZACIJA

Delovi noseće konstrukcije su pretvoreni pomoću alata Idealizations u Nastran u solid elemente. Kod svakog solida je zadat potreban materijal. Delovi noseće konstrukcije jednostavnijeg oblika nisu modifikovani, odnosno, pojednostavljenje delova nije bio potrebno. Donji deo oslonca transportera sa navrtkama M16 je zamenjen pomoću Fixed Constrain. Na prednjem modulu, prednji podsklop sa ležajevima je zamenjen pomoću koncentrisanih masa. Kod prednjeg modula postavljen je elektromotor, koji je zbog pojednostavljenja analize predstavljen kao koncentrisana masa.

Model elektromotora sastoji se od više elemenata i tokom povezivanja koncentrisane mase sa držačem motora je uzrokovao problem. Da bi se ovaj problem eliminisao, bilo je potrebno elektromotor, pomoću Shrinkwrap alata, konvertovati u jedan solid, tj. part.

5. OSNOVNA PODEŠAVANJA

Da bi se adekvatno postavile vijčane veze u modulu Nastran, model je pripremljen tako što su oko otvora za vijke definisane zone na koje naleže podloška vijčane veze. Za oslonce je izabran Fixed Constrain. Na prednjem i zadnjem modulu je Fixed Constrain postavljen na mestima spoja sa srednjim modulom i na mestima kontakta noseće konstrukcije i poda. Što se tiče srednjeg modula, Fixed Constrain je postavljen na oba kraja jer se smatra da je srednji modul u integralnoj vezi sa preostalim modulima.

Za postavljene vijčane veze koristio se alat Connection i opcija Bolt. Elementi noseće konstrukcije povezani su vijcima M12. Veza između površine elemenata je zadata pomoću Auto Contacts alata.

Opterećenje je postavljeno pomoću alata Loads na nosač oslonaca trake i na donji držač trake. Mreža konačnih elemenata je generisana automatski, zbog dobijanja tačnijih rezultata i ušteda vremena simulacije, pomoću alata Convergence Settings. Korišćena je opcija Local Refinement, koja samo kod problematičnih zona smanjuje veličinu konačnih elemenata i pravi finiju mrežu.

6. SIMULACIJA I ANALIZA

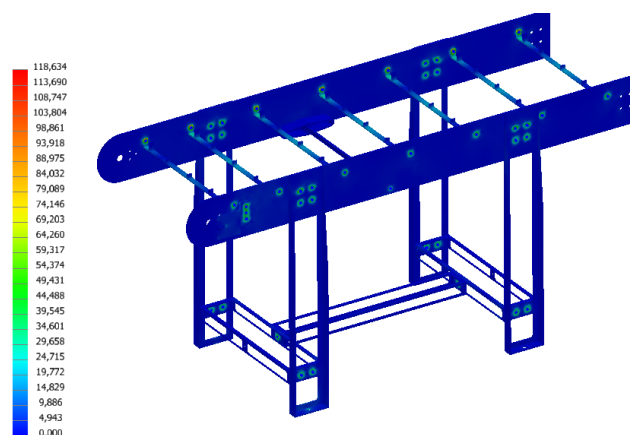
Dobijeni rezultati analize prikazani su u tabeli 2, pri čemu su dati rezultati posebno za prednji, srednji i zadnji modul analiziranog modularnog transportera.

Naponsko stanje srednjeg modula prikazano je na slici 5.

Najveći napon iznosi 149,9 MPa i javlja se samo sa jedne strane nosača oslonaca trake, dok se na drugom kraju javlja 92 MPa. Detaljnijom analizom se zaključilo da uočeni maks. napon predstavlja napon u tački, odnosno koncentrisano opterećenje, i da se može zanemariti.

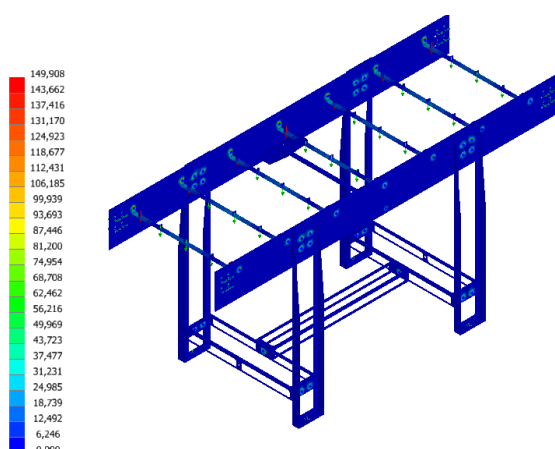
Tabela 2. Rezultati analize

	Prednji modul	Srednji modul	Zadnji modul
Stepen sigurnosti (Solid Von Mises Stress) [-]	1,98	2,54	2,19
Stepen sigurnosti (Beam Von Mises Stress) [-]	1,47	1,44	1,48
Solid Von Mises Stress [MPa]	118	93	107
Deformacija [mm]	0,936	0,908	0,844
Beam Von Mises Stress [MPa]	141	144	141
maks. aksijalna sila u vijku [N]	15210	15222	15205
maks. tangencijalna sila u lokalnom y pravcu [N]	74	117	80
maks. tangencijalna sila u lokalnom z pravcu [N]	59	229	45



Slika 7. Naponsko stanje prednjeg modula

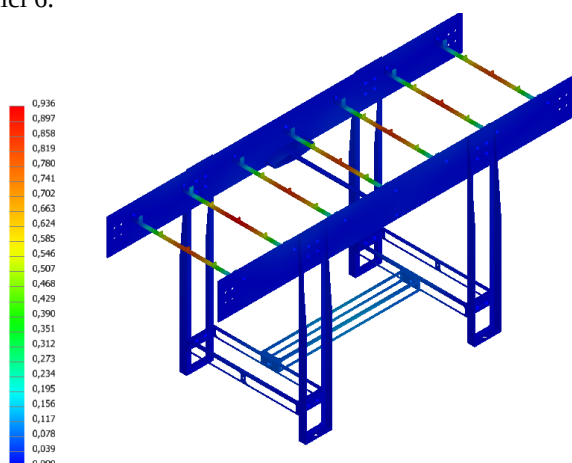
Najveća deformacije kod prednjeg modula javlja se na očekivanom mestu, na sredini oslonaca trake. Deformacija elemenata prednjeg modula prikazana je na slici 8.



Slika 5. Naponsko stanje srednjeg modula

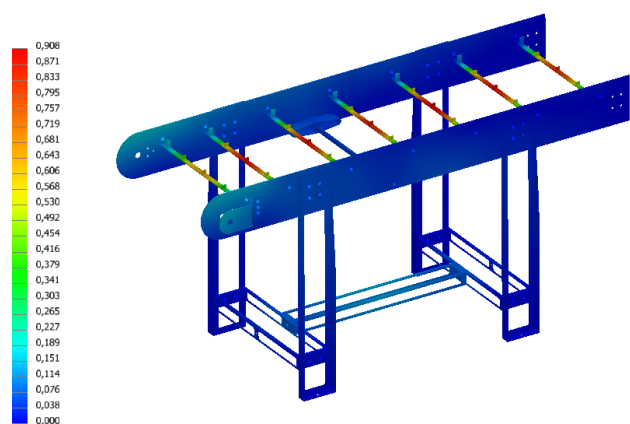
Najveća deformacija kod srednjeg modula javlja se na očekivanom mestu, tačno na sredini nosača oslonaca trake.

Deformacija elemenata srednjeg modula prikazana je na slici 6.



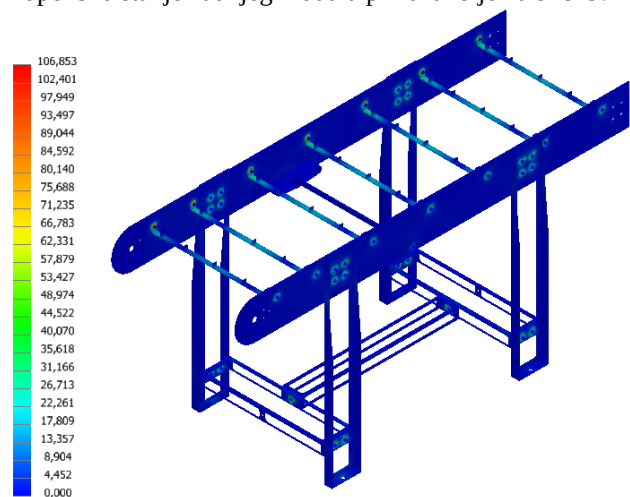
Slika 6. Deformacija srednjeg modula

Kod prednjeg modula dobijaju se najveći naponi, zbog toga što je na ovaj modul postavljen elektromotor i prednji podsklop sa ležajevima, slika 7. Najveći napon iznosi 118,6 MPa.



Slika 8. Deformacija prednjeg modula

Naponsko stanje zadnjeg modula prikazano je na slici 9.

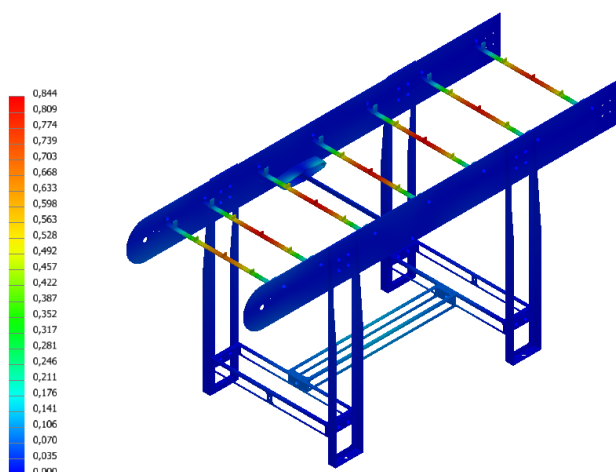


Slika 9. Naponsko stanje zadnjeg modula

Najveći napon se javlja na gornjem delu nosača oslonaca trake i iznosi 107 MPa.

Deformacija zadnjeg modula prikazana je na slici 10.

Rezultati analize kod svakog modula zadovoljavaju uslov za konvergenciju, tj. razlika između poslednje i prethodne iteracije rezultata je ispod 5%.



Slika 10. Deformacija zadnjeg modula

7. MOGUĆE KOREKCIJE PRVOBITNE KONSTRUKCIJE - OPTIMIZACIJA

Analizom dobijenih rezultata mogu se doneti određene preporuke za korekcijama pojedinih elemenata i optimizaciju konstrukcije modularnog trakastog transportera.

Preporuka je da se zameni motor sa reduktorom, zbog velikog momenta koji deluje na noseću konstrukciju, na druge elemente konstrukcije. Iz kataloga istog proizvođača može se izabrati motor i reduktor, koji imaju iste priključne mere i veličinu, tako da na nosećoj konstrukciji zbog toga nisu potrebne izmene.

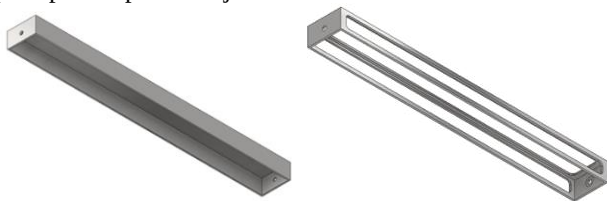
Nakon prve simulacije sa osnovnom konstrukcijom bilo je očigledno da može da se koriguje poprečni presek nekih elemenata, oduzimanjem materijala na neopterećenim ili malo opterećenim delovima [6].

Izgled gornjeg dela oslonaca transportera, pre i posle optimizacije, je prikazan na slici 11.



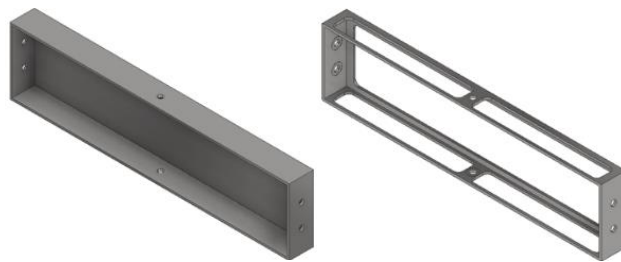
Slika 11. Izgled gornjeg dela oslonaca transportera pre i posle optimizacije

Oduzimanje materijala se preporučuje kod poprečnog i uzdužnog ukrućenja, pošto rezultati pokazuju da su naponi mali. Na slici 12 je prikazano uzdužno ukrućenje pre i posle optimizacije.



Slika 12. Poprečno ukrućenje pre i posle optimizacije

Poprečno ukrućenje pre i posle optimizacije prikazano je na slici 13.



Slika 13. Poprečno ukrućenje pre i posle optimizacije

Ležajevi mogu da se zamene odovarajućim sa slabijim karakteristikama, ali se mora voditi računa da imaju iste dimenzije i oblik, iz konstruktivnih razloga, kako vijci koji povezuju ležajeve i konstrukciju ne bi oštetili traku.

8. ZAKLJUČAK

Analiza elemenata modularnog trakastog transportera je izvršena u savremenom softveru, primenom metode konačnih elemenata. Na osnovu rezultata sprovedene analize i proračuna, zaključilo se da usvojeni elementi zadovoljavaju, ali i da postoji mogućnost optimizacije pojedinih elemenata modularnog trakastog transportera.

Vratilo je predimenzionisano, kao i usvojeni ležajevi. Na pojedinim elementima noseće konstrukcije uočene su gotovo neopterećene ili manje opterećene zone, tako da bi u tim oblastima mogao da se redukuje materijal.

9. LITERATURA

- [1] <https://eurocodeapplied.com/design/en1993/steel-design-properties>, (pristupljeno dana: 14. 3. 2022),
- [2] Inventor 2022: Material library, (pristupljeno dana: 15. 8. 2022),
- [3] <https://www.ensingerplastics.com/en/shapes/products/pa6-tecamid-6-natural>, (pristupljeno dana: 14. 3. 2022),
- [4] <https://designerdata.nl/materials/plastics/thermo-plastics/polyamide-6>, (pristupljeno dana: 14. 3. 2022),
- [5] <https://polymerdatabase.com/polymer%20physics/Poisson%20Table.html>, (pristupljeno dana: 14. 3. 2022),
- [6] A. Pinter: Modeliranje i proračun modularnog trakastog transportera, Fakultet Tehničkih Nauka, Novi Sad, 2021.

Kratka biografija:



Aron Pinter rođen je 1997. god. u Bačkoj Topoli. Diplomirao je 2021. god. na Fakultetu tehničkih nauka, smer Mehanizacija i konstrukciono mašinstvo. Iste godine upisao master akademske studije.

Kontakt: aron017@hotmail.com



Nikola Ilanković rođen je 1994. god. Završio je master akademske studije 2018. god. na Fakultetu tehničkih nauka i iste godine upisao doktorske studije mašinstva. Zaposlen je u zvanju asistent master na FTN u Novom Sadu.

ХИДРАУЛИЧКИ ПРОРАЧУН И ИЗБОР ОПРЕМЕ У РАСХЛАДНОЈ СТАНИЦИ ИНДУСТРИЈСКОГ ОБЈЕКТА ЗА ПОТРЕБЕ ПРОЦЕСНОГ ХЛАЂЕЊА ПРОИЗВОДНИХ МАШИНА

HYDRAULIC CALCULATION AND EQUIPMENT SELECTION IN COOLING STATION OF INDUSTRIAL FACILITY FOR THE NEEDS OF PROCESS COOLING OF PRODUCTION MACHINES

Јован Косановић, *Факултет техничких наука, Нови Сад*

Област- МАШИНСТВО - ЕНЕРГЕТИКА И ПРОЦЕСНА ТЕХНИКА

Кратак садржај – Тема рада јесте хидраулички прорачун и избор опреме у расхладној станици индустријског објекта за потребе процесног хлађења производних машина. Детаљно је описана усвојена опрема за припрему расхладног флуида који се користи за хлађење производних машина у погону. На основу усвојених вредности расхладног капацитета и других полазних величина урађен је хидраулички прорачун за два одвојена система хлађења – процесно хлађење машина за брушење и процесно хлађење пећница.

Кључне речи: Пумпе, цевоводи, хидраулички прорачун, расхладни агрегат, процесно хлађење

Abstract – The subject of the paper is the hydraulic calculation for defining proper equipment of the industrial building cooling station used for the cooling process of production machines. The required equipment for the coolant preparation is considered, which is used for the cooling process of production facility machines. The hydraulic calculation is shown for two different cooling systems (process cooling of grinding machines and process colling of industrial ovens) according to the cooling capacity and other input values needed for the calculation.

Keywords: Pumps, pipeline, hydraulic calculation, process cooling

1. УВОД

Висок степен иновативних технологија и капиталних инвестиција модерне индустрије подразумева све већу потребу за оптимизованим и економски одрживим расхладним процесима. Имајући у виду обим распрострањености и важност правилног, ефикасног и еколошког прихватљивог рада, расхладни системи захтевају висок степен оптимизације, резултујући мање трошкове у оперативном смислу. Да би расхладни систем могао да обезбеди жељене параметре, поред адекватног термичког прорачуна, односно прорачуна топлотних добитака, неопходно је и правилно димензионисати опрему на основу хидрауличног прорачуна.

НАПОМЕНА:

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

У раду је представљено концептуално решење, као и хидраулички прорачун и одабир одговарајуће опреме уз предефинисани расхладни капацитет.

2. ПРОЈЕКТНО РЕШЕЊЕ СИСТЕМА ЗА ХЛАЂЕЊЕ ПРОИЗВОДНИХ МАШИНА

2.1 Опис објекта индустријске производње

Објекат у којем се обавља процес производње делова за аутоиндустрију сачињен је из два дела:

- Производног погона – који чине машине за брушење, пећнице у којима се материјал термички обрађује и цевни развод инсталације технолошког хлађења производних машина у погону,
- Расхладне станице – у којој се налази опрема, односно уређаји који чине системе за технолошко хлађење производних машина. Из расхладне станице, преко цевног развода, производне машине се снабдевају расхладним флуидом.

Инсталација технолошког хлађења производног погона је подељена је на два независна система:

- Систем процесног хлађења машина за брушење,
- Систем процесног хлађења пећница.

У складу са технолошким потребама погона изведена је инсталација за технолошко хлађење машина за брушење, коју сачињавају:

Водом хлађени расхладни агрегат („chiller“), ваздушни хладњак („dry cooler“) монтиран у спољашњем простору, поред расхладне станице, растављиви плочасти измењивач топлоте за природно хлађење расхладног флуида („free cooling“), акумулациони резервоар расхладне воде запремине 2 m³, мембранске експанзионе посуде и пратећа арматура.

У складу са технолошким потребама погона изведена је инсталација за технолошко хлађење пећница као други одвојен систем хлађења којег чине:

Адијабатски хладњак монтиран у спољашњем простору, мембранска експанзиона посуда и пратећа арматура.

2.2 Технолошко хлађење машина за брушење током зимског и летњег периода

Рад постројења током зимског периода:

Концентрована смеша воде и гликола се потисним водом води од ваздушног хладњака до плочастог измењивача топлоте у којем долази до размене топлоте између концентроване смеше воде и гликола са примарне стране и воде са секундарне стране. Охлађена вода излази из плочастог измењивача топлоте и челичним цевоводом се води до расхладног агрегата. Охлађена вода циркулише кроз расхладни агрегат у коме не долази до размене топлоте и преко циркулационих пумпи се води у акумулациони резервоар.

У акумулационом резервоару долази до мешања воде са стране потиса и поврата, али је промена температуре незнатна. Хладна вода се даље преко циркулационих пумпи води кроз челични, а затим на уласку у производни погон, кроз пластични цеводод до крајњег корисника односно машина за брушење.

Хладна вода циркулише кроз хладњак машине, долази до размене топлоте између расхладне технике машине и воде, машина се хлади, а расхладна вода, чија је температура после размене топлоте сада већа, враћа се повратним цевоводом назад у расхладну станицу где се процес хлађења понавља.

Рад постројења током летњег периода:

Концентрована смеша воде и гликола се потисним водом води од ваздушног хладњака ка расхладном агрегату, у овом случају троктаки вентил се отвара и пушта проток воде директно ка примарном делу расхладног агрегата, тиме затварајући трасу тока ка плочастом измењивачу топлоте. Расхладни агрегат смањује температуру воде на страни секундара на задату вредност.

Охлађена вода се преко циркулационих пумпи води у акумулациони резервоар где се меша са водом са стране поврата, али је промена температуре незнатна. Хладна вода се даље преко циркулационих пумпи води кроз челични, а затим на уласку у производни погон, кроз пластични цеводод до крајњег корисника - машина брушење.

Хладна вода циркулише кроз хладњак машине, долази до размене топлоте између расхладне течности машине и воде, машина се хлади, а расхладна вода, чија је температура после размене топлоте сада већа, враћа се повратним цевоводом назад у расхладну станицу где се процес хлађења понавља.

2.3 Технолошко хлађење пећница

Адијабатски хладњак обезбеђује потребну температуру концентроване смеше воде и етилен-гликола који се користи за хлађење пећница. За разлику од ваздушног хладњака („dry cooler“), адијабатски хладњак поред својих вентилатора који хладе расхладни флуид, испод свог саћа измењивача има изведен цевни развод прскалица кроз које се под високом притиском избацује омекшана вода по саћу измењивача и додатно расхлађује смешу.

3. ХИДРАУЛИЧКИ ПРОРАЧУН

3.1 Дефинисање расхладних капацитета и општи прорачун

У табели 1 приказане су полазне величине неопходне за хидраулички прорачун свих расхладних кругова:

Табела 1: Полазне величине

Величине/круг	1	2	3	4
W [kW]	374	294	294	120
t_1/t_2 [°C/°C]	45/40	16/12	16/12	40/30
d_u [mm]	131,7	131,7	131,7	70,3
L [m]	40	15	140	160
$\Sigma \zeta$	6	6	6	8
ν [m ² /s]	$2 \cdot 10^{-6}$	$0,5 \cdot 10^{-6}$	$0,5 \cdot 10^{-6}$	$2 \cdot 10^{-6}$
ε [mm]	0,2	0,15	0,15	0,2
c_p [kJ/kgK]	3,9	4,2	4,2	3,9

при чему је:

W [kW] – расхладни капацитет,
 t_1/t_2 [°C/°C] – температурни режим,
 d_u [mm] – унутрашњи пречник цеви,
L [m] – укупна дужина цевовода,
 $\Sigma \zeta$ – сума локалних отпора,
 ν [m²/s] – кинематска вискозност,
 ε [mm] – апсолутна храпавост,
 c_p [kJ/kgK] – специфична топлота,

док су расхладни кругови означени на следећи начин:

- 1 – технолошко хлађење машина за брушење – кондензаторски круг,
- 2 – технолошко хлађење машина за брушење – примарни испаривачки круг,
- 3 – технолошко хлађење машина за брушење – секундарни испаривачки круг,
- 4 – технолошко хлађење пећница.

На основу полазних вредности добија се масени проток:

$$\dot{m} = \frac{W}{C_p \cdot \Delta t} \quad (1)$$

Брзина концентроване смеше кроз цевни развод износи:

$$v = \frac{\dot{m}}{(\rho \cdot d^2 \cdot \pi) \cdot 4} \quad (2)$$

Релативна храпавост износи:

$$\varepsilon_r = \frac{\varepsilon}{d} \quad (3)$$

где је:

ε – апсолутна храпавост,
 d – унутрашњи пречник цеви.

Рејнолдсов број:

$$R_e = \frac{v \cdot d}{\nu} \quad (4)$$

где је:

ν – кинематска вискозност.

Коефицијент трења се израчунава:

$$\lambda = \frac{1,325}{\ln \left(\frac{e_r}{3,7} + \left(\frac{5,74}{R_e} \right)^{0,9} \right)^2} \quad (5)$$

,па је јединични пад притиска услед трења:

$$\Delta P_i = \frac{\frac{\lambda}{d} \cdot \rho \cdot v^2}{2} \quad (6)$$

Пад притиска услед линијског отпора добија се:

$$\zeta_i = L \cdot \Delta P_i \quad (7)$$

где је:

L – дужина трасе цевовода.

Пад притиска услед локалних отпора рачуна се:

$$\zeta_i = \sum \zeta_i \cdot \rho \cdot \frac{v^2}{2} \quad (8)$$

Укупан пад притиска у цевоводу износи:

$$\Delta P = \zeta_i + \zeta_i \quad (9)$$

док је потребан проток:

$$Q = \frac{m}{\rho} \quad (10)$$

На основу протока и укупног пада притиска добија се радна тачка пумпе, на основу које се димензионише пумпа. [1][2][3]

3.2 Резултати прорачуна

На основу усвојених вредности из табеле 1 и релација [1]-[10] добијене су вредности за укупан пад притиска у цевној мрежи и потребан проток кроз сваки од наведених расхладних кругова.

Табела 2: Укупни пад притиска у кондензаторском кругу

Др притиска у цевној мрежи	12.506 Pa
Др на хватачу нечистоће DN125	12.000 Pa
Др у измењивачу топлоте сувог хладњака	35.000 Pa
Др у размењивачу топлоте расхладног агрегата	65.000 Pa
Др у неповратној клапни DN125	15.000 Pa
УКУПНО	139.506 Pa

На основу добијених вредности потребног напора (табела 2) и протока у кондензаторском кругу у оквиру технолошког хлађења машина за брушење, усвајају се две пумпе (радна + резервна) следећих карактеристика:

Произвођач: „Grundfos“

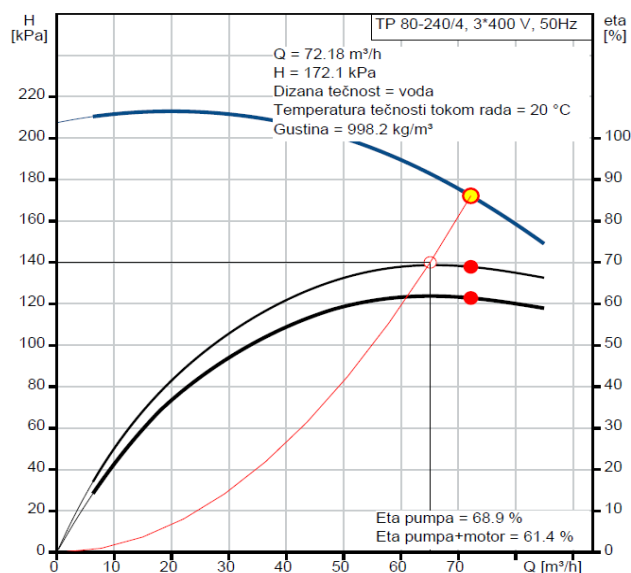
Тип: TP 80-240/4,

Радна тачка:

- проток: 65,1 m³/h

- напор: 140 kPa

Крива пумпе Grundfos TP 80-240/4 приказана је на слици 1.



Слика 1: Крива пумпе Grundfos TP 80-240/4

Табела 3: Укупни пад притиска у примарном испаривачком кругу

Др у цевној мрежи	7.105 Pa
Др на хватачу нечистоће DN125	12.000 Pa
Др у измењивачу топлоте расхладног агрегата	40.000 Pa
Др у неповратној клапни DN125	15.000 Pa
УКУПНО	74.105 Pa

На основу добијених вредности потребног напора (табела 3) и протока у примарном испаривачком кругу у оквиру технолошког хлађења машина за брушење, усвајају се две пумпе (радна + резервна) следећих карактеристика:

Произвођач: „Grundfos“

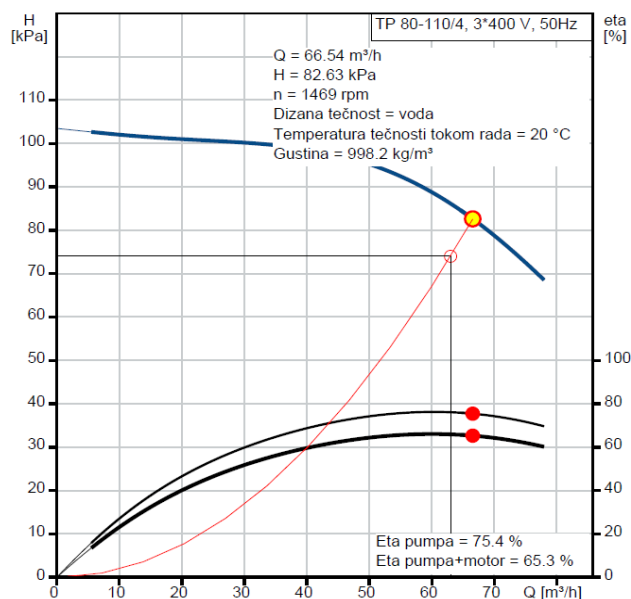
Тип: TP 80-110/4,

Радна тачка:

- проток: 63 m³/h

- напор: 74 kPa

Крива пумпе Grundfos TP 80-110/4 приказана је на слици 2.



Слика 2: Крива пумпе Grundfos TP 80-110/4

Табела 4: Укупни пад притиска у секундарном испаривачком кругу

Δр у цевној мрежи	24.081 Pa
Δр на хватачу нечистоће DN125	12.000 Pa
Δр у измењивачу топлоте машине за брушење	65.000 Pa
Δр у неповратној клапни DN125	15.000 Pa
УКУПНО	116.081 Pa

На основу добијених вредности потребног напора (табела 4) и протока у секундарном испаривачком кругу у оквиру технолошког хлађења машина за брушење, усвајају се две пумпе (радна + резервна) следећих карактеристика:

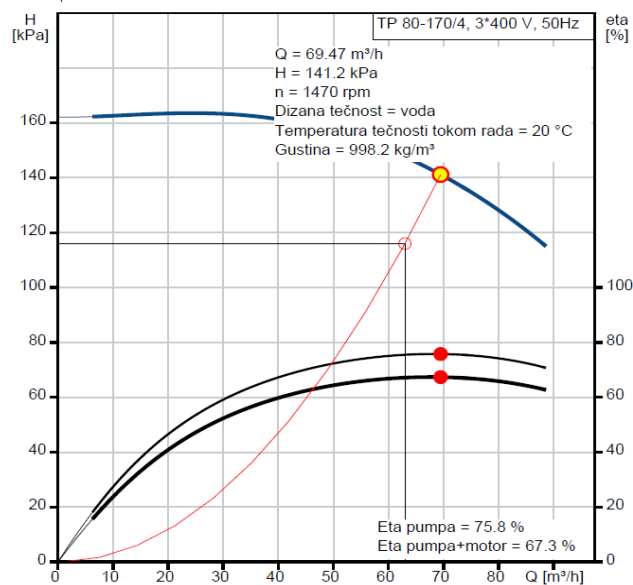
Произвођач: „Grundfos“

Тип: **TP 80-170/4**,

Радна тачка:

- проток: **63 m³/h**
- напор: **116 kPa**

Крива пумпе Grundfos TP 80-170/4 приказана је на слици 3.



Слика 3: Крива пумпе Grundfos TP 80-170/4

Табела 5: Укупни пад притиска у расхладном кругу за технолошко хлађење пећница

Δр у цевној мрежи	22.972 Pa
Δр на хватачу нечистоће DN65	12.000 Pa
Δр у измењивачу топлоте адијабатског хладњака	42.000 Pa
Δр у измењивачу топлоте пећнице	55.000 Pa
Δр у неповратној клапни DN100	15.000 Pa
УКУПНО	146.972 Pa

На основу добијених вредности потребног напора (табела 5) и протока у расхладном кругу за технолошко хлађења пећница, усвајају се две пумпе (радна + резервна) следећих карактеристика:

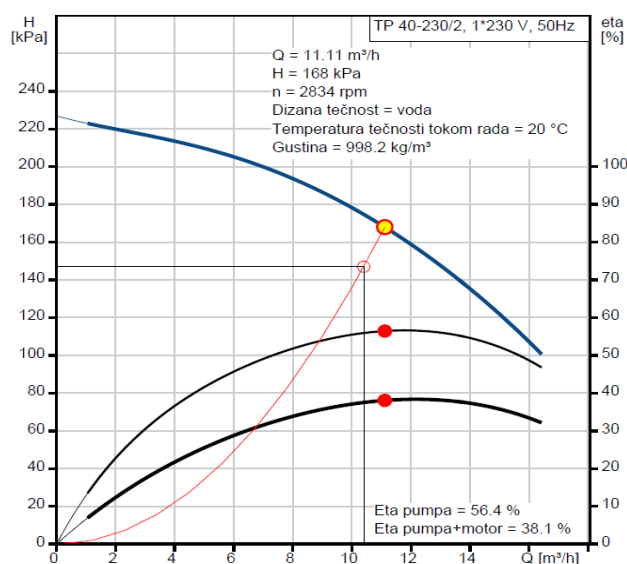
Произвођач: „Grundfos“

Тип: **TP 40-230/2**,

Радна тачка:

- проток: **10,4 m³/h**
- напор: **147 kPa**

Крива пумпе Grundfos TP 40-230/2 приказана је на слици 4 [4].



Слика 4: Крива пумпе Grundfos TP 40-230/2

4. ЗАКЉУЧАК

Инсталација за технолошко хлађење машина у производном погону подељена је у два независна система: Један је технолошко хлађење машина за брушење, а други технолошко хлађење пећница. Из хидрауличног прорачуна кондензаторског круга система технолошког хлађења машина за брушење добијен је проток од 65,1 m³/h, као и укупни пад притиска од 140 kPa, те је усвојена пумпа Grundfos TP 80-240/4.

Из хидрауличног прорачуна испаривачког круга - примар за систем технолошког хлађења машина за брушење, добијене су вредности потребног протока и напора од 63 m³/h и 74 kPa, па је усвојена пумпа Grundfos TP 80-110/4. За испаривачки круг - секундар система технолошког хлађења машина за брушење, добијене су вредности потребног протока и напора од 63 m³/h и 116 kPa и усвојена је пумпа Grundfos TP 80-170/4. За систем технолошког хлађења пећница, добијене су вредности потребног протока и напора од 10,4 m³/h и 147 kPa, па је усвојена пумпа Grundfos TP 40-230/2.

5. ЛИТЕРАТУРА

- [1] Расхладни уређаји, Миле Маркоски, Универзитет у Београду, Машински факултет, 2003. год.
- [2] Расхладни уређаји и инсталације – монтажа, пуштање у рад, одржавање и поправке, Стеван Шамшаловић, СМЕИТС, Београд, 2012. год.
- [3] Грејање и климатизација, Recknagel, Sprenger, Schramek, Чеперковић, АГМ књига, 2012. год.
- [4] Grundfos website, <https://www.grundfos.com/rs>

Кратка биографија:



Јован Косановић рођен је у Руми 1992. год. Мастер рад на Факултету техничких наука из области Енергетика и процесна техника одбранио је 2022. год.

Контакт: j.kosanovic@outlook.com

НИВЕЛИСАЊЕ ПРОЦЕСА МАШИНСКЕ ОБРАДЕ ПРИМЕНОМ HEIJUNKE**LEVELING THE MACHINING PROCESS BY USING HEIJUNKA**Мирјана Дабић, Сања Бојић, *Факултет техничких наука, Нови Сад***Област – МАШИНСТВО**

Кратак садржај – У раду су дате теоријске основе LEAN методологије и представљени су неки од основних LEAN алата. Посебан акценат је стављен на примену LEAN алата - Heijunka, у конкретном примеру за нивелисање процеса машинске обраде.

Кључне речи: Оптимизација, Lean филозофија, lean алати, Heijunka, логистика, интралогистика

Abstract – The paper presents the theoretical foundations of the LEAN methodology and some of the basic LEAN tools. A special emphasis is placed on the application of certain LEAN tool - Heijunka, in concrete example of leveling the machining process.

Keywords: Optimization, Lean, Lean tools, Heijunka, Logistics, Intralogistics

1. УВОД

У претходних неколико година многе производне организације су препознале да је један од основних стратешких елемената којим се може остварити предност у односу на конкуренцију заправо супериорност њихових услуга. У окружењу се може пронаћи велики број инструмената који су доступни, али упознавање са њима и њихово имплементирање није најлакши процес. Lean модел размишљања овде проналази своју улогу.

Такође, у протеклих 20 година, реална цена робе широке потрошње опада широм света, а с друге стране квалитет производа се повећава, а број дефеката се смањује. Организације које не успеју да уреде своје процесе и активности тако да се њима остварују одређене уштеде и елиминишу активности које не додају вредност, неће опстати на тржишту.

Порекло lean размишљања лежи у приступу lean производње типизираним Toyota производним системом (TPS) као свеобухватном стратегијом за елиминацију отпада из операција.

Елиминисање отпада постиже се на различите начине, али оно од чега би се почело јесте мапирање тока вредности операција које испоручују производе у кратком временском периоду.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је била др Сања Бојић, ванр. проф.

Циљ овог рада јесте да се прикаже могућност примене lean алата у систему, односно процесу машинске обраде.

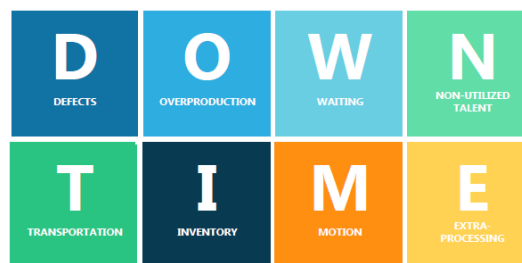
2. LEAN МЕТОДОЛОГИЈА

Примењено на пословне организације, lean означава организацију која је максимално ефектна у ономе што ради, трошећи минимум потребних ресурса, а пружајући најбољи могући квалитет производа или услуга. Lean не значи само постизање оваквог стања, већ и посвећеност низу принципа и практичних вежби које ово жељено стање континуирано одржавају. Организација која је lean успешно побеђује своју конкуренцију, бржа је, окретнија, мање троши, више постиже.

Lean је термин који описује одрживи приступ пословању (организацији) који користи мање ресурса, инвестиција, трошкова, а пружа више квалитета, профита и задовољства купаца. Lean је метода пословања базирана на задовољењу купаца испоруком квалитетних производа и услуга који су баш оно што купац жели, кад их жели, у жељеној количини, по правој цени, користећи минимум материјала, опреме, простора, рада и времена.

Главни принцип lean-а је дугорочно, обновљиво пословање, где се стално изграђују и унапређују односи са партнерима, купцима и клијентима (stakeholders) укључујући ту и запослене.

Како би се губици лакше уочили у систему и лакше елиминисали, груписани су у осам категорија:



Слика 1. Губици представљени преко акронима

- **Грешке (Defects)** Односи се на грешке на производима, сваки квар и свака поновна дорада одређеног производа се сматра грешком.
- **Прекомерна производња (Overproduction)** Односи се на производњу нечега пре него што то тржиште захтева, те се сматра за најгори тип

губитака према lean-у јер је главни узрок настанка свих осталих губитака.

- **Чекање (Waiting)** Односи се на ситуације када информације, запослени, опрема или материјали нису спремни.
- **Неискоришћен људски потенцијал (Non utilized talent)** Односи се са на неискориштење пуног потенцијала запосленог. Чињеница је да запослени на свом радном месту најбоље знају који су проблеми, а фирме једноставно не пружају могућност запосленима да учествују у решавању тих истих проблема.
- **Транспорт (Transport)** Односи се на премештање материјала са једне локације на другу. На самом производу транспорт не врши никакву промену, па самим тим не додаје вредност производу. Поред тога, транспорт често доводи до тога да операције морају чекати.
- **Превелик ниво залиха (Inventory)** Односи се на готове производе, полу-производе и сирови материјал који нису продати купцима. Залихе су увек губитак јер заровљавају обртна средства предузећа.
- **Непотребна кретања (Motion)** Односи се на непотребно кретање запослених и машина између радних станица или објеката у производним процесима у којима се обављају све операције.
- **Прекомерна обрада (Extra processing)** Односи се на стварање производа прекомерног квалитета који купац не захтева. Прекомерном обрадом сматрају се и активности поправке производа који није према захтевима купца.

3. АЛАТИ LEAN МЕТОДОЛОГИЈЕ

Најчешће праксе које се односе на lean производњу везују се за: отклањање уских грла, програм континуираних побољшања, смањење времена, правовремену производњу, унапређење одржавања, превентивно одржавање, мерење способности, Kanban, програме управљања квалитетом, реинжењеринг, програме за побољшање безбедности, итд.

5S представља основу lean производње и темељ за дисциплинован приступ приликом одржавања радног места чистим и организованим и безбедним за рад. 5S је јако користан lean алат којим можемо побољшати ефикасност не само производног погона, већ и свакодневног живота.

5S долази од јапанских ријечи Seiri (сортирати), Seiton (спремити), Seiso (средити), Seiketsu (стандардизовати) и Shitsuke (стално одржавати). Овај поступак описује како организовати радни простор ефикасно и ефективно, идентификујући и слажући коришћене предмете, одржавање предмета и простора и одржавање новог поретка. Имплементацијом 5S остварује се повећање сигурности, квалитета, продуктивности и поузданости машина.



Слика 2. Циклус 5S побољшања

Стандардизација подразумева креирање процедура и радних инструкција на основу којих оператери знају начин и време трајања операција на сваком радном месту у производном процесу. Стандардизацијом се смањују трошкови, повећава продуктивност, стабилизује време испоруке, елиминишу губици и поједностављују процеси.

Визуелни менаџмент представља континуирани процес стварања радног места на којем је елиминисан дефицит информација, тј. процес стварања окружења где су ствари очигледне од првог тренутка када су се нашли у датом подручју. Најбољи пример примене визуелног менаџмента су управо саобраћајни знаци, јер подразумева кориштење боја и сигнала уместо текста, тако да се информација преноси брже и јасније.

Овај алат служи за доношење одлуке на микро нивоу. На пример, визуелни менаџмент сликовито и знаковно помаже запосленима да уштеде време и да на лак начин приме одређену количину информација како би им рад био продуктивнији. Визуелни менаџмент користи одређена средства којима врши своју функцију, а неки од најчешћих средстава су андон светла, табле са циљевима за праћење учинка, табле за дељење идеја и за комуникацију тима, и kaizen извештаји и награде.

SMED се користи за повећање флексибилности машине, повећање капацитета машине и смањења трошкова финалног производа.

Врло је популаран начин приказивања кориштења алата и његових могућности кроз пример формуле 1. Најбржа измена гума у пит стопу реализована је у временском трајању од 1,92 секунде, слика 3.

Успоредбе ради, према доступним подацима, током 1950-их за измену гума било је потребно и више од 60 секунди.



Слика 3. Промена гуме формуле 1 за 1,92s

Push – pull принцип

Код push система производ се може изградити и када постоји мала или никаква потражња. Ови процеси могу створити велике залихе и трошкове. Ако следећи корак у процесу није спреман, могу се створити велики проблеми.

Pull систем, за разлику од претходног, значи реаговање на потражњу купца, односно разумевање захтева купца за производом, услугом. На основу тога, креира се процес који треба да одговори на тај захтев. Lean треба да обезбеди да кроз ток вредности теку само они производи и услуге које жели купац.

Непрекидни ток производње је други основни принцип „баш на време“ производње. Непрекидни ток производње се заснива на концепту елиминације заустављања и поновног кретања повезаних са производњом те ради усклађено са вучењем производње у сврху смањења међуфазних времена и залиха.

Идеално гледано, непрекидни ток производње (једнопредметни ток) се постиже производњом једног комада за другим. Једнопредметни ток је супротан серијској производњи. Уместо да се обрађује више истих производа, који након обраде чекају следећу операцију у процесу, сваки се производ појединачно производи без прекидања тока.

Heijunka је јапански назив алата LEAN методологије који у преводу значи нивелисање, уједначавање. Предност примене алата у производњи је редуковање губитака уз задовољавање потреба тржишта за производима.

Користи се за нивелисање количина код више врста производа како би се спречило претерано гомилање залиха редукујући капиталне трошкове, рад људи и водеће време производње дуж целог ланца додавања вредности. Heijunka као и остали алати, први пут је коришћена у Тојоти у склопу Toyota Production Systema. Кад је једном имплементиран, алат омогућује предвидивост нивелисањем захтева, флексибилност скраћивањем времена измене и

стабилност нивелисањем и врсте производа током дужег периода.

Да би се практично могло остварити уједначавање производње, користи се алат који се назива Heijunka кутија. Heijunka кутија је алат у облику поштанских сандучића (обично причачених на зид) који служи за заказивање планираних процеса у производњи. Сваки ред кутија представља компоненту у процесу док свака колона представља временски период (нпр. неколико сати, дана, седмица, итд.) који представља најкраће време да се достави једна јединица производа. Обојене

Канбан картице се постављају у Heijunka кутије, а свака Канбан картица представља потребу за једном јединицом производа што пружа визуелни преглед предстојећих потреба производње.

Нивелисање производње не може да се врши ако нису примењени други алати као што су 5S, визуелни менаџмент, стандардизовани процеси, SMED, непрекидни ток итд.

4. НИВЕЛИСАЊЕ ПРОЦЕСА МАШИНСКЕ ОБРАДЕ ПРИМЕНОМ HEIJUNKE

У раду је посматрана конкретна примена heijunka у систему машинске обраде у процесу ремонта турбина.

Ради се о компанији која постоји више десетина година и препозната је на тржишту по квалитету ремонта различитих врста турбина (турбине из путничких и теретних возила, као и из грађевинских машина).

Ова компанија већ има имплементиране lean алате 5S, визуелни менаџмент, стандардизацију рада, с тим да су у току увођење непрекидног тога и pull принципа.

Све то представља подлогу за највећи изазов - увођење heijunka.

4.1 Преглед тренутног стања процеса машинске обраде

Процес машинске обраде обухвата 9 процеса, а то су: демонтажа, прање, пескарење, глодачка дорада, стругарска дорада, контрола, централни склоп, монтажа и калибрација.

С обзиром да компанија ремонтује различите врсте турбина у циљу оптимизације извршено је њихово груписање у 16 категорија, при чему је извршено нормирање трајања процеса за сваку од наведе категорије. Скоро сви типови турбина пролазе кроз скоро све процесе.

Компанија ради 6 радних дана, по једну смену од 8 сати у току које постоји једна пауза од 30 мин. Потребно је формирати план за просечна времена. Такт за машинску обраду је 50 турбина дневно, односно 9 минута по турбини.

Да би се дошло до података потребних за нивелисање, за почетак је потребно израчунати укупно потребан број радника, као и потребан број радника за сваки

процес како би се могао испоштовати непрекидни ток, pull принцип и такт од 9 минута по турбини.

Анализом се дошло до закључка да би се дефинисани такт могао испоштовати са 16 радника.

4.2. Преглед изнивелисаног стања процеса машинске обраде

Време/Процеси	1	2	3	4	5	6	7	8	9
7:00:00	Долазак на радно место. Провера машина и алата. Припрема за рад.								
7:09:00	K5	K10	K7	K13	K2	K5	K9	K10	K8
7:18:00	K9	K8	K16	K7	K13	K2	K5	K9	K5
7:27:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
7:36:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
7:45:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
7:54:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
8:03:00	K5	K10	K7	K13	K2	K5	K9	K10	K8
8:12:00	K9	K8	K16	K7	K13	K2	K5	K9	K5
8:21:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
8:30:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
8:39:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
8:48:00	K10	K2	K9	K15	K8	K5	K14	K3	K10
8:57:00	K1	K10	K2	K9	K15	K8	K5	K14	K3
9:06:00	K5	K10	K2	K9	K15	K8	K5	K14	K3
9:15:00	K10	K2	K9	K15	K8	K5	K14	K3	K10
9:24:00	K5	K10	K2	K9	K15	K8	K5	K14	K3
9:33:00	K10	K2	K9	K15	K8	K5	K14	K3	K10
9:42:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
9:51:00	K5	K10	K7	K13	K2	K5	K9	K10	K8
10:00:00	K9	K8	K16	K7	K13	K2	K5	K9	K5
10:09:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
10:18:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
10:27:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
10:36:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
10:45:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
10:54:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
11:03:00	K5	K10	K7	K13	K2	K5	K9	K10	K8
11:12:00	K9	K8	K16	K7	K13	K2	K5	K9	K5
11:21:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
11:30:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
11:39:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
11:48:00	K10	K2	K9	K15	K8	K5	K14	K3	K10
11:57:00	K1	K10	K2	K9	K15	K8	K5	K14	K3
12:06:00	K5	K10	K2	K9	K15	K8	K5	K14	K3
12:15:00	K10	K2	K9	K15	K8	K5	K14	K3	K10
12:24:00	K5	K10	K2	K9	K15	K8	K5	K14	K3
12:33:00	K10	K2	K9	K15	K8	K5	K14	K3	K10
12:42:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
12:51:00	K5	K10	K7	K13	K2	K5	K9	K10	K8
13:00:00	K9	K8	K16	K7	K13	K2	K5	K9	K5
13:09:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
13:18:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
13:27:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
13:36:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
13:45:00	K10	K2	K9	K15	K8	K5	K14	K3	K10
13:54:00	K5	K10	K2	K9	K15	K8	K5	K14	K3
14:03:00	K10	K2	K9	K15	K8	K5	K14	K3	K10
14:12:00	K5	K10	K2	K9	K15	K8	K5	K14	K3
14:21:00	K10	K2	K9	K15	K8	K5	K14	K3	K10
14:30:00	K1	K9	K8	K16	K7	K13	K2	K5	K9
14:39:00	K5	K10	K7	K13	K2	K5	K9	K10	K8
14:48:00	K9	K8	K16	K7	K13	K2	K5	K9	K5
14:57:00	Чековање радног места. Провера машина и алата. Припрема радног места за наредни дан.								
15:00:00									

Слика 4. Преглед изнивелисаног стања процеса машинске обраде

5. ЗАКЉУЧАК

У данашње време се све више озбиљних компанија и производних система који производе производе или пружају услуге, опредељује управо за примену LEAN концепта и филозофије. Како се тежи унапређењу битно је кренути од начина размишљања и много дубље ући у суштину проблема целокупног система. Дакле, LEAN није само алат којим се постижу жељени резултати, већ је то животна филозофија и примењива је у многим сферама.

LEAN концепт није ни мало једноставан и треба му се посветити са пуном пажњом. Дешава се да компаније не увиде значај примене ове методологије, управо јер не разумеју срж и могућности концепта. Понекад је потребно прво променити начин размишљања, па онда приступити примерни LEAN-a.

Фокус је увек на унапређењу система, уз што мање губитке како би се остварила максимална добит. Како би се остварило управо то што је претходно наведено и како би се постигла оптималност било ког система потребно је да се остваре основе које су темељ за успешност пословања компаније. Основне ствари којима се тежи јесте да се производња одвија по принципу Just-In-Time и да буде остварен непрекидан ток материјала.

6. ЛИТЕРАТУРА

- [1] Др Бојић С.: Презентација: LEAN алати, ФТН, Нови Сад, 2020.
- [2] Бекер И., Лазаревић М., Морача С., Радловачки В., Рикаловић А., Тешкић З., Шевић Д.: LEAN систем, ФТН Издаваштво, Нови Сад, 2017.
- [3] Вомак Џ., Џонс Д.: Филозофија LEAN концепта, ФТН Издаваштво, Нови Сад, 2012.
- [4] Шинго С., „A revolution in manufacturing: The SMED system“. Productivity Press, Станфорд, 1985.
- [5] Шинго С., „A Study of the Toyota Production System“, Productivity Press, 2008.

Кратка биографија:



Мирјана Дабич рођена је у Добоју 1997. год. Дипломски рад на Факултету техничких наука из области Грађевинске и комуналне машине - „Преглед машина за ископ земљишта са непрекидним радом и подлоге за избор основних параметара“ одбранила је 2021. године.
контакт: dabic.mirjana7@gmail.com



Сања Бојић рођена је у Карловцу 1981. Докторирала је на Факултету техничких наука 2013. год, а од 2019. је у звању ванредни професор. Област интересовања су логистика, складишта и симулације.

**ERGONOMSKE KARAKTERISTIKE VILJUŠKARA NA UTICAJ RUKOVAOCA I
BEZBEDNOSNI USLOVI KORIŠĆENJA VILJUŠKARA****ERGONOMIC CHARACTERISTICS OF FORKLIFTS ON THE INFLUENCE OF THE
OPERATOR AND SAFETY CONDITIONS FOR USING THE FORKLIFT**

Ervin Molnar, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Osnovni cilj ovog rada jeste sagledavanje ergonomske karakteristike kabine viljuškara, bezbednosnih uslova korišćenja viljuškara i davanje predloga za njihovo unapređenje kako bi se smanjio zamor viljuškariste, a povećao komfor i produktivnost. U ovom radu prikazani su ergonomske aspekte unutrašnjosti kabine čeonih viljuškara na operatera - viljuškaristu, na koje delove tela najviše deluju i analizirani su praktični rezultati gde je najveći problem okarakterisan kao bol u lumbalnom delu leđa.

Ključne reči: ergonomija, viljuškar, viljuškarista, bezbednost, povrede

Abstract – The main goal of this paper is to review the ergonomic characteristics of the forklift cabin, the safety conditions of using the forklift and to make suggestions for their improvement in order to reduce the fatigue of the forklift operator and increase comfort and productivity. In this paper, the ergonomic aspects of the interior of the cabin of front-end forklifts are shown on the operator - the forklift driver, which parts of the body they affect the most. Practical results were analyzed where the biggest problem was characterized as pain in the lumbar part of the back.

Keywords: ergonomics, forklift, forklift driver, safety, injuries

1. UVOD

Viljuškar predstavlja vozilo unutrašnjeg transporta namenjen za transport tereta na kraćim relacijama koje je pogodno za podizanje, slaganje, spuštanje, prihvatanje i odlaganje tereta zahvatnim sredstvima u obliku viljušaka do svoje označene nosivosti. Akcenat će se staviti na čeonu viljuškare, jer su najzastupljeniji u praksi. Daće se osvrnuti na potencijalne izvore opasnosti usled nepravilne manipulacije teretom, kretanje po neadekvatnom terenu, kretanje velikim brzinama u krivinama, kao i mere koje je potrebno ispuniti kako bi se sve potencijalne opasnosti izbegle. Sagledaće se sve odgovornosti viljuškariste kao i signaliste za siguran i bezbedan rad. Ergonomija igra veliku ulogu u automobilske industriji pa tako i u industriji radne mehanizacije.

NAPOMENA:

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

Izvršice se pregled unutrašnjosti kabine viljuškariste i raspored komandi za upravljanje. Opisao se pozitivne i negativne stavke uticaja ergonomije na viljuškaristu, najčešći zdravstveni problemi usled loše ergonomije i predložio se rešenja da se isti smanje, tj. svedu na minimum. Izvršice se podela na direktne i indirektno faktore koji dejstvuju na delove tela viljuškariste usled korišćenja viljuškara.

Bezbedan i siguran rad je od vitalnog značaja, iz tog razloga će se prikazati praktičan rad sa utovarom cevog materijala i potrebne zaštitne mere koje je potrebno ispuniti. Izvršice se ispitivanje viljuškarista i prikazaće se koje su najčešće povrede koje im se javljaju. Izvršice se analiza rezultata dobijenih na osnovu ispitivanja osmorice viljuškarista starosne dobi od 29 do 44 godina, gde će se prikazati koja su ključna mesta na kojima se javlja bol usled manipulacije viljuškara [2].

**2. PREGLED VILJUŠKARA I KONTROLA
ISPUNJENOSTI ZA BEZBEDAN RAD
VILJUŠKAROM**

Viljuškarista pre rada mora detaljno da se upozna sa uputstvom proizvođača za rad sa viljuškarom. Pre početka svakog rada viljuškarista treba da proveri ispravnost viljuškara. Detaljno su objašnjeni koraci koje viljuškarista mora da uradi pre svake upotrebe viljuškara.

2.1. Mere zaštite

Svaki viljuškar mora biti pregledan od strane ovlašćene ustanove, bez obzira na proteklo vreme od prethodnog pregleda.

2.2. Obaveze i odgovornosti signaliste

Cilj signaliste je da olakša rad viljuškaristi i poveća bezbednost tokom manipulacije viljuškara. Signalista je lice koje je obučeno za davanje znakova viljuškaristi pri manipulaciji sa teretom, pri čemu viljuškarista sa svog radnog mesta nije u mogućnosti da delimično ili u potpunosti prati kretanje tereta pri dizanju, prenošenju i spuštanju tereta. Signalista daje viljuškaristi uobičajene odnosno ugovorene znake rukom/rukama za manevrisanje. Praktično je odrađena vežba bezbednog utovara cevog materijala dužine 9.2 m, težine 2 tone u kamion sa prikolicom. Utovar se sastoji od 6 koraka, prilazak materijalu, njegovo podizanje, kretanje ka kamionu, prilaz kamionu, spuštanje materijala i izlazak iz procesa. U tabeli 1 su detaljno objašnjeni signali [3].

Značenje	Opis	Ilustracija
A. Opšti znaci		
POČETAK Pažnja Početak radnje	Obe ruke su raširene horizontalno sa dlanovima okrenutim napred.	
STANI Prekid Zaustavljanje radnje	Desna ruka je podignuta u vis sa dlanom okrenutim napred.	
KRAJ kraj radnje	Ruke su sklopljene u visini grudi.	
B. Vertikalno kretanje		
PODIGNI	Desna ruka podignuta u vis sa dlanom okrenutim napred, polako kruži.	
SPUSTI	Desna ruka spušta se sa dlanom okrenutim nazad, polako kruži.	
VERTIKALNO RASTOJANJE	Položaj ruku pokazuje rastojanje.	
C. Horizontalno kretanje		
NAPRED	Obe ruke su savijene sa dlanovima okrenutim na gore i podlaktice se polako kreću ka telu.	
NAZAD	Obe ruke su savijene sa dlanovima okrenutim na dole i podlaktice se polako kreću od tela.	
DESNO	Desna ruka u visini ramena sa dlanom okrenutim na dole polako daje male pokrete u desno.	
LEVO	Leva ruka u visini ramena sa dlanom okrenutim na dole polako daje male pokrete u levo.	
HORIZONTALNO RASTOJANJE	Položaj ruku pokazuje rastojanje	
D. Opasnost		
OPASNOST Zaustavljanje u slučaju opasnosti	Obe ruke su podignute u vis sa dlanovima okrenutim napred.	
BRZE	Svi pokreti brže	
SPORIJE	Svi pokreti sporije	

Tabela 1. Vrste signala [3]

3. ERGONOMIJA

Objašnjen je pojam ergonomije u industriji radne mehanizacije tačnije kod viljuškara. Ciljevi ergonomije uključuju stvaranje sigurnih i zdravih radnih uslova, održavanje udobnosti i psihosocijalnog blagostanja i stvaranje dobrih performansi sistema u pogledu produktivnosti i kvaliteta.

3.1 Zdravstveni problemi viljuškarista usled loše ergonomije

Najčešći zdravstveni problemi uočeni kod viljuškarista tokom istraživanja su povreda vrata i ramena, kao i

zglobova, članaka, kolena, stopala i donjeg dela leđa. Najveći procenat bolova se odnosi na lumbalni deo leđa i iznosi preko 70%. Vozači viljuškara imaju dva puta veće šanse da zadobiju povrede donjeg dela leđa od zaposlenih koji ne voze viljuškar. Čak 67% viljuškarista navelo je radni prostor kao faktor povreda, 33% viljuškarista navelo je dizajn viljuškara kao problem nastanka povreda, dok je 8% navelo sistem čovek mašina, ličnu zaštitnu opremu i prevenciju kao uzrok povreda. Indirektno se može zaključiti da dizajn/ergonomija viljuškara ima najveći uticaj na povrede usled vožnje viljuškara. Kompanija „Linde“ je pronašla rešenje za smanjenje bola u vratu kreiranjem rotirajućeg sedišta. Sedište se može rotirati za 17° za smer kretanja unapred i za 90° stepeni pri kretanju unazad [2,4].

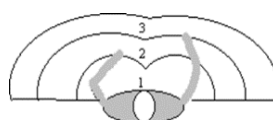
3.2 Ergonomija enterijera viljuškara – raspored komandi

Izvršen je pregled komandi i njihov domet u odnosu na poziciju viljuškariste. Komande se dele na nožne i ručne.



Slika 1 Komande viljuškara

Radni domet koji je predaleko od viljuškariste zahtevaće stresno držanje ramena, laktova, zgloba i leđa. Posezanje u stranu, iza ili predaleko ispred tela može dovesti do stresa na manje mišiće. U idealnom slučaju, radni domet bi trebalo da bude unutar normalnog dometa, sa kontrolama i materijalima kojima se rukuje najbliže telu. Takođe, idealno je imati kontrole koje obavljaju slične ili kombinovane funkcije grupisane zajedno, kako bi se smanjili neugodni položaji koji bi inače mogli da se pojave.

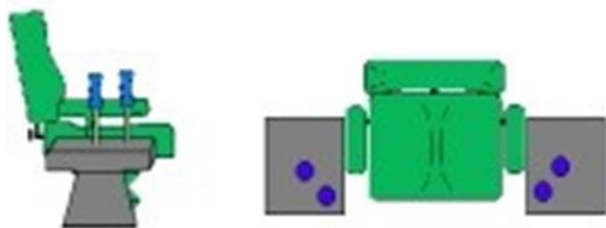


- 1 = Često korišćene kontrole/stavke
- 2 = Redje korišćene kontrole/stavke
- 3 = Najmanje korišćene kontrole/stavke

Slika 2 Radni dohvati ruku viljuškariste [4]

Generalno, najčešće korišćeni predmeti treba da budu postavljeni na dohvati podlaktice, sa manje često korišćenim predmetima na dohvati ruke, a retko korišćeni predmeti na dohvati ruke potpuno ispružene. Radni dohvati do nožnih i ručnih komandi mogu biti individualizovani za operatore obezbeđivanjem sedišta koje je podesivo napred/nazad. Ova mogućnost prilagođavanja će omogućiti operaterima

da lociraju kontrole u okviru udobnog doseg. Komande viljuškara i druge ručne komande treba postaviti sa strane rukovaoca, kako bi se sprečilo ponavljanje posezanja napred i nezgodno držanje ramena i leđa. Dodatak naslona za ruke će pomoći u podržavanju ramena dok se komande koriste [4].



Slika 3 Podešavajuće sedište i ručne komande [4]

Radna visina koja je previsoka za radnika zahtevaće stresno držanje ramena i ruku, dok će visina koja je preniska zahtevati stresno savijanje vrata i trupa. Visina radne površine treba da omogući prostor za promenu položaja i pomeranje nogu i stopala. Idealna radna stanica je podesiva po visini, što omogućava velikom percentilu stanovništva da prilagodi visinu radne površine svojim dimenzijama.

Da bi odredili odgovarajuću visinu specifičnu za rukovaoca viljuškara, potrebno je identifikovati deo tela koji najviše brine. Ako je glavna briga:

- vrat - minimizirati savijanje vrata napred povećanjem radne visine;
- ramena - minimizirati podizanje ruku spuštanjem radne visine;
- donji deo leđa - minimizirati savijanje leđa napred povećanjem radne visine;

Viljuškaristi čeonih viljuškara moraju da sede kada upravljaju opremom. Dugo sedenje povećava opterećenje kičme, što opterećuje ligamente i diske donjeg dela leđa. Opterećenje se povećava kada operator ne održava kičmu u neutralnom položaju.

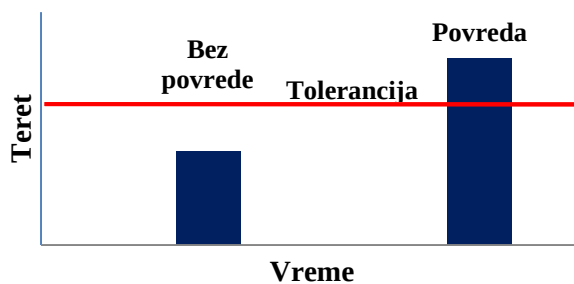
Kako bi se izbegle povrede ovog tipa potrebno je koristiti podesiva sedišta. Podesiva sedišta treba da imaju podešavanje lumbalne potpore, podešavanje napred - nazad, gore - dole kao i nagib napred - nazad. Sedišta treba da budu vazдушna ili da imaju jastuke za podešavanje vibracija. Materijal sa kojim se oblaže sedište treba da bude prozračan i neklizajući.

3.3 Povrede viljuškariste usled loše ergonomije

Povrede nastaju kada opterećenja premašuju toleranciju tkiva. Prikazaće se najčešći tipovi povreda viljuškarista u radu sa viljuškarom i podeliće se na direktne i indirektne za delove tela koji su skloni povredama. Na dijagramima 1, 2 i 3 prikazaće se vrste povreda u zavisnosti od opterećenja [4].

Prekomerna sila

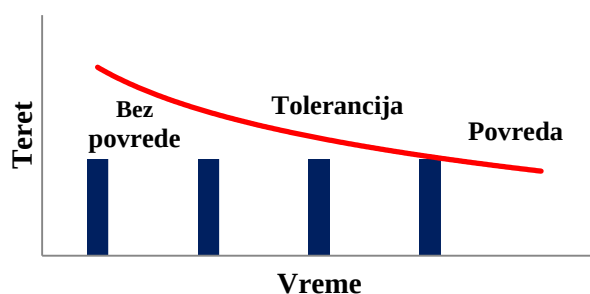
Ova vrsta povrede nastaje iz jednog događaja, gde su opterećenja ili sile toliko velike da prevazilaze toleranciju tkiva i izazivaju trenutnu povredu. Ova vrsta povreda je češća kod saplitanja i padova. Primer iz prakse jeste prelazak točkom preko članka radnika i njegovo uganuće.



Dijagram 1

Prekomerno ponavljanje

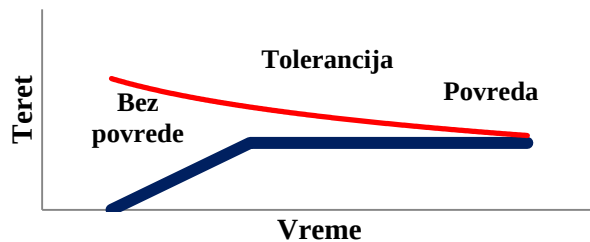
Ova vrsta povrede nastaje od ponovljenog opterećenja koje slabi tkivo do tačke otkaza. Polako napreduje do tačke u kojoj opterećenje može izazvati povredu. Ova vrsta povreda je češća kod zadataka koji se ponavljaju. Primer iz prakse kada radnik koji vuče drvenu građu sa vila i stvara mogućnost pojave diskus hernije.



Dijagram 2

Prekomerno trajanje

Ova vrsta povrede nastaje od stalnog opterećenja koje slabi tkivo do tačke otkaza. Ova vrsta povreda je češća kod zadataka koji od radnika zahtevaju da zauzmu statične ili nezgodne položaje tokom dužeg perioda. Primer iz prakse je razvoj napetosti u vratu.



Dijagram 3

4. REZULTATI ISPITIVANJA VILJUŠKARISTA U KOMPANIJI NIS

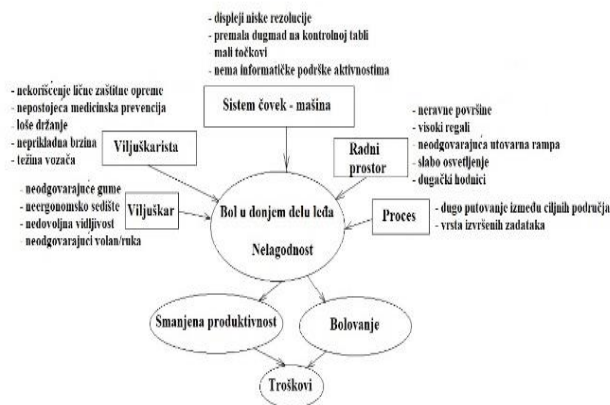
Izvršeno je anketiranje osmorice viljuškarista starosne dobi od 29 do 44 godina koji upravljaju viljuškarom svakodnevno. Imaju iskustva u ovoj branši između 5 i 10 godina. Rad je smenski 7 dana po 12h.

Čak 75% ispitanika se izjasnilo na bol u donjem delu leđa, 5% na bol u stopalima, 5% na bol u člancima, 5% na bol u kolenima i 10% na bol u ramenima i vratu [2].

Bolovi ekstremiteta kod anketiranih viljuškarista				
Donji deo leđa	Stopalo	Članak	Kolena	Vrat i ramena
75%	5%	5%	5%	10%

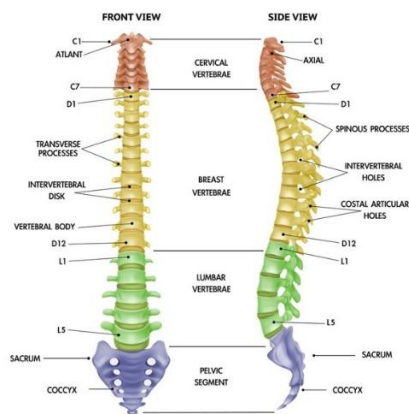
Tabela 2 Rezultati ispitivanja

Loše radne karakteristike uzrokovane su dizajnom viljuškara, radnim prostorom i sistemima čovek - mašina. Može se zaključiti da dizajn viljuškara ima najveći uticaj na povrede usled vožnje viljuškara, mada ne možemo precizirati koji deo radnih karakteristika se može dodatno dodati dizajnu viljuškara [1]. Viljuškaristi starosne dobi do 35 godina koji su fizički aktivni i vežbaju imaju značajno manju mogućnost povreda od viljuškarista starije dobi [2].



Slika 4 Uzroci bola u leđima i njegove posledice [1]

Poboljšanje sedišta i guma viljuškara skoro značajno smanjuje mogućnost povreda. Veća udobnost i manje bolova u donjem delu leđa mogu se postići pametno dizajniranim rampama za utovar. U slučajevima kada su preduzete mere kontrole (izravnavanje površine po kojoj se viljuškar kreće, smanjenje brzine), doći će do značajnog smanjenja izloženosti vibracijama celog tela. Na slici 4 su prikazani uzroci koju stvaraju bol u leđima i nelagodnost.



Slika 5 Položaj kičmenog stuba u neutralnom položaju [2]

Viljuškarista stalno sedi na vibrirajućem sedištu što dovodi do bola u leđima. Kičma se sastoji od 33 kosti koje se nazivaju pršljenovi, slika 5. Svaki od ovih pršljenova je posebno dizajniran da zaštiti kičmenu moždinu i pruži podršku leđima. Između svakog od pršljenova nalaze se diskovi. Diskovi imaju čvrste elastične zidove koji su ispunjeni vodenom supstancom nalik gelu.

Ovi diskovi su kao žele krofne, kada su pritisnuti sa jedne strane, druga strana se izboči i povećava pritisak na zid diska. Da bi se održala ravnomerna raspodela pritiska na diskove, kičma se mora držati u neutralnom položaju. Sedenje će dovesti do rotacije karlice iz neutralnog položaja, jer će se lumbalna kičma spljoštiti [4].

5. ZAKLJUČAK

Napretkom tehnologije i ergonomija je dovedena do zavidnog nivoa, komande su grupisane prema svrsi, njihova upotreba zahteva minimalne pokrete ruku i nogu viljuškariste. Bolovi kod viljuškarista se najčešće javljaju u predelu leđa i donjeg dela leđa, vrata, ramena, zglobova na rukama, šaka i članaka. Unapređenjem ergonomije teži se ka eliminisanju ovih bolova. U oblasti ergonomije viljuškara još uvek postoji potencijal za nova istraživanja. Najveću pažnju je potrebno obratiti na zdravlje viljuškariste, produktivnost i ekonomske efekte.

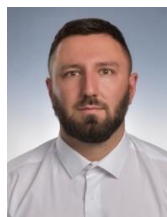
Što je ergonomija bolja, produktivnost će biti veća. Anketiranjem zaposlenih u kompaniji NIS došlo se do očekivanog zaključka da se kao najveće žarište bola javlja bol u leđima, ramenom i vratnom delu viljuškarista. Predlog za smanjenje bola je upotreba vazdušnih sedišta sa lumbalom potporom ili podešavajućim jastucima, povećanje visine sedišta za oslanjanje ramenog dela i ugradnja podesivog naslona za ruke. U viljuškarima koji rade u hladnijim delovima sveta bi se kao rešenje predložila ugradnja grejača sedišta, sedalnog i naslonog dela. Dolazi se do zaključka da se razvojem ergonomije i tehnologije, u poređenju sa periodom iz 1980 godine kada je ergonomija bila na začetku, dosta uticalo na razvoj kabine viljuškara.

Smanjeni su bolovi u ljudskom organizmu, smanjen je zamor viljuškariste, povećan je komfor vožnje, povećana je produktivnost kao i profit.

6. LITERATURA

- [1] Gajšek B., Dukić G., Opetuk T.: *Review of ergonomic solutions to protect from injuries of lower back in case of forklift drivers*, conference paper, 2015
- [2] Molnar E.: *Ergonomske karakteristike viljuškara na uticaj rukovaoca i bezbednosni uslovi korišćenja viljuškara*, Master rad, Fakultet tehničkih nauka, Novi Sad, 2022
- [3] Radović D., : *Prilog 2 UP-47.04.01-006_v2_sr –Rad viljuškarom*, NAFTA GAS Transport, 2015
- [4] <https://www.safer.ca/docs/Forklift.pdf> (Oktobar 2022)

Kratka biografija:



Ervin Molnar rođen je u Kikindi, 1991. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mašinstva - Mehanizacije i konstrukcionog mašinstva odbranio je 2022.god. kontakt: molnar.ervin.ki@gmail.com

ODREĐIVANJE PARAMETARA REZANJA PRI IZRADI TELA VENTILA VISOKOG PRITISKA

DETERMINATION OF MACHINING PARAMETERS DURING MANUFACTURING HIGH-PRESSURE VALVE BODY

Nikola Grdić, Borislav Savković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – U okviru ovog rada prikazana je kompletna tehnologija obrade sa izabranim parametrima kod obrade vrlo složenog komada od nerđajućeg čelika i osvrt na najznačajnije probleme u svim operacijama, kao i rešenje svih problema obrade rezanjem sa inovativnim zahvatima procesa rezanja.

Ključne reči: Tehnologija obrade rezanjem, obrada nerđajućeg čelika, parametri obrade, inovacija procesa obrade

Abstract – This paper present the complete processing technology with selected parameters for processing a very complex piece of stainless steel and a review of the most significant problems in all operations, as well as the solution of all cutting processing problems with innovative cutting process interventions.

Keywords: Cutting processing technology, stainless steel processing, processing parameters, innovative cutting process

1. UVOD

Obrada materijala skidanjem strugotine je jedan od najvažnijih proizvodnih procesa u savremenoj industriji. Ova grupa metoda obrade na svetskom tržištu učestvuje sa oko 70% od ukupnih proizvodnih procesa. Taj procenat dosta govori koliko je važan proces obrade skidanjem strugotine.

Nove generacije proizvodnih sistema sa kraja 20. i početkom 21. veka treba da poseduju sledeće karakteristike [1]:

- Fleksibilnost;
- Integrabilnost;
- Koordinaciju;
- Inteligenciju.

Od fabrike XXI veka očekuje se da bude sposobna da blagovremeno reaguje na sve promene koje nastaju u relativno kratkom vremenu. Da ima organizacionu fleksibilnost koja omogućava prilagođavanje raspoloživim resursima, zahtevima tržišta i ostvarivanju povoljnih efekata.

Savremeno tržište zahteva visoko kvalitetne proizvode, proizvedene za što kraće vreme i što je moguće jeftinije. Zbog toga kompanije moraju biti spremne da ispoštuju ove zahteve tržišta da bi bile konkurentne i uspešne. Da bi ovo postigla, industrija se bazira na skraćanju vremena projektovanja i proizvodnje [1].

NAPOMENA:

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

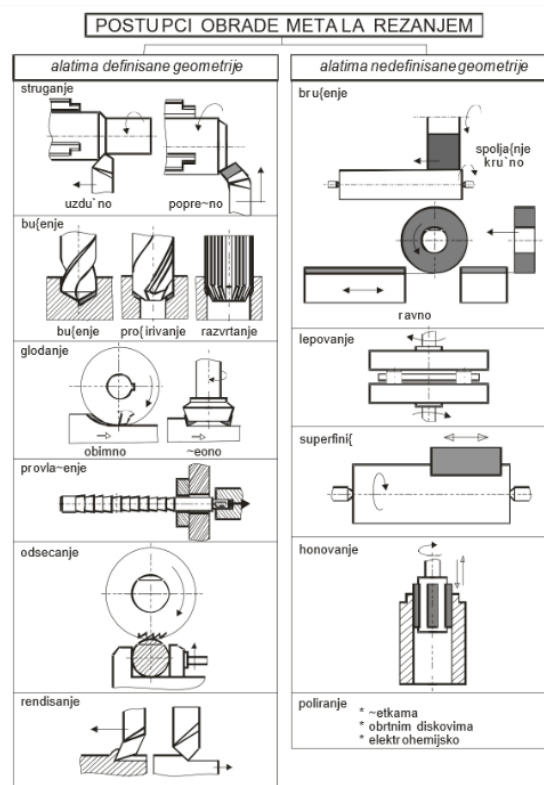
U oblasti tehnologije obrade skidanjem strugotine, akcentat se stavlja skraćanju pomoćnih i glavnih vremena [1].

2. OBRADA METALA REZANJEM

Postupci obrade metala rezanjem su postupci oblikovanja (promene oblika, dimenzija, hrapavosti obrađene površine i karakteristika površinskog sloja) uklanjanjem viška materijala mehaničkim dejstvom reznog alata na predmet obrade. Najčešće se razvrstavaju na postupke [2]:

- Prethodne - grube obrade;
- Završne - fine obrade.

Postupci prethodne obrade (struganje, bušenje, glodanje, rendisanje ...) imaju prvenstveno cilj da uklone što veću količinu materijala. Postupcima završne obrade (razvrtanje, provlačenje, brušenje, honovanje, lepovanje ...) se ostvaruje zahtevani kvalitet obrade (tačnost i kvalitet obrađene površine). Osnovni postupci obrade materijala rezanjem su: struganje, bušenje, glodanje, testerisanje (odsecanje), rendisanje, provlačenje, brušenje i glačanje (lepovanje, superfiniš, honovanje i poliranje) [2].



Slika 1. Postupci obrade metala rezanjem [2]

3. TEHNOLOŠKE OSOBINE I OBRADIVOST MATERIJALA

Pod pojmom apsolutne obradivosti materijala se podrazumeva skup analitičkih izraza koji opisuju propratne pojave u zoni rezanja ispitivanog materijala. Pod pojmom relativne obradivosti podrazumeva se poređenje stanja obrade nekog materijala s poznatim etalonskim materijalom. Obradivost materijala je veoma važna u industriji prerade metala rezanjem.

Od nje zavisi brzina rezanja, troškovi potrošnje alata, traženi kvalitet obrađene površine i drugo. Uopšteno se obradivost materijala može definisati kao njegova sposobnost da se može obrađivati skidanjem materijala.

Bolju obradivost ima materijal [3]:

- koji se određenim alatom obrađuje većim brzinama rezanja,
- koji se obrađuje uz što manje sile rezanja,
- kod kojeg je vek trajanja alata što duži,
- kod kojeg se dobija strugotina što povoljnijeg oblika,
- kod kojeg se postiže što bolji kvalitet obrađene površine,
- koji pri obradi ne uzrokuje visoku radnu temperaturu,
- i drugi manje važni kriterijumi.

4. OBRADIVOST I KARAKTERISTIKE NERĐAJUĆEG ČELIKA

4.1 Austenitni nerđajući čelici

Ovi tipovi imaju visoku čvrstoću, u hladnom radnom stanju njihova čvrstoća ima tendenciju da izazove brzo habanje alata. Njegova duktilnost je dovoljna da izazove stvaranje dugih strugotina koje imaju tendenciju da se zalepe na alat. To može rezultovati formiranjem sloja na ivici alata (fenomen naslage) koji može povećati opterećenje rezanja alat i kao posledicu ima rizik od loma alata. Zbog toga su velika opterećenja rezanja, što dovodi do brzog trošenja alata, sa rizikom od loma zbog oštećenja oštrica. Provodljivost ovih legura je oko tri puta niža nego kod običnih ugljeničnih čelika što dovodi do visokih temperatura koje smanjuju vek trajanja alata. Iz ovih razloga, austenitne klase namenjene za obradu uopšteno imaju kontrolisan sadržaj sumpora između 0,15 i 0,35%. U zavisnosti od predviđene primene, odnosno od zahtevanih svojstava usluga, zajedno sa uslovima realizacije, razvijene su brojne druge legure sa posebnim kombinacijama osobina [4].

5. ALATNI MATERIJAL ZA OBRADU TEŠKO OBRADIVIH ČELIKA

Odgovarajući izbor alata je od najveće važnosti za mašinsku obradu nerđajućeg čelika i direktno određuje produktivnost. Četiri različite kategorije od materijala mogu se razlikovati [4]:

- Obloženi i nepresvučeni brzorezni čelici
- (HSS);
- Obloženi karbidni umeci;
- Kermeti;
- Keramika, umetci ojačani silikonskim vlaknima.

Uopšteno se koriste brzorezni čelici za bušenje i narezivanje navoja, dok umetci obloženi karbidom se koriste za struganje i glodanje nerđajućih čelika pri višem rezanju brzina. Premazi koji se najčešće koriste su napravljeni od titanijum nitrida (TiN), titan karbonitrida $\{Ti(C,N)\}$ i aluminijum oksida (Al₂O₃).

Premazi se mogu nanositi pomoću dva procesa:

- Fizičko taloženje pare (PVD);
- Hemijsko taloženje pare (CVD).

5.1 Izbor alata i režima obrade

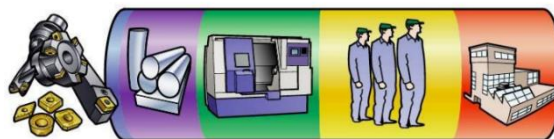
Ekonomičnost i produktivnost mašinske obrade razlikuje dva tipa troškova:

- Varijabilni troškovi

Troškovi nastali u procesu proizvodnje: rezni alati, potrošni materijal (3%), materijal radnih komada (17%)

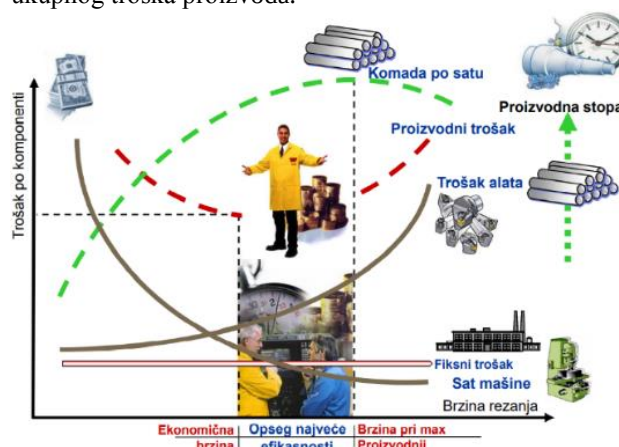
- Fiksni troškovi

Prateći troškovi: mašine i pribori (27%), ljudstvo (31%), aneksi, administracija itd. (22%)



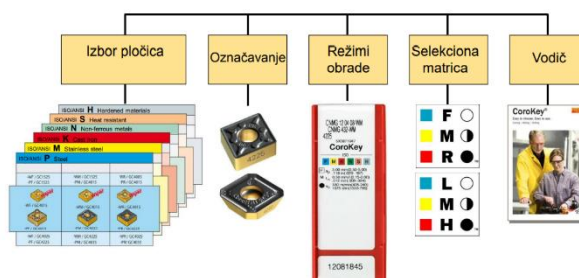
Slika 2. Ilustrativni prikaz troškova mašinske obrade [5]

Prikaz dijagrama troškova po komponenti u odnosu na brzinu rezanja kao jednog od najznačajnijih faktora ukupnog troška proizvoda.



Slika 3. Zavisnost brzine rezanja i troškova [5]

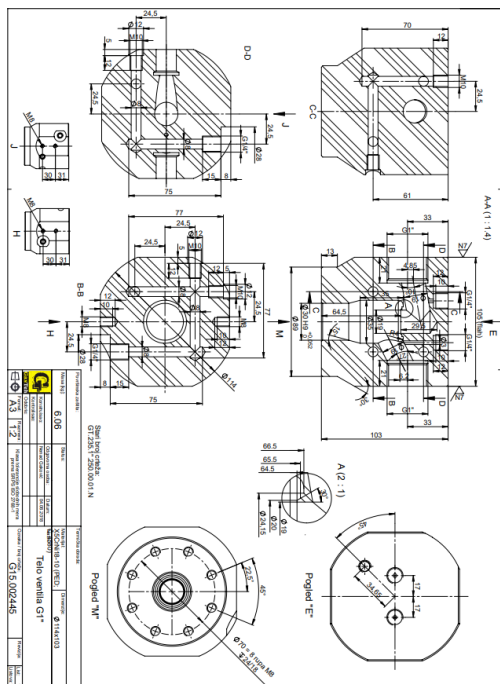
Procedura izbora alata i režima u ilustrovanom primeru proizvođača Sandvik-Coromant kao jednom od vodećim kompanija u proizvodnji reznog alata.



Slika 4. Izbor pločice za rezanje i režima obrade [5]

6. PARAMETRI I TEHNOLOGIJA OBRADE TELA VENTILA VISOKOG PRITISKA

Radionički crtež

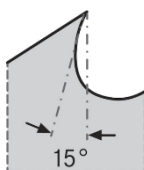


Na osnovu sadržaja tehnološkog procesa formira se "Tehnološka lista" razradjena po svim operacijama obrade uključujući i međukontrolu, kontrolu i ispitivanje. Tehnološka lista definiše sve potrebne informacije za izradu bilo kog komada. Definisani su materijal i masa priprema, dimenzije priprema, interne oznake, nazive i šifre pozicije. Takođe definiše i radno mesto, opis operacije i broj programa, potrebne merne pribore i vreme izrade po svakoj operaciji. Iz tehnološke liste se takođe vidi gde je smešten pribor za određenu operaciju.

6.1 Tehnologija i parametri obrade NC sečenja

Zbog čestog ponavljanja serije komada blokadnog mehanizma visokog pritiska, priprema se dobija iz šipkastog materijala od nerđajućeg čelika X5CrNi18-10 prečnika Ø120. Zbog veličine prečnika, broja komada, karakteristika obrade nerđajućeg čelika, komadi se seku na NC testeri.

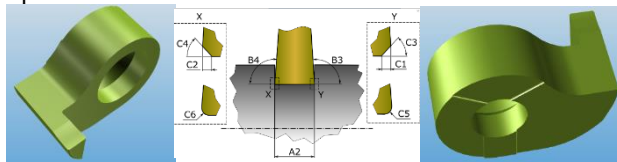
Za odsecanje punih materijala nerđajućeg čelika u praksi su se najbolje pokazali listovi za CNC testeru sa visokoproduktivnim umetkom. Visokoproduktivni list sa geometrijom brušenog zuba, optimizovan za dug radni vek alata na visokoproduktivnim trakastim testerama. Izuzetno pozitivna geometrija zuba obezbeđuje optimalnu produktivnost rezanja materijala karakteristika nerđajućih čelika. Ranije korišćen umetak je od 0°, daljom analizom utvrdeno je da umetak od 15° daje mnogo bolju postojanost uz bolje parametre.



Slika 5. Konstruisani umetak bimetalne testera

6.2 Tehnologija i parametri obrade CNC struganja

Fino struganje dizne u nerđajućem čeliku u ovoj uzanoj zoni rezanja zahteva i povoljne geometrijske karakteristike reznog alata, dobro hlađenje i obezbeđivanje odvođenja strugotine iz procesa rezanja. Inovacija u procesu obrade jesu nosači napravljeni od tvrdog metala i pločica od tvrdog metala, time se smanjuju vibracije što je jako bitno kod pločica malih veličina jer one ne trpe oscilacije u procesu obrade.



Slika 6. Izbor adekvatne pločice i parametara rezanja [6]

Prilikom obrade struganjem zahvaljujući geometriji pločice gore prikazanoj na slici, odvija se proces rezanja koji obezbeđuje male otpore rezanja, a time i povećanje postojanosti alata. Ključna stvar kod tehnologije izrade jeste u odvođenju strugotine iz zone rezanja, do god se strugotina ne zadržava u toj zoni imaćemo stabilniji proces rezanja, bolji kvalitet i postojanost. Brzina rezanja u ovim uslovima iznosi 60 m/min i 0,05 mm/o.



Slika 7. Nosač, prikaz dizne i tipa obrade struganjem [6]

6.3 Tehnologija i parametri obrade NC sečenja

Posebnu pažnju treba napomenuti kod bušenja otvora pod uglom u ovoj operaciji, nekada bilo je nezamislivo bušiti burgijom u "čošak", a da se prethodno ne poravna čelo gde treba da bušimo rupu ili otvor. Ovaj zahvat ispod moguć je sa istim prethodno pomenutim tipom burgija sa pločicama od tvrdog metala.



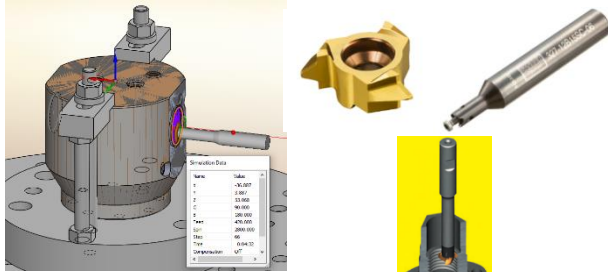
Slika 8. Operacija izrade otvora po uglom i prikaz alata [7]

Brzina rezanja burgije sa pločicama Ø19 je 143 m/min uz pomak od 0,05 po zubu. Režimi obrade su uzeti u srednjoj vrednosti preporuke proizvođača. Karakteristično za ovaj zahvat je korišćenje ciklusa bušenja po 1mm uz vraćanje na sigurnosnu poziciju i vreme čekanja izduvavanja strugotine iz otvora. Vreme izrade Ø19 kose rupe u nerđajućem čeliku iznosi 3 minuta i 04 sekunde.

Poslednji zahvat u ovoj operaciji je izrada navoja G1", kako su navoji sami za sebe nauka, tako može biti i njihova izrada zadati dosta problema, pogotovo u nerđajućem čeliku. Prvenstveno ovaj navoj izrađen je sa HSS mašinskim ureznikom, gde je dolazilo do kidanja

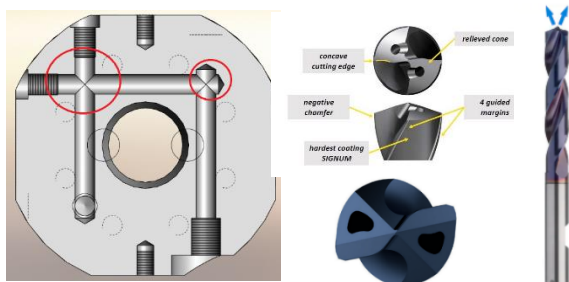
navoja, lepljenja materijala za alat i izrada lošeg navoja. Naredni korak je bilo urezivanje sa ručnim ureznicima.

Problem kod ručnog urezivanja jeste dodatna operacija zbog kojeg se produžava vreme izrade komada, kao i veliki fizički napor zbog velikog prečnika HSS ureznika i velikih otpora rezanja u nerđajućem čeliku. Uvođenjem glodanja navoja rešen je problem izrade navoja.



Slika 8. Izrada navoja G1'' glodanjem uz prikaz alata [7]

Pored problema temperatura, dubokog bušenja, nastaje problem ukrštanja rupa prilikom dubokog bušenja, na označenim mestima na slici ispod dolazi do nakupljanja strugotine i loma burgije.



Slika 9. Duboko bušenje, izbor parametara i prikaz alata [8]

Geometrija alata pogodna za ovu vrstu dubokog bušenja u nerđajućem čeliku je ima 4 fazete koje daju dodatnu stabilnost, ugao od 135° koji omogućuje da burgija ostane u svom centru, negativnu oborenu ivicu koja daje dodatnu izdržljivost, polirani rezni ceo smanjuje unos toplote u alat jer se smanjuje trenje i toplota brže odlazi i konkavnu rezu ivicu sa ojačanom prevlakom.

Odvodjenje strugotine kao jedan od najbitnijih faktora kod dubokog bušenja poboljšava se sa hlađenjem kroz telo burgije, hlađenje se izvodi pod visokim pritiskom i pomaže burgiji da odvodi strugotinu iz nepovoljne zone rezanja koja oštećuje i rezu ivicu i fazete. Slika 10. ispod prikazuje izrađeno Telo ventila u realnosti.



Slika 10. Telo ventila visokog pritiska

7. ZAKLJUČAK

Savremeno tržište zahteva stalno praćenje trenda razvoja tehnologije i alata za obradu rezanjem iz više razloga, a neki od njih su:

- Produktivnost
- Optimizacija

- Dobit
- Inovacija
- Konkurentnost na tržištu itd.

Pre početka svakog rada, tako i pri analizi i projektovanju potrebno je sagledati prednosti i nedostatke koje se mogu desiti novim tehnološkim idejnim rešenjem.

Analiza parametara obrade i tehnologije donosi dobrobit preduzeću jer daje jasan uvid na trenutnu situaciju troškova izrade nekog proizvoda, a njenim razvojem mogu se postići znatne prednosti, kao što je prikazano u primerima. Izborom adekvatnog rešenja, podešavanjem parametara na određenoj mašini sa što adekvatnijim izborom tehnologije omogućava se produktivnija obrada, niža cena izrade a time i niz benefita za neku kompaniju. Imajući više varijanti idejnog rešenja tehnologije i više varijanti odabira alata, podešavanjem parametara obrade za svako rešenje lako se sagledavaju prednosti i nedostaci. Eliminisanjem neadekvatnog rešenja sve smo bliže pravom rešenju.

8. LITERATURA

- [1] Zeljković M., Tabaković S., Živković A.: CAD/CEA/CAM i CIM sistemi/ Autorizovani rukopis predavanja, Fakultet tehničkih nauka, Novi Sad, 2015/2016
- [2] Kovač P., Milikić D.: Rezanje metala, univerzitetski udžbenik, Fakultet Tehničkih nauka, 1998.
- [3] Kovač P., Milikić D., Gostimirović M., Savković B., Sekulić M.: Tehnologija obrade rezanjem, zbirke rešenih zadataka, Univerzitet u Novom Sadu, 2013.
- [4] Nedić, B., Lazić M.: Obrada metala rezanjem, Mašinski fakultet- Kragujevac, Stanić, J., Teorija procesa obrade I, Mašinski fakultet, Beograd, 1995
- [5] http://www.dpm.ftn.uns.ac.rs/predmeti/Projektovanje%20tehnoloskih%20procesu/PTP_Vezba7.pdf pristupljeno 5.10.22
- [6] https://www.phorn.co.uk/fileadmin/user_upload/media/PDF/2015_Catalogue_Sub-Sections_EN/2015_Super-Mini_Catalogue_Chapters/Supermini_catalogue_chapters_1_06_107_108_10P_111_11P_114_116_18P.pdf pristupljeno 03.10.22
- [7] https://www.sandvik.coromant.com/en-us/products/corodril_880/pages/how-to-use.aspx pristupljeno 05.10.22
- [8] <https://www.guhring.com/ProductsServices/ViewProductLine/RT-100-XF> pristupljeno 2.10.22

Kratka biografija



Nikola Grdić rođen je u Vrbasu 1990. god. Diplomirao je na Fakultetu tehničkih nauka 2016. godine. Zasnovaio je radni odnos u kompaniji Gasteh u kojoj je napredovao od radnog mesta inženjera-tehnologa do mesta rukovodioca sektora tehnologije.



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

RAZVOJ NANOFLUIDA VODA/ Al_2O_3 PRIMENOM RAČUNARSKE DINAMIKE FLUIDA DEVELOPMENT OF NANOFLUID WATER/ Al_2O_3 USING COMPUTATIONAL FLUID DYNAMICS

Danijel Đurković, *Fakultet tehničkih nauka, Novi Sad*

Oblast- MAŠINSTVO

Kratak sadržaj – Predmet rada bila je prinudna konvekcija nanofluida voda/ Al_2O_3 . Cilj rada bio je da se za potrebe razvoja nanofluida urade numeričke simulacije prinudne konvekcije nanofluida voda/ Al_2O_3 . Numeričke simulacije urađene su primenom računarske dinamike fluida, metodom konačnih zapremina. Numeričke simulacije su realizovane u komercijalnom softveru Star CCM +.

Ključne reči: nanofluid, prinudna konvekcija, metod konačnih zapremina

Abstract - The subject of this paper is forced convection of water/ Al_2O_3 nanofluid. The aim was to perform numerical simulations of forced convection of water/ Al_2O_3 nanofluid for the purpose of the development of nanofluids. The numerical simulations were carried out by means of computational fluid dynamics, using the finite volume method. The numerical simulations were performed using the commercial Star CCM+ software.

Keywords: nanofluid, forced convection, finite volume method

1. UVOD

U novije vreme procesi grejanja i hlađenja predstavljaju najveći izazov za razvoj efikasnosti uređaja za razmenu toplote. Do sada su u industriji najčešće korišćeni konvekcionalni fluidi kao što su voda, mineralna ili sintetička ulja i etilen – glikol.

Konvekcioni prelaz toplote korišćenjem konvekcioničkih tečnosti je često ograničen njenom niskom toplotnom provodljivošću i malom mogućnošću akumulacije veće količine toplotne energije.

U poslednje vreme radilo se dosta na poboljšanju fizičkih osobina fluida kao što su toplotna provodljivost i toplotni kapacitet fluida. Za iznalaženje rešenja problema koje bi zadovoljilo potrebe savremene industrije, veoma velika pažnja posvetila se razvoju nanofluida.

Nanofluidi su tečne suspenzije koje sadrže čestice koje su veličine manje od 100 nm. Radi se o česticama koje imaju veću toplotnu provodljivost od osnovne tečnosti.

Koeficijent prelaza toplote zavisi od toplotnih svojstava nanofluida, zapreminskog udela nanočestica u nanofluidu, oblika i vrste čvrste čestice. Imajući to u vidu došlo se do ideje o dodavanju čvrstih čestica u tečnosti koje mogu da povećaju vrednost koeficijenta prelaza toplote u poređenju sa konvekcionalnim fluidima.

NAPOMENA:

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

Materijali koji doprinose povećanju ovog koeficijenta su čisti metali (Al, Cu, Ni), oksidi metala (Al_2O_3 , TiO_2 , CuO, SiO_2 , Fe_2O_3 i Fe_3O_4), ugljenične nanocevi i keramika [1].

Istraživači su na početku koristili suspenzije vode sa čvrstim česticama prečnika od nekoliko milimetara ili nekoliko mikrometara. Poređenjem dobijenih rezultata sa rezultatima dobijenim u slučaju kada je radni fluid voda, primećeno je povećanje koeficijenta prelaza toplote radnog fluida.

Prilikom upotrebe fluida sa dispergovanim česticama javili su se problemi kao što su erozija zida cevi, loša stabilnost suspenzije, sedimentacija i začepljenje cevi, a to su prilično ozbiljni problemi za sisteme koji koriste npr. mini i mikro kanale [1].

2. PREGLED STANJA U OBLASTI

Mnogi naučni radovi ukazuju da postoji poboljšanje toplotne provodljivosti i konvektivnog prenosa toplote kod nanofluida, pri niskim koncentracijama nanočestica [2].

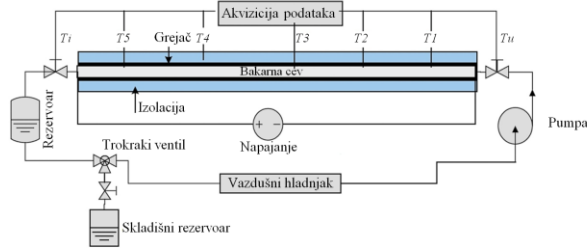
Zanimljivo je da u istraživačkoj studiji iz 1873. godine Maksvel (Maxwell) analizirao dispergovane čestice u tečnosti. Međutim 1904. godina bi mogla da se smatra kao početak nanotehnologije [3].

Velike čestice mogu uzorkovati različite probleme u opremi za prenos toplote. Velike čestice brzo teže da se istalože tako da se veliki pad pritiska može pojaviti npr u mikro kanalima. Takođe abrazivne aktivnosti čestica mogu da uzrokuju eroziju zidova cevi. Čestice male zapremine i koncentracija sprečavaju zagušenje i povećanje pada pritiska nanofluida [4].

Istraživači su se najviše u prošlosti bavili konvektivnim prenosom toplote nanofluida. Wen i Ding [5] eksperimentalno su istražili konvektivni prenos toplote nanofluida kroz bakarnu cev pri laminarnom režimu strujanja. Na slici 1. šematski je prikazano eksperimentalno postrojenje koje su autori rada koristili za istraživanje prinudne konvekcije nanofluida. Nanofluid su formirali suspendovanjem čestica Al_2O_3 u demineralizovanoj vodi kao osnovnom fluidu. Suspendovane su nanočestice $\gamma - \text{Al}_2\text{O}_3$ veličine od 27-56 nm. Kao surfaktant korišćen je natrijum dodecilbensensulfanat $\text{C}_{18}\text{H}_{29}\text{H}_a\text{O}_3\text{S}$ (jedna desetina od mase nanočestica).

Razmatrani su nanofluidi sa nekoliko različitih koncentracija nanočestica Al_2O_3 . Nanofluidi su pripremljeni metodom iz dva koraka. Nanočestice su prvo pomešane sa vodom u određenom odnosu. Zatim je nanofluid tretiran u ultrazvučnom kupatilu u trajanju od 16 do 20 h.

Uočeno je da nanofluidi sa koncentracijom nanočestica manjom od 4% imaju veoma dobru stabilnost. Ovi nanofluidi su bili stabilni više od nedelju dana. Konvektivni prenos toplote nanofluida na ulazu u cev nije u dovoljnoj meri istražen do predmetnog istraživanja. Iz tog razloga je fokus istraživanja stavljen na ponašanje nanofluida na ulazu u cev.



Slika 1. Šematski prikaz eksperimentalnog postrojenja [5]

Konvektivni prenos toplote analiziran je duž cevi putem vrednosti lokalnih koeficijenata prelaza toplote i Nuseltovog broja koji su računati iz sledećih jednačina

$$\alpha(x) = \frac{q}{(T_w(x) - T_f(x))}, \quad (1)$$

$$Nu(x) = \frac{\alpha(x) \cdot D}{\lambda}, \quad (2)$$

Profil temperature fluida duž cevi računat je iz energetskog bilansa:

$$\dot{Q}(x) = q \cdot S(x) = c_p \cdot \dot{m} \cdot (T_f(x) - T_{in}), \quad (3)$$

$$\dot{m} = \rho \cdot v \cdot A, \quad (4)$$

$$T_f(x) = T_{in} + \frac{q \cdot S(x)}{c_p \cdot \rho \cdot v \cdot A}, \quad (5)$$

Jednačina (5) zasniva se na pretpostavci da nema gubitka toplote kroz zidove cevi.

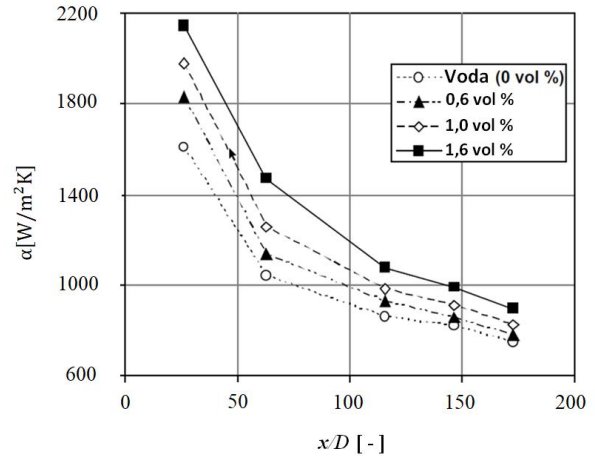
Rejnoldsov i Prantlov broj definišu se kao:

$$Re = \frac{\rho \cdot v \cdot D}{\mu}, \quad (6)$$

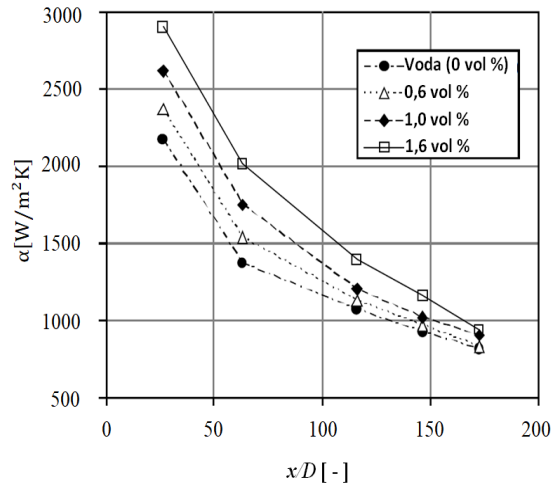
$$Pr = \frac{\nu}{a}, \quad (7)$$

Na slikama 2 i 3 prikazani su aksijalni profili lokalnog koeficijenta prelaza toplote za dve razmatrane vrednosti Rejnoldsovog broja. Rezultati pokazuju da primenom predmetnog nanofluida i povećanjem Rejnoldsovog broja može značajno da se poboljša konvektivni prenos toplote na ulazu u cev. Sasvim očekivano koeficijent prelaza

toplote opada udaljavanjem od ulaza u cev. Može da se uoči da je potrebna veća dužina cevi za razvijanje termičkog graničnog sloja kod nanofluida u odnosu na čistu vodu.



Slika 2. Aksijalni profil lokalnog koeficijenta prelaza toplote (\$Re = 1050 \pm 50\$) [6]



Slika 3. Aksijalni profil lokalnog koeficijenta prelaza toplote (\$Re = 1600 \pm 50\$) [6]

3. MATERIJAL I METOD

Cilj master rada bio je da se ispita prinudna konvekcija nanofluida kroz pravu cev kružnog poprečnog preseka primenom računarske dinamike fluida. Nanofluid je dobijen suspendovanjem nanočestica Al_2O_3 u vodi kao osnovnom fluidu. Koncetracija nanočestica iznosila je 1%.

Numerički model urađen je tako da odgovara fizičkom modelu prikazanom u radu [5]. Cev je dužine 970 mm i unutrašnjeg prečnika 4,5 mm. Razmatran je laminaran režim strujanja sa dve vrednosti Rejnoldsovog broja \$Re=1050\$ i \$Re=1600\$. Na zid cevi dovođena je konstantna toplotna snaga koja je iznosila 300 W i smatralo se da su uslovi strujanja nanofluida nestacionarni.

Verifikacija rezultata računarske dinamike fluida urađena je rezultatima dostupnim u literaturi [5].

Za potrebe računarske dinamike fluida izračunate su fizičke osobine nanofluida Al_2O_3 čije suspendovane čestice Al_2O_3 imaju zapreminsku koncentraciju od (1%) u demineralizovanoj vodi.

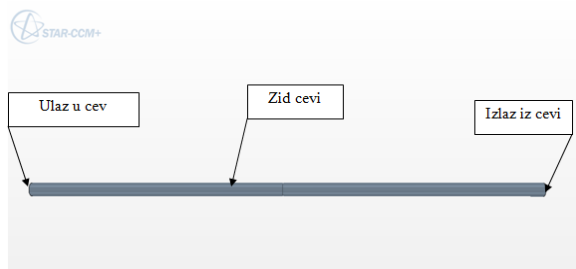
Fizičke osobine su izračunate za temperaturu nanofluida na ulazu u cev iz jednačine za energetski bilans (5). Uvedene su četiri aproksimacije numeričkog u odnosu na fizički model:

- smatralo se da je brzina strujanja fluida konstantna po površini poprečnog preseka na ulazu u cev;
- fizičke osobine sa kojima su podešene numeričke simulacije izračunate su za temperaturu nanofluida na ulazu u cev koja je iznosila 295 K;
- smatrano je da je nanofluid jednofazni fluid i
- zanemaren je efekat prirodne konvekcije.

3.0 Podešavanje numeričke simulacije

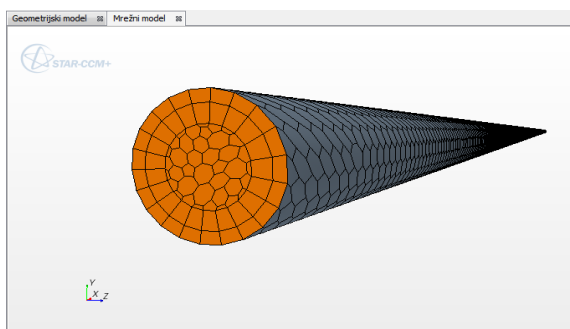
3.1 Crtanje CAD modela za formiranje geometrijskog modela to jest prava cev

3.2 Definisanje tipa granica u softverskom paketu STAR CCM+ sve granice su predpodešene da budu zidovi (**Wall**). Na slici 4 prikazane su granice domena. Na ulazu u cev podešavana je brzina strujanja, dok je na izlazu iz cevi podešavano da razlika pritiska bude na granici $\Delta p=0$. Kako su predpodešeni tipovi svih granica tipa zida, tipovi granica na **Ulazu** i **Izlazu** iz cevi su promenjeni u **Velocity Inlet** i **Pressure Outlet**.



Slika 4. Prikaz graničnih uslova

3.3 Generisanje mrežnog kontinuuma slika 5.



Slika 5. Izgled mrežnog kontinuuma

3.7 Podešavanje kriterijuma zaustavljanja numeričke simulacije

Kao kriterijum zaustavljanja numeričke simulacije podešeno je maksimalno fizičko vreme, pri čemu je deaktiviran kriterijum zaustavljanja numeričke simulacije maksimalnim brojem iteracija.

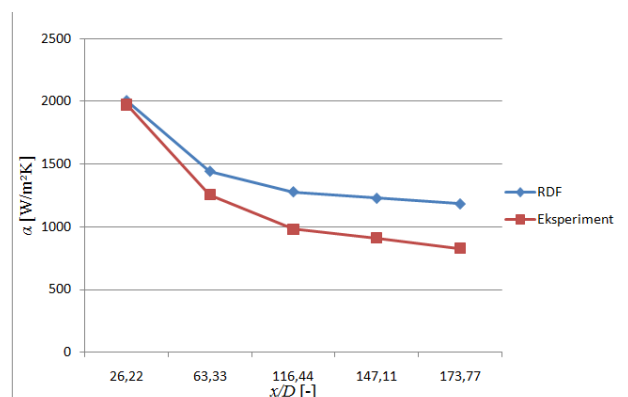
Maksimalno fizičko vreme trajanja simulacije je 100 s, a vremenski korak pri rešavanju je 0,1 s.

4. REZULTATI RADA I DISKUSIJA

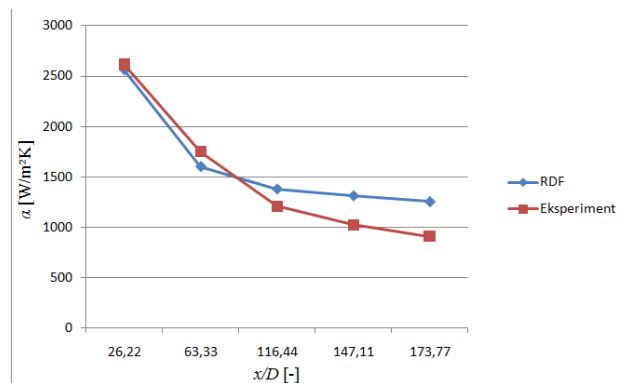
Na osnovu razlike temperature na zidu cevi i osrednjene temperature fluida dobijene numeričkim simulacijama na definisanim mestima x duž cevi izračunat je lokalni koeficijent prelaza toplote i Nuseltov broj.

Primenom jednačine (1) izračunate su vrednosti lokalnih koeficijenta prelaza toplote duž cevi pri vrednosti Reynoldsovog broja $Re = 1050$ i $Re = 1600$:

Na slici 6 i slici 7 prikazan je dijagram sa poređenjem koeficijenta prelaza toplote dobijenih računarskom dinamikom fluida i eksperimentom.



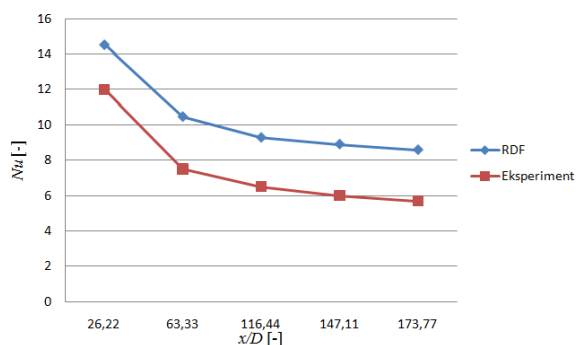
Slika 6 Grafički prikaz promena lokalnog koeficijenta prelaza toplote duž cevi pri strujanju 1vol% Al_2O_3 nanofluida pri vrednosti Reynoldsovog broja $Re = 1050$.



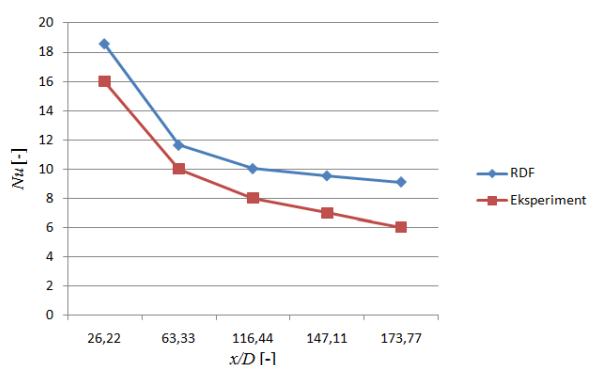
Slika 7 Grafički prikaz promena lokalnog koeficijenta prelaza toplote duž cevi pri strujanju 1vol% Al_2O_3 nanofluida pri vrednosti Reynoldsovog broja $Re = 1600$.

Primenom jednačine (2) izračunate su vrednosti lokalnog Nuseltovog broja duž cevi pri vrednosti Reynoldsovog broja $Re = 1050$ i $Re = 1600$:

Na slici 8 i slici 9 upoređene su vrednosti Nuseltovog broja dobijene računarskom dinamikom fluida i eksperimentalnim putem.



Slika 8 Grafički prikaz vrednosti lokalnog Nusseltovog broja duž cevi pri strujanju 1vol% Al_2O_3 nanofluida pri vrednosti Reynoldsovog broja $Re = 1050$.



Slika 9 Grafički prikaz vrednosti lokalnog Nusseltovog broja duž cevi pri strujanju 1vol% Al_2O_3 nanofluida pri vrednosti Reynoldsovog broja $Re = 1600$.

5. ZAKLJUČAK

Verifikacija rezultata numeričke simulacije urađena je analizom lokalnih vrednosti Nusseltovog broja i koeficijenta prelaza toplote duž cevi. Primećeno je da sa povećanjem brzine strujanja odnosno Reynoldsovog broja dolazi i do povećanje Nusseltovog broja, povećava se značajno količina prenesene toplotne energije sa zida cevi na fluid. Najveći lokalni koeficijent prelaza toplote je na ulazu u cev, a zatim drastično opadaju vrednosti sa povećanjem razdaljine odnosno na samom izlazu bakarne cevi. Radi se o tendenciji koja je sasvim očekivana pri prinudnoj konvekciji fluida kroz pravu cev.

Poređenjem vrednosti koeficijenata prelaza toplote dobijenih računarskom dinamikom fluida i eksperimentom za razmatrane vrednosti Reynoldsovog broja dobijena su sledeća relativna odstupanja:

- o za vrednost Reynoldsovog broja $Re = 1050$ na ulazu u cev dobijena su najmanja odstupanja u odnosu na eksperimentalne rezultate i iznosila su 1,63%, a na izlazu iz cevi dobijena su najveća odstupanja koja iznose 43%
- o za vrednosti Reynoldsovog broja $Re = 1600$ gde su relativna odstupanja na ulazu u cev iznosila najmanja 2,07%, a na izlazu iz cevi su dobijena najveća odstupanja koja su iznosila 38%.

Uočeno je da se povećanjem Reynoldsovog broja smanjuju odstupanja rezultata računarske dinamike fluida i eksperimentalnih rezultata.

U budućnosti bi trebalo nastaviti istraživanje prinudne konvekcije nanofluida voda/ Al_2O_3 kroz pravu cev kružnog poprečnog preseka primenom računarske dinamike fluida. Potrebno je da se uradi analiza nezavisnosti mreže, ispita uticaj fizičkih modela i šema diskretizacije na numerička rešenja, kako bi se smanjila razlika između rezultata dobijenih računarskom dinamikom fluida i eksperimentom.

6. LITERATURA

- [1] Choi S.U.S.(1995). *Enhancing thermal conductivity of fluids with nanoparticles*, ASME Publ. Fed, 231 99-106.
- [2] Yang Y., Zhang Z.G., Grulke E.A., Anderson W.B., Wu G. (2005). *Heat transfer properties of nanoparticle-in-fluid dispersions (nanofluids) in laminar flow*, Int.J.Heat Mass Transfer, 48 1107-1116.
- [3] Bashirnezhad K., Bazri S., Safaei M.R., Goodarzi M., Dahari M., Mahian O., Dalkilica A.S., Wongwises. (2016). *Viscosity of nanofluids: A review of recent experimental studies*, International Communications in Heat and Mass Transfer, 73 (02), 114-123.
- [4] Khanafer K., Vafai K., Lightstone M. (2003). *Buoyancy-driven heat transfer enhancement in a two-dimensional enclosure utilizing nanofluids*. Int. J. Heat Mass Transf, 46 (19), 3639-3653.
- [5] Wen D. end Ding Y. (2004). *Exsperimental investigation into convective heat transfer of nanofluids at the entrance region under laminar flow conditions*, International Journal of Heat and Mass Transfer, 47 (24), 5181-5188.

Kratka biografija:



Danijel Đurković rođen je u Novom Sadu, 1983. godine. Fakultet tehničkih nauka u Novi Sad, odsek Mašinstvo - Energetika i procesna tehnika osnovne studije završio 2016. godine, odbranivši diplomski rad na temu "Numeričke simulacije blendi za merenje protoka prirodnog gasa".

PRIMENA CAN I SPI KOMUNIKACIONIH PROTOKOLA PRILIKOM IZVRŠAVANJA INTEGRACIONIH TESTOVA APLIKACIJE INSTRUMENT TABLE**USAGE OF CAN AND SPI COMMUNICATION PROTOCOLS DURING THE EXECUTION OF INTEGRATION TESTS FOR INSTRUMENT CLUSTER APPLICATION**

Boris Deak, Boris Dumnić, *Fakultet tehničkih nauka, Novi Sad*

Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu obrađena je tematika daljinskog testiranja instrument tabli i komunikacioni protokoli koji se koriste u automobilskoj industriji. Opisani su CAN i SPI komunikacioni protokoli i način njihove primene prilikom izvršavanja integracionih testova aplikacije instrument table.

Ključne reči – Instrument tabla, CAN komunikacija, SPI komunikacija

Abstract – In this paper, the topic of remote testbox for instrument cluster, and also CAN and SPI communication protocols are discussed. Description and practical implementation of mentioned communication protocols are given for instrument cluster application.

Keywords – Instrument cluster, CAN communication, SPI communication

1. UVOD

Automobilska industrija predstavlja jednu od najbrže rastućih u 21. veku. Danas, automobil predstavlja složen tehnički sistem kog karakteriše mnoštvo podсистема sa specifičnim funkcijama. Instrument tabla predstavlja jednu od najvažnijih elektronskih kontrolnih jedinica (engl. *Electronic Control Unit - ECU*) savremenog automobila u pogledu komunikacije sa vozačem. Današnji proizvođači automobila opremaju svoja nasavremenija vozila složenim instrument tablama koji vozaču pružaju bitne informacije [1].

Razvojni put instrument table od njene prve pojave u masovnoj proizvodnji do instrument tabli kakve danas prepoznajemo u vozilima novijih generacija je bio veoma dug. Prvi proizvedeni automobili nisu posedovali instrument table. Brzinometar je prva proizvedena komponenta koja čini današnje instrument table i prvenstveno je osmišljena iz bezbednosnih razloga.

Digitalne instrument table brzo su zamenile konvencionalne analogne table u vozilima tokom protekle decenije. Ova promena otvorila je nove oblike kreativnosti za proizvođače automobila, jer digitalizovane kontrolne table mogu da izgledaju mnogo atraktivnije. Ranije, instrument tabla je bila ograničena na nekoliko osnovnih indikatora

kao što su brzinometar, merač goriva, itd. Sa napretkom tehnologije, vozaču je dostupno više informacija koje su od velike pomoći tokom vožnje. Funkcije kao što su pomoć pri parkiranju, indikatori kilometraže, preostalih kilometara, uslovi vožnje, dijagnostika kvarova, signali upozorenja (poruke), itd, samo su neki od savremenih dodataka. Pored toga što su nove funkcije dizajnirane da vozači budu u potpunosti informisani o performansama automobila, mogu imati i značajnije nadogradnje u oblastima kontrole pokreta i prepoznavanja glasa. Dakle, digitalizacija instrument tabli u vrhunskim automobilima će se pojaviti kao jedan od glavnih trendova na tržištu u narednim godinama [2].

U ovom radu obrađen je značaj i razvoj softvera za instrument table u automobilskoj industriji. Predstavljeni su različiti nivoi aplikacije i njihova međusobna interakcija, kao i interakcija instrument table sa ostalim ECU u vozilu. Predstavljen je razvoj i značaj instrument table kao podсистема u složenom sistemu svakog vozila.

2. INSTRUMENT TABLE

Kompleksni sistemi u sklopu svakog vozila se sastoje od velikog broja različitih ECU. Svaki ECU predstavlja zaseban sistem koji je u stalnoj komunikaciji sa ostalim podсистemima putem različitih protokola i standarda komunikacije. Instrument tabla predstavlja jedan od najbitnijih ECU sistema vozilu. Glavna i osnovna funkcija instrument table jeste komunikacija i interakcija između vozača i vozila.

Instrument tabla kao finalni proizvod predstavlja kolaboraciju više različitih disciplina koje se paralelno razvijaju i međusobno usko sarađuju tokom perioda razvoja instrument table. Neke od tih disciplina su razvoj softver aplikacije instrument table, razvoj vizuelnog dela aplikacije (engl. *Human to Machine Interface - HMI*) i dizajna interfejsa, razvoj hardvera i elektronskih komponenti, itd. Aplikativni deo softvera instrument tabli obuhvata dve manje celine:

- viši nivo aplikacije i
- niži nivo aplikacije.

3. APLIKATIVNI DEO INSTRUMENT TABLI

Viši nivo aplikacije koji predstavlja interakciju između samog vozača i vozila se sastoji iz dva osnovna dela: aplikativni deo i HMI deo.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Boris Dumnić, vanr. prof.

Aplikativni deo višeg sloja aplikacije se sastoji iz primene različitih algoritama i logičkih funkcija koji su korišćeni u cilju postizanja efikasne implementacije zahteva proizvođača. Aplikacija instrument table se sastoji iz velikog broja funkcionalnosti i modula koji zajedno predstavljaju kompleksnu celinu, a neki od najznačajnijih su:

1. konačni automat (engl. *state machine*),
2. mehanizam upravljanja stanjima i
3. mehanizam upravljanja greškama.

3.1 Konačni automat

Glavna i osnovna funkcionalnost koju svaka instrument tabla mora imati je konačni automat. Njena uloga je da omogući vozaču da se kreće kroz navigacioni meni i da uspešno pristupi svim mogućim stranicama i podstranicama koje instrument tabla podržava. Usklađenost konačnog automata sa svim ostalim aplikativnim modulima ima krucijalan značaj za normalan i nesmetan rad svake instrument table.

3.2 Mehanizam upravljanja stanjima

Uloga mehanizma upravljanja stanjima jeste da obezbedi normalan rad instrument table i pravilan prolazak kroz različita stanja u kojima instrument tabla može biti. Stanja kroz koja instrument tabla prolazi i u kojima se može naći takođe zavise od proizvođača i njegovih zahteva. Neka od stanja su:

- aktivno stanje (engl. *active state*),
- stanje paljenja instrument table (engl. *start up state*),
- neaktivno stanje (engl. *sleep state*) i
- polu-aktivno stanje (engl. *doze state*).

Prilikom tranzicije kroz različita stanja instrument tabla neprestano komunicira sa ostalim ECU podsystemima u kompleksnom sistemu koje čini vozilo.

3.3 Mehanizam upravljanja greškama

Mehanizam upravljanja greškama (engl. *warning handler*) predstavlja funkcionalnost koja vozača obaveštava o aktuelnom stanju vozila, potencijalnim opasnostima, korisnim informacijama i sl.

Signalizacija obaveštenja i grešaka se mogu na različite načine prikazati vozaču, a neki od načina su iskačući prozori, uključivanje odgovarajućih led dioda (engl. *telltails*), ikonice na displeju instrument table (engl. *virtual telltalls*), interni i eksterni zvukovi, itd.

Neki primeri grešaka, upozorenja i obaveštenja su: upozorenje za sigurnosni pojas, nizak nivo goriva, neadekvatan pritisak u gumama, pokazivači pravca, ABS (engl. *Anti-lock Braking System* - ABS) upozorenje, itd.

3.4. Najčešće korišćeni algoritmi u razvoju aplikacije instrument tabli

Tokom razvoja aplikacije za instrument table i njene implementacije koristi se veliki broj različitih algoritama koji omogućavaju kretanje kroz navigacioni meni, konverziju mernih jedinica, animacije, i sl. Neki od algoritama koji se primenjuju su:

- algoritam beskonačne liste,
- algoritam za prigušenu promenu vrednosti i
- interpolacija u dve tačke.

4. KOMUNIKACIJA INSTRUMENT TABLE SA OSTATKOM SISTEMA U VOZILIMA

Svaki komunikacioni protokol koji se koristi u automobilske industriji mora obezbediti sigurnu, brzu i bezbednu komunikaciju u realnom vremenu.

Procenjuje se da u prosečnom vozilu današnjice složeni sistem elektronskih komponenti sadrži preko 70 ECU koje međusobno razmenjuju nekoliko hiljada različitih signala. Povećanjem složenosti i kompleksnosti komunikacije u složenom sistemu vozila javljaju se i prvi problemi za proizvođače. Glavni razlog za uspostavljanje standarda za efikasniju i bržu komunikaciju je smanjenje materijala potrebnog za prenosne puteve unutar sistema.

4.1. CAN komunikacija

CAN je najzastupljeniji komunikacioni protokol koji se koristi u automobilske industriji. Preko 90% aplikacija koristi upravo CAN protokol.

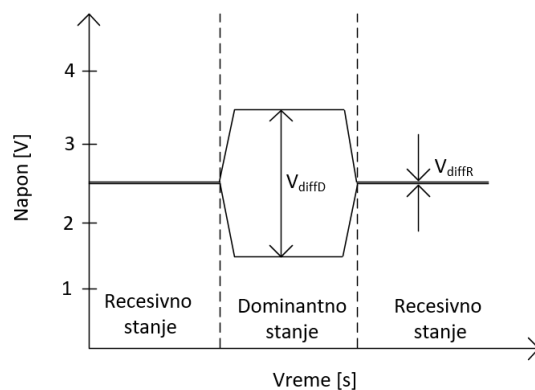
4.1.1 Osnovni principi CAN komunikacije

CAN je serijski protokol koji se sastoji od samo dve namenske žice od kojih je jedna višeg (engl. *CAN High* – *CANH*), a druga nižeg naponskog nivoa (engl. *CAN Low* – *CANL*). Bitna karakteristika CAN protokola jeste što omogućava brz, robusan i stabilan protok informacija, brzine do 1 MB/s (uz dužinu mrežnog metra). CAN komunikaciona mreža se sastoji od više čvorova (engl. *Node*) koji međusobno razmenjuju informacije putem jedinstvene komunikacione linije (engl. *CAN Bus*). Svi čvorovi su ravnopravni na mreži, ne postoji master/slave princip komunikacije, već se CAN protokol naziva još i multi-master protokolom [3].

4.1.2 Fizički sloj CAN komunikacionog protokola

CAN je serijski komunikacioni protokol koji se ostvaruje sa dve žice pomoću kojih se informacija prenosi tehnologijom diferencijala napona. Ova tehnologija podrazumeva da se podaci šalju jedan po jedan bit preko dva komplementarna signala – *CANH* i *CANL*. U CAN komunikacionom protokolu postoje jasno definisana dva moguća stanja magistrale: dominantno i recesivno stanje, kao što je prikazano na slici 2.

Dominantno stanje predstavlja situaciju kada se na ulazni pin primopredajnika dovede logička nula. Do recesivnog stanja na komunikacionoj magistrali dolazi kada se na ulazni pin primopredajnika dovede logička jedinica [4].



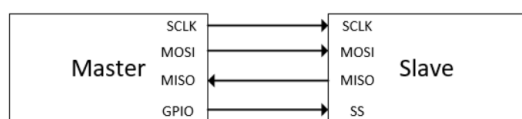
Slika 2. Grafički prikaz dominantnog i recesivnog stanja

CAN komunikacioni protokol podržava četiri tipa poruka:

1. poruka sa podacima (*engl. Data frame*),
2. poruka sa zahtevom za podacima (*engl. Remote frame*),
3. poruka o grešci (*eng. Data frame*) i
4. poruka sa zahtevom za podacima (*engl. Remote frame*).

4.2. SPI komunikacija

SPI komunikacija (*engl. Serial Peripheral Interface*) je sinhroni serijski komunikacioni protokol u kojem učestvuje jedan *master* kontroler i jedan ili više *slave* uređaja. Pod *slave* uređajima se podrazumevaju različiti senzori, aktuatori, memorijske kartice, pa čak i drugi kontroleri koji dobijaju instrukcije od *master* kontrolera. SPI komunikacioni protokol zahteva četiri pina za ostvarenje komunikacije između *master* i *slave* kontrolera, kao što je prikazano na slici 3.



Slika 3. Blok šema povezivanja pinova u SPI protokolu

MISO (*engl. Master In Slave Out*) pin služi za prenos podataka u *slave* režimu. Kada *slave* šalje informacije *master* uređaju koristi MISO izlazni pin. Sa druge strane *master* poslate informacije prima preko svog MISO pina koji je definisan kao ulazni u *master* režimu rada. Pin koji se takođe koristi za prenos podataka je MOSI (*engl. Master Out Slave In*) pin. Njegova uloga je prenos podataka od strane *master* uređaja prema određenom *slave* uređaju. Prenos podataka se mora sinhronizovati sa serijskim taktom (*engl. Serial Clock*) koji se generiše na SCLK pinu. Serijski takt generiše *master* uređaj i šalje ga ka *slave* uređaju preko izlaznog SCLK pina u *master* režimu, odnosno preko ulaznog SCLK pina u *slave* režimu. Ukoliko imamo više *slave* uređaja neophodno je na neki način odrediti kojem tačno *slave* uređaju se *master* uređaj obraća i to se postiže preko SS (*engl. Slave Select*) pina [5].

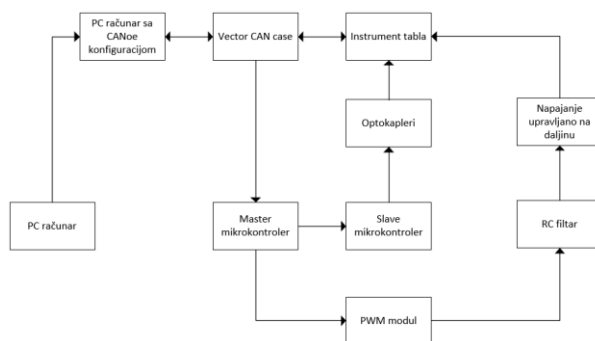
5. TESTIRANJE INSTRUMENT TABLI PRIMENOM CAN I SPI KOMUNIKACIONIH PROTOKOLA

Neizostavni deo razvoja softvera za instrument table podrazumeva i izvršavanje različitih vrsta testova kojima se potvrđuje ispravnost i učinkovitost implementirane aplikacije. Testovi se definišu i izvršavaju na osnovu definisanih zahteva proizvođača u kojima je opisano na koji način se očekuje da se određene funkcionalnosti ponašaju.

5.3 Ideja daljinskog testiranja instrument table

Prilikom postavljanja zahteva i ciljeva ovog projekta jasno je definisano da je glavni cilj omogućiti testiranje aplikacije instrument table bez direktnog posedovanja opreme za testiranje. Najbrojniji ulazni parametri u aplikaciju instrument table su različite CAN poruke i signali, ali pored njih veoma bitnu ulogu imaju i digitalni ulazi, koji su realizovani dovođenjem tačno definisanog

naponskog nivoa na određeni ulazni pin instrument table. Softver koji se koristi za simuliranje CAN komunikacione mreže u realnom okruženju je Vector CANoe i u njemu se kreira CANoe konfiguracija koja se sastoji od svih poruka koje se šalju ka instrument tabli od strane drugih elektronski kontrolnih jedinica vozila. Pored ulaznih CAN poruka, bitnu ulogu u testiranju aplikacije instrument table imaju i digitalni ulazi. Na slici 4 prikazana je blok šema postavke za daljinsko testiranje instrument table.



Slika 4. Blok šema daljinskog testiranja instrument table

PC računar sa CANoe konfiguracijom je povezan sa instrument tablom i *master* mikrokontrolerom preko CANCase hardvera, koji prilagođava CAN komunikaciju na USB ulaz PC računara. CAN komunikacionim protokolom se instrukcije šalju od strane PC računara prema *master* mikrokontroleru, čija je uloga obrada informacija koje stižu sa CAN magistrale i prosledjivanje stanja digitalnih ulaza ka *slave* mikrokontroleru preko SPI protokola. Digitalni izlazi *slave* mikrokontrolera se vezuju sa optokaplerima koji obezbeđuju galvansku izolaciju između mikrokontrolera i instrument table, optokapleri u provodnom stanju dovode potrebni napon na digitalni ulaz instrument table.

Stanje digitalnog izlaza *slave* mikrokontrolera zavisi od vrednosti koja je zadata u CANoe konfiguraciji i zatim prosledjena CAN komunikacionim protokolom preko *master* mikrokontrolera. Pored digitalnih ulaza moguće je regulisati i napon napajanja instrument table. *Master* kontroler putem CAN signala dobija vrednost faktora ispuhe kojim se reguliše vrednost napona napajanja instrument table.

Vrednost faktora ispuhe je u opsegu 0-255 i na taj način se menja širina izlaznog PWM signala iz *master* kontrolera. Povorka signala u opsegu 0-5V se dovodi na RC filter koji povorku pretvara u jednosmerni signal konstantne vrednosti uz minimalna odstupanja. Filtrirani napon koji može biti u opsegu 0-5V, u zavisnosti od vrednosti faktora ispuhe, se dovodi na laboratorijsko napajanje koji ima mogućnost kontrole napona napajanja sa ulaznim naponom, gde ulazni napon 0-5V odgovara izlaznom naponu 1-32V.

6. ZAKLJUČAK

Digitalnim razvojem i tehnološkim napretkom stvoreni su uslovi rada u kojima se može sve lakše realizovati rad na daljinu. Potreba za mobilnošću posebno je došla do izražaja 2020. godine izazvana globalnom pandemijom. U svakom smislu, ovo je bio veliki izazov za mnoge kompanije i preduzeća. U radu je priložen projekat kojim je ostvarena mogućnost lakšeg rada na daljinu prilikom

razvoja aplikacije i izvršavanja integracionih testova instrument tabli, gde se praktičnom primenom CAN i SPI komunikacionih protokola postižu isti rezultati kao i direktnim korišćenjem opreme za testiranje.

7. LITERATURA

- [1] S. Nimara, D. B. Popa, and R. Bogdan, "Automotive instrument cluster screen content validation," in *2017 25th Telecommunication Forum (TELFOR)*, Belgrade, Nov. 2017, pp. 1–4. doi: 10.1109/TELFOR.2017.8249425.
- [2] S. Kalkar, I. Bose, B. Saloni, S. Anilkumar, and S. Tavhare, "Machine Learning based Instrument Cluster inspection using Camera," presented at the 10TH SAE India International Mobility Conference.
- [3] D. Ibrahim, *Controller Area Network projects*. United Kingdom: Elektor International Media, 2011.
- [4] *From theory to practical applications*. New York: Springer, 2013.
- [5] Dawoud Shenouda Dawoud, Peter Dawoud, *Serial Communication Protocols and Standards*, River Publishers 2020.

Kratka biografija:



Boris Deak rođen je u Somboru 1997. godine. Osnovne studije završio je na Fakultetu tehničkih nauka 2020. godine. iz oblasti Elektrotehnika i računarstvo, smer Energetska elektronika i električne mašine. Master rad, na istom fakultetu, je odbranio 2022. godine.
Kontakt: boris.deak@gmail.com



Boris Dumnić rođen je 1976. godine. Zaposlen je na Fakultetu tehničkih nauka, Departmanu za energetiku, elektroniku i telekomunikacije, Katedri za energetsku elektroniku i pretvarače. Oblasti interesovanja su mu električne mašine, pogoni, energetska elektronika i obnovljivi izvori električne energije.

EKSPLOATACIJA BATTERY ENERGY STORAGE-A U DERMS-U**OPERATION OF BATTERY ENERGY STORAGE IN DERMS**Luka Bosanac, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je opisan model kao i parametri baterije za skladištenje električne energije (engl. Battery Energy Storage). Opisane su oblasti korišćenja, najpoznatiji tipovi i degradacija baterije. U praktičnom delu rada razvijena je odgovarajuća aplikacija kojom je prikazano korišćenje Battery Energy Storage-a u kombinaciji sa obnovljivim izvorima energije. Aplikacija poseduje Dashboard sa prikazom proizvodnje, potrošnje, cene i količinu energije u Battery energy storage-u u toku 24 časa. Svaki storage može raditi u tri režima: peak shaving, price arbitrage i defaultni („life save“) režim. Kao najkonkurentniji Battery Energy storage posmatra se litijum-jonska baterija spram koje se računa i degradacija u aplikaciji.

Ključne reči: model, degradacija, obnovljivi izvori energije, Battery Energy Storage

Abstract This paper describes the model as well as the parameters of the battery for electrical energy. Areas of use, the most well-known types and degradation of the battery are described. In the practical part, an appropriate application was developed which shows the use of Battery Energy Storage in combination with renewable energy sources. The application has a dashboard with a display of production, consumption, price and amount of energy in Battery Energy Storage during 24 hours. Each storage can operate in three modes: peak shaving, price arbitrage and default mode. The most competitive Battery Energy Storage is considered to be the lithium-ion battery against which the degradation in the application is calculated.

Keywords: model, degradation, renewable sources, Battery Energy Storage

1. UVOD

Battery Energy Storage (BES) je tehnologija za skladištenje električne energije koja koristi elektro-hemijske reakcije. Cilj BES-a jeste skladištenje ili upotreba električne energije na zahtev što donosi velike benefite za elektroenergetske sisteme. BES je jedna od najupotrebljivijih i najkorisnijih tehnologija u elektroenergetskim sistemima.

Kako se sve više koriste obnovljivi izvori energije, sistem podleže novim izazovima. S obzirom na varijabilnost proizvodnje obnovljivih izvora energije, predviđanje proizvodnje uveliko se usložnjava.

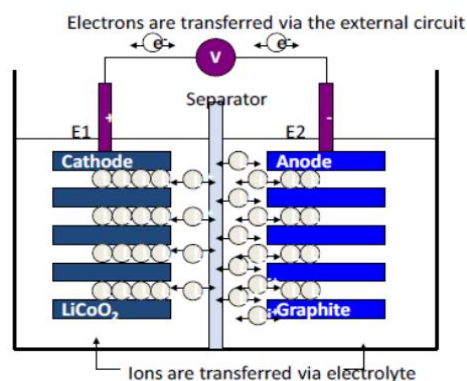
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Bošković, docent

Svaki elektroenergetski sistem teži balansu između potrošnje i proizvodnje električne energije. Ključna tehnologija za energetska održivost upravo je BES. Brojni su benefiti korišćenja BES-a kao što je energetska arbitraža, regulacija frekvencije, regulacija napona, *peak shaving* kao i korišćenje u *data* centrima i „pametnim“ građevinama [1].

2. MODEL I PARAMETRI BATERIJA

Svaki element u električnom kolu ima svoje električne parametre, a svaka od osnovnih ćelija jedne baterije ima svoju elektromotornu silu i unutrašnju otpornost. Čelije se mogu međusobno redno vezivati, čime se postiže povećanje ukupnog napona na krajevima baterije (akumulatora). Što je veći broj ćelija vezan na red to je veći napon na krajevima baterije. Kada je baterija neopterećena kroz nju ne teče struja i na krajevima baterije vlada elektromotorna sila praznog hoda. Kada se baterija opteret, strujno kolo se zatvara i struja teče kroz potrošač i bateriju. Princip rada Li-Ion ćelije baterije je prikazan na slici 1.



Slika 1. Princip rada ćelije Li-Ionske baterije

2.1. Kapacitet

Kapacitet baterije predstavlja jedan od veoma važnih parametara sa aspekta opterećenja same baterije kao izvora jednosmernog napona (struje). Jedinica koja se koristi za kapacitet baterije je amper-čas (Ah). Jedan amper-čas (1Ah) predstavlja struju baterije od jednog ampera koja protekne za vremenski interval od jednog sata (1h).

2.2. Energija

Osnovna namena svake baterije jeste da skladišti električnu energiju. Za energiju Battery Energy Storage-a, koristi se jedinica vat-čas (Wh). Energija koja se izražava jednim amper-časom predstavlja snagu od 1W koja izvrši rad za 1 sat.

2.3. Specifična energija

Specifična energija baterije je parametar koji se definiše kao uskladištena električna energija po kilogramu baterije. Jedinica kojom se izražava ova veličina jeste vat-čas po kilogramu (Wh/kg).

2.4. Gustina energije

Gustina električne energije kod baterija definiše se kao odnos uskladištene električne energije po kubnom metru (Wh/m^3).

2.5. Specifična snaga

Specifična snaga baterija izražava se u vatima po kilogramu (W/kg) ili ($\text{W}\cdot\text{kg}^{-1}$). Specifična snaga se često definiše i kao gustina energije koju je baterija u stanju da isporuči u jedinici vremena. Neke baterije imaju veoma veliku specifičnu energiju ali sa druge strane imaju lošu specifičnu snagu. Ovo znači da mogu uskladištiti veoma veliku količinu energije ali tu istu uskladištenu energiju predaju potrošaču u dužem vremenskom intervalu [4].

2.6. Ostali parametri

Pored prethodno navedenih parametara koji predstavljaju bitne podatke prilikom analize i odabira odgovarajućih, ništa manje važni su i sledeći parametri:

1. Amper-čas efikasnost
2. energetska efikasnost baterija
3. samopražnjenje baterija
4. geometrija baterija
5. temperatura, grejanje i hlađenje baterija
6. životni vek baterije
7. broj ciklusa punjenja baterija

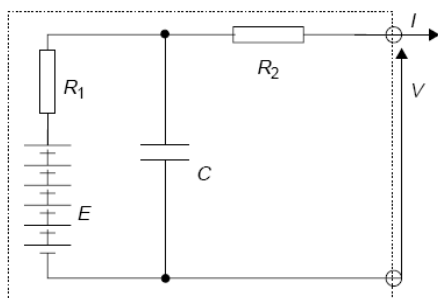
2.7 Model baterije

Modelovanje je veoma važan i koristan deo savremene inženjerske prakse. Veliki trud se danas ulaže na razvijanje dobrih i tačnih modela sistema, pre svega, kako bi se što bolje opisao njihov rad i primena istih u različitim režimima rada.

3. EKVIVALENTNO ELEKTRIČNO KOLO BATERIJE

Jedan od osnovnih parametara električnog kola baterije jeste elektromotorna sila. Stoga su inženjeri i istraživači koji se bave navedenom problematikom posvetili mnogo pažnje načinu opisivanja odnosno modelovanja elektromotorne sile baterije kao jednog od njenih osnovnih parametara.

Na slici 2. prikazano je jedno ekvivalentno električno kolo koje daje dobre rezultate prilikom modelovanja rada same baterije.



Slika 2. Ekvivalentno električno kolo baterije

4. OBLASTI KORIŠĆENJA

Battery Energy Storage-i imaju široku primenu. Najčešće se koriste u kombinaciji sa obnovljivim izvorima energije, pri regulaciji frekvencije i napona te u momentima *peak*-ova potrošnje električne energije. Takođe, uz pomoć *BES*-a, moguće je arbitrirati cenu električne energije što dovodi do velikih ušteda kako sistema za proizvodnju, tako i samog potrošača.

Pored velike podrške za obnovljive izvore energije, *BES* održava stabilnost mikromreža i *behind the meter* usluga.

4.1 Regulacija frekvencije

Usled neusklađenosti između proizvodnje i potrošnje dolazi do varijacije frekvencije. Varijacija frekvencije podrazumeva odstupanje frekvencije od nominalne vrednosti 50/60Hz (zavisno od dizajna elektroenergetskog sistema). Dodatni uticaj na varijaciju frekvencije uzrokovalo je korišćenje obnovljivih izvora energije za proizvodnju električne energije.

Kako bi sistem zadovoljilo potrebno snabdevanje energijom, neophodno je posedovati dodatnu količinu energije u rezervi. Takva se rezerva naziva *Spinning reserve*. Pored toga što treba da sadrži potrebnu količinu energije za regulaciju frekvencije, ova rezerva treba da sadrži i dovoljno energije kako bi zamenila rad najveće proizvodne jedinice u slučaju ispada iste. Na osnovu istorijskih podataka podešava se kapacitet rezerve.

4.2. Peak shaving

Jedan od najvećih i najkompleksnijih "problema" predstavlja predviđanje potrošnje električne energije. Na osnovu istorijskih podataka može se doneti veliki broj zaključaka. Brojni faktori utiču na potrošnju kao što su: temperatura, vlaga, vetar, padavine, gromovi kao i nasumični događaji koji su često nepredvidivi. Nasumični događaji zahtevaju brzu reakciju. Kod *Peak shaving-a Battery Energy Storage-a (BES)* se upotrebljava pri vršnim opterećenjima gde se koristi energija iz *BES*-a kako bi se izbegao uvoz električne energije po skupoj ceni zbog nepredviđene situacije u poslednji čas. Primeri mogu biti popularni događaji ili ispadi većih postrojenja koji imaju uticaj na potrošnju električne energije.

4.3. Integracija obnovljivih izvora energije

Najpoznatiji obnovljivi izvori energije su vetar i sunce zbog svoje održivosti i koristi za životnu sredinu. Međutim, veoma brze i slučajne promene stvaraju velike probleme za mrežu. Postoji problem u predviđanju električne energije kao i uzrokovanje različitih tehničkih problema u vidu zaštite i kvalitete električne energije. Potrebno je posedovati različite pristupe za ublažavanje variranja izlazne snage izazvane energijom vetra i sunca.

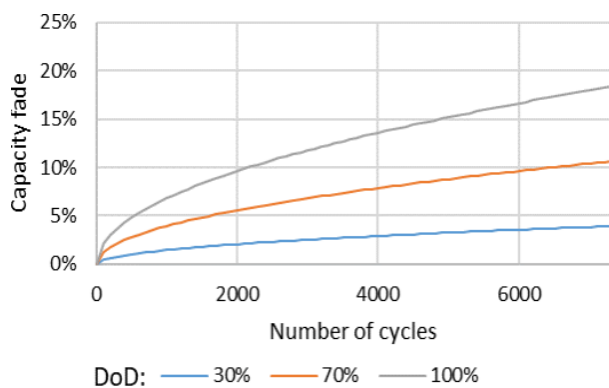
Utvrđeno je da je sistem za skladištenje električne energije iz *BES*-a najprikladniji i najefikasniji način za rešenje problema te vrste ali pod uslovom da se primenjuju efikasne strategije upravljanja kao i optimalno korišćenje *BES*-a [3].



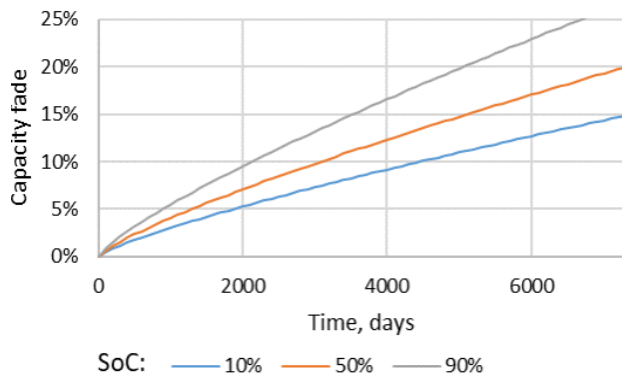
Slika 3. Rezultati simulacije

5. SIMULACIJA UPOTREBE BES-A SA OBNOVLJIVIM IZVORIMA ENERGIJE

Prikazana je demonstracija korišćenja Battery Energy Storage-a u različitim režimima tokom 24 časa kao i prikaz preostalog života BES-a uz pomoć formula za degradaciju baterije.



Slika 4. Gubitak kapaciteta pri ciklusima [2]



Slika 5. Gubitak kapaciteta BES-a pri mirovanju [2]

Na slici 3. ilustrovan je *dashboard* putem kog aplikacija pruža prikaz proizvodnje obnovljivih izvora energije, potrošnje električne energije od strane korisnika, energiju BES-a i cenu električne energije u toku 24 časa. BES radi u tri različita režima. Prvi režim je *default*-ni u kom BES čuva život baterije, drugi je *peak shaving* u kom se BES prilagođava vršnim opterećenjima isporučivanjem veće snage u toku jednog časa i *price arbitrage* režim koji manipuliše BES-om u odnosu na cenu električne energije u odgovarajućem satu. Aplikacija poseduje i mogućnost manuelnog komandovanja snage BES-a u trenutnom satu za željeni BES.

Za predviđanje preostalog života BES-a koriste se dva glavna mehanizma. Posmatra se BES u praznom hodu, tj. vreme kada se BES ne puni niti prazni i BES u toku ciklusa tj. kada je aktivno ponašanje BES-a.

Slika 4. prikazuje pad kapaciteta BES-a u funkciji broja ciklusa za različite nivoe dubine pražnjenja.

Na slici 5. prikazan je pad kapaciteta BES-a u funkciji vremena pri različitim nivoima napunjenosti BES-a.

Zbirom ove dve situacije se dobija kompletna formula za degradaciju BES-a:

$$\delta^{CF} \approx \delta^{idl} + \delta^{cyc} [2]$$

δ^{cyc} predstavlja gubitak života pri ciklusima

δ^{idl} predstavlja gubitak života u stanju mirovanja

$$\delta^{cyc} = 0.005678e^{-1.943SoC} DoD^{0.7162} \sqrt{n},$$

$$\delta^{idl} = 0.000112e^{0.7388SoC} \tau^{0.8}$$

SoC (State of Charge) - nivo napunjenosti BES-a

DoD (Depth of Discharge) - dubina pražnjenja BES-a

n – broj ciklusa

τ – vreme (u danima)

Formule su ustanovljene pri konstantnoj temperaturi od 25°C. Razni faktori utiču na termodinamiku BES-a kao što su uslovi okoline, faktor formiranja sistema, hemijski faktori, dizajn rashladnog sistema BES-a i drugi. Dubina pražnjenja utiče na stopu degradacije na netrivialan način. Sa istim protokom energije i manjom dubinom pražnjenja štedi se kapacitet BES-a. Na primer, 5000 ciklusa sa 30% DoD dovodi do manjeg propadanja od 1500 ciklusa na 100% DoD [2].

6. ZAKLJUČAK

Battery Energy Storage je sistem za skladištenje električne energije. Brojni su razlozi za upotrebu BES-a kao što je regulacija frekvencije, regulacija napona, *peak shaving*, arbitraža cene i mnoge druge. BES je i dalje u fazi istraživanja i razvoja i neophodno je da napreduje po pitanju cene, kapaciteta i dužine životnog veka. U poslednjim godinama zabeležen je veliki napredak kao i napredak ostalih modernih tehnologija i celokupnog elektroenergetskog sistema. BES učestvuje u opskrbljivanju kvalitetnom energijom kao i sigurnošću celokupnog sistema. S obzirom da je manipulisanje BES-om veoma kompleksno i može biti mnogo skuplje pogrešnim manipulisanjem, potrebno je posedovati menadžment za BES. Kreirani BES sadrži određen broj ciklusa i potrebno je postupati pažljivo kako bi se postigli što bolji rezultati za dugoročniji život. Zbog degradacije BES-a, potrebno je posedovati što tačniji model i predviđanja o životnom veku.

7. LITERATURA

- [1] Bolan Xu, Alexandre Oudalov, Andreas Ulbig, Goran Andersson, „Modeling of Lithium-Ion Battery Degradation for Cell Life Assessment“, IEEE Transactions on Smart Grid · June 2016
- [2] Timur Sayfutdinov, Petr Vorobev, Member IEEE, “Optimal Operation of Li-ion Battery Storage ”
- [3] Muhammad Jabir, Safdar Raza, Hazlee Azil Illias, Hazlie Mokhlis, „Intermittent Smoothing Approaches for Wind Power Output: A Review“, *Energies* 2017
- [4] Ivana Stojković Simatović, “Materijali za konverziju energije: ispitivanje i primena”

Kratka biografija:



Luka Bosanac rođen je 1996. godine u Vukovaru. Nakon završetka srednje škole 2015. godine, Fakultet tehničkih nauka upisuje iste godine, smer Primenjeno softversko inženjerstvo. Osnovne akademske studije je završio 2019. godine. Master akademske studije upisuje 2019. godine, smer Primenjeno softversko inženjerstvo.

Kontakt: boske_woo@live.co.uk

PARALELIZACIJA DFA ALGORITMA ZA OBUKU DUBOKIH NEURONSKIH MREŽA**PARALLELISATION OF DFA ALGORITHM FOR DEEP NEURAL NETWORK TRAINING**

Aleksandar Grahovac, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu predstavljena je paralelizacija DFA algoritma obuke neuronskih mreža na multiprocesorskom sistemu. Paralelizacija je implementirana za duboku neuronsku mrežu proizvoljnih dimenzija.

Ključne reči: DFA, duboke neuronske mreže, multiprocesori

Abstract – This paper presents a multiprocessor parallelization of the DFA algorithm for neural network training. Parallelization is implemented for a deep neural network of arbitrary dimensions.

Keywords: DFA, deep neural networks, multiprocessors

1. UVOD

U životu neuronskih mreža prisutne su dve razdvojene faze – obučavanje i upotreba. Obučavanje neuronskih mreža zapravo predstavlja postepeno profinjenje parametara mreže tako da se teži što manjoj grešci mreže. Pod greškom mreže, podrazumevamo poređenje izlaza mreže sa očekivanim izlazom.

Najčešće korišten algoritam za treniranje većine neuronskih mreža, među kojima su i duboke neuronske mreže (eng. Deep Neural Network, kratko DNN), je algoritam baziran na „propagaciji greške unazad“ (eng. backpropagation algorithm, kratko BP) [1].

BP radi tako što nakon poređenja očekivanog i dobijenog izlaza (npr. sa cross-entropy funkcijom) izračunava grešku poslednjeg sloja mreže. Ta greška se zatim, nakon algebarskih manipulacija, prosleđuje sloju pre i tako kaskadno sve do ulaznog sloja.

Kada se izračuna greška za svaki sloj, daljim algebarskim operacijama se od trenutnih parametara sloja i greške računaju novi parametri sloja. Opisana radnja se ponavlja veliki broj puta dok se ne iscrpe svi poznati parovi (ulaz, izlaz), ili kad više nema napretka u smanjenju greške i tada prestaje faza učenja.

Kada se završi faza učenja, započinje faza rada gde se kroz mrežu propuštaju ulazi za koje nisu poznati izlazi, već jedino dobijamo izlaze mreže koje dalje koristimo za druge potrebe.

2. PROBLEMI SA BP ALGORITMOM

Iako je BP algoritam široko zastupljen, postoje određeni problemi sa njim. Naime, da bi smo dobili grešku određenog sloja, moramo izračunati sve greške slojeva nakon njega. Postoje dva problema sa tim, koji su navedeni i diskutovani u delovima teksta, 2.1. i 2.2.

2.1. Problem verodostojne reprezentacije bioloških nervnih sistema

Neuronske mreže su zamišljene kao algoritamska reprezentacija pravih nervnih sistema kod bioloških organizama, i sa jedne strane njihov cilj je da ih što verodostojnije predstave (mimikuju). Problem sa BP je taj da u biološkim nervnim sistemima ne postoji mogućnost propagiranja informacija u povratnom smeru, tj. nije moguće da sloj dobije grešku od narednog sloja.

Iako ovaj problem nije ključan za inženjersku primenu, svakako predstavlja važnu kritiku, jer ukoliko se navodimo prethodnim rezultatima, verodostojno mimikovanje prirode je do sada dalo veoma dobre rezultate, tako da nije nerealno očekivati da bi sa algoritmom učenja koji bolje modeluje biološki nervni sistem rezultati bili kvalitetniji.

2.2. Problem paralelizacije algoritma učenja

Mnogo ozbiljniji problem za inženjersku primenu je činjenica da se BP algoritam nikako ne može paralelizovati po slojevima. Naime, pošto je za računanje greške datog sloja potrebna već izračunata greška narednog sloja, BP se mora izvršavati sekvencijalno po slojevima. Treba napomenuti da i dalje postoji mogućnost paralelizacije algebarskih operacija unutar sloja, npr. može se paralelizovati matrično množenje i sabiranje kojim se dobija greška sloja.

U slučaju DNN sa malim brojem slojeva, ovaj problem i nije toliko od značaja, već je značajnija paralelizacija unutar samih slojeva. Isto ne važi za mreže koje poseduju značajanu dubinu.

Napretkom industrije, nastali su veliki modeli koji nisu DNN, ali često sadrže DNN kao podstrukturu. Na primer, popularan OpenAI (doduše ne DNN) GPT-3 model sadrži 175 milijardi parametara [2], pa svakako da paralelizacija po slojevima igra mnogo bitniju ulogu.

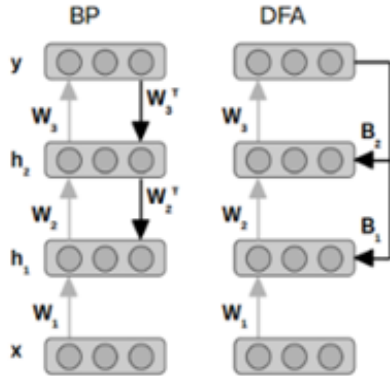
3. DFA ALGORITAM

Problem verodostojne mimike bioloških nervnih sistema još nije uspešno rešen. Jedno od manje uspešnih rešenja je tzv. „Indirect Feedback Alignment“ (krako IFA), gde se

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Staniša Dautović, vanr. prof.

za računanje greške datog sloja, greška izlaza sistema prosleđuje sloju pre, pa se algebarskim operacijama dolazi do željene greške. Iako je ovaj algoritam veoma verodostojan biološkom uslovu da nema vraćanja informacija među slojevima, on ipak za sada ne daje dovoljno dobre rezultate.



Slika 1. Idejna razlika između BP i DFA [1]

Algoritam koji rešava problem paralelizacije jeste tzv. „Dynamic Feedback Alignment“, skraćeno DFA. On se bazira na BP, uz izmenu da se greške slojeva ne računaju algebarskim operacijama nad greškom narednog sloja i trenutnih parametara, već se umesto greške narednog sloja koristi greška izlaznog sloja.

Preciznije, kada se odredi greška mreže, upravo tu vrednost prosleđujemo svakom sloju. Dok se kod BP za potrebe algebarskih operacija koriste iste matrice težina kao i kod propuštanja ulaza kroz mrežu, DFA koristi nove matrice B_i , koje se randomizovano generišu na početku rada programa i ne menjaju se tokom učenja.

Formalno, DNN je potpuno opisan sledećim jednačinama

$$\begin{aligned} a_i &= W_i h_{i-1} + b_i, i = 1 \dots N \\ h_0 &= x \\ h_i &= f(a_i), i = 1 \dots N \\ h_N &= y, \end{aligned} \quad (1)$$

gde su redom:

- a – izlaz pre aktivacione funkcije;
- W – težinski parametri mreže;
- b – bias parametri mreže;
- h – izlaz sloja, tj. izlaz nakon aktivacione funkcije;
- f – odabrana aktivaciona funkcija;
- i – redni broj sloja.

U slučaju DNN sa cross-entropy funkcijom greške, greška izlaznog sloja e se može lako odrediti

$$e = y^{\wedge} - y, \quad (2)$$

gde je $y^{\wedge}$ očekivan izlaz, y izračunat izlaz mreže. Za DFA, greška i -tog sloja iznosi

$$\delta a_i = (B_i e) \odot f'(a_i), \quad (3)$$

gde je $\odot$ Hadamardov proizvod; dok za težine važi

$$\begin{aligned} \delta W_i &= -\delta a_i h_{i-1}^T \\ \delta b_i &= -\delta a_i. \end{aligned} \quad (4)$$

Upravo jednačine (3) i (4) predstavljaju osnovu implementacije DFA.

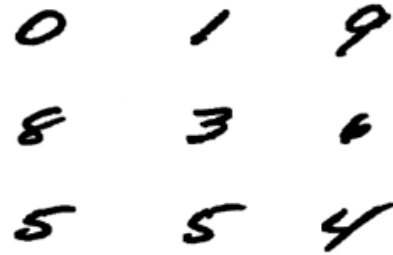
4. IMPLEMENTACIJA DFA ALGORITMA

Za potrebe ovog rada implementiran je DFA algoritam za obučavanje fully-connected DNN proizvoljnih dimenzija, sa cross-entropy funkcijom greške i ReLU aktivacionom funkcijom. Takođe, korišten je i bias u svakom sloju.

Implementacija na jednom procesoru je izvršena u programskom jeziku Python, dok je multiprocesorska verzija implementirana takođe u jeziku Python, ali i u programskom jeziku C.

4.1 Implementacija na jednom procesoru

Za potrebe implementacije na jednom procesoru, za početak, neophodno je bilo odabrati skup parova (ulaz, izlaz). Za potrebe ovog rada odabrana je poznata cifarska MNIST baza podataka [3].



Slika 2. Primeri MINST cifarskih ulaza [3]

Nakon što su slike učitane, one se skaliraju da bi odgovarale ulazu mreže i konačno se prosleđuju proceduri za obučavanje mreže.

Procedura za obučavanje mreže od parametara prihvata sledeće:

- trening podaci, gde su ulazi matrice dimenzija 14x14 sa vrednostima u opsegu (-0.5, 0.5) – koristi se za obučavanje mreže;
- podaci za validaciju, istih karakteristika kao prethodni – koristi se za proveru preciznosti mreže nakon svake epohe učenja; ključno svojstvo ovog niza je da njegove elemente mreža ne koristi tokom popravljavanja parametara, tj. predstavljaju uzorak koji mreža nije videla ranije;
- broj epoha – epoha predstavlja broj interakcija algoritma, tj. broj ponavljanja istih trening podataka;
- dimenzije mreže – date su nizom koji sadrži broj neurona za svaki od slojeva mreže; veličina niza jedino je ograničena hardverom;
- koeficijent p – udeo elementa niza (ulaz, izlaz) za obučavanje koji se koristi za validaciju; nakon svake epohe elementi za validaciju se ponovo biraju; pretpostavljena vrednost iznosi 0.2;
- koeficijent učenja – predstavlja broj sa kojim se množi izračunata promena parametra, da bi se izbegle nagle promene; pretpostavljena vrednost iznosi 0.05.

U funkciji za obučavanje mreže, prvo se izvrši faza inicijalizacije gde se, sa jedne strane, inicijalizuju matrice težina i matrice za računanje greške, a sa druge strane od prosleđenih parametara funkciji se računaju često korištene veličine koje predstavljaju rezultat algebarskih opera-

cija nad ulaznim veličinama, što se radi da bi se ubrzalo izvršenje programa.

Inicijalizacija matrica težina i matrica za računanje greške se najčešće vrši sa normalnom distribucijom, mada se sa odabirom drugih distribucija mogu dobiti bolji rezultati.

Zatim, vrši se iteracija po broju epoha, gde se unutar svake epohe prvo niz za obučavanje mreže deli na podniz koji se koristi u ovoj epohi za obučavanje i podniz koji se u ovoj epohi koristi za procenu tačnosti mreže. Nakon toga se svih $(1-p)$ slika namenjenih za obuku u datoj epohi propušta kroz mrežu i dobijaju se izlazi mreže.

Pošto su sada poznati očekivani izlazi i dobijeni izlazi, algebarskim operacijama opisanim pod (3) i (4) dobijamo vrednosti za popravku parametara. Pre korekcije parametara, ove vrednosti se množe koeficijentom učenja i konačno se dodaju na postojeće parametre.

Na kraju, kroz mrežu se propuštaju podniz za proveru preciznosti mreže konstruisan od niza za obuku mreže, kao i podaci za validaciju. Rezultati propuštanja se ispisuju u konzoli.

Nakon svih epoha, trenutna stanja parametara se pamte i program se završava.

Takođe, implementirano je, kao zasebni program, procedura za propuštanje ulaza kroz mrežu. Ona koristi dobijene parametre nakon treniranja mreže i izvršava matrično množenje po slojevima i primenjuje aktivacionu funkciju.

Osim procedure propuštanja, implementirana je i test funkcija koja ispisuje procenat tačno predviđenih izlaza.

4.2. Multiprocesorska implementacija DFA algoritma

Za podrebe multiprocesorske implementacije odabrana je tehnika komunikacije porukama, tačnije MPI standard.

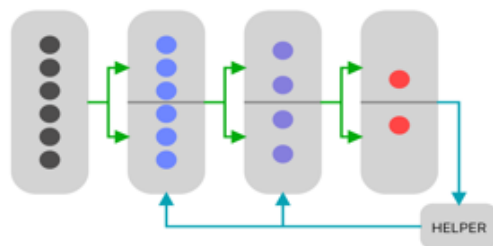
Podela procesa na procesore je učinjena na sledeći način – svaki sloj mora da sadrži barem jedan procesor koji upravlja datim slojem, i dodatno mora da postoji tzv. pomoćni procesor koji upravlja prosleđivanjem greške izlaza ka slojevima. Ukoliko je prisutno više procesora od minimalnog potrebnog broja, oni se redom dodeljuju slojevima, tako da više procesora ravnomerno deli sloj. Šema opisane implementacije je data na slici 3. Na slici, crnom bojom su predstavljeni ulazni, plavom duboki, a crvenom izlazni neuroni. Za duboke slojeve, kao i izlazni sloj, sivom linijom je predstavljena podela na procese po sloju. Dodatno, prisutan je i pomoćni proces koji upravlja povratom spregom.

Sam program radi na sledećem principu – tokom prostog propuštanja kroz mrežu, svaki procesor čeka da dobije svoje ulazne podatke, nad kojima izvrši algebarske operacije množenja matrica i primeni aktivacionu funkciju.

Funkcija obučavanja je takođe modifikovana. Faza inicijalizacije dodatno sadrži računanje pozicije procesora, tj. kom sloju i kom delu sloja dati procesor pripada. Zbog bržeg izvršavanja, ulazni podaci se ne šalju porukama, već ih svaki procesor kojem su potrebni sadrži lokalno, tako što ih prilikom pokretanja programa svi procesori učitaju zasebno.

Zatim procesori ulaznog sloja kreću sa računanjem izlaza, dok ostali procesori čekaju na rezultate. Kada procesori

ulaznog sloja završe, svaki od procesora narednog sloja dobija rezultate svakog od procesora prethodnog sloja i neophodno ih je spojiti u jedan niz. Opisan proces se ponavlja do poslednjeg sloja, koji izlaz prosleđuje pomoćnom procesoru.



Slika 3. Idejna šema podele sistema za više jezgara

Pomoćni procesor izlaze spaja u jedan niz i prosleđuje ih svim procesorima zarad računanja greške.

Važno je primetiti da do ovog trenutka još ne postoji paralelizam.

Nakon što su svi procesori dobili grešku, oni konkurentno računaju za koliko treba da poprave svoje parametre i zatim ih i poprave.

Čim procesor završi popravku koeficijenta, nastavlja sa prethodnim radom, tj. procesori ulaznog sloja uzimaju nove ulaze, dok ostali procesori čekaju svoj red.

Na kraju svake epohe, pomoćni procesor ispisuje rezultate validacije, tj. procenat pogodenih izlaza, kao i u prethodnoj, sekvencijalnoj verziji.

Nakon završetka programa, svaki procesor lokalno čuva parametre, a moguće ih je i spojiti u jednu celinu i čuvati na taj način.

Ukoliko je odabrano lokalno, tj. izdvojeno čuvanje, potreban je poseban multiprocesorski program za propuštanje ulaza kroz distribuiranu mrežu. Taj program je takođe implementiran, a dodatno, slično kao i ranije, poseduje test funkciju koja izbacije procenat pogodenih ulaza.

5. ANALIZA REZULTATA

5.1. Vremensko ubrzanje

Posmatrajmo prvo teorijski odnos brzine verzije na jednom i na više procesora.

Za jedan procesor, vremenska kompleksnost učenja je jednostavna, pošto je izvršavanje po slojevima sekvencijalno, što rezultuje sa

$$T_{single} = \sum_{i \in \text{slojevi}} pass_i + \sum_{i \in \text{slojevi}} update_i. \quad (5)$$

Za multiprocesorsku verziju, ako pretpostavimo da svaki sloj ima tačno jedan procesor, tj. imamo minimalan potreban broj procesora, očigledno je da će vremenska kompleksnost učenja biti

$$T_{multi} = \sum_{i \in \text{slojevi}} pass_i + \max_{i \in \text{slojevi}} \{update_i\}. \quad (6)$$

Ako pretpostavimo zarad lakše analize, da su svi slojevi jednakih dimenzija i shodno tome da zahtevaju jednako vremena, dobijamo sledeći odnos

$$\frac{T_{single}}{T_{multi}} = \frac{N \cdot pass + N \cdot update}{N \cdot pass + update}, \quad (7)$$

što za funkcije propuštanja i funkcije korekcije parametara istog reda kompleksnosti u limesu daje poboljšanje od 2.

Jasno je da to nije idealno, ali nažalost propuštanje kroz mrežu je fundamentalno sekvencijalno po slojevima i samim tim se ne može popraviti bolje od opisanog.

Jedan način da se ovo ograničenje prevaziđe jeste paralelizacija unutar samog sloja, koja se u opisanoj implementaciji i dešava ukoliko je na raspolaganju broj procesora veći od minimalnog. Jasno je da ovo ubrzanje nije nužno vezano za DFA algoritam, već se može primeniti i za BP.

Kada se primeni i drugi vid paralelizacije, smanjuje se i vreme izvršavanja i vreme obuke, oba k puta

$$\begin{aligned} \frac{T_{single}}{T_{multi}} &= \frac{N \cdot pass}{\frac{N}{k} \cdot pass} = k, \\ \frac{T_{single}}{T_{multi}} &= \frac{N \cdot pass + N \cdot update}{\frac{N}{k} \cdot pass + \frac{1}{k} \cdot update}. \end{aligned} \quad (8)$$

Primitimo da isto vredi i za BP, tako da u limesu BP i DFA daju isto ubrzanje ako se koriste sve pomenute paralelizacije. Ali u realnom slučaju, DFA daje ipak bolje vreme jer na dva načina paralelizuje popravljavanje parametara.

Što se tiče tačnosti, pokazano je da DFA ima nešto lošiji učinak od BP [4], ali potrebno je dalje istraživanje da bi se donela konačna odluka.

U našoj implementaciji, za veoma kratko treniranje (do 1s) na 1000 slika, dobija se preciznost od 90% za slike koje nisu prethodno videne.

5.2. Eksplozija parametara

Analizom različitih arhitektura ustanovljeno je da nisu sve arhitekture stabilne. Naime, ukoliko između dva sloja postoji nagla promena broja neurona, prilikom obučavanja mreže vrednosti parametara divergiraju (eksplozija parametara), nakon čega se dolazi do prekoračenja opsega i dalja obuka mreže nije moguća.

Precizna definicija nagle promene još nije poznata, ali je empirijski utvrđeno da velik broj arhitektura koje sadrže dva susedna sloja od kojih je jedan veći za barem 50% poseduju opisan problem.

5.3. Kvantitativni rezultati

Implementirana je mreža dimenzija (br_ulaza, br_neur_sloja_1, br_neur_sloja_2, br_neur_sloja_3, br_izlaza) = (196, 100, 50, 30, 10) sa 10000 podataka, za slučaj multiprocesorske verzije u C-u, nad 15 procesora. Maksimalna dosegnuta preciznost iznosi 92%. Preciznost od 90% je dosegnuta nakon 10 iteracija algoritma. Vreme izvršavanja jedne iteracije je 1.3 sekunde.

U poređenju sa rezultatima dobijenim sa bibliotekom Keras za BP algoritam sa identičnim parametrima imamo maksimalnu dosegnutu preciznost od 92%, gde je preciznost od 90% dosegnuta nakon 3 iteracije. Ključni parametar čini vreme izvršavanja jedne iteracije koje u proseku iznosi čak 14,4s.

Jasno je da postoji značajno unapređenje što se tiče vremena izvršavanja algoritma, oko 11 puta, ali je prisutna degradacija broja iteracija potrebnih da se dosegne preciznost od 90%. Sama preciznost mreže je ostala gotovo nepromenjena.

Treba naglasiti da biblioteka Keras koristi posebne algoritme optimizacije klasičnog BP algoritma kao i da se program izvršavao na Google Colab GPU mašinama.

6. ZAKLJUČAK

U ovom radu opisan je rad DFA algoritma za obuku neuronskih mreža i objašnjen je na implementaciji na jednom procesoru, kao i na multiprocesorskom sistemu.

Iako sam DFA algoritam ne može ubrzati funkciju propuštanja kroz neuronsku mrežu, on paralelizacijom može ubrzati obučavanje iste.

Nažalost, DFA poseduje dva problema – pošto je propuštanje ulaza kroz mrežu jedan od delova obučavanja mreže, postoji limit koliko se može ubrzati učenje paralelizacijom baziranom na svojstvima DFA, ali moguća je paralelizacija unutar slojeva.

Drugi problem predstavlja eksplozija parametara, tj. činjenica da za određene strukture, vrednosti parametara divergiraju. Ovaj problem nema jednostavno rešenje, već se jedino može zaobići ili upotrebom potpuno drugog algoritma koji se ne bazira na BP algoritmu (npr. Hebovo učenje) ili konstrukcijom mreže bez naglih promena dimenzija između slojeva.

7. LITERATURA

- [1] Nøklund A., “Direct Feedback Alignment Provides Learning in Deep Neural Networks”, 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.
- [2] OpenAI, “Language Models are Few-Shot Learners”, Advances in Neural Information Processing Systems 33 (NeurIPS 2020).
- [3] Grother P. J., “MNIST Special Database 19”, National Institute of Standards and Technology, 1995.
- [4] Launay J., Poli I., Boniface F., Krzakala F., “Direct Feedback Alignment Scales to Modern Deep Learning Tasks and Architectures”, 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Kratka biografija:

Aleksandar Grahovac rođen je u Subotici 1999. god. Diplomirao je 2021. god. na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva, smer Mikroracunarska elektronika, grupa Embeded sistemi i algoritmi. Iste godine na istoj grupi upisuje master akademске studije.
kontakt: aleksandar.g1999@gmail.com

PRORAČUN NESIMETRIČNIH TOKOVA SNAGA DISTRIBUTIVNIH MREŽA SA DISTRIBUTIVNIM GENERATORIMA ZASNOVANIM NA UREĐAJIMA ENERGETSKE ELEKTRONIKE

CALCULATION OF UNBALANCED POWER FLOW OF DISTRIBUTION NETWORKS WITH DISTRIBUTED GENERATORS BASED ON POWER ELECTRONIC DEVICES

Tijana Terzić, Luka Strezoski, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu su prikazane osnovne informacije o ponašanju distribuiranih resursa u normalnim režimima, a potom i upravljačke strategije koje su korišćene u njima. Cilj je bio proračunati tokove snaga distributivnih mreža sa distributivnim generatorima, pa su se morali uvesti i novi tipovi čvorova.

Ključne reči: Distributivne mreže, Distribuirani energetske resursi, Nesimetrični tokovi snaga

Abstract: The paper presents basic information on the behavior of distributed resources in normal regimes, and then the management strategies used in them. The goal was to calculate the power flow in distribution network with distributive generators, so new types of nodes had to be introduced.

Keywords: Distribution networks, Distributed energy resources, Unbalanced power flow

1. UVOD

U današnje vreme suočavamo se sa ekološkim problemima, koji dovode do pitanja kako namiriti sve veće potrebe za proizvodnjom električne energije sa smanjenim uticajem na životnu sredinu. Utvrđeno je da proizvodnja električne energije iz obnovljivih izvora ima znatno manji uticaj na zagađenje okoline od proizvodnje iz neobnovljivih izvora.

Distribuirani energetske resursi (DER) su distribuirani generatori (DG), distribuirana skladišta električne energije i hibridni DER. Priključenjem sve većeg broja distribuiranih energetske resursa na distributivne mreže, one postaju složenije, kao i procesi koji se u njima odvijaju. Tradicionalni model za proračun tokova snaga se više ne može koristiti, pa se i tradicionalna klasifikacija čvorova mora upotpuniti. Jednofaznim čvorovima tipa θV , PQ i PV su pridruženi novi trofazni čvorovi.

U ovom radu je urađen proračun tokova snaga za datu mrežu koju čini osam čvorova u programskom jeziku Fortran i razvojnom okruženju Microsoft Visual Studio 2010 i dobijeni rezultati prikazani su tabelarno i grafički.

2. NORMALIZACIJA KOLA

Normalizacija je proces transformacije veličina iz domena apsolutnih vrednosti u domen relativnih vrednosti. Opšti

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Luka Strezoski.

oblik transformacije u domen relativnih jedinica prikazan je u izrazu (1). U izrazu (2) predstavljen je skup mogućih veličina u apsolutnim jedinicama, a u (3) skup baznih veličina koje na osnovu (1) moraju biti različite od nule.

$$\hat{x} = \frac{\tilde{x}}{X^b}, \quad X^b \neq 0 \quad (1)$$

$$\tilde{x} \in \{S, U, I, Z, Y\} \quad (2)$$

$$X^b \in \{S^b, U^b, I^b, Z^b, Y^b\} \quad (3)$$

Osnovne bazne veličine su bazna snaga i bazni naponi i oni moraju biti različiti od nule. Prilikom izbora baznih napona treba voditi računa da je cilj normalizacije eliminisati što veći broj transformatora, odnosno odnosa transformacije. Na osnovu osnovnih baznih veličina, izvođe se odnosi transformacije, struje, impedanse, admitanse i to za svaki naponski nivo [3].

3. PONAŠANJE DISTRIBUIRANIH RESURSA U NORMALNIM REŽIMIMA

Neke od tehnologija za proizvodnju električne energije su: fotonaponski sistemi, vetroelektrane, mikroturbine, gorivne ćelije, motori sa unutrašnjim sagorevanjem. Oni mogu biti direktno vezani na mrežu, mogu biti vezani delimično preko uređaja energetske elektronike ili potpuno preko uređaja energetske elektronike.

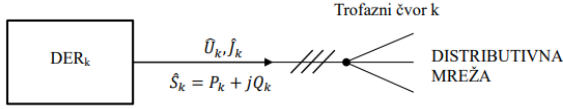
Obnovljivi izvori energije imaju promenljivu proizvodnju električne energije koju je teško kontrolisati i koja zavisi od vremenskih prilika, godišnjih doba itd. Oni su teško upravljivi kada je neophodno kontrolisati aktivnu snagu. Kako bi se ti nedostaci prevazišli, u energetske sistem se uvode distribuirana skladišta električne energije. Njihov rad se zasniva na tome da kada se javi period niske potrošnje, višak energije se skladišti u distribuirana skladišta energije.

Za razliku od toga, kada se javi period povećane potrošnje, energija se iz distribuiranih skladišta transformiše u električnu energiju i isporučuje gde je neophodna. Primer su baterije akumulatora i zamajci [1, 2].

4. UPRAVLJAČKE STRATEGIJE IMPLEMENTIRANE U DER

Posmatra se DER_k koji je na nesimetričnu distributivnu mrežu priključen u trofaznom čvoru k i ta situacija je prikazana na slici 1 na osnovu [1]. Čvor k je opisan osnovnim i izvedenim faznim veličinama. Osnovne

veliĉine su fazni naponi $\hat{U}_k$, aktivne snage P_k i reaktivne snage Q_k , a izvedene veliĉine su injektirane struje $\hat{J}_k$.



Slika 1 – DER_k prikljuĉen u ĉvoru k

Oni mogu biti izraĉeni u faznom domenu, kao i u domenu simetriĉnih komponenti. Naponi su izraĉeni modulima i uglovima, a snage su u kompleksnom obliku. Veza izmeĉu ova dva domena ostvaruje se matricama transformacije.

Trofazne snage u domenu simetriĉnih komponenti su jednake:

$$\begin{aligned}\hat{S}_{k\Sigma} &= P_{k\Sigma} - jQ_{k\Sigma} = 3(\hat{S}_k^d - \hat{S}_k^i - \hat{S}_k^o) \\ &= 3(\hat{U}_k^{d*} \hat{J}_k^d - \hat{U}_k^{i*} \hat{J}_k^i - \hat{U}_k^{o*} \hat{J}_k^o) \\ &= 3[(\text{Re}\{\hat{U}_k^{d*} \hat{J}_k^d\} + j\text{Im}\{\hat{U}_k^{d*} \hat{J}_k^d\}) \\ &\quad - (\text{Re}\{\hat{U}_k^{i*} \hat{J}_k^i\} + j\text{Im}\{\hat{U}_k^{i*} \hat{J}_k^i\}) \\ &\quad - (\text{Re}\{\hat{U}_k^{o*} \hat{J}_k^o\} + j\text{Im}\{\hat{U}_k^{o*} \hat{J}_k^o\})] \quad (4)\end{aligned}$$

Energetski resursi zasnovani na energetskoj elektronici imaju upravljaĉke strategije koje su sloŹenije od upravljaĉkih strategija tradicionalnih naizmeniĉnih maŹina. Oni omogućavaju da se u nesimetriĉnim režimima injektiraju simetriĉne struje, tako Źto inverzna i nulta komponenta struje dobijaju vrednosti nula. Mogu obezbediti i da se odrŹavaju simetriĉni naponi u nesimetriĉnim režimima mreŹe. Ovo dovodi do velikih promena kada je u pitanju model za proraĉun tokova snaga. Tri pofazna ĉvora bila su dovoljna za proraĉun simetriĉnih režima, pa se sada moraju proširiti.

5. MODELI ĆVOROVA ZA PRORAĆUN TOKOVA SNAGA

U novim mreŹama ima $3N$ kompleksnih napona, odnosno $2N$ realnih i $2N$ kompleksnih veliĉina stanja. Promenljive stanja prikazane u domenu simetriĉnih komponenti su direktnog, inverznog i nultog redosleda. Naponi direktnog redosleda su opisani pomoću $2N$ realnih veliĉina, odnosno N veliĉina koje odgovaraju vrednostima modula i N veliĉina koje odgovaraju vrednostima ugla napona. Naponi inverznog i nultog redosleda su u kompleksnom domenu i opisani su sa po N kompleksnih veliĉina.

Za predstavljanje tradicionalnih naizmeniĉnih maŹina u nesimetriĉnim režimima uvode se ĉvorovi tipa: $(\theta V)_\Sigma$, $P_\Sigma Q_\Sigma$ i $P_\Sigma V$. Ostali ĉvorovi modeluju savremene distribuirane energetske resurse. Moguće je obezbediti simetriĉne fazne napone u nesimetriĉnom režimu mreŹe i to zahvaljujući ĉvorovima tipa: $P_\Sigma Q_\Sigma V_{sim}$ i $P_\Sigma V_{sim}$. Ćvorovi $P_\Sigma Q_\Sigma I$ i $P_\Sigma VI$ modeluju upravljaĉke strategije kojima se moŹe kontrolisati inverzna komponenta struje u nesimetriĉnim režimima. Dva trofazna ĉvora su izvedena iz tradicionalnih tipova: $3\theta V$ i $3PQ$. Ova klasifikacija ĉvorova sa njihovim zadatim i nepoznatim veliĉinama data je u tabeli 1 [1].

Tabela 1 – Nova klasifikacija ĉvorova

Tip ĉvora	Zadate veliĉine	Nepoznate veliĉine
θV	θ, U	/
PQ	P, Q	θ, U
PV	P, U	θ
$(\theta V)_\Sigma$	θ^d, U^d	$\hat{U}^i, \hat{U}^o$
$P_\Sigma Q_\Sigma$	P_Σ, Q_Σ	$\theta^d, U^d, \hat{U}^i, \hat{U}^o$
$P_\Sigma V$	P_Σ, U^d	$\theta^d, \hat{U}^i, \hat{U}^o$
$P_\Sigma Q_\Sigma V_{sim}$	$P_\Sigma, Q_\Sigma, \hat{U}^i, \hat{U}^o$	θ^d, U^d
$P_\Sigma V_{sim}$	$P_\Sigma, U^d, \hat{U}^i, \hat{U}^o$	θ^d
$P_\Sigma Q_\Sigma I$	$P_\Sigma, Q_\Sigma, \hat{I}^i$	$\theta^d, U^d, \hat{U}^i, \hat{U}^o$
$P_\Sigma VI$	$P_\Sigma, U^d, \hat{I}^i$	$\theta^d, \hat{U}^i, \hat{U}^o$
$3\theta V$	$\theta_{ka}, U_{ka}, \theta_{kb}, U_{kb}, \theta_{kc}, U_{kc}$	/
$3PQ$	$P_{ka}, Q_{ka}, P_{kb}, Q_{kb}, P_{kc}, Q_{kc}$	$\theta^d, U^d, \hat{U}^i, \hat{U}^o$

6. PRORAĆUN NESIMETRIĆNIH TOKOVA SNAGA DISTRIBUTIVNIH MREŽA SA DG

Analizira se trofazna mreŹa koja ima N trofaznih ĉvorova. Prvi ĉvor je oznaĉen kao balansni ĉvor. Ukupan broj ĉvorova mreŹe i brojevi elemenata skupova mreŹe povezani su relacijom (5), a skupovi indeksa razliĉitih tipova ĉvorova sa skupom $\{2, 3, \dots, N\}$ povezani su relacijom (6). Model tokova snaga je prikazan na osnovu [1,2].

$$\begin{aligned}1 + N_{P_\Sigma Q_\Sigma} + N_{P_\Sigma V} + N_{P_\Sigma Q_\Sigma V_{sim}} + N_{P_\Sigma V_{sim}} + N_{P_\Sigma Q_\Sigma I} \\ + N_{P_\Sigma VI} + N_{3PQ} \\ = N\end{aligned} \quad (5)$$

$$\begin{aligned}\alpha_{N_{P_\Sigma Q_\Sigma}} \cup \alpha_{N_{P_\Sigma V}} \cup \alpha_{N_{P_\Sigma Q_\Sigma V_{sim}}} \cup \alpha_{N_{P_\Sigma V_{sim}}} \cup \alpha_{N_{P_\Sigma Q_\Sigma I}} \\ \cup \alpha_{N_{P_\Sigma VI}} \cup \alpha_{N_{3PQ}} \\ = \{2, 3, \dots, N\} \quad (6)\end{aligned}$$

Sada se mogu izdvojiti vektori nepoznatih veliĉina stanja mreŹe direktnog, inverznog i nultog redosleda:

$$\begin{aligned}X^d = [\theta_k^d, k \in \{2, 3, \dots, N\}, U_k^d, k \\ \in \alpha_{P_\Sigma Q_\Sigma} \cup \alpha_{P_\Sigma Q_\Sigma V_{sim}} \cup \alpha_{P_\Sigma Q_\Sigma I} \cup \alpha_{3PQ}]^T \quad (7)\end{aligned}$$

$$\hat{X}^i = [\hat{U}_k^i, k \in \{2, 3, \dots, N\}, k \notin \alpha_{P_\Sigma Q_\Sigma V_{sim}} \cup \alpha_{P_\Sigma V_{sim}}]^T \quad (8)$$

$$\hat{X}^o = [\hat{U}_k^o, k \in \{2, 3, \dots, N\}, k \notin \alpha_{P_\Sigma Q_\Sigma V_{sim}} \cup \alpha_{P_\Sigma V_{sim}}]^T \quad (9)$$

Vektor X^d se sastoji od $(N - 1 + N_{P_\Sigma Q_\Sigma} + N_{P_\Sigma Q_\Sigma V_{sim}} + N_{P_\Sigma Q_\Sigma I} + N_{3PQ})$ realnih nepoznatih veliĉina stanja, a vektori $\hat{X}^i$ i $\hat{X}^o$ od $2(N - 1 - N_{P_\Sigma Q_\Sigma V_{sim}} - N_{P_\Sigma V_{sim}})$ kompleksnih nepoznatih veliĉina stanja. Kako bi se odredile nepoznate veliĉine, potrebno je obezbediti isti broj jednaĉina modela. On je predstavljen u domenu simetriĉnih komponenti. Model nesimetriĉnih tokova snaga ima sledeći izgled:

$$P_k^d = \text{Re} \left\{ \hat{U}_k^{d*} \sum_{l=1}^N \hat{y}_{kl}^d \hat{U}_l^d \right\} = g_k'(X^d), \quad k \in \{2, 3, \dots, N\} \quad (10)$$

$$\begin{aligned}Q_k^d = -\text{Im} \left\{ \hat{U}_k^{d*} \sum_{l=1}^N \hat{y}_{kl}^d \hat{U}_l^d \right\} = g_k''(X^d), \quad k \\ \in \alpha_{P_\Sigma Q_\Sigma} \cup \alpha_{P_\Sigma Q_\Sigma V_{sim}} \cup \alpha_{P_\Sigma Q_\Sigma I} \\ \cup \alpha_{3PQ} \quad (11)\end{aligned}$$

$$-f_k^i - \hat{y}_{k1}^i \hat{U}_1^i = \sum_{l=2}^N \hat{y}_{kl}^i \hat{U}_l^i, \quad k \in \{2, 3, \dots, N\} \text{ i } k \notin \alpha_{P_{\Sigma}Q_{\Sigma}V_{sim}} \cup \alpha_{P_{\Sigma}V_{sim}} \quad (12)$$

$$-\hat{f}_k^o - \hat{y}_{k1}^o \hat{U}_1^o = \sum_{l=2}^N \hat{y}_{kl}^o \hat{U}_l^o, \quad k \in \{2, 3, \dots, N\} \text{ i } k \notin \alpha_{P_{\Sigma}Q_{\Sigma}V_{sim}} \cup \alpha_{P_{\Sigma}V_{sim}} \quad (13)$$

Model se sada svodi na $(N - 1 + N_{P_{\Sigma}Q_{\Sigma}} + N_{P_{\Sigma}Q_{\Sigma}V_{sim}} + N_{P_{\Sigma}Q_{\Sigma}I} + N_{3PQ})$ nelinearnih realnih jednačina i $2(N - 1 - N_{P_{\Sigma}Q_{\Sigma}V_{sim}} - N_{P_{\Sigma}V_{sim}})$ kompleksnih linearnih jednačina. Rešavanjem ovih jednačina dobija se $(N - 1 + N_{P_{\Sigma}Q_{\Sigma}} + N_{P_{\Sigma}Q_{\Sigma}V_{sim}} + N_{P_{\Sigma}Q_{\Sigma}I} + N_{3PQ})$ realnih veličina direktnog redosleda i $2(N - 1 - N_{P_{\Sigma}Q_{\Sigma}V_{sim}} - N_{P_{\Sigma}V_{sim}})$ kompleksnih veličina inverznog i nultog redosleda.

U kraćem obliku model se može zapisati:

$$(\Delta S^d)^h = JAC^h(\Delta X^d)^h \quad (14)$$

$$(\hat{f}^i)^h = \hat{f}^i(\hat{X}^i)^{h+1} \quad (15)$$

$$(\hat{f}^o)^h = \hat{f}^o(\hat{X}^o)^{h+1} \quad (16)$$

Postupak proračuna tokova snaga se ponavlja sve dok model ne konvergira i ispuni sledeće uslove:

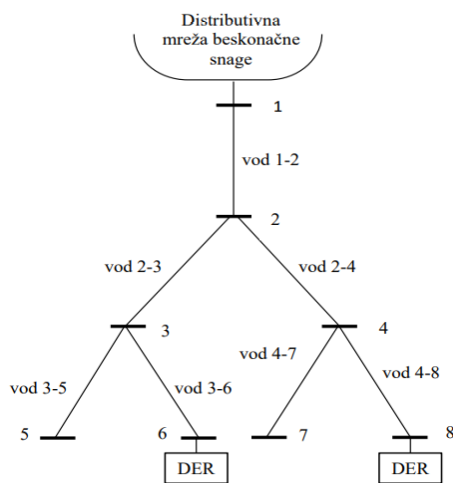
$$(\Delta \theta_k^d)^h < \varepsilon_1, \quad k \in \{2, 3, \dots, N\} \quad (17)$$

$$(\Delta U_k^d)^h < \varepsilon_2, \quad k \in \left\{ \alpha_{P_{\Sigma}Q_{\Sigma}} \cup \alpha_{P_{\Sigma}Q_{\Sigma}V_{sim}} \cup \alpha_{P_{\Sigma}Q_{\Sigma}I} \cup \alpha_{3PQ} \right\} \quad (18)$$

7. REZULTATI PRORAČUNA TOKOVA SNAGA

1.1 Projektni zadatak

U ovom radu za proračun nesimetričnih tokova snaga distributivnih mreža sa distributivnim generatorima zasnovanim na uređajima energetske elektronike analizira se trofazna mreža koja ima osam čvorova i sedam vodova. Njena konfiguracija je prikazana na slici 2.



Slika 2 – Konfiguracija mreže

Mreža se napaja iz korena mreže, koji je povezan na mrežu beskonačne snage.

Prvi čvor je balansni, tipa $3\theta V$, čvorovi 2, 3, 4, 5 i 7 su potrošački, tipa $3PQ$, a generatorski čvorovi su 6 i 8 tipa

$P_{\Sigma}V$ i $P_{\Sigma}Q_{\Sigma}I$. Svi vodovi su jednakih dužina po 1 km i jednakih parametara koji su dati u tabeli 2.

Tabela 2 – Podaci o vodovima

$r^d = r^l$	$x^d = x^l$	$g^d = g^l$	$b^d = b^l$	r^o	x^o	g^o	b^o
$\left[\frac{\Omega}{km} \right]$	$\left[\frac{\Omega}{km} \right]$	$\left[\frac{S}{km} \right]$	$\left[\frac{\mu S}{km} \right]$	$\left[\frac{\Omega}{km} \right]$	$\left[\frac{\Omega}{km} \right]$	$\left[\frac{S}{km} \right]$	$\left[\frac{\mu S}{km} \right]$
0,40	0,36	0,00	300,00	2,00	1,80	0,00	60,00

Ovi podaci omogućavaju formiranje matrice admitansi sva tri redosleda, što je urađeno na osnovu [3]. Režim mreže je nesimetričan zbog nesimetrije napona faza balansnog čvora i različitih vrednosti snaga po fazama potrošačkog područja. Potrebni podaci se nalaze u narednim tabelama. Modul i ugao napona balansnog čvora faze A su: 20,00 kV i 0°, faze B: 20,50 kV i -115°, faze C: 21,00 kV i -220°.

Tabela 3 – Podaci o klasičnom DER

S_n [MW]	V_n [kV]	x^d [%]	x^i [%]	x^o [%]
5	20	110	20	10

Tabela 4 – Podaci o potrošačkom području

P_a	P_b	P_c	Q_a	Q_b	Q_c
[MW]	[MW]	[MW]	[MVar]	[MVar]	[MVar]
1,50	1,00	0,50	0,75	0,50	0,25

U čvoru 6 su trofazna aktivna snaga i direktna komponenta napona redom jednake $P_{\Sigma} = 1$ MW i $U^d = 20\,500$ V. U čvoru 8 trofazna aktivna i reaktivna snaga su $P_{\Sigma} = 1$ MW i $Q_{\Sigma} = 0,5$ MVar.

2.1 Rezultati proračuna

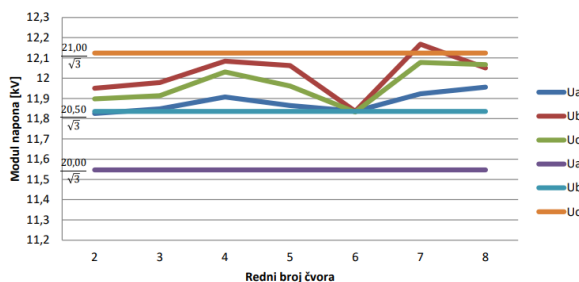
Cilj proračuna tokova snaga je da se izračunaju nepoznate veličine vektora stanja. Proračun se završava kada se zadovolje kriterijumi konvergencije, a u ovom slučaju je bilo neophodno šest iteracija. Vrednosti modula i uglova napona po fazama za svaki čvor u poslednjoj iteraciji su predstavljene u tabeli 5 i grafički na slici 3.

Tabela 5 – Vrednosti faznih napona po čvorovima

Redni broj čvora	U_a	θ_a	U_b	θ_b	U_c	θ_c
	[kV]	[°]	[kV]	[°]	[kV]	[°]
2	11,826	0,713	11,950	-118,900	11,898	121,073
3	11,848	1,388	11,978	-118,251	11,913	121,797
4	11,907	0,806	12,084	-118,602	12,031	121,226
5	11,865	1,331	12,062	-118,122	11,961	121,980
6	11,837	2,186	11,837	-117,853	11,833	122,160
7	11,923	0,749	12,168	-118,479	12,078	121,409
8	11,956	1,019	12,050	-118,550	12,067	121,017

Najveće odstupanje modula napona faze A u odnosu na nominalni napon mreže i napon balansnog čvora je u generatorskom čvoru 8 i on je viši za 3,54 %. Modul napona faze B u čvoru 7 je za 5,38 % viši u odnosu na nominalni napon, a za 2,80 % viši u odnosu na napon balansnog čvora.

U čvoru 6 je najniži modul napona faze C u odnosu na napon balansnog čvora i on za 2,40 % ima manju vrednost. Vidi se da su vrednosti u sve tri faze dosta bliske zadatom naponu u ovom čvoru. U čvoru 7 napon za 4,60 % viši u odnosu na nominalni napon. Može se zaključiti da su moduli napona u opsegu oko 5,50 % u odnosu na nominalni napon mreže, a oko 3,00 % u odnosu na napone balansnog čvora.



Slika 3 – Vrednosti modula napona po čvorovima

Uglovi napona faza A i B se razlikuju samo za par stepeni od uglova napona balansnog čvora, dok je u fazi C dosta veća razlika. Primeti se da su te vrednosti dosta slične, uz ne veća odstupanja od po $1,5^\circ$ u svakoj fazi.

Dobijene vrednosti struja po fazama u poslednjoj iteraciji postupka se mogu očitati u tabeli 6.

Tabela 6 – Vrednosti faznih struja po čvorovima

Redni broj čvora	I_a [A]	θ_a [°]	I_b [A]	θ_b [°]	I_c [A]	θ_c [°]
2	61,948	-25,305	122,714	-132,494	109,855	94,239
3	61,835	-24,595	122,539	-131,793	109,615	94,891
4	60,991	-25,059	121,601	-132,441	108,938	94,412
5	61,215	-24,316	122,233	-131,805	109,222	94,861
6	242,993	69,777	215,388	-21,012	300,642	-151,179
7	60,379	-24,785	121,308	-132,452	108,557	94,380
8	30,994	-25,402	30,994	-145,402	30,994	94,597

Vrednosti modula struja u potrošačkim čvorovima imaju približne vrednosti po svakoj fazi, koje se razlikuju najviše do $1,5 A$. U generatorskom čvoru 6, koji je upravljačke strategije $P_\Sigma V$, moduli struja su redom po fazama oko 4,2 i 3 puta veće od modula struja potrošačkih čvorova. Čvor 8 je generatorski, tipa $P_\Sigma Q_\Sigma I$, što znači da su struje inverznog i nultog redosleda jednake $0 A$. U faznom domenu moduli struja su nekoliko puta manji u odnosu na struje u potrošačkim čvorovima. Moduli struja u sve tri faze imaju identične vrednosti, odnosno po $30,994 A$.

Uglovi struja u potrošačkim područjima takođe imaju približne vrednosti po fazama. Približne vrednosti su i u čvoru 8, ali nešto niže u fazama A i B u odnosu na potrošačke čvorove. U čvoru 6 su uglovi pomereni za oko 95° u fazi A u odnosu na ostale čvorove, u fazi B su pomereni za oko 118° , dok je u fazi C najveća promena i iznosi oko 245° .

8. ZAKLJUČAK

Poslednjih decenija, usled klimatskih promena, uvođenjem distribuiranih energetske resursa, došlo je do velikih promena u elektroenergetskim sistemima. Sada mreže bivaju nesimetrične, a tokovi snaga mogu biti i dvosmerni. Distribuirani energetski resursi imaju upravljačke strategije koje se dosta razlikuju od tradicionalnih mašina, jer imaju mogućnost da kontrolišu inverzne komponente struja u nesimetričnim režimima, ali i da održavaju simetrične napone trofaznog čvora u nesimetričnom režimu. Zbog toga su uvedeni i novi tipovi čvorova.

Cilj rada bio je da se postupkom za proračun nesimetričnih tokova snaga sa distributivnim generatorima odrede nepoznate varijable stanja, jer se dalje na osnovu njih

mogu izračunati sve ostale potrebne veličine. Data je mreža koja se sastoji od osam čvorova, od toga je prvi čvor balansni, dva su generatorska i ostali čvorovi su potrošački. Postupak za proračun tokova snaga je razvijen u programskom jeziku Fortran. Izračunati su naponi i struje u faznom domenu i predstavljeni preko modula i uglova. Uglovi napona su približnih vrednosti kao i uglovi balansnog čvora u fazama A i B, uz razlike u fazi C, dok se moduli razlikuju i oni su uglavnom u opsegu između $\frac{20,50}{\sqrt{3}} kV$ i $\frac{21,00}{\sqrt{3}} kV$. U čvoru tipa $P_\Sigma V$ sve tri faze imaju približne vrednosti. Moduli struja u potrošačkim čvorovima imaju slične vrednosti. Isti slučaj važi i za uglove struja. U generatorskim čvoru $P_\Sigma V$ struje rastu i do nekoliko puta u odnosu na potrošačke, a uglovi se pomeraju drastično u svakoj fazi. U generatorskom čvoru $P_\Sigma Q_\Sigma I$ moduli padaju, a uglovi su slični kao u potrošačkim čvorovima. Razlog za to leži u činjenici da je moguće kontrolisati inverznu komponentu struje i u nesimetričnim režimima.

9. LITERATURA

- [1] Nikola Vojnović, „Proračun tokova snaga neuravnoteženih mreža sa energetske resursima priključenim na mrežu preko uređaja energetske elektronike“, Novi Sad 2018.
- [2] Luka Strezoski, „Modelovanje i fundamentalni proračuni aktivnih distributivnih mreža“, ISBN 978-86-6022-352-6, Izdavač: Novi Sad: Fakultet tehničkih nauka, 2021 (Novi Sad : FTN, Grafički centar GRID); (Polje rezultata: Tehničko-tehnološke nauke)
- [3] Vladimir Strezoski, „Osnovni proračuni elektroenergetskih sistema. Tom 1, Elementi“, Novi Sad 2017.

Kratka biografija:



Tijana Terzić rođena je u Loznici 1997. godine. Osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu završila je 2020. godine.

RAZVOJ SISTEMA ZA VIDEO KONFERENCIJE**DEVELOPMENT OF VIDEO CONFERENCING SYSTEMS**

Ana Rudić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – SOFTVERSKO INŽENJERSTVO I INFORMACIONE TEHNOLOGIJE

Kratak sadržaj – Tema rada jeste izrada sistema za video konferencije. U tu svrhu implementirana je Android aplikacija nalik aplikacijama slične namjene. Za potrebe razmjene audio-video stream-ova učesnika video poziva implementiran je server kao čvor na koji se klijenti kače i preko kojeg se razmjena tih stream-ova i obavlja. Server je razvijen u Java programskom jeziku, verzije 1.8, korišćenjem Spring razvojnog okruženja. Za skladištenje podataka korištene su relacione baze podataka i to MySQL za potrebe skladištenja podataka neophodnih serveru i Firebase realtime baza podataka za potrebe čuvanja podataka neophodnih za funkcionisanje same Android aplikacije. Video-conference Android aplikacija nudi niz funkcionalnosti poput ostvarivanja video poziva, dodavanja novih prijatelja, pruža uvid u pristigle notifikacije i nudi mogućnost ažuriranja ličnih podataka. Pored ovih, tu su i niz funkcionalnosti koji olakšavaju korišćenje same aplikacije, kao što su mijenjanje teme, podešavanje dnevnog i noćnog režima rada aplikacije i podešavanje željenog jezika.

Ključne reči: Vonage Video API, Session, Subscriber, Publisher, Java, Spring Boot

Abstract – This paper deals with the development of a video conference system. For demonstration purposes, a mobile application is developed, as well as a server through which the exchange of audio-video streams of the video call participants takes place. The mobile application is an Android application, while the server is implemented in the Java programming language, using the Spring development environment. The application enables a number of functionalities, such as making video calls, adding new friends, reviewing incoming notifications, and updating personal data, but also a number of functionalities that will make it easier for users to use the application itself.

Keywords: Vonage Video API, Session, Subscriber, Publisher, Java, Spring Boot

1. UVOD

Zadatak rada jeste kreiranje video-conference mobilne aplikacije, koja će omogućiti uspostavljanje komunikacije među ljudima širom svijeta, uprkos činjenici da su fizički razdvojeni i hiljadama kilometara. Poseban značaj ovakve

aplikacije stekle su nažalost u situacijama kakva je danas zadesila cio svijet. Svjedoci smo da su ljudi iako locirani na istim lokacijama, onemogućeni da se viđaju i svoje svakodnevne poslovne obaveze obavljaju unutar svojih korporacija. Međutim zahvaljujući postojanju ovakvih aplikacija, poslovanje je uspješno nastavljeno bilo da je vanredna situacija spriječila direktan kontakt ili je to velika kilometarska razdaljina.

Video-conference jeste Android aplikacija, ali je za realizaciju njene ključne funkcionalnosti, tačnije za realizaciju video poziva bilo neophodno razviti i serversku aplikaciju. Serverska aplikacija je Spring Boot [2] aplikacija, čiji je razvoj ostvaren uz pomoć serverske Video Vonage SDK [3] biblioteke. Svi podaci sa kojima server manipuliše smješteni su unutar MySQL relacione baze podataka. Klijentska iliti Android aplikacija razvijena je uz pomoć klijentske Video Vonage SDK [3] biblioteke, dok su svi podaci neophodni za njen rad smješteni u realtime Firebase [1] bazu podataka.

2. SPECIFIKACIJA

Zadatak rada obuhvata demonstraciju razmjene audio-video stream-ova između dva učesnika video poziva. U tu svrhu implementirana je mobilna aplikacija po uzoru na aplikacije slične namjene. Shodno tome video-conference aplikacija pruža mogućnost dodavanja novih prijatelja, kao i mogućnost prihvatanja ili odbijanja zahtjeva za prijateljstvo od strane drugih korisnika. Naravno, aplikacija omogućava i ostvarivanje video poziva, sto je ujedno i centralna funkcionalnost ove aplikacije.

Prilikom pokretanja aplikacije korisniku je omogućena prijava na sistem ukoliko ima otvoren nalog na istoj. Ako to nije slučaj, neophodno je da prvo izvrši registraciju.

Nakon uspješne prijave, korisniku se prikazuje glavni prozor aplikacije u okviru kojeg može da izvrši nekoliko funkcionalnosti: doda prijatelje, obavi poziv sa željenom osobom ili ima uvid u notifikacije, tačnije uvid u pristigle zahtjeve za prijateljstvo.

U cilju brže pretrage, u samom vrhu ekrana nalazi se polje za pretragu gdje je omogućena pretraga korisnika po nekoliko parametara, po imenu, prezimenu i broju telefona.

U cilju stvaranja što prijatnijeg orkuženja za korisnike, aplikacija nudi mogućnost promjene jezika, teme, kao i dnevnog ili noćnog režima rada.

Aplikacija je trenutno prevedena na tri jezika: srpski, engleski i njemački. Što se tiče odabira teme, trenutno je dostupna paleta sa 15 različitih boja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Vidaković, red. prof.

Aplikacija naravno nudi i mogućnost ažuriranja podataka sa korisničkog profila. Korisnik može da mijenja lične podatke, kao što su ime, prezime, slika, broj telefona, korisničko ime, ali i lozinku koju koristi za prijavu na sistem.

Ukoliko se prilikom korišćenja ove aplikacije, korisnicima javi bilo kakva nedoumica u smislu koraka koje je potrebno da preduzmu da bi se određena funkcionalnost ostvarila, dovoljno je samo da zatraže pomoć. Pomoć jeste zapravo vodič kroz cijelu aplikaciju sa detaljnim pojašnjenjima, koracima, kao i redoslijedom kojim se dati koraci moraju izvršiti kako bi korisnika doveli do željenog cilja.

3. IMPLEMENTACIJA

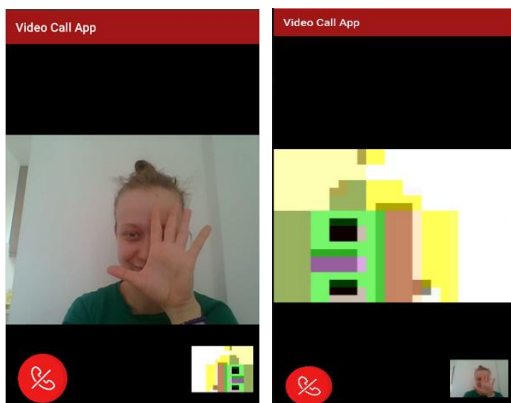
Aplikaciju čine dvije cjeline, serverski i klijentski dio. Serverski dio aplikacije je implementiran u programskom jeziku Java, verzije 1.8, korišćenjem Spring razvojnog okruženja. Klijentski dio jeste Android aplikacija. Za realizaciju video poziva korištena je Video Vonage API platforma i to klijentska Video Vonage SDK na klijentskoj strani i serverska Video Vonage SDK biblioteka na serverskoj strani.

3.1. Klijentski dio

Klijentski dio aplikacije predstavlja view komponentu u arhitekturi ove aplikacije. Kao što je već spomenuto, za razvoj klijentskog dijela aplikacije korišćena je klijentska Video Vonage SDK biblioteka za objavljivanje korisničkih audio-video stream-ova, kao i za pretplaćivanje na audio-video stream-ove drugih učesnika unutar video poziva. Klijentski dio aplikacije pored metoda za iscrtavanje komponenti korisničkog interfejsa, sadrži i metode za komunikaciju sa serverskim dijelom aplikacije, ali i metode za komunikaciju sa Firebase bazom podataka u cilju dobavljanja podataka, dodavanja novih i manipulacije nad postojećim, ali i za korisničku autentifikaciju.

3.1.1. Metode za prikaz audio-video poziva

Pored gore pomenutih metoda, kao centralne metode ove aplikacije izdvojile su se upravo metode za prikaz audio-video poziva. Rezultat izvršavanja ovih metoda jeste prikaz vidljiv na slici 1.



Slika 1. Ekranu zaduženi za prikaz video poziva

Za realizaciju gore prikazanog video poziva prije svega neophodno je zatražiti dozvole za korišćenje kamere i mikrofona sa korisničkog uređaja.

Kada su dozvole odobrene potrebno je povezati se sa OpenTok sesijom, koja zapravo i identifikuje svaki video poziv, nakon čega je korisnik u mogućnosti da objavi svoj audio-video stream ili da se pretplati na stream drugog učenika tog poziva, što je i vidljivo na slici 2.

```
@AfterPermissionGranted(RC_VIDEO_APP_PERM)
private void requestPermissions(){
    String perms[] = {Manifest.permission.INTERNET, Manifest.permission.CAMERA,
        Manifest.permission.RECORD_AUDIO};

    if(EasyPermissions.hasPermissions(this, perms)){
        mPublisherViewController = findViewById(R.id.publisher_container);
        mSubscriberViewController = findViewById(R.id.subscriber_container);

        fetchSessionConnectionData();
    }else{
        EasyPermissions.requestPermissions(this, "Hey this app needs Camera, Please
        allow.", RC_VIDEO_APP_PERM, perms);
    }
}
```

Slika 2. Kod koji se izvršava kada su dozvole za korištenje kamere i mikrofona odobrene/nisu odobrene

Povezivanje sa sesijom ostvaruje se preko AsyncTask-a [4] koji omogućava da se u pozadinskoj niti dobavi sesija sa serverske strane, kao i tokeni koji jedinstveno identifikuju korisnike. AsyncTask jeste mehanizam koji glavnu nit oslobađa izvršavanja dugih operacija, tako što se to izvršavanje delegira pozadinskoj niti.

U `doInBackground()` metodi `SessionAsyncTask` klase ostvarena je konekcija ka serveru, odnosno ka <http://10.0.2.2:8080/sessions> endpoint-u koji vraća session ID, token ID i API_KEY, ključne parametre na osnovu kojih se instancira Session klasa definisana unutar Video Vonage API biblioteke, a zatim se ostvaruje i korisnikova konekcija sa sesijom (slika 3).

```
public class SessionAsyncTask extends AsyncTask<Void, Void, Void> {

    private Context context;
    private Session session;

    public SessionAsyncTask(Context videoChatActivity){
        this.context = videoChatActivity;
    }

    @Override
    protected Void doInBackground(Void... voids) {
        RequestQueue reqQueue = Volley.newRequestQueue(context);
        reqQueue.add(new JsonObjectRequest(Request.Method.GET,
            //10.0.2.2
            "http://10.0.2.8080/sessions",
            null, new Response.Listener<JSONObject>() {

                @Override
                public void onResponse(JSONObject response) {
                    try {
                        String API_KEY = response.getString("apiKey");
                        String SESSION_ID = response.getString("sessionId");
                        String TOKEN = response.getString("token");

                        Log.i("API_KEY: ", API_KEY);
                        Log.i("SESSION_ID: ", SESSION_ID);
                        Log.i("TOKEN: ", TOKEN);

                        session = new Session.Builder(context, API_KEY, SESSION_ID).build();
                        session.setSessionListener((Session.SessionListener) context);
                        session.connect(TOKEN);
                        processValue(session);
                        Log.d("SESSION_IN_ASYNC", session.toString());

                    } catch (JSONException error) {
                        Log.e("Web Service error: ", error.getMessage());
                    }
                }, new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Log.e("Web Service error: ", error.getMessage());
                    }
                }));
                return null;
            }
        }
    }
}
```

Slika 3. Ostvarivanje konekcije sa sesijom posredstvom AsyncTask-a

Dio koda zadužen za objavljivanje audio-video stream-a implementiran je u okviru `onConnected` metode koja se poziva nakon što je uspostavljena konekcija sa sesijom.

OnConnected je metoda SessionListener interfejsa koji je definisan unutar klijentske Video Vonage API biblioteke, a za čiju je implementaciju zadužena VideoChatActivity aktivnost (slika 4). Publisher je klasa koja je takođe definisana unutar prethodno pomenute biblioteke.

```
@Override
public void onConnected(Session session) {
    Log.i(LOG_TAG, "Session Connected");

    mPublisher = new Publisher.Builder(this).build();
    mPublisher.setPublisherListener(this);
    mPublisher.setCatcher(new CameraVideoCatcher(VideoChatActivity.this,
        Publisher.CameraCaptureResolution.MEDIUM, Publisher.CameraCaptureFrameRate.FPS_30));
    mPublisherViewController.addView(mPublisher.getView());

    if(mPublisher.getView() instanceof GLSurfaceView){
        ((GLSurfaceView) mPublisher.getView()).setZOrderOnTop(true);
    }

    mSession.publish(mPublisher);
}
```

Slika 4. Funkcija zadužena za objavljivanje korisničkih audio-video stream-ova unutar sesije

Unutar onStreamReceived metode, metode SessionListener interfejsa, implementiran je kod koji omogućava korisniku da se pretplati na stream drugog učesnika video poziva (slika 5). Subscriber jeste klasa klijentske Video Vonage API biblioteke.

```
@Override
public void onStreamReceived(Session session, Stream stream) {
    Log.i(LOG_TAG, "Stream Received");
    if(mSubscriber == null){
        mSubscriber = new Subscriber.Builder(this, stream).build();
        mSession.subscribe(mSubscriber);
        mSubscriberViewController.addView(mSubscriber.getView());
    }
}
```

Slika 5. Funkcija zadužena za pretplaćivanje na audio-stream-ove drugih korisnika unutar sesije

OnStreamDropped metoda jeste metoda SessionListener interfejsa koja se poziva kada učesnik poziva prekine objavljivanje svog audio-video stream-a (slika 5).

```
@Override
public void onStreamDropped(Session session, Stream stream) {
    Log.i(LOG_TAG, "Stream Dropped");

    if(mSubscriber != null){
        mSubscriber = null;
        mSubscriberViewController.removeAllViews();
    }
}
```

Slika 6. Funkcija koja se izvršava kada korisnik prestrane objavljivanje svog audio-video stream-a

3.2. Serverski dio

Kao što je već rečeno serverski dio je razvijen za potrebe generisanja sesija koje identifikuju video pozive, kao i za potrebe generisanja tokena koji identifikuju učesnike tih poziva. Kreiranje se obavlja uz pomoć serverske Video Vonage API biblioteke. Na serverskoj strani se nalazi sloj kontrolera, servisa i repozitorijuma. Za manipulaciju nad podacima korišćena je MySQL relacionalna baza podataka.

3.2.1. Controller

Budući da sloj controller-a sadrži isključivo metode koje se odnose na kreiranje sesija i tokena, sve one grupisane su u jednu klasu SessionController. Servisni sloj sastoji se od dvije klase SessionService i TokenService koje sadrže metode za procesiranje podataka potrebnih kontroleru. Sloj repozitorijuma se sastoji od dva interfejsa SessionRepository i TokenRepository koje redom implementiraju servisi SessionService i TokenService. Na

slici 7 prikazan je endpoint kontrolera zadužen za generisanje sesije i tokena.

```
@RequestMapping(
    method = RequestMethod.GET,
    produces = MediaType.APPLICATION_JSON_VALUE
)
public ResponseEntity<JSONObject> createSession() throws OpenTokException,
    InterruptedException{
    numOfTokens += 1;
    JSONObject response = null;
    if(saveSessionId()){
        ArrayList<SessionDTO> sessions = getSortedListOfSessions();
        SessionDTO sessionDTO = sessions.get(sessions.size() - 1);
        String value = "";
        System.out.println("num of tokens: " + numOfTokens);
        for(int i = 0; i<numOfTokens; i++){
            CreateTokenThread createToken = new CreateTokenThread(sessionDTO);
            Thread thread = new Thread(createToken);
            thread.start();
            thread.join();
            value = createToken.getValue();
            TokenDTO tokenDTO = new TokenDTO();
            tokenDTO.setTokenId(value);
            sessionDTO.getTokens().add(tokenDTO);
            System.out.println(sessionDTO.getTokens());
            sessionService.save(sessionDTO);
        }

        response = new JSONObject();
        response.put("apiKey", API_KEY);
        response.put("sessionId", sessionDTO.getSessionId());
        response.put("token", value);
    }

    return new ResponseEntity<JSONObject>(response, HttpStatus.OK);
}
```

Slika 7. Metoda SessionController-a zadužena za generisanje sesija i tokena

Kao što je već rečeno, generisanje sesija i tokena se obavlja uz pomoć serverske Video Vonage API biblioteke, tačnije pomoću instance klase OpenTok. Pozivi metoda nad ovom instancom koji se koriste u tu svrhu su asinhroni. Međutim, kako je za kreiranje korisničkih tokena neophodno postojanje sesije, bilo je potrebno sinhronizovati te pozive. To je učinjeno dodavanjem ključne riječi *synhronized* [5] ispred naziva metode saveSessionId(), kao što je i vidljivo na slici ispod (slika 8).

```
private synchronized boolean saveSessionId() throws OpenTokException{
    if(!isSessionCreated()){
        System.out.println("Kreiranje sesije");
        SessionProperties sessionProperties = new
        SessionProperties.Builder().mediaMode(MediaMode.ROUTED).build();
        Session session = sdk.createSession(sessionProperties);
        String sessionId = session.getSessionId();
        SessionDTO sessionDTO = new SessionDTO();
        sessionDTO.setSessionId(sessionId);

        DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
        Date date = new Date();

        sessionDTO.setDateOfCreation(date);
        Date expiryDate = DateUtils.addHours(sessionDTO.getDateOfCreation(), 24);
        sessionDTO.setExpire(false);
        sessionService.save(sessionDTO);
        System.out.println("Sesija sacuvana");
        sessionCreated = true;
    }
    return sessionCreated;
}
```

Slika 8. Metoda za generisanje sesije

Neposredno prije kreiranja sesije potrebno je provjeriti da li u bazi postoji aktivna, tačnije sesija čiji datum važenja nije istekao. Ta provjera je izvršena tako što se svi podaci iz baze dobave, zatim se sortiraju od najranije kreiranih pa do onih najkasnije. Ukoliko je datum isteka najkasnije kreirane sesije neposredno prije trenutnog datuma, ona je nevažeća i neophodno je kreirati novu. To je potkrepljeno sljedećim parčetom koda (slika 9).


```
private Boolean isSessionCreated(){
    ArrayList<SessionDTO> sessions = getSortedListOfSessions();
    System.out.println("Size: " + sessions.size());
    if(!sessions.isEmpty()){
        SessionDTO sessionDTO = sessions.get(sessions.size() - 1);
        if(sessionDTO.getExpire()){
            System.out.println("Isticuci datum prije trenutnog");
            return false;
        }else{
            System.out.println("Isticuci datum posle trenutnog");
            return true;
        }
    }else{
        return false;
    }
}

private ArrayList<SessionDTO> getSortedListOfSessions(){
    ArrayList<SessionDTO> sessions = (ArrayList<SessionDTO>)
    sessionService.findAll();
    Collections.sort(sessions, new Comparator<SessionDTO>(){
        public int compare (SessionDTO s1, SessionDTO s2){
            return s1.getDateOfCreation().compareTo(s2.getDateOfCreation());
        }
    });
    return sessions;
}
```

Slika 9. Metode kojima se utvrđuje validnost sesije

Sinhronizovanjem prethodno pomenutih metoda obezbijedili smo da se prvo izvrši poziv metode zadužene za kreiranje sesije. Međutim time smo onemogućili da se pozivi metoda za kreiranje korisničkih tokena izvršavaju paralelno. To je riješeno uvođenjem niti [6] koje asinhrono izvršavaju te pozive. Klasa koja reprezentuje niti prikazana je na listingu ispod (slika 10).

```
public class CreateTokenThread implements Runnable{
    private volatile String tokenId;
    private int API_KEY = 47474691;
    private String API_SECRET = "0f3fc8e0dfd732d8d8e84b07de071c98a741ae7";
    private OpenTok sdk = new OpenTok(API_KEY, API_SECRET);
    private SessionDTO session;

    public CreateTokenThread(SessionDTO session) {
        this.session = session;
    }

    @Override
    public void run() {
        System.out.println("SessionDTO: " + session.getSessionId());

        try {
            tokenId = sdk.generateToken(session.getSessionId());
            System.out.println("TOKEN: " + tokenId);
        } catch (OpenTokException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public String getValue() {
        return tokenId;
    }
}
```

Slika 10. Nit zadužena za generisanje tokena

Rest endpoint će se pozivati onoliko puta koliko je učesnika prisutno unutar određenog video poziva, a to automatski znači da će toliko biti kreiranih i pokrenutih niti koje će kao rezultat svog izvršavanja vratiti vrijednost jedinstvenih korisničkih tokena (tokenId). Ova aplikacija podržava maksimalno dva učesnika, tako da će po svakom pozivu za generisanje sesije (identifikatora svakog video poziva) zapravo biti generisana dva korisnička tokena (slika 11).

```
for(int i = 0; i<numOfTokens; i++){
    CreateTokenThread createToken = new CreateTokenThread(sessionDTO);
    Thread thread = new Thread(createToken);
    thread.start();
    thread.join();
    value = createToken.getValue();
    TokenDTO tokenDTO = new TokenDTO();
    tokenDTO.setTokenId(value);
    sessionDTO.getTokens().add(tokenDTO);
    System.out.println(sessionDTO.getTokens());
    sessionService.save(sessionDTO);
}
```

Slika 11. Broj generisanih tokena

Kada korisnik aplikacije odluči da završi poziv, serveru se upućuje zahtjev da izvrši sledeće parče koda (slika 12).

```
@RequestMapping(
    value =("/{sessionId}",
    method = RequestMethod.GET,
    produces = MediaType.APPLICATION_JSON_VALUE
)
public ResponseEntity<JSONObject> setSessionExpire(@PathVariable String sessionId){
    System.out.println("Set session expire");
    JSONObject response = new JSONObject();
    SessionDTO sessionDTO = sessionService.findById(sessionId).get();
    System.out.println("Session: " + sessionDTO);
    if(sessionDTO != null){
        sessionDTO.setExpire(true);
        sessionService.save(sessionDTO);
        response.put("sessionId", sessionId);
        response.put("updated", "successfully");
        response.put("expire", String.valueOf(sessionDTO.getExpire()));
    }
    return new ResponseEntity<JSONObject>(response, HttpStatus.OK);
}
```

Slika 12. Metoda SessionController-a koja se izvršava po okončanju video poziva

3. ZAKLJUČAK

Cilj ovog master rada bio je demonstracija razmjene audio-video tokova između učesnika video poziva. Osnovna ideja je bila da se implementira jednostavan server koji će predstavljati čvor na koji bi se klijenti kačili i preko kojeg bi ta razmjena tekla. Jedan učesnik koristeći kameru i zvuk mobilnog telefona šalje svoj tok ka serveru, a ujedno preko komunikacije sa serverom se i pretplaćuje na audio-video tok drugog učesnika, što se sve prikazuje na ekranu mobilnog telefona u vidu video poziva.

Aplikacija je razvijena u trenutno aktuelnim tehnologijama i ima dobru osnovu za dalji razvoj. Ključno je napomenuti da je trenutno razvijena video-conference aplikacija koja podržava video poziv između dva učesnika. S obzirom na tu činjenicu dalji pravci razvoja i unapređivanja tekli bi u smjeru proširenja broja učesnika, tako da on ne bude ograničen.

4. LITERATURA

- [1] <https://firebase.google.com/docs/android/setup>
- [2] <https://spring.io/projects/spring-boot>
- [3] <https://tokbox.com/developer/guides/basics/>
- [4] <https://developer.android.com/reference/android/os/AsyncTask>
- [5] <https://www.geeksforgeeks.org/synchronization-in-java/>
- [6] <https://www.javatpoint.com/how-to-create-a-thread-in-java>

Kratka biografija:



Ana Rudić rođena je u Bijeljini 1995. 2014 god. je upisala Fakultet tehničkih nauka u Novom Sadu, smjer softversko inženjerstvo i informacione tehnologije. Nakon završenih osnovnih studija upisala je master studije na istom fakultetu. Položila je sve ispite predviđene planom i programom. kontakt:ana.rudic95@gmail.com

ИМПЛЕМЕНТАЦИЈА TASK QUEUE СЕРВИСА У ОКВИРУ ПЛАТФОРМЕ ЗА ИВИЧНО РАЧУНАРСТВО**IMPLEMENTATION OF TASK QUEUE SERVICE WITHIN EDGE COMPUTING PLATFORM**

Вељко Максимовић, Факултет техничких наука, Нови Сад

Област – ПРИМЕЊЕНЕ РАЧУНАРСKE НАУКЕ И ИНФОРМАТИКА

Кратак садржај – У раду се описује улога и имплементација Task Queue сервиса у оквиру микросервисне архитектуре. Сервис је имплементиран у оквиру пројекта „Constellations”, који представља управљачку јединицу за кластере рачунара који омогућују ”рачунарство на ивици”, то јест везу између корисника и data center-a.

Кључне речи: Task Queue, дистрибуирани системи, микросервисне архитектуре

Abstract – This paper describes the purpose and implementation of Task Queue within the microservice architecture. Service is implemented as a part of the constellations project, which is software for managing micro-clusters on the edge of the network, between the end users and data centers.

Keywords: Task Queue, distributed systems, microservice architectures.

1. УВОД

Предмет истраживања рада јесте успостављање ”редова задатака” (енгл. *task queue*) чија намена је да учествује у расподели задатака на доступне сервисне инстанце унутар информационог система микросервисне архитектуре. Task Queue сервис је имплементиран у оквиру пројекта *Constellations*, чији циљ је лакша имплементација рачунарства на ивици (енгл. *Edge computing*).

1.1. Рачунарство у облаку

Почетком 1960. године појавила се нова парадигма за развој информационих система – рачунарство у облаку (енгл. *cloud computing*). Идеја која је стајала иза ове парадигме подразумевала је дистрибуцију рачунарских ресурса, као што су процесори, RAM меморије и хард дискови, путем интернет мреже. Касније, дистрибуција самих ресурса прелази у дистрибуцију комплетних платформи, попут оперативних система, база података, или сервиса [1].

НАПОМЕНА:

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

Данас постоји велики број компанија које нуде услуге рачунарства у облаку, од којих се као најпопуларније издвајају: Amazon Web Services, Microsoft Azure и Google Cloud Platform.

Међутим, овај приступ има и неколико мана: Крајњи корисници информационих система су често физички удаљени од рачунарских центара на којима су ти информациони системи покренути. То значи да се перформансе ових услуга деградирају због времена које је неопходно за комуникацију путем интернет мреже између корисничких рачунара и рачунара на којима се налази информациони систем [2].

У одређеним ситуацијама, у зависности од области којом се информациони систем бави, слање корисничких података изван граница државе у којој се корисник налази законом није дозвољено [3]. У оваквим ситуацијама није уопште могуће користити рачунарство у облаку, јер оно подразумева комуникацију са рачунарским центрима који се често налазе изван држава у којој су корисници тих софтверских производа.

1.2. Рачунарство на ивици

Рачунарство на ивици (енгл. *edge computing*) подразумева проширење инфраструктуре модерних информационих система чији је циљ решавање горепомнутих недостатака тренутног рачунарства у облаку: свести комуникацију између корисника и рачунарских центара на минимум. Понуђачи интернет услуга (енгл. *Internet service providers*) ове сервере на ивици својих мрежа користе за привремено чување - кеширање (енгл. *caching*) информација, како не би приликом сваког захтева морали да комуницирају са серверима изван њихове мреже, и тиме смањују кашњења и своје трошкове комуникације [4].

Сличан приступ може да се примени и на рачунарство у облаку, где би се „на ивици” мрежа, на којима се налазе крајњи корисници, налазили кластери рачунара који би радили једноставну обраду података и извршавали једноставније функционалности, а комуникација са серверима унутар рачунарских центара би се радила само када је то неопходно. Ови кластери рачунарских ресурса, који се налазе у физичкој околини крајњих корисника, називају се микро-облаци (енгл. *micro-clouds*) [5].

Коришћење рачунарских ресурса унутар микро-облака решава и проблем обраде сензитивних

података, јер би се сервери који обрађују податке налазили у истим државама где и корисници.

1.3. Рачунарство на ивици као услуга

Имплементацијом управљачког система за креирање и управљање микро групама чворова – кластерима (енгл. *cluster*) према потреби корисника добијају се све предности „аутоматског“ одржавања инфраструктуре које нуде понуђачи услуга рачунарства у облаку, само са мање физичких ресурса и на више локација [6].

2. МИКРОСЕРВИСНЕ АРХИТЕКТУРЕ

Раздвајањем логичких целина (модула) софтверског производа на засебне програме који се могу покретати независно једни од других, почетком 2000. година појављује се сервисно оријентисана архитектура софтвера (енгл. *Service Oriented Architecture - SOA*) [7]. Током растављања ових сервиса на све једноставније компоненте ради лакшег развоја и скалирања, настаје микросервисна архитектура, популаризована од стране компаније Netflix која 2012. године отпочиње миграцију својих система.

Неке од предности које овај приступ развоју софтвера нуди су отпорност на грешке у систему, једноставно скалирање, слобода при избору технологије за развој сваке од компонената и развој у мањим међусобно независним тимовима.

2.1. Комуникација између сервиса

Невезано од приступа развоју софтвера који користимо, неизбежна је подела истог на подкомпоненте. Негде ће те компоненте бити имплементирани у виду модула, негде у виду пакета, а негде у виду засебних сервиса. Критеријуми на основу којих одређујемо како да поделимо систем на мање целине се такође разликују, а један од њих јесте комуникација између различитих делова софтвера [8]. Ако различити делови софтвера комуницирају често, то је знак да би вероватно требало да буду део исте компоненте. Иако се трудимо да имплементирамо компоненте тако да што мање зависе једне од других, међусобна комуникација често је неизбежна.

2.2. Проблеми мрежне комуникације

Имплементација комуникације преко мреже представља неколико нових изазова. За почетак, време потребно за размену информација је неколико редова величине веће у поређењу са комуникацијом унутар истог процеса [9]. Док је за читање и писање података у меморију потребно између 1ns и 100ns, комуникација преко мреже унутар истог рачунарског центра (енгл. *data center*) траје између 1μs и 10μs. Узета је претпоставка да ће сви сервиси унутар нашег система бити у истом рачунарском центру.

Поред времена потребног за комуникацију, поузданост мреже је такође значајан фактор. За размену података унутар истог процеса можемо да претпоставимо да ће се увек извршити, то јест шанса да се подаци неуспешно упишу у меморијску локацију је занемарива. Са друге стране, такве

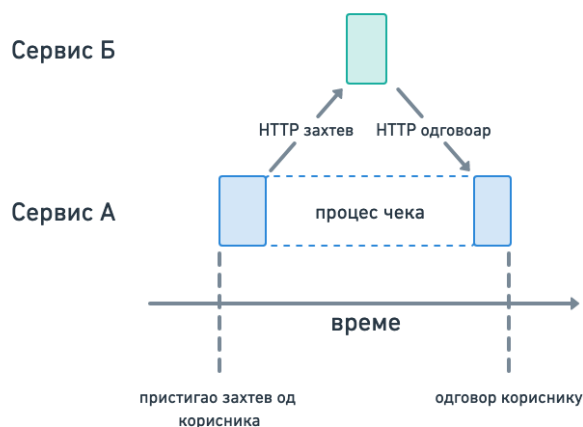
претпоставке не можемо имати ако је у питању комуникација преко мреже. Поред могућности да сама мрежна инфраструктура није исправна, постоји такође шанса да сервис са којим хоћемо да комуницирамо у том моменту није доступан.

3. КОМУНИКАЦИЈА У ДИСТРИБУИРАНОМ СИСТЕМУ

У овом поглављу се описују различити начини имплементације комуникације преко мреже.

3.1. Синхрона комуникација

Синхрона комуникација између софтверских компоненти подразумева да компонента која иницира комуникацију чека у неактивном стању (енгл. *idle state*) све док не добије одговор на свој захтев, и тек онда наставља са извршавањем функционалности. Уколико те компоненте комуницирају преко мреже може доћи до два проблема.

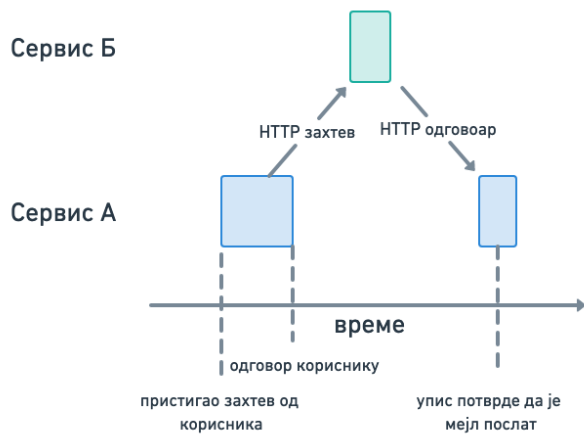


Слика 1: Илустрација комуникације преко мреже између два сервиса

Прво, време потребно за саму комуникацију (слање захтева и одговора) често представља већину времена неопходног за извршавање функционалности (слика 1), иако у том периоду ни један сервис не ради ништа, те је одговор система на захтеве корисника веома спор. Проблем постаје још израженији ако функционалност захтева комуникацију између више од 2 сервиса. Други проблем настаје ако сервис Б није доступан, те се функционалност унутар сервиса А неће извршити због неуспешне комуникације са сервисом Б.

3.2. Асинхрона комуникација

Шта ако одговор који враћа сервис Б није неопходан за извршавање функционалности унутар сервиса А? На пример: сервис А обавља регистрацију нових корисника. Слика 2 приказује ток извршавања функционалности, где сервис А не чека одговор сервиса Б, већ наставља даље са извршавањем инструкција. Овај вид комуникације назива се асинхрона комуникација.



Слика 2: Приказ асинхроне комуникације

Асинхроним слањем захтева решили смо један од проблема синхроне комуникације, спор одговор система због времена изгубљеног за пренос информација преко мреже. Међутим, и даље остаје питање како реаговати ако сервис Б није доступан, а ми смо већ одговорили кориснику да је успешно регистрован? Да ли памтити поруке које нису успешно послате и покушати поново? Колико пута треба покушати, и колико често? Шта ако у међувремену и сервис А неочекивано престане са радом, и изгубимо информације о порукама које нису успешно послате?

Решавање ових проблема није тривијално, нити има везе са доменом нашег система, те нема ни смисла имплементирати ове функционалности у оквиру сваког сервиса који иницира асинхрону комуникацију. Било би боље увести нову компоненту у систем, која ће бити посредник за слање асинхроних порука између сервиса, и имплементирати руковање са порукама у случају када неки од сервиса није доступан.

4. РЕД ЗА РАЗМЕНУ ПОРУКА

Ред за размену порука (енгл. *Message Queue*) је компонента која се користи за асинхрону комуникацију између процеса. Своје место је, између осталог, пронашла у дистрибуираним системима, под које спадају и микросервиси. Основни задатак ове компоненте је да комуникацију унутар система учини отпорнијом на неочекиване грешке. Ако сервис до ког треба да дође информација тренутно није доступан, ред за размену порука ће исту чувати и покушати са поновним слањем касније. У зависности од имплементације самог реда можемо конфигурисати параметре попут очекиване величине и структуре поруке, колико дуго порука може да буде задржана у систему пре него што се обрише, колико често треба покушавати слати поруке недоступним сервисима, итд.

Редови отварају нови модел комуникације, где се једна порука пропада кроз читав систем, и доставља на више различитих дестинација, које се динамички мењају током рада апликације. Овај модел комуникације се назива објављивач/претплатник

(енгл. *publish/subscribe* или *pub/sub*) и омогућује нам лакше скалирање микросервисног система [10].

5. ИМПЛЕМЕНТАЦИЈА

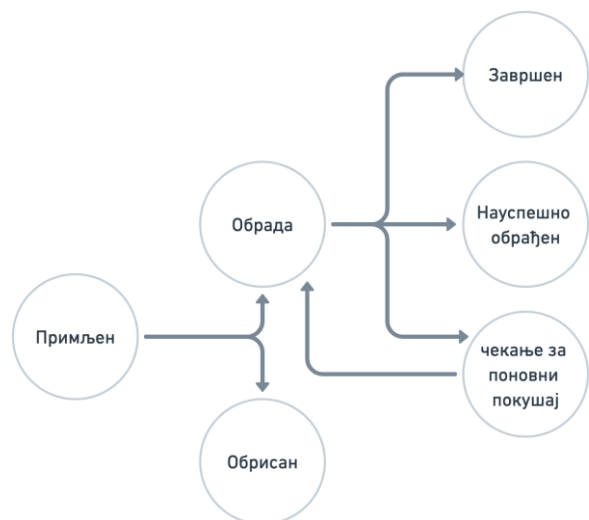
Следећа секција описује имплементацију реда за размену порука о задацима (енгл. *task queue*).

5.1. Дефиниција *task*-а

Задатак (енгл. *task*) је основна структура података на основу које ред зна са којим сервисом треба да комуницира, и како да се понаша ако комуникација не буде успешна. У овој имплементацији задатак садржи следеће податке: Јединствену идентификацију, назив, идентификацију реда у који задатак треба да буде уписан, дестинацију сервиса који треба да изврши задатак, у виду URL адресе и HTTP методе која треба да се користи приликом слања захтева, Додатне податке (ако су потребни), колико пута ред треба да покуша да пошаље задатак, колико дуго ред треба да чека на потврду од сервиса и колико времена треба да прође да би ред поново покушао да пошаље неуспешно извршен задатак. Ред за извршавање задатака се понаша као машина стања (енгл. *state machine*), у којој сваки задатак пролази кроз следећа стања (слика 3): примљен, обрада, завршен, чекање за поновни покушај, неуспешно обрађен и обрисан.

5.2. Дефиниција *Task Queue*-а

Ред за задатке (енгл. *task queue*) је компонента направљена над истоименом структуром података - редом. Сваки овакав ред има: идентификацију, назив, максималан број "жетона" – токена (енгл. *token*) и фреквенција обнављања токена (ови појмови су објашњени у поглављу 5.3). Начин чувања задатака је могуће мењати у зависности од потреба самог низа, јер је ова компонента дефинисана у виду интерфејса.



Слика 3: Стања кроз која пролази задатак током обраде унутар реда

За потребе овог рада подаци се чувају у меморији процеса, користећи динамичке низове које нуди језик Go (енгл. *slices*). Овим низовима управља компонента под називом „Менаџер за редове задатака“ (енгл.

TaskQueueManager) чија функционалност је упис пристиглих задатака у одговарајући низ, и касније прослеђивање задатака из низа до радника на даљу обраду.

5.3. Алгоритам корпе жетона - „*Token bucket*“

У системима који се користе за преношење информација, међу које спадају и редови, постоји потреба за контролом и ограничењем количине података који могу да прођу у одређеном временском периоду. За то користимо *token bucket* алгоритам који је настао за потребе телекомуникационих мрежа како би се ограничио проток информација. Сваки ред добија своју корпу са токенима, приликом чега је дефинисано која је максимална количина токена унутар корпе, и колико често пристижу нови токени. Са ова два параметра можемо индиректно да дефинишемо приоритете редова. Пре него што било који задатак из реда пређе у обраду, мора да добије токен. Ако у том тренутку нема слободних токена, задатак чека док се не појави нови, и тек онда прелази у обраду [11].

5.4. Обрада задатака

За крај је, након добијања токена, остала само још обрада задатка. С обзиром да ова операција може да потраје, треба имплементирати засебну компоненту која ће да обради задатак, како менаџер за редове не би био оптерећен тиме. Ова компонента се назива „радник“ (енгл. *worker*), треба да извуче информације које се налазе у задатку, на основу њих конструише HTTP захтев и исти пошаље сервису на извршавање. Ако не успе да пошаље захтев, или сервис не потврди успешно извршавање задатка у предефинисаном временском периоду, радник проверава да ли треба поново да покуша слање задатка, или не. Ако је одговор не, пребацује задатак у „неуспешно извршен“ стање, а ако је одговор да, интервал предвиђен за чекање пре поновног слања задатка проводи у неактивном стању.

Како би компонента која управља редом за обраду задатака и радници могли да раде паралелно, неопходно је да се извршавају у засебним рутинама. Уместо да креирамо и уништавамо нове рутине за сваки посао који пристиже, било би ефикасније да имамо покренут скуп рутина које ће нон-стоп радити (енгл. *worker pool*), и којима ће се слати задаци на обраду. У зависности од потреба система и ресурса које имамо на располагању, можемо мењати број радника који ће у сваком тренутку бити доступни.

6. ЗАКЉУЧАК

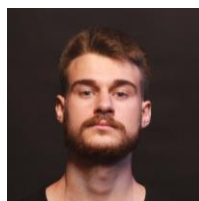
Кроз овај рад приказано је једно решење комуникације између сервиса унутар дистрибуираног система у оквиру *Constellations* платформе која нуди рачунарство на ивици као услугу (енгл. *as a service*). Правци даљег истраживања могу бити поређења перформанси *task queue* компоненте у зависности од броја радника који су доступни, и у зависности од слоја за чување самих задатака.

Колико ће писање задатака у фајл систем, или у базу података успорити комуникацију, и како ће утицати на отпорности система на грешке. Такође може да се уради мерење перформансе компоненте у зависности од броја сервиса који се налазе у систему, те количине порука које се размењују.

7. ЛИТЕРАТУРА

- [1] Sheth, Ashwini, et al. “Research paper on cloud computing.” *Contemporary research in India* (ISSN 2231-2137), no. April, 2021, 2021.
- [2] Rogier, Boris. “What is network latency? How it works and how to reduce it?” *Kadiska*, 10 November 2020, <https://kadiska.com/what-is-network-latency-how-it-works-and-how-to-reduce-it/>. Accessed 4 October 2022.
- [3] Cory, Nigel. “Cross-Border Data Flows: Where Are the Barriers, and What Do They Cost?” *Information Technology and Innovation Foundation (ITIF)*, 1 May 2017, <https://itif.org/publications/2017/05/01/cross-border-data-flows-where-are-barriers-and-what-do-they-cost/>. Accessed 4 October 2022.
- [4] Desertot, Mikael, et al. “Towards an autonomic approach for edge computing.” *Concurrency and Computation: Practice and experience*, vol. 19, 2007.
- [5] Симић, Милош, et al. “Towards Edge Computing as a Service: Dynamic Formation of the Micro Data-Centers.” *IEEE Access*, vol. 9, 2021.
- [6] Симић, Милош, et al. “Infrastructure as Software in Micro Clouds at the Edge.” *Sensors*, vol. 21, 2021.
- [7] Dragoni, Nicola, et al. “Microservices: Yesterday, Today, and Tomorrow.”
- [8] Wals, Donny. “Breaking an app up into modules – Donny Wals.” 15 December 2019, <https://www.donnywals.com/breaking-an-app-up-into-modules/>. Accessed 30 September 2022.
- [9] Popescu, Diana Andreea. “Latency-driven performance in data centres.” *Technical Report*, University of Cambridge, vol. 937, 2019.
- [10] Särelä, Mikko, et al. “RTFM: Publish/Subscribe Internetworking Architecture.” *ICT-MobileSummit 2008 Conference Proceedings Paul Cunningham and Miriam Cunningham*, 2008.
- [11] Tang, Puqi Perry, and Tsung-Yuan Charles Tai. “Network Traffic Characterization Using Token Bucket Model.” *Intel Architecture Labs*, 1997.

Кратка биографија:



Вељко Максимовић рођен је 1997. године у Суботици. Смер Рачунарство и аутоматика на Факултету техничких наука у Новом Саду уписује 2016. године. Основне студије завршава октобра 2020. године. Исте године, на истом факултету уписује мастер студије, од када је ангажован као сарадник у настави на Катедри за информатику.

AUTOMATIZOVANO TESTIRANJE KORISNOSTI VEB APLIKACIJA POMOĆU JEST OKVIRA**AUTOMATED USABILITY TESTING OF WEB APPLICATIONS USING THE JEST FRAMEWORK**

Aleksandar Savić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – U ovom radu je opisan Jest alat za automatizovano RESTful API testiranje veb aplikacije. Na studiji slučaja aplikacije za manipulaciju korisnika je ilustrovano korišćenje ovog alata za automatizovano testiranje realnih veb aplikacija. Objašnjeni i implementirani su API testovi za svaku funkcionalnost aplikacije.

Ključne reči: Jest, automatizovano testiranje, API, REST, Express

Abstract – This paper describes the Jest tool for automated RESTful API testing of a web application. A user manipulation application case study illustrates the use of this tool for automated testing of real web applications. API tests for each application functionality are explained and implemented.

Keywords: Jest, automated testing, API, REST, Express

1. UVOD

Testiranje softvera je važan deo razvoja svih aplikacija. Za veb aplikacije, međutim, to je neizbežno. To je zbog činjenice da je razvoj veb aplikacija skup i dugotrajan, ali pažljivo biranje poboljšanih metoda testiranja dovodi do jeftinijeg razvoja i održavanja [1]. Jedan od načina za smanjenje troškova razvoja i testiranja aplikacija je automatizovano testiranje. Sadašnji razvoj skripti za automatizovano testiranje veb aplikacija, međutim, pati od mnogih nedostataka. Stoga, često mala promena u aplikaciji dovodi do mnogih promena u testovima, a samim tim pruža više mogućnosti za unošenje grešaka u test. Razvoj i održavanje takvih testova je stoga skupo [2]. Zato je Jest predložen kao idealan alat za automatizaciju testova.

2. AUTOMATIZOVANO TESTIRANJE

Automatizovano testiranje predstavlja metodu kojom se test procedure automatizuju pomoću nekog alata za izvršavanje testova. Na ovaj način, test koji se jednom isplanira i za koji se kreira skripta može se uvek pustiti bez trošenja vremena na ponavljanje koraka testa. Za automatizovano testiranje postoji preduslov da tester poseduje znanje nekog programskog jezika.

NAPOMENA:

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

2.1. Motivacija

Tradicionalno, proces testiranja softvera je uglavnom bio ručni. Promene koda koje je napravio jedan programer je pregledao drugi, a sam softverski proizvod je prošao ručno regresijsko testiranje od strane timova za osiguranje kvaliteta (QA), tek nakon što je proces razvoja završen. Ručni testeri su pratili softversku dokumentaciju korak po korak kako bi potvrdili da nove i postojeće funkcionalnosti rade kako je definisano. Svi problemi koje je otkrio QA su prosleđeni nazad programerima da ih poprave. Kada je testiranje softvera odvojeno od razvoja i sprovodi se samo nekoliko puta mesečno ili godišnje, programeri često uče o greškama koje su napravili nedeljama ili mesecima nakon što su ih napravili.

Do tog trenutka teško je setiti se logike koda koji ima grešku, a greška postaje spora i teško ju je rešiti i iz nje učiti. Osim sporih povratnih informacija za programere, ručno testiranje usporava tempo kojim nove verzije softvera mogu biti objavljene klijentima. Pouzdanost ručnog pristupa takođe se može dovesti u pitanje, pošto su ručni zadaci koji se ponavljaju skloni ljudskim greškama. Štaviše, kako softverski sistem koji se testira (SUT) raste i evoluira, dokumentacija koja se koristi za ručno testiranje takođe zahteva da se stalno ažurira, što košta dodatno vreme i trud.

Automatizovano testiranje rešava gore opisane probleme. Danas je široko prihvaćena praksa da programeri pišu automatizovane testove koji ne uspeju u izgradnji aplikacije ako otkriju regresije. Lokalno pokretanje automatizovanih testova omogućava programerima da rano primele greške i da ih brzo poprave, često pre nego što se greške provere u kontroli verzija.

2.2. Tipovi automatizovanih testova

Automatizovano testiranje softvera može se vršiti na različitim nivoima, od najjednostavnijih gradivnih blokova aplikacije do načina na koji aplikacija funkcioniše kao celina. Test piramida, koncept koji je popularisao Kon, je način vizuelizacije strategije testiranja koja uključuje više slojeva različitih tipova automatizovanih testova.

Piramida obezbeđuje ravnotežu brzine i složenosti postavljajući veliki broj malih testova koji se brzo izvršavaju na dnu, sa sve sporijim i složenijim testovima ka vrhu. Tri tipa automatizovanih testova pomenutih u piramidi su testovi jedinica, integracije i end-to-end testovi.

Postoji više terminologija koje definišu različite vrste automatizovanih testova.

2.3. Ciljevi test automatizacije

Testovi bi trebalo da poboljšaju kvalitet softvera. Kvalitet softvera se može utvrditi odgovorom na dva pitanja: da li je softver pravilno napravljen i da li je napravljen pravi softver. Praksa koja pomaže da se odgovori na ova pitanja nezavisno jedno od drugog je pisanje testova pre pisanja softvera. Ovaj pristup olakšava razmatranje šta softver treba da radi odvojeno od načina na koji to treba da radi. Pristup pisanju testova kao prvi test, gde testovi pomažu u dizajniranju softvera, široko je rasprostranjen u industriji. Prvo pisanjem testova lako je verifikovati ponašanje sistema pre verifikacije njegove implementacije.

Test-driven development je tehnika razvoja softvera gde se proces razvoja vodi pisanjem testova. Svaki put kada treba implementirati novi deo funkcionalnosti, prvo se napiše test za tu funkcionalnost. Zatim se sama funkcija implementira i prolazi test. Konačno, i karakteristika i test se refaktorisu radi strukture i jasnoće. Pisanje testova pre softvera daje jasnu definiciju uspeha. Na ovaj način, testovi deluju kao specifikacija za softver.

2.4. Pokrivenost koda

Pokrivenost koda je metrika koja se koristi za izražavanje koje delove softvera testovi ne koriste, koliki deo izvornog koda je testiran. Ova metrika se može primeniti na različite vrste automatizovanih testova uključujući testiranje jedinica.

Iako 100% pokrivenost testom može izgledati kao dobar cilj, u stvarnosti to ne garantuje nedostatak nedostataka. Pokrivenost od 100% samo znači da je sav kod koji se testira bio izvršen, a ne da se sistem ponaša u skladu sa specifikacijom.

2.5. Razvoj vođen ponašanjem i testiranje u stilu specifikacije

Razvoj vođen testovima se zalaže za korišćenje jedne pune rečenice za svaki naziv metode testiranja. Korišćenje punih rečenica omogućava izradu softverske dokumentacije koja ima smisla ne samo za programere, već i za menadžere projekata, analitičare, testere i druge poslovne korisnike. Na ovaj način BDD može da obezbedi ažurnu dokumentaciju za softver.

Automatizovani testovi se proveravaju u kontrolu verzija zajedno sa kodom aplikacije, a programeri koji žele da razumeju kako sistem funkcioniše mogu da pročitaju test paket kao primer kako da koriste API sistema. Držanje jedne rečenice po nazivu testa čini testove fokusiranim, jer se samo relativno mali deo ponašanja može opisati u rečenici. Nazivi rečenica su takođe od pomoći u slučajevima kada testovi ne uspeju – željeno ponašanje je odmah jasno.

2.6. Test doubles

Softver koji se testira sastoji se od mnogo jedinica. Svaka jedinica ima određeno ponašanje. Neke jedinice su sadržane u jednoj klasi ili funkciji. Međutim, većina jedinica zahteva interakciju sa drugim jedinicama. Prilikom testiranja određene jedinice, ovo oslanjanje na druge jedinice (koje se nazivaju saradnici) može biti problematično – ili saradnik nije dostupan, ili saradnik vraća rezultate koji nisu prikladni za test, ili izvršavanje saradnika ima nepotrebne nuspojave. U ovim situacijama, saradnici mogu biti zamenjeni kako bi se stekla potpuna kontrola nad okruženjem u kojem se testira jedinica. Test double je alat koji se koristi za izolaciju jedinice od njenih saradnika kako bi se testirala.

Postavljanje testnih duplikata je obično prvi korak u testu. Uspostavljeni obrazac za strukturiranje testova jedinica je raspored-postupi-potvrđivati. Prvo se postavljaju objekti potrebni za test. Zatim se izvršava ponašanje koje se testira. Na kraju, daju se tvrdnje o rezultatu izvršenja. Isti obrazac je takođe opisan kao četvorofazni test. Četiri koraka su podešavanje, vežbanje, verifikacija i demontaža.

3. REST

Termin REST je skraćenica za Representational State Transfer i definisao ga je Roy Thomas Fielding u svojoj doktorskoj disertaciji: „Arhitektonski stilovi i dizajn mrežnih softverskih arhitektura“, objavljen 2000. REST nije sama softverska arhitektura, već „koordinirani skup arhitektonskih ograničenja koja pokušava da minimizira kašnjenje i mrežnu komunikaciju, dok maksimizira nezavisnost i skalabilnost implementacije komponenti“. Ovo poglavlje predstavlja REST: uvod u to kako Fielding pristupa REST-u kao arhitektonskom stilu, opis specifičnih ograničenja i elemenata koji čine REST, apstrakcije RESTful sistema i konačno, objašnjenje RESTful sistema koristeći HTTP.

3.1. REST pristup

REST je arhitektonski stil za distribuirane hipermedijske sisteme. To je hibridni stil izveden iz različitih arhitektonskih stilova zasnovanih na mreži i kombinovan sa dodatnim ograničenjima kako bi se definisao uniformni interfejs konektora. Fokusira se na ograničenja koja se moraju postaviti na semantiku konektora, dok se drugi stilovi fokusiraju na ograničenja semantike komponenti.

Dodatna ograničenja REST-a potiču od sledećih arhitektonskih stilova: ClientServer, Client-Stateless-Server, Client-Cache-StatelessServer, Layered Sistem i Code On Demand. REST takođe uključuje koncepte resursa i uniformnog interfejsa.

3.2. Ograničenja

REST je arhitektonski stil za distribuirane hipermedijske sisteme. To je hibridni stil izveden iz različitih arhitektonskih stilova zasnovanih na mreži i kombinovan sa dodatnim ograničenjima kako bi se definisao uniformni interfejs konektora. Fokusira se na ograničenja koja se moraju postaviti na semantiku konektora, dok se drugi stilovi fokusiraju na ograničenja semantike komponenti.

Dodatna ograničenja REST-a potiču od sledećih arhitektonskih stilova: *ClientServer*, *Client-Stateless-Server*, *Client-Cache-StatelessServer*, *Layered Sistem* i *Code On Demand*. REST takođe uključuje koncepte resursa i uniformnog interfejsa.

Interfejs Klijent-Server zahteva postojanje klijentske komponente koja šalje zahteve i serverske komponente koja prima zahteve i može da izda odgovor. Ovo ograničenje je zasnovano na principu razdvajanja briga. Jedinstveni interfejs razdvaja interfejs klijenata i servera. Ovo razdvajanje briga znači da klijenti, na primer, nisu povezani sa skladištenjem podataka koje je problem servera, a serveri nisu povezani sa korisničkim interfejsom ili korisničkim stanjem koje se tiče klijenta. Ovo poboljšava prenosivost interfejsa na više platformi i skalabilnost pojednostavljujući komponente servera. Takođe podržava nezavisnu evoluciju logike na strani klijenta i logike na strani servera, jer se

svaka komponenta može zameniti i razvijati zasebno sve dok se interfejs među njima ne menja. Klijent-Server je možda najosnovnije ograničenje jer se sva druga ograničenja pozivaju na njegove artefakte i tako se nadograđuju na ovo ograničenje.

Statelessness ograničenje se dodaje ograničenju Klijent-Server. U svakoj interakciji, komunikacija između klijenta i servera mora biti bez stanja. To znači da svaki zahtev bilo kog klijenta treba da sadrži sve informacije neophodne kako bi server razumeo značenje zahteva. Zatim, sve podatke koji se tiču stanja sesije treba vratiti klijentu. Dakle, stanje sesije se u potpunosti čuva na klijentu i server ne može ponovo da koristi informacije iz prethodnih zahteva.

Da bi se ublažilo smanjenje performansi mreže zbog ograničenja bez stanja, dodato je još jedno ograničenje na stil klijent-bez stanja-server prikazan ranije. REST uključuje ograničenje keša tako da naredni zahtevi serveru ne moraju da se upućuju ako su potrebni podaci već u lokalnom kešu na strani klijenta. Tako se formira klijent-keš-server bez stanja.

3.3. REST elementi

REST arhitektonski stil ograničava arhitekturu na arhitekturu klijent/server i dizajniran je da koristi komunikacioni protokol bez stanja, obično HTTP. U stilu REST arhitekture, klijenti i serveri razmenjuju reprezentacije resursa korišćenjem standardizovanog interfejsa i protokola. Svaka softverska arhitektura je sastavljena od komponenti, konektora i podataka. „REST ignoriše detalje implementacije komponenti i sintakse protokola kako bi se fokusirao na uloge komponenti, ograničenja njihove interakcije sa drugim komponentama i njihovu interpretaciju značajnih elemenata podataka“.

Uloga REST komponenti je uspostavljanje komunikacije. Korisnički agent koristi klijentski konektor da pokrene zahtev i postaje krajnji primalac odgovora. Izvorni server koristi serverski konektor da upravlja prostorom imena za traženi resurs. Proksi je posrednik koji je izabrao klijent da obezbedi inkapsulaciju interfejsa drugih usluga, prevod podataka, poboljšanje performansi ili bezbednosnu zaštitu. Mrežni prolaz je posrednik koji nameće mreža ili izvorni server da bi obezbedio inkapsulaciju interfejsa drugih usluga, za prevođenje podataka, poboljšanje performansi ili sprovođenje bezbednosti.

Konektori predstavljaju apstraktni interfejs za komunikaciju komponenti, poboljšavajući jednostavnost pružanjem jasnog razdvajanja briga i skrivanjem osnovne implementacije resursa i komunikacionih mehanizama.

Komponente REST komuniciraju tako što prenose reprezentaciju resursa u formatu koji odgovara jednom od evoluirajućeg skupa standardnih tipova podataka koji se biraju dinamički na osnovu mogućnosti ili želja primaoca i prirode resursa. Dakle, elementi podataka se mogu sumirati na sledeći način: resursi, identifikatori resursa, reprezentacije, reprezentacije i metapodaci resursa i kontrolni podaci.

3.4. REST API

Akronim API dolazi iz Application Programming Interface. API je skup funkcija i procedura koje ispunjavaju jedan ili više zadataka u svrhu da ih koristi drugi softver.

Omogućava implementaciju funkcija i procedura koje su u skladu sa API-jem u drugom programu bez potrebe da se ponovo programiraju.

3.5. REST kroz HTTP

The Hypertext Transfer Protocol (HTTP) je protokol zahteva/odgovora na nivou aplikacije bez stanja koji koristi proširivu semantiku i samoopisno opterećenje poruka za fleksibilnu interakciju sa hipertekstualnim informacionim sistemima zasnovanim na mreži. U RESTful sistemu, klijenti i serveri pregovaraju o predstavljanju resursa preko HTTP-a. RESTful sistemi primenjuju četiri osnovne funkcije trajnog skladištenja, CRUD (Create, Read, Update, Delete), na skup resursa. U smislu HTTP standarda, te radnje se mogu prevesti na HTTP metode (poznate i kao glagoli): POST, GET, PUT i DELETE (Slika 2). Druge HTTP metode koje se takođe koriste, ali ne tako često kao one koje se koriste su OPTIONS, HEAD, TRACE, PATCH i CONNECT.

CRUD actions	HTTP method equivalence
Create	POST
Read	GET
Update	PUT
Delete	DELETE

Slika 2. CRUD i HTTP ekvivalentnost

4. JEST ALAT ZA AUTOMATIZOVANO TESTIRANJE VEB APLIKACIJA

Jest je JavaScript okvir za testiranje otvorenog koda koji je kreirao i održava Facebook. To je jedan od mnogih JavaScript okvira za testiranje, pored Mocha, Jasmine, Karma i drugih. U godišnjoj anketi o stanju JavaScript-a za 2018. u kojoj je anketirano preko 20.000 programera, Jest je pokretač testova, biblioteka tvrdnji i biblioteka za mokovanje. Jest takođe nudi ugrađeni alat za pokrivanje testa i funkciju koja se zove testiranje snimka. Za razliku od nekih svojih prethodnika, kao što je Jasmine, Jest ne zahteva pravo okruženje pretraživača za pokretanje. Umesto toga, pokreće se unutar procesa Node.js gde se API-ji pretraživača emuliraju pomoću jsdom-a.

Simulirano okruženje pretraživača ubrzava izvršavanje testova i omogućava pokretanje testova na različitim sistemima sa istim rezultatima. Isti testovi se mogu izvršiti na mašini programera i na serveru za kontinuiranu isporuku. Kao test runner, Jest pronalazi i izvršava testove. Tačnije, traži test datoteke u datom projektu i izvršava testove unutar tih datoteka. Podrazumevano, Jest proverava svaku datoteku u osnovnom direktorijumu projekta. Precizna lista fascikli za proveru može da se obezbedi kao roots svojstvo konfiguracione datoteke Jest.

Kao biblioteka tvrdnji, Jest obezbeđuje skup alata za pisanje testova na jednostavan, čoveku čitljiv način. Ovi alati su funkcije koje se nazivaju uparivači. Uparivači omogućavaju testiranje vrednosti na različite načine. Da bi se testirala određena vrednost, potrebno je kreirati objekat očekivanja. Zatim se poziva uparivač na objektu očekivanja. Kao biblioteka za mokovanje, Jest obezbeđuje alat koji se zove lažne funkcije za kreiranje testnih doubles. Lažne funkcije omogućavaju zamenu zavisnosti unutar testa jedinice nečim što se može kontrolisati i pregledati. Jest uključuje Istanbul biblioteku za

pokrivenost JavaScript koda. Istanbul obezbeđuje veliki broj reportera koji omogućavaju generisanje izveštaja u različitim formatima.

5. STUDIJA SLUČAJA - TESTIRANJE VEB APLIKACIJE ZA MANIPULACIJU KORISNIKA

Za demonstraciju automatskog RESTful API testiranja modernog softverskog projekta koristiće se jedan projekat pod nazivom „USER-API“ koji treba da pruži manipulaciju korisnika. Veb aplikacija omogaćava da se kreira korisnik, koji ima neophodne atribute kao što su ime, email i lozinka, da vrše izmene svojih atributa, kao i mogućnost brisanja. Fokus ovog poglavlja je API testiranje sa Jestom. U nastavku će biti testirane sve funkcionalnosti koje aplikacija nudi. Aplikacija je implementirana pomoću Express radnog okvira.

Funkcionalnosti koje pokriva naša aplikaciju su kreiranje, izmenu, brisanje i dobavljanje korisnika, kao i dobavljanje svih korisnika iz baze podataka. Test scenario za brisanje korisnika je potrebno sprovesti radi utvrđivanja validnosti implementacije funkcije za brisanje korisnika. Slika 3 prikazuje funkciju za brisanje korisnika.

```
router.delete("/deleteUser/:id", (req, res, next) => {
  db.run(
    'DELETE FROM user WHERE id = ?',
    req.params.id,
    function (err, result) {
      if (err) {
        res.status(400).json({ "error": res.message });
        return;
      }
      if (this.changes !== 0) {
        res.json({ "message": "deleted", rows: this.changes, "data": result });
      } else {
        res.status(404).send("User doesn't find!");
      }
    }
  );
});
```

Slika 3. Prikaz funkcije za brisanje korisnika DELETE metodom

RESTful API automatizovani test za brisanje postojećeg korisnika proverava uspešnost brisanja korisnika. Slika 4 prikazuje test za brisanje korisnika.

```
test("Should delete a user successfully", async () => {
  const EMAIL = randomEmail({ domain: 'gmail.com' });
  const requestBody = getBodyRequest(NAME, EMAIL, PASSWORD);
  const res = await request(app)
    .post('/user/addUser')
    .send(requestBody)
    .set('Accept', 'application/json');
  const userId = res.body.id;
  const response = await request(app)
    .delete(`/user/deleteUser/${userId}`)
    .expect('Content-Type', /json/)
    .expect(200);
  const responseForGet = await request(app)
    .get(`/user/getUser/${userId}`)
    .set('Accept', 'application/json')
    .expect('Content-Type', /json/);
  expect(responseForGet.statusCode).toBe(404);
  expect(responseForGet.body).toEqual("User doesn't find!");
});
```

Slika 4. Prikaz testa za verifikaciju brisanja postojećeg korisnika

6. ZAKLJUČAK

U ovom radu opisano je automatizovano testiranje u Express aplikaciji. Objašnjen je JEST radni okvir sa fokusom na RESTful API testove. Objašnjen je pojam REST. Opisani su korišćeni alati i tehnologije za razvoj aplikacije.

7. LITERATURA

- [1] James Bach. Test Automation Snake Oil. http://www.satisfice.com/articles/test_automation_snake_oil.pdf, 1999. [Online; accessed 13-May-2012].
- [2] Dale H. Emery. Writing Maintainable Automated Acceptance Tests. http://dhemery.com/pdf/writing_maintainable_automated_acceptance_tests.pdf, 2009. [Online; accessed 14-May-2012]
- [3] M. Henning. API: Design Matters. ACM Queue, 5(4), 2018.
- [4] B. A. Myers. Human-Centered Methods for Improving API Usability. In 2017 IEEE/ACM 1st International Workshop on API Usage and Evolution (WAPI), pages 2–2, May 2017.

Kratka biografija:



Aleksandar Savić rođen je u Šapcu 1998. god. Osnovne akademske studije završio je 2021. godine na Fakultetu tehničkih nauka u Novom Sadu. Master rad na Fakultetu tehničkih nauka iz oblasti Računarstvo i automatika – Računarska grafika odbranio je 2022. godine

kontakt: acasavic2409@gmail.com

PREDIKCIJA ŽANRA PESME NA OSNOVU TEKSTA PESME POMOĆU ALGORITAMA MAŠINSKOG UČENJA**PREDICTION OF SONG GENRE BASED ON SONG LYRICS USING MACHINE LEARNING ALGORITHMS**Marko Jovanović, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSTVO I AUTOMATIKA**

Kratak sadržaj – Muzički svet je u procesu eksponencijalnog rasta, naročito u proteklih desetak godina kada se zahvaljujući digitalizaciji dogodila značajna ekspanzija. Broj novih umetnika i novih pesama je u konstantnom porastu, što je dovelo do potrebe za kreiranjem inteligentnog i efikasnog načina da se slušaoci snađu u moru različitih izbora. Jedan od kvalitetnih načina za filtriranje numera je filtriranje na osnovu žanra pesme. U ovom radu će biti opisan sistem za klasifikaciju pesama u različite žanrove na osnovu teksta pesme. Podaci za potrebe obučavanja i testiranja sistema su prikupljeni iz više različitih izvora i transformisani tako da čine jedan jedinstven skup podataka. Najbolji rezultati su postignuti upotrebom GloVe vektorskih reprezentacija reči i LSTM arhitekture mreže. Postignuta je tačnost od 78%, preciznost od 78%, odziv od 77% i F-mera od 77%.

Ključne reči: mašinsko učenje u muzici, duboko učenje u muzici, predikcija žanra pesme

Abstract – The world of music is in the process of exponential growth, especially in the past ten years when a significant expansion took place thanks to the digitalization of the world. The number of new artists and songs is constantly increasing, which has led to the need to create an intelligent and efficient way for listeners to navigate the myriad of choices. One of the proven ways to filter tracks is to filter based on song genre. This paper will describe a system for classifying songs into different genres based on the song's lyrics. Data for training and testing the system is collected from several different sources and transformed into a single data set. The best results were achieved using GloVe word embeddings and LSTM network architecture. This model achieved 78% accuracy, 78% precision, 77% recall, and 77% F-measure.

Keywords: machine learning in music, deep learning in music, song genre prediction

1. UVOD

Automatska klasifikacija muzike po žanrovima jedan je od glavnih zadataka u oblasti *music information retrieval*-a (MIR). Popularizacijom mobilnih telefona i drugih prenosivih uređaja, platforme za reprodukciju muzike kao

što su *Spotify* [1] i *Apple music* [2] dobijaju na značaju i konstantno se takmiče u pokušajima da svojim korisnicima pruže kvalitetne muzičke preporuke.

Kvalitet preporučene pesme mnogo zavisi od raspoloživih meta-podataka o pesmi. Generisanje meta-podataka o pesmama je u prošlosti bio manuelni zadatak i često je veoma dugo trajao. Kako je sve više sadržaja dostupno na internetu, a i dinamika konzumiranja sadržaja postaje sve kompleksnija, moderna rešenja su postala potrebna kako bi anotacija numera bila kvalitetna i brza. Razvojem različitih algoritama mašinskog učenja, kao i razvojem samog hardvera računara, omogućeno je obučiti različite modele koji će automatski obavljati posao generisanja meta-podataka o pesmama.

U ovom radu su, za rešavanje problema automatske klasifikacije pesme po žanrovima na osnovu teksta pesme, korišćene tehnike obrade prirodnog jezika (engl. *Natural Language Processing* - NLP). Prilikom rešavanja NLP problema, cilj je da se tekstu pesme pridruže odgovarajuće labele sa semantičkim značenjem, koje u ovom slučaju predstavljaju žanr pesme.

Moderna rešenja za klasifikaciju tekstualnog sadržaja su najbolje opisana u radu [6]. Sva moderna rešenja, na kojima se zasnivaju modeli opisan u ovom radu, se temelje na kompleksnim vektorskim prostorima za reprezentaciju reči, dubokom učenju i na arhitekturama konvolucionih neuronskih mreža i LSTM mreža.

Glavni cilj ovog rada je da detaljno opiše proces pripreme podataka da bi bili u odgovarajućem formatu za uspešno treniranje modela dubokog učenja. Takođe, predstavljeni su različiti algoritmi dubokog učenja kojima je vršena predikcija žanra pesme na osnovu teksta pesme. Na ovaj način se obogaćuje skup meta-podataka i rešava se prethodno opisan problem. Sistem predstavljen u radu uspešno uočava različite skupove reči karakteristične za pojedine žanrove i pomoću tih informacija klasifikuje pesme u jedan od deset mogućih žanrova. Pokazano je da zbog prirode problema i teškoće učenja pravila u govornom jeziku kompleksniji modeli dubokog učenja postižu znatno bolje rezultate.

2. PRETHODNA REŠENJA

Slični radovi su odabrani tako što je uzeta u obzir metodologija i slični modeli mašinskog učenja. Prethodna rešenja su takođe rešavala problem automatske klasifikacije žanra pesme na osnovu teksta pesme.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Jelena Slivka, vanr. prof.

Takođe, uzeti su u obzir radovi koji koriste skupove podataka generisane na različite načine.

Upotreba hijerarhijskih mreža sa mehanizmom pažnje (engl. *Hierarchical Attention Network* - HAN) je predložena u radu [3]. HAN uzimaju u obzir unutrašnju strukturu i organizaciju podataka kako bi se unapredio model. Primećeno je da se muzičke numere sastoje od reči koje čine segmente, a segmenti čine pesmu u celosti. HAN se primenjuje za klasifikaciju pesme na osnovu više pomenutih nivoa teksta pesme, počev od nivoa reči, do nivoa segmenta. Cilj rada [3] je bio da se pokaže da ovakva arhitektura pruža bolje rezultate u odnosu na arhitekture koje ne uzimaju u obzir hijerarhijsku organizaciju. Skup podataka koji je korišćen u radu [3] predstavlja kolekciju tekstova pesama prikupljenih upotrebom licenciranog softvera *LyricFind* [4]. Čitav skup podataka se sastojao od preko milion pesama koje su bile sačuvane u JSON formatu. Pošto u ovom skupu podataka nisu bile dostupne informacije o žanru pesme, početni skup podataka je proširen. Informacije o žanru pesme su preuzete pomoću aplikacije *iTunes Search API* [5]. Iz skupa podataka su izbačeni oni žanrovi koji su imali manje od 50 pesama predstavnika tog žanra. Ove transformacije su dovele do toga da se finalni skup podataka sastojao od oko pola miliona tekstova pesama koje su pripadale jednom od 20 različitih žanrova. Obučena neuronska mreža sa mehanizmom pažnje je postigla preciznost od 43.66%.

Jedan od zanimljivijih radova na ovu temu bio je rad [6] koji poredi više različitih algoritama mašinskog učenja i njihove rezultate. Korišćeni su model nasumične šume (engl. *random forest*), kao i konvolucione i LSTM arhitekture dubokih neuronskih mreža. Opisani su samo najbolji rezultati koji su postignuti LSTM arhitekturom mreže. Broj žanrova za predikciju je bio sedam. Tačnost za ovaj pristup je iznosila 62.4%.

Rad [7] se zasniva na pretpostavci da se tekstovi pesama dovoljno razlikuju da bi automatska detekcija žanra pesme, kvaliteta pesme, kao i godine izdavanja pesme bili mogući. Pokazano je da klasifikatori koji uzimaju u obzir informacije o tekstu pesme pružaju kvalitetnije predikcije od onih koji koriste samo zvuk pesme. Pored detekcije žanra, u radu [7] je predstavljeno i rešenje koje razlikuje dobre i loše pesme. Zbog nedostatka dovoljno velikog skupa podataka autori su koristili tekstove sa *LyricsMode* [8] veb stranice zbog velike pokrivenosti različitih žanrova i konzistentnosti u podacima. Primenjene su metode pretprocesiranja teksta, uklanjanje duplikata i tokenizacija reči pesama. U finalni skup podataka su uključene samo pesme koje su na engleskom jeziku. Konačni skup podataka se sastojao od 400 hiljada tekstova pesama i pratećih žanrova. Za klasifikaciju je korišćen *n-gram* model, a broj žanrova za predikciju je bio osam.

3. SKUPOVI PODATAKA

Za potrebe ovog rada, podaci su prikupljeni iz više različitih izvora. Informacije o imenu pesme i žanrovima su preuzete iz skupa podataka *Million Song Dataset* - MSD [9], dok su informacije o tekstovima pesme preuzete sa *Genius* [10] API-ja. Proces preuzimanja

tekstova pesme je bio iterativan. Za svako ime pesme i ime izvođača iz skupa MSD poslat je zahtev na *Genius* API koji je vraćao tekst tražene pesme. Ukoliko nekim slučajem tekst pesme nije bilo moguće pronaći, pesma bi bila izbačena iz skupa podataka. Dešavali su se slučajevi gde je za određenu pesmu postojalo više mogućih tekstova pesme. Ovaj problem je rešen tako što je uzet u obzir samo prvi predloženi tekst pesme, a svi ostali su odbačeni.

Za pretprocesiranje podataka korišćene su različite tehnike za obradu prirodnog jezika. Prvo su svi karakteri prebačeni u mala slova. Nakon toga su izbačeni znaci interpunkcije. Da bi se uklonile nevažne reči u tekstovima iskorišćen je predefinisani skup stop reči iz NLTK [11] biblioteke. Lematizacijom su reči prebačene u svoj osnovni oblik. Na kraju su sve reči predstavljene pomoću vektorskih reprezentacija. Algoritmi koji su korišćeni za ovaj zadatak su bili *Word2Vec* i *GloVe* [12].

Da bi se model mogao obući, da bi se podesili hiper-parametri i da bi se evaluirao model, neophodno je imati tri skupa podataka. Trening skup podataka od 80% početnog skupa podataka je korišćen za obučavanje modela, validacioni skup od 10% početnog skupa je korišćen za podešavanje hiper-parametara, a test skup koji sadrži 10% početnog skupa podataka je korišćen za evaluaciju rešenja. Zbog kompleksnosti korišćenih modela i dugackog trajanja obučavanja različitih modela nije iskorišćena višestruka unakrsna validacija koja bi verovatno bila bolji način za pronalaženje optimalnih vrednosti hiper-parametara.

4. METODOLOGIJA

U ovom poglavlju će biti opisana arhitektura modela za predikciju žanra pesme kroz tri poglavlja: (1) vektorske reprezentacije reči, (2) konvolucione neuronske mreže, (3) LSTM arhitektura konvolucione neuronske mreže. Takođe, u odvojenom poglavlju će biti opisan proces obučavanja modela.

4.1. Vektorske reprezentacije reči

Word2Vec je algoritam pomoću kog se kreiraju vektorske reprezentacije reči. Algoritam je zasnovan na modelu neuronske mreže za učenje asocijacija reči iz velikog korpusa teksta. Ovako kreirane reprezentacije reči uzimaju u obzir i kontekst, tako da su slične reči ili sinonimi bliže u vektorskom prostoru.

Pretrenirane *GloVe* vektorske reprezentacije reči su dotrenirane nad postojećim korpusom. Broj dimenzija je bio podešen na 100, dok je veličina rečnika iznosila 400 hiljada reči.

4.2. Konvolucionna neuronska mreža

Za konstrukciju arhitekture konvolucione neuronske mreže korišćena je *Keras* [13] biblioteka.

Funkcija gubitka koja se koristila je unakrsna entropija (engl. *cross entropy*). Unakrsna entropija se tipično koristi kod problema klasifikacije. *Adam* je korišćen kao optimizacioni algoritam stohastičkog gradijentnog spusta. Aktivaciona funkcija na izlazu je bila *SoftMax* i klasifikacija se vršila za deset mogućih žanrova.

Ulaz u mrežu je bio 2D niz ograničen na dužinu od 2000 vektorskih reprezentacija reči u tekstu pesme, uključujući i *padding*. Svaka reč je predstavljena odgovarajućom vektorskom reprezentacijom čiji je broj dimenzija 100. *Pooling* sloj nije korišćen jer je bitna pozicija reči u tekstu za predikciju žanra.

Izlaz konvolucione neuronske mreže je vektor od deset vrednosti. Svaka vrednost predstavlja verovatnoću pripadanja određenom muzičkom žanru

4.3. LSTM arhitektura mreže

Za potrebe treniranja i konstrukcije arhitekture LSTM neuronskih mreža korišćena je *Keras* [13] biblioteka. Ova biblioteka je upotrebljena zbog velike fleksibilnosti i lakoće kreiranja različitih modela neuronskih mreža.

Kao funkcija gubitka korišćena je unakrsna entropija. Za potrebe algoritma stohastičkog gradijentnog spusta korišćen je *Adam* optimizacioni algoritam. Ovaj algoritam karakteriše adaptivna stopa učenja. Maksimalna dužina sekvence reči podešena je na 2000, dok je broj dimenzija vektora reprezentacije reči bio 100. Aktivaciona funkcija koja se koristila u poslednjem sloju je *SoftMax* funkcija, koja je pogodna za rešavanje problema klasifikacione prirode.

Cilj, odnosno izlaz, ove neuronske mreže je predikcija odgovarajućeg žanra pesme na osnovu ulaza koji predstavlja sekvencu vektorskih reprezentacija reči. Izlazna vrednost je vektor od deset vrednosti verovatnoće pripadanja svakom od žanrova.

4.4. Obučavanje modela

Za obučavanje, validaciju i testiranje modela za klasifikaciju žanra pesme korišćen je hibridni skup podataka. Ime pesme, ime izvođača, žanr i tekst pesme su preuzeti iz skupa podataka sa *Million songs dataset* [9] i *Genius* [10] API-ja. Skup podataka je sadržao oko million različitih numera i meta-podatke o njima. Inicijalno je postojalo 60 različitih žanrova, ali su modeli obučavani na podskupu od 10 najrelevantnijih žanrova.

Kod obučavanja modela konvolucione neuronske mreže broj epoha je bio podešen na 102, iz razloga što je za svako treniranje bilo potrebno ~14 časova da se završi. *Batch size* hiper-parametar je bio podešen na 64.

Obučavanje LSTM arhitekture mreže je bio veoma slično kao i obučavanje konvolucione neuronske mreže. Broj epoha je bio podešen na 102, dok je *batch size* podešen na 64.

5. EVALUACIJA REŠENJA I REZULTATI

Mere evaluacije sistema za klasifikaciju pesama korišćene u ovom radu su F1-mera, tačnost, odziv i preciznost. Za računanje ovih metrika neophodno je imati poznate vrednosti iz matrice konfuzije (engl. *confusion matrix*) [14].

Tačnost se računa kao količnik ukupnog broja tačnih predikcija i ukupnog broja predikcija:

$$Tačnost = \frac{\text{broj tačnih predikcija}}{\text{ukupan broj predikcija}}$$

Odziv daje informaciju koliko stvarno pozitivnih primera je model uspeo tačno da predvidi i računa se:

$$Odziv = \frac{\text{stvarno pozitivni}}{\text{stvarno pozitivni} + \text{lažno negativni}}$$

Preciznost predstavlja meru koliko je model tačan pri davanju procene da je neki od primera pozitivan u odnosu na sve primere koji su procenjeni kao pozitivni:

$$Preciznost = \frac{\text{stvarno pozitivni}}{\text{stvarno pozitivni} + \text{lažno pozitivni}}$$

F1-mera se računa kao kombinacija mere odziva i preciznosti:

$$F1\text{-mera} = 2 * \frac{\text{preciznost} * \text{odziv}}{\text{preciznost} + \text{odziv}}$$

Za svaki od obučanih modela izračunate su vrednosti za tačnost, odziv, preciznost i F1-meru. Tabela 1 prikazuje rezultate primene:

- Kombinacije konvolucionih neuronskih mreža i *GloVe* vektorskih reprezentacija reči
- Kombinacije LSTM arhitekture neuronske mreže i *Word2Vec* vektorskih reprezentacija
- Kombinacije LSTM arhitekture mreže i *GloVe* vektorskih reprezentacija reči

Tabela 1. Performanse obučanih modela

Naziv modela	Tačnost	Preciznost	Odziv	F-mera
CNN + GloVe	68%	64%	63%	62%
LSTM + Word2Vec	72%	72%	71%	70%
LSTM + GloVe	78%	78%	77%	77%

U tabeli 1 je istaknut u kom se nalaze rezultati modela koji je postigao najbolje vrednosti metrika za evaluaciju. Iako je korišćen manji skup podataka, postignuti su veoma dobri rezultati. Teško je bilo pronaći relevantan naučni rad sa kojim bi mogli da se direktno uporede dobijeni rezultati zbog činjenice da je početni skup podataka kreiran kombinacijom različitih tehnika za prikupljanje podataka.

U tabeli 2 se nalazi detaljni prikaz performansi LSTM modela i *GloVe* vektorskih reprezentacija reči po žanrovima. Neki od žanrova za koje je muzika često tačno klasifikovana pripadaju "hip hop", "metal" i "pop" žanrovima. Ova pojava se objašnjava činjenicom da postoji mnogo karakterističnih reči i načina izražavanja u pomenutim žanrovima. Ovo znatno olakšava klasifikaciju i samim tim su i rezultati bolji od očekivanih. Sa druge strane, muzika koja pripada "rege" žanru i "klasična muzika" su klasifikovane sa slabijom tačnošću. Razlog je

mali broj pesama predstavnika ovih žanrova i činjenica da su ove pesme često na stranom jeziku.

Tabela 2. Detaljni prikaz performansi LSTM + GloVe modela

Žanr	F1-mera
Bluz	67%
Klasika	62%
Kantri	79%
Disko	78%
Hip hop	93%
Džez	57%
Metal	91%
Pop	88%
Rege	56%
Rok	77%

6. ZAKLJUČAK

U ovom radu predstavljeno je i upoređeno više različitih pristupa za rešavanje problema automatske predikcije žanra pesme na osnovu teksta pesme. Pokazano je da, zbog prirode problema i teškoće učenja pravila u govornom jeziku, kompleksniji modeli dubokog učenja postižu bolje rezultate. Takođe, opisan je i proces pripreme podataka da bi bili u odgovarajućem formatu za uspešno treniranje modela dubokog učenja. Dat je i uvid u arhitekturu različitih modela koji su bili obučavani. Iz svega priloženog se može zaključiti da se najbolji rezultati postižu primenom GloVe vektorskih reprezentacija reči i LSTM arhitekture neuronske mreže.

Iako se celokupan sistem može opisati kao uspešan, neki od narednih koraka za dalji razvoj i unapređenje postojećeg rešenja su opisani u nastavku.

Prva ideja bi bila da se ceo skup podataka poveća tako da uključuje više pisama iz slabije zastupljenih žanrova. Ovaj proces može biti i sintetičke prirode gde bi se obučili modeli mašinskog učenja koji bi generisali tekstove pesama određenih žanrova.

Druga ideja jeste da se uzmu u obzir i meta-podaci o samoj pesmi i da se na taj način robusnijim skupom podataka poboljšaju rezultati predikcije. Neke od dodatnih osobina koje bi mogle da se koriste su dužina trajanja pesme, podaci o instrumentima koji se koriste, kao i karakteristike samog audio zapisa poput glasnoće i opsega frekvencija.

Treća ideja jeste da se upotrebom moćnijeg hardvera isproba više različitih arhitektura neuronskih mreža. Na ovaj način bi bilo moguće pronaći optimalne hiper-parametre mnogo brže i efikasnije.

Četvrta ideja jeste da se pokuša sa primenom transformer arhitektura poput BERT-a i GPT-3.

7. LITERATURA

- [1] Spotify <https://www.spotify.com/> [datum pristupa 18.09.2022.]
- [2] Medium <https://www.apple.com/apple-music/> [datum pristupa 18.09.2022.]
- [3] Alexandros T. 2017. *Lyrics-based music genre classification using a hierarchical attention network*
- [4] LyricFind <https://www.lyricfind.com/> [datum pristupa 18.09.2022.]
- [5] iTunes Search API <https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/index.html> [datum pristupa 18.09.2022.]
- [6] Ciao Luiggy R., et. al. 2019. *Combining Diverse Models for Lyrics-based Music Genre Classification*
- [7] Michael Fell, Caroline Sporleder, *Lyrics-based Analysis and Classification of Music*
- [8] Wu, Chuhan & Wu, Fangzhao & An, Mingxiao & Huang, Jianqiang & Huang, Yongfeng & Xie, Xing. (2019). NPA: Neural News Recommendation with Personalized Attention.
- [9] Lyrics mode <https://www.lyricsmode.com/> [datum pristupa 18.09.2022.]
- [10] Million Song Dataset <http://millionsongdataset.com> [datum pristupa 19.09.2022.]
- [11] Genius <https://genius.com> [datum pristupa 19.09.2022.]
- [12] NLTK <https://www.nltk.org/> [datum pristupa 19.09.2022.]
- [13] GloVe: Global Vectors for Word Representation <https://nlp.stanford.edu/projects/glove/> [datum pristupa 19.09.2022.]
- [14] Keras <https://keras.io/> [datum pristupa 19.09.2022.]
- [15] Multi-class Classification: Extracting Performance Metrics From The Confusion Matrix <https://towardsdatascience.com/multi-class-classification-extracting-performance-metrics-from-the-confusion-matrix-b379b427a872> [datum pristupa 19.09.2022.]

Kratka biografija:

Marko Jovanović rođen je 07.03.1997. godine u Novom Sadu, Republika Srbija. Upisao se na Fakultet tehničkih nauka, odsek Računarstvo i automatika 2016. godine. Diplomirao je 2020. godine i iste godine upisuje se na master akademske studije, odsek Računarstvo i automatika, smer Elektronsko poslovanje.

UTICAJ DISTRIBUTIVNIH GENERATORA NA REGULACIJU NAPONA DISTRIBUTIVNIH MREŽA**IMPACT OF DISTRIBUTED GENERATORS ON VOLTAGE REGULATION IN DISTRIBUTION NETWORK**

Kristina Milić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je razmatran problem regulacije napona u distributivnim mrežama (DM) sa distributivnim generatorima (DG). Usled proizvodnje aktivne snage u DM, merenja modula struje i napona samo na sekundaru regulacionog transformatora nisu dovoljna za korektan rad klasičnih automatskih regulatora napona. Za korektnu regulaciju napona aktivnih DM potrebna je modifikacija klasične regulacije napona. U radu su date teoretske osnove o klasičnoj i modifikovanoj regulaciji napona. Njihove mogućnosti analizirane su na primeru jednostavne aktivne DM.

Ključne reči: Regulacija napona, aktivne distributivne mreže, regulacioni transformatori sa regulacijom pod opterećenjem

Abstract – This paper discusses the problem of voltage regulation in networks with distributed generators. Measurement of current and voltage on secondary winding aren't sufficient for correct operation of classical automated regulation transformers, during their active power supply to the distribution network. In order to correctly regulate the voltage of an active distributed network, modification is necessary to achieve classic voltage regulation. The paper presents the basic theory of classical and adaptive voltage regulation. Their capabilities are analyzed by using the example of simple active distribution network.

Keywords: Voltage control, Active distribution networks, on-load tap-changing transformer

1. UVOD

Kao posledica tokova snaga na čvorovima DM se realizuju naponi različitih vrednosti. Te vrednosti više ili manje odstupaju od nominalnih vrednosti za koje su potrošači konstruisani. Ako se na potrošačima realizuje napon čija je vrednost niža od nominalne: smanjuju se stepen iskorišćenja i produktivnost rada. Kao posledice napona čija je vrednost veća od nominalne, povećava se naprezanje izolacije opreme, odnosno smanjuje se vek trajanja električnih aparata i stepen iskorišćenja. Prema tome, potrošači trpe štetu kad god vrednost napona odstupa od nominalne. Zbog toga je regulacija vrednosti napona u sistemu jako važna.

NAPOMENA:

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

U skladu sa savremenim trendovima u elektroenergetici (deregulacija tržišta, očuvanje životne sredine, Kyoto protokol, dekarbonizacija i napredak tehnologije) radikalno se menja struktura izvora električne energije. Posledica je da u DM intenzivno raste broj i uticaj DG. Klasična regulacija napona [1-3], zasnovana na automatskom regulatoru napona i unapred podešenom zakonu regulacije, nije spremna da odgovori na nove izazove [4,5].

Nakon Uvoda u drugom delu rada opisani su regulacioni transformatori i postupak proračuna tokova snaga. U trećem delu je izvršena je postavka problema regulacije napona u pasivnim i aktivnim DM. Njihova verifikacija na primeru jednostavne aktivne DM prikazana je u četvrtom delu rada. Zaključci rada i referentno navedena literatura koja je korišćena pri izradi ovog rada, dati su u petom i šestom delu rada, respektivno.

2. OPŠTI POJMOVI

U ovom delu prikazani su regulacioni transformatori, automatski regulatori napona i postupak za proračun tokova snaga, korišćeni za izradu ovog rada.

2.1. Regulacioni transformatori

Regulacioni transformatori (RTr) su energetske transformacije sa mogućnošću podešavanja odnosa transformacije oko nominalne vrednosti, odnosno regulacije napona na sabirnicama njegovog sekundara. U tu svrhu kod njih je promenljiv broj navojaka jednog od namota. Promenljiv broj navojaka se dobija tako što jedna strana namotaja na svakom stubu ima više izvoda. Ti izvodi mogu biti na primarnoj, ili na sekundarnoj strani. Zbog lakšeg preklapanja manjih struja, regulaciona sklopka se postavlja na stranu višeg napona [2].

Prema mogućnostima regulacije izlaznog napona, RTr mogu da budu:

- sa regulacijom pod opterećenjem,
- sa regulacijom u beznaponskom stanju.

2.2. Automatski regulatori napona

Uloga automatskog regulatora napona (ARN) je da se na osnovu izmerene vrednosti modula struje na sekundaru RTr definiše željena (referentna) vrednost modula napona na kontrolisanim SN sabirnicama. Željena vrednost napona se poredi sa izmerenom vrednošću. Ako je njihova razlika veća od dozvoljene, inicira se promena položaja regulacionog otcepa na RTr preko kontrolnih releja.

2.3. Algoritam sumiranja struja

Algoritam sumiranja struja, odnosno Shirmohammadi-ev algoritam, je iterativni postupak za proračun tokova snaga. Algoritam počinje sa inicijalizacijom postupka. Ova inicijalizacija se sastoji u učitavanju podataka o DM, numeraciji grana i čvorova po slojevima i inicijalizaciji indeksa iteracija (h) na početnu vrednost ($h=1$). Nakon inicijalizacije započinje iterativni postupak, gde se svaka iteracija sastoji se od sledeća tri koraka [6]:

1. **Proračun injektiranih struja** – Proračun se izvodi za svaki čvor, počevši od čvorova u prvom sloju, prema sledećoj relaciji, (i)-indeks čvora:

$$i_i^{(h)} = \left(\frac{S_{p,i}}{V_i^{(h-1)}} \right)^* + y_{0,i} \cdot V_i^{(h-1)}, i = 1, \dots, n_{cv} \quad (1)$$

2. **Proračun struja po granama** ("zamena unazad") – Proračun se izvodi za svaku granu, počevši od grana u poslednjem sloju, prema sledećoj relaciji:

$$j_i^{(h)} = i_i^{(h)} + \sum_{j \in I_j} j_j^{(h)}, i = n_{gr}, \dots, 1, \quad (2)$$

3. **Proračun napona u čvorovima** ("zamena unapred") – Proračun se izvodi za svaki čvor, počevši od čvora u prvom sloju, prema sledećoj relaciji:

$$V_i^{(h)} = V_{i-1}^{(h)} - z_i \cdot j_i^{(h)}, i = 1, \dots, n_{cv}. \quad (3)$$

- Proveriti vrednosti napona u dve uzastopne iteracije:
 $\Delta V(h) < \varepsilon. \quad (4)$

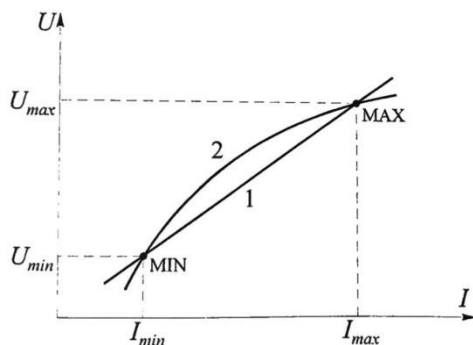
3. REGULACIJA NAPONA U DM

Osnovni cilj regulacije napona je da održava napone u svim čvorovima mreže u granicama propisanih graničnih vrednosti, pritom, ako je moguće što bliže nominalnim vrednostima.

3.1 Klasična regulacija napona

Regulacija napona u DM, pomoću RTr zasniva se na principu kompenzacije pada napona jedinstvenog voda. Ovaj princip se sadrži u ideji da se potrošaču priključenom u određenom čvoru mreže obezbediti konstantan napon, nezavisno od promene opterećenja, ali i nezavisno od promene stanja na visokonaponskim sabirnicama transformatora. Ovakva kontrola, odnosno regulacija naziva se **kompandovana regulacija napona** [1,2].

Optimalna, minimalna i maksimalna vrednost napona na SN sabirnicama RTr, određena je za režime maksimalnog i minimalnog opterećenja. Na osnovu te dve tačke, definisan je zakon regulacije.



Slika 1. Zakon regulacije i karakteristike regulacije za dva tipa regulatora: (1) linearna i (2) nelinearna

3.2 Regulacija napona u aktivnim DM

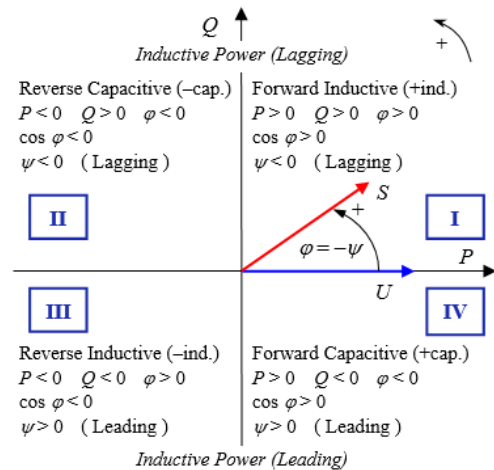
Ugradnjom DG današnje pasivne DM postaju aktivne DM. Pritom, smer tokova aktivne snage više nije samo u jednom smeru, od primara ka sekundaru RTr. Posledica je da merenjem samo modula struje na strani sekundara Tr nije moguće kvalitetno proceniti vrednost opterećenja DM, a samim tim ni optimalnu vrednost napona na sekundaru Tr.

Kada DG injektira snagu u DM, na sekundaru RTr će se realizovati napon čija je vrednost niža od optimalne U_{opt} .

Za razumevanje problema rada klasične regulacije napona u DM sa DG potrebno je razmatrati četiri ekstremna primera:

- 1) DG ne radi, potrošnja je minimalna (maksimalna);
- 2) maksimalna proizvodnja DG, potrošnja je minimalna (za različite vrednosti faktora snage);
- 3) maksimalna proizvodnja DG, potrošnja je maksimalna i
- 4) proizvodnja DG jednaka je potrošnji.

Na slici 2 prikazani su režimi DM, koji mogu da se ostvare priključenjem DG u DM. U pasivnim DM podrazumeva se da su tokovi aktivnih snaga jednosmerni, od korena DM ka njenim potrošačima, I i IV kvadrant. U aktivnim DM tokovi aktivne snage mogu biti dvosmerni (reverse), II i III kvadrant [5].



Slika 2 – P-Q dijagram DM sa DG [5]

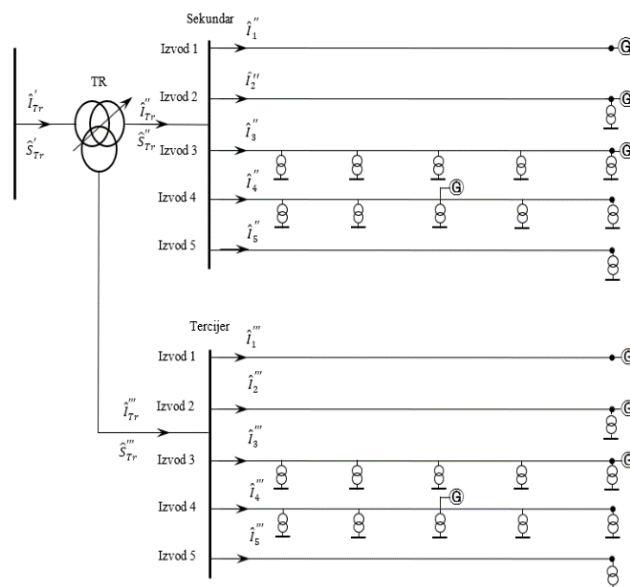
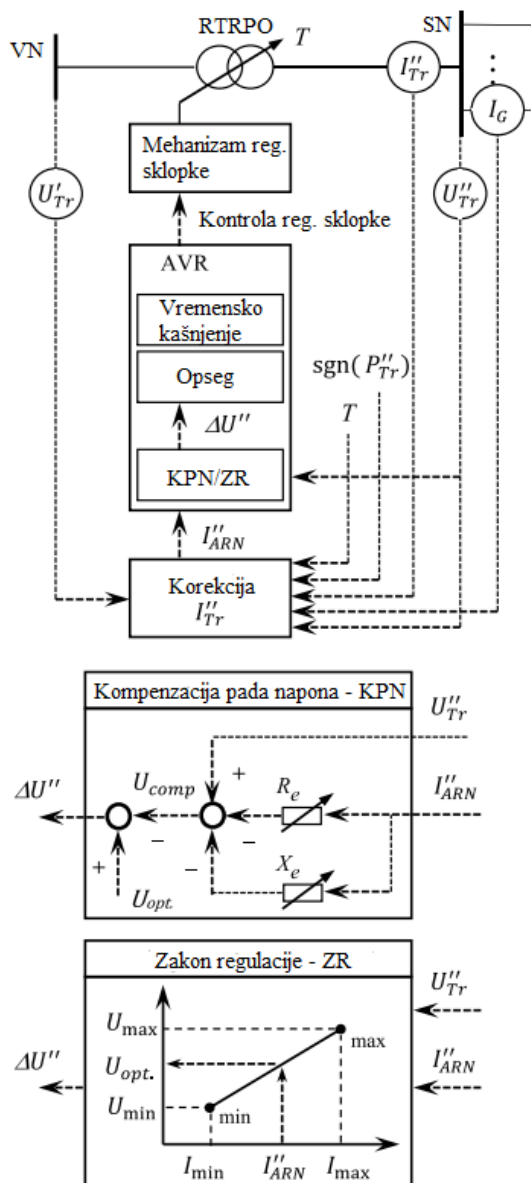
3.3 Modifikovana klasična regulacija napona

Modifikovana KRN zasniva se na korekciji vrednosti modula struje I''_{Tr} sa sekundara RTr. Za njenu korekciju potrebne su aktuelne vrednosti modula napona sa primara i sekundara RTr, modula struje sa sekundara RTr, pozicije regulacione sklopke t i smer aktivne snage kroz RTr. Blok dijagram modifikovane klasične regulacije napona prikazan je na slici 3 [5].

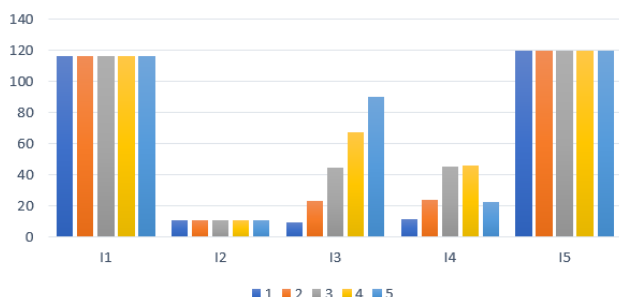
Korekcija struje I''_{Tr} vrši se na osnovu procene stanja na izvodu i rada DG, korišćenjem RTr pod opterećenjem i ukupne struje potrošnje. Korigovana struja I''_{ARN} se šalje na AVR relej.

U zavisnosti od tipa AVR releja i struje koja u njega ulazi, izmerena vrednost napona na sekundaru U''_{Tr} biva korigovana sa procenjenim padom napona u DM. Taj napon obeležava se sa U_{comp} – komparativni napon. Kada se odredi njegova vrednost uzima se optimalna vrednost napona U_{opt} koja je određena direktno pomoću karakteristike releja. Kada se te dve vrednosti dobiju,

U drugom slučaju, uzima se optimalna vrednost napona definisana karakteristikom zakona regulacije U_{opt} i poredi se sa izmerenom vrednošću napona na sekundaru U''_{Tr} . Na osnovu njihove razlike određuje se da li je potrebno da se promeni prenosni odnos RTr.



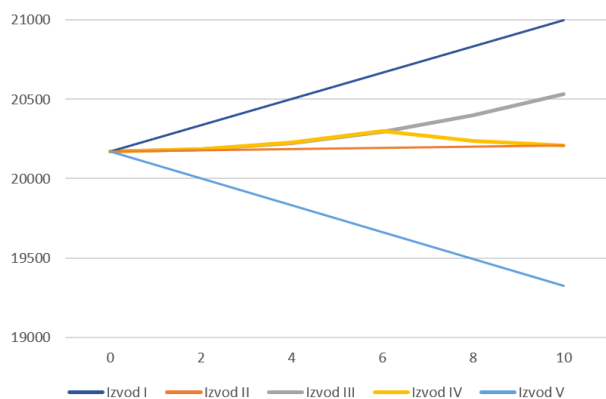
proizvodnja na kraju izvoda. Na trećem izvodu (I_3) struja je najmanja na početku izvoda i onda se sve više povećava do kraja izvoda jer je tu priključen DG. Na četvrtom izvodu (I_4) je struja najveća na sredini izvoda gde je priključen DG, struja se smanjuje udaljavanjem od sredina i najmanja je na krajevima.



Slika 5 – Promena struje po deonici voda za svaki izvod na sekundaru RTr

Na slici 6 prikazane su vrednosti promene napona duž izvoda. Plavom, narandžastom, sivom, žutom i svetloplavom bojom prikazane su promene duž I, II, III, IV i V izvoda, respektivno.

Može se uočiti da vrednost napona značajno raste na I izvodu (samo DG na kraju izvoda); blago raste na II izvodu (isti potrošač i DG na kraju izvoda); raste na III izvodu (ravnomerno raspoređena potrošnja i DG na kraju izvoda); raste do sredine izvoda IV gde je povezan DG i posle opada (ravnomerno raspoređena potrošnja i DG na sredini izvoda); značajno opada na V izvodu (samo potrošač na kraju izvoda). Vrednost napona se povećava od početka ka kraju izvoda.



Slika 6 – Vrednosti napona duž izvoda

5. ZAKLJUČAK

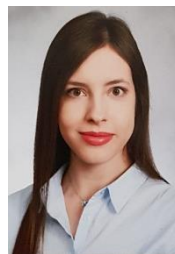
Ugradnjom DG tokovi aktivne snage na sekundaru regulacionog transformatora ne samo da više nisu jednaki opterećenju DM, već i njihov smer više nije unapred poznat. U takvim okolnostima, ARN donosi odluke na osnovu pogrešnih informacija o opterećenju DM čime je onemogućena kvalitetna procena optimalne vrednosti napona na sekundaru RTr, odnosno onemogućen je njen kvalitetan rad i tada se potrošačima isporučuje električna energija sa naponom čija je vrednost ispod vrednosti koja je definisana zakonom regulacije.

Jedan od načina za rešavanje ovih problema jeste modifikacijom klasične regulacije napona koja se sastoji samo u korekciji lažne slike o opterećenju DM.

6. LITERATURA

1. G.Švenda: *Specijalizovani softveri u elektroenergetici*, skripta sa predavanja iz istoimenog predmeta na master studijama, Fakultet Tehničkih Nauka, Novi Sad, 2022.
2. V.Strezoski: *Sistem regulacije napona distributivnih mreža*, Fakultet Tehničkih Nauka, Novi Sad, 1997.
3. G.S.Švenda, Z.Simendic, V.Strezoski: Advanced Voltage Control Integrated in DMS, Electrical Power & Energy Systems, Vol. 43, Issue 1, December 2012, pp. 333-343, DOI: [10.1016/j.ijepes.2012.05.014](https://doi.org/10.1016/j.ijepes.2012.05.014)
4. G.Švenda, I.Krstić, S.Kanjuh, M.Jajčanin: Voltage Improvement in Distribution Networks with High Penetration of Electric Vehicles and Renewable Energy Sources; *CIREN 2022 workshop on E-mobility and power distribution systems*, 2-3 June 2022, Porto, Portugal, Paper No. 1225, DOI: [10.1049/icp.2022.0728](https://doi.org/10.1049/icp.2022.0728)
5. 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)
6. D.Shirmohammadi, H.W.Hong, A.Semlyen, G.X.Luo: A compensation- based power flow method for weekly meshed distribution and transmission networks; *IEEE Trans. on PS*; 1988; Vol.3, pp. 753-762, DOI: [10.1109/59.192932](https://doi.org/10.1109/59.192932)
7. S.Bosančić: Regulacija napona klasične distributivne mreže sa distributivnim generatorom; master rad, FTN, Novi Sad, septembar 2021.

Kratka biografija:



Kristina Milić rođena je u Novom Sadu 1997. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetika, elektronika i telekomunikacije odbranila je 2020.god. Kontakt: kristinamilic15@gmail.com

ANALIZA BEZBEDNOSNIH NAPADA U NODEJS EKOSISTEMU CYBERSECURITY ATTACK ANALYSIS IN THE NODEJS ECOSYSTEM

Boris Šulichenko, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO

Kratak sadržaj – U ovom radu opisano je kako rade NodeJS i NPM i koje su im mane. Opisani su neki od poznatih napada na NPM ekosistem i NodeJS kao platformu za izvršavanje JavaScript koda na serveru. Uz opise kako napadi mogu da se izvrše prikazane su i moguće mitigacije.

Ključne reči: NodeJS, NPM, izvršavanje napada, mehanizmi odbrane.

Abstract – This paper describes how NodeJS and NPM work and what their disadvantages are. Some of the most popular attacks on NPM and NodeJS as a platform for executing JavaScript code on servers are described here. Also, with descriptions of how attacks are executed, this paper shows how users can mitigate those attacks.

Keywords: NodeJS, NPM, executing attacks, defence mechanisms.

1. UVOD

Veliki faktor u razvoju uspešnih web aplikacija igra uloga bezbednosti i otpornosti na različite napade. Samim tim autori aplikacija moraju biti upoznati sa tehnikama kako se napadi izvršavaju, ali i kako se od njih mogu odbraniti.

JavaScript je slabo tipizirani i interpretirani programski jezik visokog nivoa koji je 1995. godine kreirao Brendan Eich za potrebe Netscape web pretraživača. Zbog svoje jednostavnosti i efikasnosti stekao je veliku popularnost među programerima koji se bave razvojem *frontend* delova aplikacija. Iz želje da se JavaScript izvršava van web pretraživača je 2009. godine nastao NodeJS.

Zbog razvoja celog ekosistema napadačima je NodeJS postao zanimljiv koji zbog toga ulažu svoje resurse i vreme u istraživanje i smišljanje novih tehnika za probijanje bezbednosti aplikacija zasnovanih na ovoj tehnologiji od kojih su neke prikazane u ovom radu.

2. INTERNI RAD NODEJS-a

NodeJS je okruženje za izvršavanje JavaScript koda (eng. *runtime environment*) na serveru koji je 2009. godine kreirao Ryan Dahl [1].

Node Package Manager (skraćeno NPM) [2] je alat i registar za upravljanje zavisnostima u projektima koji se razvijaju korišćenjem JavaScript jezika.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Nikola Luburić.

Da bismo shvatili zašto su neki od napada na NodeJS uspešni i kako oni zapravo rade i utiču na njega potrebno je da razumemo od kojih komponenti se NodeJS sastoji i kako one komuniciraju međusobno.

Tri glavne komponente su:

- JavaScript runtime – V8 engine [3]
- Petlja događaja (eng. *event loop*) – *libuv* biblioteka [4]
- Standardna biblioteka funkcija.

2.1. JavaScript runtime

NodeJS spada u okruženja koja rade na principu petlji događaja pa se sa visoke tačke gledišta skoro sva izvršavanja dešavaju u jednoj niti procesora.

Kao *runtime*, NodeJS koristi V8 engine. V8 je softver otvorenog koda i visokih performansi napisan u C++-u, a za potrebe razvoja NodeJS-a je adaptiran da može da radi izvan pretraživača. Njegova uloga je da vrši parsiranje, interpretiranje, kompajliranje i optimizaciju JavaScript koda koji korisnik želi da izvrši.

2.2. Petlja događaja

Prevedeni kod se izvršava u petlji događaja. U te svrhe NodeJS koristi biblioteku *libuv* koja je napisana u C++-u i služi kao apstrakcija nad operativnim sistemom. Pored toga, odgovorna je za asinhrono izvršavanje koda, pristup *file system*-u, mreži, itd.

Petlja događaja nije ništa drugo nego beskonačna *while* petlja koja unutar sebe izvršava kod u nekoliko faza. Gledano sa visokog nivoa apstrakcije ceo proces izvršavanja programa u NodeJS-u se izvršava na jednoj niti i samim tim se postavlja pitanje kako je moguće uslužiti više zahteva istovremeno?

Odgovor je u ne-blokirajućim I/O procesima ili nitima. To znači da proces koji je započeo neku akciju neće zauzeti 100% resursa i time onemogućiti korišćenje aplikacije drugim korisnicima. Proces ili nit će samo uspostaviti komunikaciju sa korisnikom i zatraženu akciju će smestiti u red za izvršavanje. Uz samu akciju korisnik treba da preda i funkciju povratnog poziva (eng. *callback function*) koju proces poziva nakon izvršenja zadatka. Ona se može posmatrati kao odgovor procesa samom korisniku.

Zadatak koji je dobijen od strane korisnika se prebacuje na *worker* nit koja se izvršava u pozadini. Nit će javiti glavnoj niti da je akcija izvršena i radnju obaveštavanja korisnika će prebaciti na nju.

Za ceo ovaj proces izvršavanja zadataka je zadužen *libuv* jer on sadrži *thread pool* i samim tim omogućava *multi-threading* način rada u NodeJS-u. *Libuv* je takođe zadužen i za upravljanje *worker* nitima.

2.3. Standardna biblioteka funkcija

Treća komponenta *NodeJS*-a je biblioteka standardnih funkcija koja pruža gotova rešenja za neke od najčešćih problema sa kojima programeri mogu da se susretnu prilikom razvoja aplikacija. Na primer rad sa događajima, *file system*-om, HTTP protokolom, procesima i nitima.

3. PREGLED PROBLEMA

Većina paketa je napisana od strane ljudi koji se bave razvojem softvera otvorenog koda iza kojih ne stoji velika kompanija koja može da ulaže sredstva u bezbednost tih paketa. Samim tim javlja se veliki broj ranjivosti koje napadači mogu da iskoriste.

3.1. Napadi na NPM ekosistem

Krhkost NPM ekosistema i zavisnosti među paketima se može videti na primeru kada je u martu 2016. godine autor paketa *left-pad* obrisao paket sa repozitorijuma što je dovelo do toga da veliki broj drugih paketa postane nedostupan, tj. neupotrebljiv dok se ne nađe i ne implementira zamena za taj paket [5].

U julu 2018. godine desilo da su kredencijali jednog od autora *eslint-scope* paketa omogućili napadaču da objavi malicioznu verziju paketa koja je slala lokalne fajlove na udaljeni server.

NPM ekosistem ima veliki broj slabih tačaka, a neke od njih su [6]:

- Velika površina napada - prosečan NPM paket koristi oko 80 drugih paketa u stablu zavisnosti i ima oko 40 programera koji ga održavaju ili imaju pristup njemu.
- Popularni paketi direktno i indirektno utiču na više od 100000 paketa.
- Postoje nalozi korisnika koji imaju pristup i do 100000 paketa, na direktan i indirektan način.
- Skoro 40% paketa zavise od drugih paketa gde je poznata bar jedna ranjivost.

Modeli pretnje NPM ekosistema su kreiranje malicioznih paketa, eksploatacija koda koji se više ne održava, preuzimanje kredencijala za upravljanje paketom, itd.

Neke od osnovnih stvari koje se mogu učiniti da se spreče napadi su podizanje svesti ljudi koji učestvuju u izradi paketa o stvarima koje im se mogu desiti i kako da se zaštite od njih. NPM *cli* pruža alat *audit* [7] koji služi za skeniranje paketa u projektu naspram baze podataka ranjivih paketa. Ukoliko se pronađe neki ranjiv paket obavestiće korisnika o riziku pri njegovom korišćenju. Mitigacije koje NPM kao organizacija može da ponudi su provera paketa pre objavljivanja, obuka i provera autora pre dobijanja dozvole za objavljivanje paketa. Ove mitigacije su teške za realizaciju jer zahtevaju veliku količinu resursa.

3.2. Primeri napada

Napad na paket *event-stream*, desio se tako što je napadač dobio *maintainer* pristup kodu uz pomoć socijalnog inženjeringa nakon čega je dodao malicioznu zavisnost u ovaj projekat - *flatmap-stream* paket koji je bez znanja krajnjih korisnika krao *Bitcoin*-e. Ovaj maliciozni kod je opstao dva ipo meseca u produkciji i bio je skinut skoro osam miliona puta [8].

Početkom 2022. godine *maintainer* paketa *colors* i *FakerJS* je objavio verzije u kojima je namerno narušio rad paketa. Kao rezultat došlo je do *Denial of Service* napada na bilo kom serveru koji koristi *NodeJS* i ove pakete. U tom momentu *colors* je imao 20 miliona skidanja nedeljno i koristio se u 4 miliona projekata, a *FakerJS* oko 2 miliona skidanja nedeljno. Mitigacije za ovakvu vrstu napada mogu biti: vraćanje paketa da koristi prethodnu, stabilnu verziju, korišćenje alternativnih paketa sa kojima se može ostvariti isti učinak i pre korišćenja paketa i njegovog nadograđivanja pregledati da li nova verzija donosi neke štetne promene [9].

Ne moraju svi napadi na pakete da budu destruktivne prirode. Primer ovakvog napada je korišćenje paketa *peacenotwar* koji je namenjen u svrhe protesta protiv rata u Ukrajini i koji na standardni izlaz ispisuje aktuelne vesti o ratu. Najveći uticaj se desio na korisnike *vue-cli* paketa i *Unity game engine*-a gde je preko indirektnih zavisnosti u *node-ipc* paket ubačen *peacenotwar* [10].

4. ANALIZA NAPADA

Ovo poglavlje sadrži detaljne analize napada na NPM i *NodeJS*, kao i tehnike i metode koje se mogu primeniti za njihovu realizaciju. Pored opisa kako se napad izvršava prikazane su i tehnike uz pomoć kojih se korisnici mogu odbraniti ili preventirati sam napad.

4.1. Typosquatting napad

Typosquatting napad radi na principu tako što napadač za ime paketa odabere nešto što veoma liči na ime već nekog poznatog paketa i postavi ga na NPM. Oslanja se na principe socijalnog inženjeringa i na to da korisnik neće izvršiti proveru paketa pre njegovog instaliranja, već će se osloniti na svoje sećanje da dobije ime paketa. *Typosquatting* napad se takođe oslanja na to da korisnik napravi grešku u kucanju prilikom instaliranja paketa gde može greškom da instalira maliciozni paket koji je namerno tako nazvan [11].

Jedan od popularnijih primera ovog napada se desio na paket *crossenv* [12]. Ovo ime paketa je slično sa *cross-env* koji je predstavljao metu napada. Maliciozni paket je obuhvatio sve funkcionalnosti originalnog paketa, tako da se razlika u korišćenju nije mogla ni primetiti. Jedina razlika je bila ta što je maliciozni paket slao podatke o varijablama okruženja na udaljeni server tj. napadaču.

Neke od mitigacija koje se mogu iskoristiti da bi se zaštitili su korišćenje *—ignore-scripts* argumenta komandne linije za sprečavanje izvršavanja proizvoljnih komandi, poverljive informacije skladištiti van varijabli okruženja i koristiti alate poput *npq*-a [13] za skeniranje zavisnosti pre instalacije.

4.2. Supply chain i Dependency confusion napad

Paketi koji se koriste kao zavisnosti se nalaze u centralnom registru koji može biti javni i privatni. Za krajnje korisnike taj registar predstavlja *supply chain*. *Supply chain attack* je napad gde napadač pokušava da kompromituje izvorni kod paketa sa ciljem da ubaci maliciozni kod direktno u mesto gde se taj paket koristi ili indirektno kroz stablo zavisnosti.

Istraživač bezbednosti i etički haker Alex Birsan je napravio prototip napada na *supply chain* pod imenom *dependency confusion* tako što je uspeo da ubaci maliciozni kod u centralne repozitorijume kompanija poput *Apple*-a, *Microsoft*-a, *Uber*-a i *Yelp*-a. [14]

Dependency confusion napad za cilj ima da napravi zabunu prilikom instaliranja stabla zavisnosti na mašinama gde se projekat izvršava tako što se oslanja na greške da osoblje firme neće dobro podesiti mašinu da preuzima pakete sa privatnog registra. U takvom slučaju će se, podrazumevano, dobiti maliciozni paket koji nosi isti naziv i koji je okačen na javni registar.

Mitigacije koje se mogu primeniti su korišćenje imenskog prostora (eng. *scoped namespace*) na NPM-u, kreirati ispravne konfiguracije, isključiti *install* i *uninstall* komande radi sprečavanja izvršavanja proizvoljnih skripti, dodatno istražiti pakete, sprečiti izloženost ranjivih informacija itd.

4.3. Backdoor napad

Backdoor je vrsta napada gde korisnici mogu da zaobiđu bezbednosne mere i dobiju pristup resursima kojima ne bi smeli da pristupaju na računaru, mreži ili aplikaciji, a u nekim slučajevima i pristup celim uređajima [15].

Svaki *backdoor* se sastoji iz dva dela: sam maliciozni kod koji se izvršava na zaraženoj mašini i komunikacioni kanal koji omogućava napadaču da ima pristup mašini. U *NodeJS*-u se može napraviti jednostavan *backdoor* korišćenjem ugrađenog modula *child_process* i funkcije *exec* koja omogućava izvršavanje proizvoljnih komandi na mašini i *ExpressJS* biblioteke za pravljenje HTTP servera.

Načini na koje možemo izbeći *backdoor* napade su da koristimo poznate i dobro održavane pakete i da koristimo alate poput *npq* i *snyk* koji nas mogu upozoriti ako će instalacija paketa ubaciti neki maliciozni kod.

4.4. Denial of Service napad

Denial of Service (skraćeno - DoS) napadi za cilj imaju da onemoguće određeni servis na neki vremenski period ili trajno tako što će na njega slati veliku količinu zahteva i na taj način mu preopteretiti dostupne resurse ili naneti veliku novčanu štetu zbog upotrebe tih resursa.

Neki od principa zaštite su da se koriste asinhronne metode gde god je to moguće, da se ne čuva previše podataka u memoriji u jednom trenutku i da se ranjive metode, koje mogu da prouzrokuju DoS napad, dodatno zaštite [16].

4.5. Prototype pollution napad

Po definiciji *prototype pollution* napad je *denial of service* i *injection* napad gde napadač dobija kontrolu nad vrednostima *property*-a nekog objekta. Ovo omogućava napadaču da kontroliše samu biznis logiku aplikacije. Posledice ovog napada mogu biti nedostupnost servera, a u najgorem slučaju može da dođe i do *remote code execution* napada [17].

Da bismo bolje razumeli zašto je ovaj napad efektivan potrebno je objasniti kako prototipovi u *JavaScript*-u rade. Prototipovi predstavljaju mehanizam uz pomoć kojih *JavaScript* objekti mogu da nasleđuju osobine drugih objekata. Svaki *JavaScript* objekat u sebi ima

ugrađeno jedno polje pod nazivom `__proto__` koje sadrži polja „roditeljskog“ objekta. Prilikom pristupanja polju objekta pretraga se prvo vrši u datom objektu, ako polje nije pronađeno u njemu pretraga se nastavlja u prototipu, zatim u prototipu prototipa i tako dalje dok se ne dođe do korenskog prototipa. Za izvršavanje ovog napada bitna je informacija da je *JavaScript* dinamički jezik što znači da je tokom izvršavanja programa moguće vršiti izmene nad prototipovima svih objekata [18].

U sklopu istraživanja za ovaj rad napravljena je demonstracija koja prikazuje kako može da se napadne *web server* koji koristi *ExpressJS* biblioteku u pozadini. Prikazana su dva načina napada. Prvi način koristi propust u funkciji *merge* iz popularne *lodash* [19] biblioteke i dovodi do *denial of service*-a. Treba napomenuti da je ovaj propust popravljen u novijim verzijama biblioteke, ali može biti dostupan u drugim bibliotekama koje implementiraju isti algoritam jer ovakva funkcija predstavlja najčešći način za realizaciju *prototype pollution*-a. Pored spajanja običnih atributa ukoliko ova funkcija nije ispravno implementirana izvršiće i spajanje prototipova. Ukoliko se u aplikaciji radi *merge* operacija nad objektom koji predstavlja korisnički unos i proizvoljnog objekta iz aplikacije može doći do napada na server. Na listingu 4.1 prikazan je pojednostavljeni *endpoint* i primer malicioznog podatka koji se šalje na njega.

```
// Primer ExpressJS endpoint-a sa POST metodom
app.post("/", (req, res) => {
  merge({}, req.body)
  res.send("OK")
})

// Maliciozni korisnički ulaz
const payload = {
  "__proto__": {
    "toString": 123,
    "valueOf": "It works"
  }
}
```

Listing 4.1 *ExpressJS endpoint* sa POST metodom i maliciozni korisnički ulaz

Pošto se metode *toString* i *valueOf* često implicitno koriste prilikom izvršavanja koda njihovo ponašanje je sada izmenjeno i samim tim dolazi do nedostupnosti servera zbog učestalih grešaka na najjednostavnijim *JavaScript* operacijama.

Druga vrsta napada za rezultat ima izvršavanje proizvoljnih komandi na napadnutom računaru. Do njegovog izvršavanja može doći ukoliko se negde u kodu nepažljivo koristi *child_process* modul i *exec* funkcija. Primer ovakvog koda se može videti na listingu 4.2 gde se dinamički, bez prethodnih provera, izvršavaju proizvoljne komande. Takođe je prikazan i maliciozni korisnički ulaz koji uz prethodno korišćenje *merge* funkcije može da doda malicioznu komandu na listu komandi za izvršavanje.

```
// Primer ExpressJS endpoint-a sa POST metodom
app.post("/", (req, res) => {
  const commands = [
    script1: "ls",
    script2: "touch test.js"
  ]
})
```

```

}

for (const key in commands) {
  childProcess.execSync(commands[key], { stdio: "inherit" })
}

res.send("OK")
})

// Maliciozni korisnički ulaz
const payload = {
  "__proto__": {
    "my command": "echo You are attacked!"
  }
}

```

Listing 4.2 Izvršavanje komandi korišćenjem `exec` funkcije iz `child_process` modula i maliciozni korisnički ulaz

Neki od načina odbrane su da se koriste bezbedne verzije `merge` metoda, da se objekti kreiraju bez prototipa sa `Object.create(null)` metodom i korišćenje `Object.freeze()` metode koja sprečava menjanje atributa objekata.

5. ZAKLJUČAK

Prilikom kreiranja aplikacija bezbednost treba da se shvati ozbiljno i da se primenjuju odgovarajuće tehnike zaštite jer u suprotnom posledice napada mogu biti ozbiljne.

Da bi napadač mogao da osmisli novi napad, a i autor aplikacije da se zaštiti od potencijalnih napada bitno je da se razume interni rad tehnologije u kojoj je aplikacija realizovana. U ovom radu je opisano kako *NodeJS* interno funkcioniše i opisane su njegove slabe tačke kroz primere napada.

Fokus je bio na opisivanju nekih od poznatih napada koji su se desili u prethodnim godinama na NPM ekosistem i sam *NodeJS* kao platformu za izvršavanje *JavaScript* koda na serveru. Pored samih napada, dati su i mogući načini zaštite od napada sa kojima bi svaki autor *web* aplikacija trebao biti upoznat. Pored toga, prikazane su i ideje koje NPM kao organizacija može da primeni za bolju zaštitu.

6. LITERATURA

- [1] Node.js [Na mreži] [Citirano 30 9 2022.] <https://en.wikipedia.org/wiki/Node.js>
- [2] NPM [Na mreži] [Citirano 30 9 2022.] <https://www.npmjs.com/>
- [3] V8 [Na mreži] [Citirano 30 9 2022.] <https://v8.dev/>
- [4] Libuv [Na mreži] [Citirano 30 9 2022.] <https://libuv.org/>
- [5] Serdar Yegulalp, *How one yanked JavaScript package wreaked havoc* [Na mreži] [Citirano 30 9 2022.] <https://www.infoworld.com/article/3047177/how-one-yanked-javascript-package-wreaked-havoc.html>
- [6] Markus Zimmermann, Cristian-Alexandru Staicu, *Small World with High Risks: A Study of Security Threats in the npm Ecosystem*. 2019
- [7] NPM Audit [Na mreži] [Citirano 30 9 2022.] <https://docs.npmjs.com/cli/v8/commands/npm-audit>

[8] Danny Grander, *Malicious code found in npm package event-stream downloaded 8 million times in the past 2.5 months* [Na mreži] [Citirano 30 9 2022.] <https://snyk.io/blog/malicious-code-found-in-npm-package-event-stream/>,

[9] Liran Tal, Assaf Ben Josef, *Open source maintainer pulls the plug on npm packages colors and faker, now what?* [Na mreži] [Citirano 30 9 2022.] <https://snyk.io/blog/open-source-npm-packages-colors-faker/>

[10] Liran Tal, *Alert: peacenotwar module sabotages npm developers in the node-ipc package to protest the invasion of Ukraine* [Na mreži] [Citirano 30 9 2022.] <https://snyk.io/blog/peacenotwar-malicious-npm-node-ipc-package-vulnerability/>

[11] Liran Tal, *What is typosquatting and how typosquatting attacks are responsible for malicious modules in npm* [Na mreži] [Citirano 30 9 2022.] <https://snyk.io/blog/typosquatting-attacks/>

[12] Snyk – crossenv [Na mreži] [Citirano 30 9 2022.] <https://security.snyk.io/package/npm/crossenv>

[13] Github – npq [Na mreži] [Citirano 30 9 2022.] <https://github.com/lirantal/npq>,

[14] Alex Birsan, *Dependency Confusion: How I Hacked Into Apple, Microsoft and Dozens of Other Companies* [Na mreži] [Citirano 30 9 2022.] <https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610>

[15] Ulises Gascón, *What is a backdoor? Let's build one with Node.js* [Na mreži] [Citirano 30 9 2022.] <https://snyk.io/blog/what-is-a-backdoor/>

[16] Karl Döüna, *Secure Your Node.js Web Application, Keep Attackers Out and Users Happy*

[17] Prototype pollution [Na mreži] [Citirano 30 9 2022.] <https://learn.snyk.io/lessons/prototype-pollution/javascript/#pgwwpvrcchiwtb>

[18] Object prototypes [Na mreži] [Citirano 30 9 2022.] https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object_prototypes

[19] Lodash [Na mreži] [Citirano 30 9 2022.] <https://www.npmjs.com/package/lodash>

Kratka biografija:



Boris Šuličenko rođen je 22.8.1997. godine u Novom Sadu, Republika Srbija. Godine 2016. upisao je Fakultet tehničkih nauka u Novom Sadu smer Softversko inženjerstvo i informacione tehnologije. 2020. godine je diplomirao na osnovnim akademskim studijama i iste godine upisuje master akademske studije na istoimenom smeru.

PRILOG KREIRANJU OPTIMALNOG PLANA SADNJE KUKURUZA UPOTREBOM METODE DINAMIČKOG PROGRAMIRANJA

AN ENCLOSURE TO THE CREATION OF THE OPTIMAL PLANTING PLAN OF CORN USING THE DYNAMIC PROGRAMMING METHOD

Milica Cvjetićanin, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – Razvijen je model za kreiranje optimalnog plana sadnje kukuruza na osnovu fizički mogućih opsega sadnje date biljke, tako da nedeljni prinos rezultuje maksimalnom popunjenošću raspoloživih skladišta. Problem je rešen primenom metode dinamičkog programiranja nazvane Problem ranca. Pomenutom metodom izvršena je minimizacija gubitaka useva na godišnjem nivou.

Ključne reči: Teorija optimalnog upravljanja, Dinamičko programiranje, Optimizacija, Problem ranca

Abstract – Based on physical range of corn planting, an optimal corn planting model is being developed, so that the weekly yield results in the maximum occupancy of the available warehouses. The problem was solved using Dynamic Programming method called Knapsack Problem. With this method, yearly crop losses were minimized.

Keywords: Optimal control theory, Dynamic Programming, Optimization, Knapsack problem

1. UVOD

Dinamičko programiranje u kombinaciji sa analizom podataka predstavlja jedan od najmoćnijih alata za optimizaciju problema različitog tipa. Pomenuta kombinacija svoju primenu sve više nalazi i u poljoprivredi s obzirom na to da su klimatske promene rapidne, a konvencionalne tehnike reagovanja ne pružaju mogućnost za vršenje modelovanja na tako preciznom nivou i ujedno iziskuju mnogo više napora i vremena.

Cilj ovog rada jeste omogućavanje proizvodnje kvalitetnijih useva kukuruza i doprinos postizanju održivih lanaca ishrane kao posledice kreiranja planske sadnje ove biljke.

Ideja je da se korišćenjem algoritma problema ranca, tehnike dinamičkog programiranja, kreira optimalan plan sadnje različitih sorti kukuruza uzimajući u obzir vremenske opsege u toku godine kad isti moraju biti posađeni. Na ovaj način postiže se minimizacija biljnih otpada, maksimalna iskorišćenost datog zemljišta, kao i maksimalan prinos useva, čime se pruža doprinos umanjenju uticaja krize gladi u svetu.

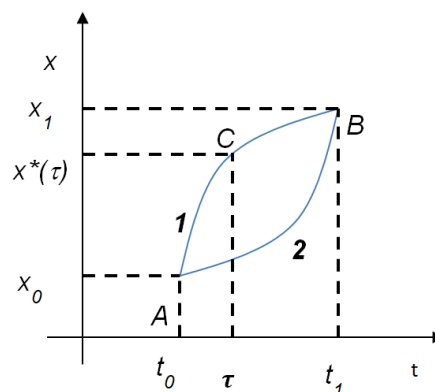
NAPOMENA:

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

2. OSNOVNI TEORIJSKI POJMOVI

2.1. Dinamičko programiranje

Dinamičko programiranje [1] je optimizaciona metoda koja predstavlja mehanizam za rešavanje optimalnih upravljačkih problema. Konceptualnu ideju dinamičkog programiranja prvi je uveo R. Belman. Belmanova teorija se temelji na poznatom postulatu: “Optimalno upravljanje ima svojstvo da, bez obzira na početno stanje i početnu odluku, preostale donete odluke moraju činiti optimalnu politiku u odnosu na stanje koje proizilazi iz prve odluke” [2]. Fizički smisao ove definicije mogao bi se definisati zaključkom da je svaki deo optimalne putanje takođe optimalan. Rešavanjem Hamilton-Jakobi-Belmanove (HJB) parcijalne diferencijalne jednačine dobija se optimalno upravljanje u zatvorenoj povratnoj sprezi [3]. Problem optimalnog upravljanja odnosi se na pronalaženje upravljačkog signala $u^*(t, x)$ koji sistem definisan modelom $\dot{x} = f(t, x, u)$ vodi po trajektoriji $x^*(t) \in X$ takvoj da kriterijum optimalnosti $I = \int_{t_0}^{t_1} F(t, x, u) dt$ ima minimalnu/maksimalnu vrednost. Takođe poznat je početni uslov $x(t_0) = x_0$.



Slika 1. Grafički prikaz putanje AB

Ukoliko se putanja obeleženu brojem 2 (Slika 1) definiše kao optimalna, može se zapisati u sledećem obliku:

$$S(t_0, x_0) = \int_{t_0}^{t_1} F(t, x_{opt}(t), u_{opt}(t)) dt \quad (1)$$

Sa druge strane putanja 1 definisana je tako da deo putanje od tačke A do tačke C nije optimalan, dok je drugi deo putanje, od tačke C do tačke B optimalan. Shodno tome putanja obeležena brojem 1 modelovana je kao:

$$\int_{t_0}^{t_1} F(t, x, u) dt + S[\tau, x^*(\tau)], \quad (2)$$

gde važi da je $t_0 \leq t \leq \tau$.

Prateći osnovnu ideju Belman-ove teorije, a to je da svaki deo optimalne putanje mora biti optimalan, kao i činjenicu da postoji isključivo jedna optimalna putanja, na osnovu (1) i (2) uvodi se:

$$S(t_0, x_0) = \min \left\{ \int_{t_0}^{\tau} F(t, x, u) dt + S(\tau, x^*(\tau)) \right\} \quad (3)$$

Uzimajući u obzir da su vremenski trenuci t_0 i τ veoma bliski i razvojem izraza $S(\tau, x^*(\tau))$ u Tejlorov red dobija se opšti oblik HJB jednačine:

$$\frac{\partial S(t, x)}{\partial t} + \min \left\{ F(t, x, u) + \sum_{i=1}^n \frac{\partial S(t, x)}{\partial x_i} f_i(t, x, u) \right\} = 0 \quad (4)$$

Prethodno opisani postupak jasno ukazuje na osnovnu ideju dinamičkog programiranja na kojoj se bazira čitava Belmanova teorija optimalnog upravljanja, a to je da se složen problem rešava razlaganjem na skup potproblema, čijom se optimizacijom i potom kombinovanjem formira jedinstveno optimizaciono rešenje početnog složenog problema.

Osnovna ideja primene dinamičkog programiranja sastoji se iz četiri koraka. Prvo je potrebno odrediti strukturu optimalnog rešenja, zatim je potrebno rekursivno definisati optimalno rešenje, potom izračunati vrednosti optimalnog rešenja i na kraju konstruisati optimalno rešenje. Rekurzija se može sprovoditi unapred ili unazad, dok se vrednost optimalnog rešenja može računati tehnikom odozgo na dole ili tehnikom odozdo na gore. Nezavisno od izabrane tehnike, svaki potproblem se rešava samo jednom, a jedina razlika je u tome na koji način se rešenje potproblema čuva i u kom momentu se ponovo koristi.

Postoji veliki broj algoritama koji se zasnivaju na dinamičkom programiranju. U okviru ovog rada korišćena je jedna od primena, takozvani Problem ranca (*eng. Knapsack problem*). It tog razloga će ovom algoritmu biti nešto više reči u nastavku.

2.2. Problem ranca

Problem ranca [4] predstavlja algoritam koji pomaže u rešavanju problema kombinatorike. Od svih varijacija ovog algoritma u ovom radu je, nakon niza testova, kao najpogodnija odabrana je primena 0-1 algoritma ranca i to koristeći pristup koji se oslanja na dinamičko programiranje. Metoda 0-1 bazira se na tome da objekat može biti ili izabran u celosti pri čemu dobija vrednost 1 ili ne biti izabran uopšte i dobiti vrednost 0 [5].

Definicija algoritma: Ukoliko postoji skup od n objekata obeleženih redom $x_1, \dots, x_n$, gde svaki objekat i ima težinu w_i i vrednost v_i , pri tom uzimajući u obzir da je maksimalni kapacitet W , treba odrediti:

$$\max \left\{ \sum_{i=1}^n v_i x_i \right\}, \text{ tako da } \sum_{i=1}^n w_i x_i \leq W \text{ i da } x_i \in \{0, 1\} \quad (5)$$

Algoritam problema ranca može se sprovesti prateći četiri koraka implementacije dinamičkog programiranja, gde je ključni korak rekurzija. Primenjuje se tako što se isključivo jednom rešeni potproblemi pamte u vidu

tabele. Na taj način postiže se ponovno korišćenje pomenutih međurešenja i daleko bolja efikasnost [5].

Tabela se može zamisliti kao matrica $V[0, \dots, n, 0, \dots, W]$, pri čemu važi da je $1 \leq i \leq n$ i $0 \leq w \leq W$. Početni korak popunjavanja tabele primenjuje se na sledeći način: $V[0, w] = 0$, za $0 \leq w \leq W$, nema selektovanih objekata $V[i, w] = -\infty$, za $w < 0$, nemoguće.

Rekursivni korak primenjuje se koristeći sledeću formulu:

$$V[i, w] = \max\{V[i-1, w], v_i + V[i-1, w-w_i]\} \quad (6)$$

Prilikom popunjavanja tabele uvek se koristi prethodni red za popunjavanje narednog.

Polje od najvećeg interesa u tabeli je $V[n, W]$, jer ono sadrži maksimalnu moguću vrednost dobijenu izborom svih n objekata sa ograničenjem težine W .

- Ukoliko važi izraz $V[n, W] = V[n-1, W]$ onda objekat n nije selektovan koristi se $V[n-1, W]$
- Ukoliko važi izraz $V[n, W] \neq V[n-1, W]$ onda se zaključuje da optimalna selekcija uzima u obzir objekat n i koristi se $V[n-1, W-w_n]$

Prethodno opisani korak potrebno je sprovoditi sve dok se opet ne dođe do početnog reda čime je lista odabranih objekata kompletirana.

Asimptotsko vreme izvršavanja ove metode je $O(n * W)$, gde je n broj objekata, a W maksimalni kapacitet ranca.

3. MATERIJALI I METODE

3.1. Baza podataka

Početni skupovi podataka korišćeni u ovom radu dobijeni su u saradnji sa kompanijom SciO [6]. U cilju prilagođenja rešavanju ovog problema date skupove podataka bilo je potrebno transformisati vršenjem pojedinih operacija. Takođe, svaki od skupova posmatra dešavanja na dve lokacije: *lokacija 0* ($W = 7000$ klipova kukuruza) i *lokacija 1* ($W = 6000$ klipova kukuruza).

1. DATASET #1: Obuhvata ulazne promenljive za model optimizacije. Sadrži identifikator sorte tj. populaciju, opseg u kom je moguća sadnja, broj jedinica stepeni rasta neophodnih za sazrevanje biljke (GDUs) u °C, kao i prinos nakon žetve.
2. DATASET #2: Sadrži broj jedinica °C koje se akumuliraju svaki dan na obe lokacije u periodu od 01.01.2009. godine do 31.12.2019. godine.
3. DATASET #3: Koristi se za prikaz rezultata modela i u njega je naknadno za svaku biljku potrebno uneti optimalan opseg sadnje. Nakon što se ovaj skup podataka popuni predstavljaće plan sadnje.

Iz razloga što je svako od skladišta potrebno popuniti maksimalnim brojem klipova svake nedelje, glavni cilj jeste doći do informacije u kojim nedeljama u godini je posmatranu biljku moguće ubrati kako bi se potom od svih mogućih biljaka za tu nedelju napravila kompozicija onih koje se najbolje uklapaju. Ideja realizacije u radu jeste kreiranje matrice čije će kolone biti nedelje u toku godine, a vrste popis svih populacija. Matrica treba da daje informaciju o opsegu nedelja u kom je moguće ubrati svaku od populacija tako što će se u tim nedeljama nalaziti vrednosti prinosa date biljke, dok će u svim ostalim biti napisana nula. Ovaj raspored žetve kreira se tako što se od

graničnih vrednosti opsega dozvoljene sadnje vrši sumiranje predviđenih toplotnih jedinica za buduću godinu (dobijeni usrednjavanjem dnevnog toplotnog prinosa za svaki isti datum iz dataset 2) koje je moguće da biljka apsorbira svaki dan sve dok suma ne bude jednaka GDUs faktoru zrenja date biljke. Sve transformacije skupova podataka vršene su u okviru alata MS Excel.

3.2. Implementacija algoritma

Implementacija algoritma realizovana je u okviru Visual Studio Code softverskog okruženja korišćenjem Python programskog jezika.

Nakon dobijenih mogućih opsega žetve svake biljke, data matrica je importovana u okruženje. Glavna ideja realizacije temelji se na primeni prilagođenog algoritma problema ranca, pri čemu su u svakoj mogućoj nedelji izabrane vrste tako da je apsolutni gubitak minimizovan.

Inicijalno je u radu razvijena jedna verzija centralnog dela koda za obe lokacije. Uveden je jedinstven skup vrednosti svake biljke. Početno, vrste su sortirane u rastućem redosledu u odnosu na broj nedelja kad je njihova žetva moguća. Ovo je izvršeno usled činjenice da ukoliko postoji više vrsta sa istim prinosom i ukoliko problem ranca izabere taj prinos kao pogodan, izabraće se ona populacija na koju algoritam prvu naiđe. Imajući to u vidu, usled izbegavanja gubitka nekih populacija, bolje je izabrati onu, koja u budućim nedeljama ima manje prostora za odabir. Takođe, implementiran je inicijalni odabir onih populacija kojima je posmatrana poslednja nedelja kad mogu biti izabrane. Kao posledica toga maksimalan kapacitet skladišta koji se prosleđuje problemu ranca umanjen je za sumu prinosa ovako izabranih populacija.

Nakon što je u okviru problema ranca dobijena lista najpogodnijih populacija u svakoj nedelji, izvršena je konkatencija listi tako što je ovoj listi dodata lista inicijalno odabranih populacija za tu nedelju. Zatim je kreirana izlazna matrica tako što je u nedelji u kojoj je populacija izabrana upisan njen prinos žetve, dok je u svim ostalim nedeljama u godini za ovu populaciju upisana vrednost 0.

Od ovako dobijenog plana žetve, plan sadnje je kreiran sumiranjem vrednosti dnevnih toplotnih prinosa od date nedelje unazad sve dok se ne dostigne vrednost GDUs faktora date populacije. Ovakvo sumiranje izvršeno je od prvog i poslednjeg dana posmatrane nedelje u godini, čime je dobijen opseg najpogodnije sadnje koji je upisan u izlazni dataset 3 (Slika 2).

Population	Planting dates	
p_67	27-Jan	31-Jan
p_68	27-Jan	31-Jan
p_61	27-Jan	31-Jan
p_59	20-Jan	23-Jan

Slika 2. Plan sadnje

Naknadno su u cilju poboljšanja rezultata izvršene određene modifikacije u kreiranju plana žetve. Modifikacije obuhvataju redukciju maksimalnog kapaciteta skladišta, promenu nedelje početka žetve kao i preraspodelu plana žetve u nedeljama sa velikim gubicima i to koristeći ponovljenu simulaciju, preraspodelu od nazad ka napred i/ili metodu ugnježdenog problema ranca.

4. REZULTATI

Kvantitativna mera pogodnosti rezultata predstavlja apsolutni gubitak broja klipova. Ukoliko je maksimalan kapacitet skladišta W , broj nedelji žetve n , a vrednost dobijena sumiranjem prinosa u svih n nedelja žetve obeležena sa c , onda je ukupan gubitak broja klipova:

$$l = |W * n - c| \quad (7)$$

Uzimajući u obzir prosečnu masu jednog klipa kukuruza od 102g [7], ukupan gubitak u kilogramima iznosi:

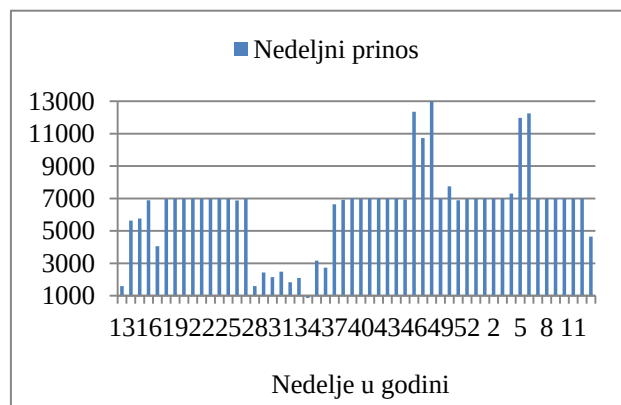
$$lm = \frac{102 * l}{1000} \quad (8)$$

4.1 Lokacija 0

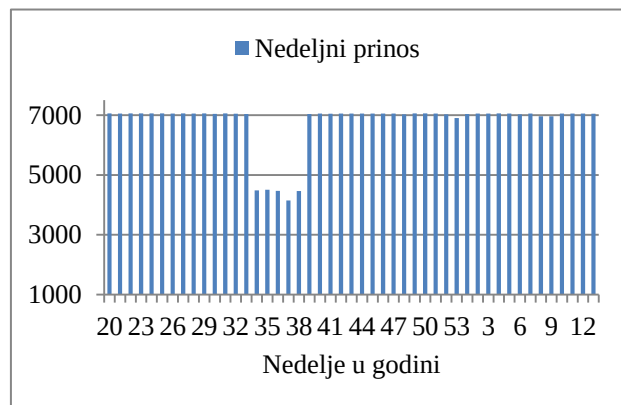
Prilikom primene osnovnog algoritma, bez poboljšanja, uočen je veoma veliki gubitak od nešto više od 3.5t na godišnjem nivou (Slika 3).

S druge strane, primenom algoritma sa svim navedenim poboljšanjima ukupan godišnji gubitak evidentiran je kao vrednost nešto veća od 1t (Slika 4).

Relativno velika vrednost gubitka evidentirana je usled nebalansiranosti fizički dozvoljenih opsega sadnje biljke što se posebno uočava u nedovoljnim prinosima žetve u nedeljama 34-40. Međutim, primenom poboljšanja gubitak je smanjen više nego trostruko.



Slika 3. Nedeljni prinos, lokacija 0, bez poboljšanja



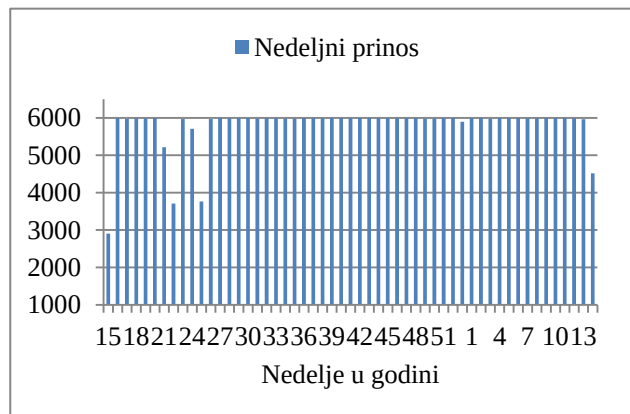
Slika 4. Nedeljni prinos, lokacija 0, sa poboljšanjima

4.1 Lokacija 1

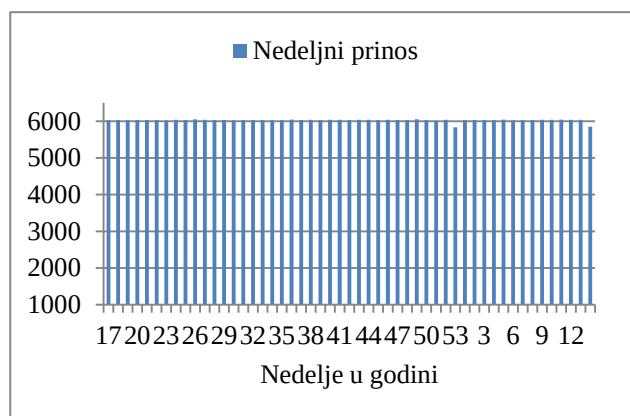
Prilikom primene osnovnog algoritma, bez poboljšanja, uočen je manji inicijalni gubitak nego na lokaciji 0 i u ovom slučaju iznosi nešto manje od 1t (Slika 5).

S druge strane, primenom algoritma sa određenim navedenim poboljšanjima ukupan godišnji gubitak evidentiran je kao vrednost od svega 136,5kg (Slika 6).

Može se evidentirati da je algoritam sa poboljšanjima dao izvanredne rezultate.



Slika 5. Nedeljni prinos, lokacija 1, bez poboljšanja



Slika 6. Nedeljni prinos, lokacija 1, sa poboljšanjima

5. ZAKLJUČAK

Globalna kriza hrane je stanje koje već vekovima predstavlja jedan od najvećih globalnih problema.

Na osnovu gore prezentovanog rada mogu se izvesti sledeći zaključci:

- U ovom radu izvršeno je pravljenje modela sposobnog da kreira optimalan plan sadnje biljaka na nekoj lokaciji, tako da su gubici minimalni.
- Kreiranjem optimalnog plana sadnje na najbolji mogući način koriste se vremenski uslovi koji su svakoj biljci potrebni za zrenje. Žetvom biljke onda kada ona dostigne potpunu zrelost maksimizuje se prinos, dok se potpunim popunjavanjem skladišta izbegavaju novčani gubici. Dodatno, pražnjenjem skladišta svakih 7 dana izbegava se gubitak mase klipova kukuruza koji je posledica skladištenja u dužem vremenskom periodu.
- Imajući u vidu da bi primenom efikasnijeg plana sadnje rasprostranjenost pojave gladi širom sveta bila umanjena, može se reći da je ovim pristupom obezbeđen doprinos eliminisanju, ili bar umanjenu efekta krize hrane.

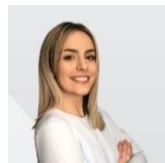
U cilju unapređenja rada predlaže se adaptacija velikih gubitaka u pojedinim nedeljama na lokaciji 0 kreiranjem plana rentiranja preostalog prostora, čime se novčani gubici eliminišu.

Takođe, može se i unaprediti računanje prediktovanih dnevnih toplotnih prinosa uvođenjem faktora zaboravljanja.

6. LITERATURA

- [1] R. Bellman, Dynamic programming, New Jersey: Princeton University, 1957.
- [2] J. Réveillac, Optimization tools for logistics, Elsevier, 2015.
- [3] M. a. J. Z. Rapaić, Projektovanje linearnih regulatora i estimatora u prostoru stanja, Novi Sad: Univerzitet u Novom Sadu, Fakultet tehničkih nauka, 2014.
- [4] P. a. G. R. Gilmore, The theory and computation of knapsack functions, 1966.
- [5] "0/1 Knapsack problem," 6 August 2022. [Online]. Available: <https://www.guru99.com/knapsack-problem-dynamic-programming.html>. [Accessed 22 September 2022].
- [6] "SCiO Analytics," [Online]. Available: <https://scio.systems/>. [Accessed 2022].
- [7] S. D. G. H. Kimenju, "Crop loss in % of weight of stored maize," 2022.

Kratka biografija:



Milica Cvjetičanin rođena je u Novom Sadu 1998. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Mehatronika –robotika i automatizacija odbranila je 2021. godine. kontakt: milica.manmm@gmail.com

МИКРОФРОНТЕНД АПЛИКАЦИЈЕ**MICROFRONTEND APPLICATIONS**

Никола Мандић, Факултет техничких наука, Нови Сад

Област – РАЧУНАРСТВО И АУТОМАТИКА

Кратак садржај – У овом раду представљен је развој једне веб апликације, за чију имплементацију се користе концепти микрофронтенд архитектуре. Рад прати комплетан развој једне такве апликације, од постављања структуре и конфигурисања овакве архитектуре, пратећих проблема до континуираног достављања и испоруке једне такве апликације.

Кључне речи: Веб апликација, React, Микрофронтенд

Abstract – This paper presents development of one web application, which implementation is based on concepts of microfrontend architecture. The paper follows complete development of application. At development of application is meant to build a structure and configuration of application architecture, solving issues with configurations. Continuous integration and delivery for application are covered in this paper.

Keywords: Web application, React, Microfrontend

1. УВОД

Развојем интернет технологија долази до све већег развоја писања софтвера као веб апликација. Најчешћи модел писања представља клијент сервер модел. Веб апликација представља систем који је направљен користећи клијент-сервер архитектуру [1]. Клијент-сервер модел је значајно смањио развој развој апликације тако што долази до раздвајања функционалности на клијентску и серверску страну [2]. У почетку и клијентска и серверска страна писане су као монолитне апликације. Повећањем пословних захтева система и све тежег одржавања серверске стране модела долази до појаве микросервисне архитектуре. Микросервисна архитектура подразумева рашчлањивање монолитне апликације на више мањих апликација зарад лакшег одржавања и даљег развоја апликација. По узору на микросервисне апликације јавља се идеја да се такав приступ примени и на клијентску страну, која је у почетку формирана као монолитна апликација.

2. МИКРОФРОНТЕНД АПЛИКАЦИЈЕ

Претходних година током развоја веб апликација, започето је коришћење микросервисне архитектуре на серверској страни.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Александар Купусинац, ред. проф.

Микросервисна архитектура представља још један стил архитектуре софтверских система у коме се велике сложене апликације компонују склапањем појединачних сервиса [3].

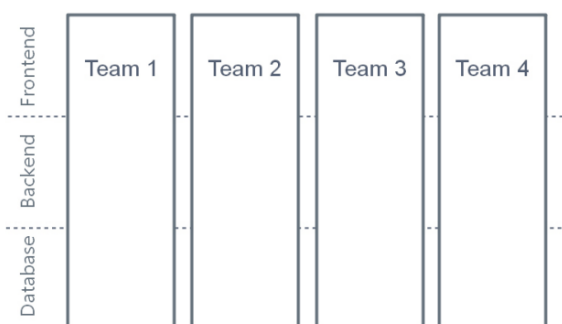
Критеријуми које сервиси задовољавају су [3, 4]:

- Лаки за тестирање и одржавање.
- Слабо спрегнути.
- Сваки сервис обавља један задатак, неки део пословне логике система.
- Могу бити независно *deployovani*.
- Сваки микросервис је могуће развијати у програмском језику који је погодан захтевима.
- Не морају бити свесни имплементације осталих сервиса.
- Комуникација се обавља програмским интерфејсима *API-ijima* независним од програмског језика (нпр. *Representational State Transfer (REST)*)

Микрофронтенди наслеђују концепте микросервиса на страни веб претраживача. Под овим се подразумева да се апликација на клијентској страни раздваја на више мањих апликација [5]. Свака од тих апликација може да се покреће, развија и има независан *deployment* [5]. Бенефите које је донела микросервисна архитектура, попут модуларности, оптимизације, лакшег раста и развоја апликације доводе до развоја идеје да се исти такав концепт примени и на клијентској страни. Идеја на коју се ослањају микрофронтенди је та да се веб апликација третира као комбинација функционалности везаних за различит тим, то подразумева да сваки тим има задужење да развија одређен сет повезаних пословних функционалности апликације [5]. Тимови су унитарно функционални и развијају целокупну функционалност, а под тим се подразумева да се развија клијентски и серверски део апликације.

Концепти и идеје на којима се базирају микрофронтенди [1, 5]:

- Сваки тим има слободан избор технологије коју користи.
- Код сваког тима мора бити изолован, подразумева да се не користе глобалне варијабле.
- Коришћење префикса код варијабли да се зна власништво варијабле да се избегну конфликти.
- Коришћење емитовања догађаја веб претраживача зарад комуникације између апликација



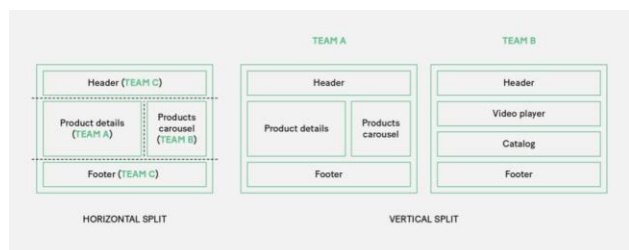
Слика 1. Организација тимова са микрофронтенд архитектуром [5]

На слици 1. представљен је начин поделе тимова унутар организације као последица коришћења микрофронтенд архитектуре, сваки тим је задужен за одређен домен система. Под тим се подразумева да тим развија и клијентску и серверску страну у складу са доменом који им је додељен.

Важно је нагласити да овакав приступ у архитектури клијентске апликације отвара потенцијалне проблеме који требају да се реше. Неки од њих су [1]:

- Оркестрација – како решити учитавање апликације само када је потребна.
- Рутирање – како поставити логику рутирања и одлучити коју апликацију учитати.
- Изолација – како избећи колизије у окружењу.
- Комуникација – на који начин апликације треба да комуницирају једна између друге.
- Конзистентан кориснички интерфејс – обезбедити да једна апликација не утиче на стилове друге апликације.
- Менаџмент билбблиотека – како избећи да се једна библиотека учитава више пута.

Микрофронтенд апликације се могу организовати на два начина хоризонтално и вертикално (слика 2). Под хоризонталном организацијом подразумева се да може више апликација бити присутно на једној страници унутар веб претраживача, док под вертикалном организацијом једна апликација се извршава по страници [6]. Једна од напомена приликом коришћења хоризонталног приступа је нарушавање приципа да би свака апликација требала имати задужење на основу одређене пословне логике. Вертикална организација је веома слична начину на који функционишу традиционалне SPA (Single Page Application), при чему сваки тим води рачуна о специфичним деловима апликације [6].

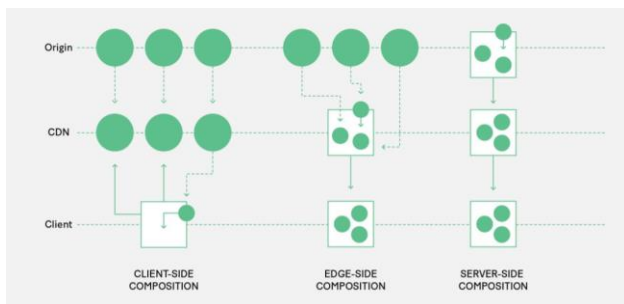


Слика 2. Хоризонтална и вертикална организација микрофронтенда [6]

Постоје различити приступи имплементације решења микрофронтенд апликације (слика 3) [1]:

- Композиција на серверској страни – приликом генерисања страница на серверу, врши се оркестрација неопходних апликација. Проблем овог приступа је тај што се губе могућности SPA (Single Page Application), јер се при сваком захтеву са клијентске стране морају генерисати странице.
- Build-time интеграција – подразумева да се свака апликација генерише као независан JavaScript пакет и интегрише се током завршне компилације модула. Главна мана овог приступа је та што сви микрофронтенди морају бити рекомпајлирани ако се било који од њих измени.
- Run-time интеграција помоћу iframe-a - унутар iframe тага се добија изолација апликација. Мана овог приступа је немогућност дељења истих библиотека између апликација, па као последица тога апликација заузима више простора унутар веб претраживача. Такође једини начин комуникације представља API од iframe што аутоматски смањује флексибилност на пољу комуникације између апликација.
- Run-time интеграција помоћу JavaScript-a – представља једно од најоптималнијих решења. При овом решењу се подразумева да се свака апликација прави као одвојен модул и да се учитава једино уколико је потребно. Сваки модул се развија независно што омогућава тимовима да развијају функционалности независно од осталих. Са овим приступом омогућено је коришћење заједничких библиотека између апликација.
- Run-time интеграција помоћу веб компоненти - представља најновији приступ који постаје могућ због нових HTML стандарда. По овим стандардима дозвољава се креирање свог HTML елемента, дефинисање понашања тог елемента и динамичног учитавања кода тог елемента. Елемент је по питању изолације на истом нивоу као и iframe таг. Предност овог приступа је у комуникацији између апликација што су у овом случају веб компоненте. Проблем код овог приступа је његова старост, јер су стандарди на ком се базира нови и велики број веб претраживача није прилагођен раду по тим стандардима. Постоји начин да се и овај проблем заобиђе, коришћењем polyfill-a, који представљају библиотеке које трансформишу у прилагодљив код старијим веб претраживачима. Додатан проблем је што се самим тим повећава количина садржаја сервирана веб претраживачу.

Постоји још и приступ дистрибуције апликација на веб серверу, где се различита апликација учитава у зависности од руте којој се приступа. Мана овог приступа је да се сваки пут страница мора освежити када се приступа новој рути на веб претраживачу.



Слика 3. Различити приступи имплементације микрофронтенд апликација [6]

3. КЛАУД КОМПЈУТИНГ

Постоји доста дефиниција за клауд компјутинг, једна од најчешће коришћених је та да клауд компјутинг представља групу дистрибуираних компјутера који пружају сервисе и ресурсе кроз интернет [2].

Постоје три врсте сервиса које нуди клауд компјутинг:

- *IaaS (Infrastructure as a Service)*, кориснику је пружена могућност коришћења рачунарске инфраструктуре у виду виртуелне платформе.
- *PaaS (Platform as a Service)*, са овом опцијом корисник не треба да води више рачуна попут хардвера, оперативног система. Омогућава кориснику потпуни развој апликације.
- *SaaS (Software as a Service)*, под овом тачком пружа се опција коришћења готових софтверских решења који су одржавани од стране пружаоца услуге.

Развој клауд компјутинга допринео је у многоме развоју скалабилности серверских страна веб апликација, убрзава се процес развоја софтвера избацивањем неких раније неопходних ствари попут сервера на физичким локацијама и осталог. Предности клауд компјутинга преставља лаган начин избора технологија и услуга у зависности од проблема који се решава. Поред наведеног, врло лако у зависности од оптерећења система се може скалирати и велика уштеда се јавља по погледу трошкова (нису потребни више физички присутни центри података и др.).

4. ИМПЛЕМЕНТАЦИЈА

Систем представља апликација за преглед и додавање знаменитости. Корисници имају могућност да прегледају споменике на мапи и да позиционирају мапу у односу на споменик, такође имају могућност додавања нових споменика у систем.

Архитектура овог система се базира на клијент сервер моделу. При чему клијентска страна је имплементирана по концептима микрофронтенд архитектуре што представља тему овог рада. Серверска страна система имплементирана је у *Node.js* платформи са коришћењем *Express* радног оквира погодног за развијање бекенд дела веб апликација. База коришћења за чување података је *MongoDB*, која спада у ред *NoSQL* база података.

Клијентски део састоји се од три апликације које чине заједно микрофронтенд архитектуру. Једна од аплика-

ција задужена је за приказ споменика на мапи, друга апликација служи за функционалности додавања споменика и преглед листе свих споменика. Намена треће апликације служи зарад интегрисања горе наведених апликација и њиховог функционисања и пружања одговарајуће логике рутирања и приказа интегрисаних апликација на основу руте. Трећа апликација се назива и контејнер апликација.

```

1  const devConfig = {
2    mode: 'development',
3    output: {
4      publicPath: 'http://localhost:8082/',
5    },
6    devServer: {
7      port: 8082,
8      historyApiFallback: {
9        index: '/index.html',
10      },
11    },
12    plugins: [
13      new ModuleFederationPlugin({
14        name: 'monuments',
15        filename: 'remoteEntry.js',
16        exposes: {
17          './MonumentsApp': './src/bootstrap',
18        },
19        shared: packageJson.dependencies,
20      }),
21      new HtmlWebpackPlugin({
22        template: './public/index.html',
23      }),
24    ],
25  };
26
27  module.exports = merge(commonConfig, devConfig);

```

Слика 4. Конфигурација *Webpack-a*

Конфигурација *Webpack-a* (слика 4) користи се унутар апликација зарад њихове комуникације са осталим апликацијама и формирања извршивог кода разномљивог широког спектру веб претраживача. На основу те конфигурације свака апликација има могућност да се покреће у изолацији са чим је постигнут ефекат да сваки тим може независно у односу на цео систем развијати део пословне логике везане за свој домен. Поред тога све апликације су подешене да раде заједно једна са другом. Поље *name* и *filename* представљају локацију и идентификатор на основу ког контејнер апликација долази до неопходних доступних модула *remote* апликације. *ModuleFederationPlugin* представља функционалност *Webpack-a* која омогућава да се формира микрофронтенд апликација и постизање комуникације између различитих апликација и њиховог извршавања. Унутар конфигурације приказане на слици 4. поље *exposes* служи да се прикажу делови апликације који су доступни контејнер апликацији за коришћење. То значи када се унутар контејнера захтева приступ апликацији споменика тај захтев ће се мапирати на *bootstrap* датотеку унутар апликације за споменике.

Приступ који се користи за имплементацију клијентске стране представља интеграција у *Run-time* помоћу *JavaScript-a*. Приликом имплементације ових апликација коришћена је *React* библиотека за исцртавање корисничког интерфејса.

Током развоја апликације посвећена је пажња на аспект континуиране интеграције и честог достављања апликације кориснику. За подешавање *CI/CD pipeline* коришћене су *Github Actions*. Конфигурацијом ових акција постиже се резултат да на основу сваког догађаја приликом промене кода унутар апликација окида се акција у којој се ивршавају одређени кораци. Неки од корака су креирање *build* верзије апликације и синхронизација нових датотека са неким од клауд провајдера. За клауд провајдера коришћени су Амазонови веб сервис и широк спектар њихових функционалности за веб апликације. Ова тачка представља један од услова формирања архитектуре где промене унутар једне од апликације које чине клијента ће бити увек освежене и неће утицати на рад осталих апликација.

5. ЗАКЉУЧАК

Овај рад представља креирање, упознавање са процесима и концептима микрофронтенд апликација на клијентској страни веб апликација. Објашњени су приступи начина имплементације оваквих апликација. Решење је имплементирано у виду *JavaScript Run-time* приступа. Предности приступа имплементiranог унутар решења су:

- Изолован развој апликација које чине систем.
- Промене настале приликом развоја једне од апликација не захтевају рекомпајлирање свих осталих апликација.
- Све апликације се интегришу у реалном времену.
- Могућност коришћења различитих технологија при имплементацији апликације, интеграција је одрађена на генеричан начин.

Приликом имплементације приказане су неопходне технологије за реализацију апликације. У моменту писања рада, микрофронтенд апликације спадају у новије приступе креирања веб апликација и по мишљењу аутора побољшање апликације и генерално целог приступа представља проналазак што генеричнијег начина комуникације између апликација и напредније конфигурације *Webpack-a* за постизање што оптимизованијих апликација.

6. ЛИТЕРАТУРА

- [1] Pavlenko, Andrey, et al. "Micro-frontends: application of microservices to web front-ends." J. Internet Serv. Inf. Secur. 10.2 (2020): 49-66.
- [2] Oluwatosin, Haroon Shakirat. "Client-server model." IOSR Journal of Computer Engineering 16.1 (2014): 67-71.
- [3] Мирослав Зарић, *Микросервиси*, <https://enastava.ftninformatika.com/courses/257/files/folder/predavanja?preview=112844>, Факултет техничких наука, Нови Сад, (Приступљено: 09.2022.)
- [4] What are microservices ?
Доступно: <https://microservices.io/>
- [5] Yang, Caifang, Chuanchang Liu, and Zhiyuan Su. "Research and application of micro frontends." IOP conference series: materials science and engineering. Vol. 490. No. 6. IOP Publishing, 2019.
- [6] Micro-frontends in context, Luca Mezzalana,
Доступно: <https://increment.com/frontend/micro-frontends-in-context/>

Кратка биографија:



Никола Мандић рођен је у Вуковару 1996. год. Основне академске студије завршио је 2019. године на Факултету техничких наука у Новом Саду. Мастер рад на Факултету техничких наука из области Рачунарство и аутоматика – Електронско пословање одбранио је 2022. године.

TESTNO OKRUŽENJE ZA RUKOVANJE METEOROLOŠKIM PODACIMA POMOĆU SPARQL UPITA**TEST ENVIRONMENT FOR HANDLING METEOROLOGICAL DATA USING SPARQL QUERIES**Stefan Stojaković, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj - U radu je prikazano proširenje SPARQL Playground aplikacije koje omogućuje rad sa geoprostornim podacima, kao i njihovo iscrtaavanje. Dizajniran je i implementiran WeatherService koji je testiran na primeru podataka o vremenskim prilikama, pri grafičkom prikazu opasnosti od nevremena.

Ključne reči: RDF, SPARQL, GeoSPARQL, WeatherService

Abstract: The paper presents an extension of the SPARQL Playground application that supports work with weather geospatial data. It also provides support for geospatial data drawing. WeatherService was designed, implemented and tested on the example of weather data, and hazard graphical representation.

Keywords: RDF, SPARQL, GeoSPARQL, WeatherService

1. UVOD

Aplikacija SPARQL Playground nudi mogućnosti učenja SPARQL jezika kroz direktan rad sa SPARQL upitima. Kao takva predstavlja veoma koristan alat za studente i razvojne inženjere i predstavlja dobru bazu za dodatna proširenja koja se oslanjaju na SPARQL jezik. Jedno takvo proširenje prikazano je u ovom radu.

Originalna SPARQL Playground aplikacija ne podržava rad sa geoprostornim podacima, a takođe ne podržava ni rad sa podacima o vremenskim opasnostima, kao ni grafičku simulaciju istih, te je ideja rada da se podrže nedostajuće funkcionalnosti.

Implementiran je WeatherService za rad sa meteorološkim podacima o potencijalnim opasnostima od nevremena, koji su u TTL formatu, kao i simulator servisa, kako bi se grafički prikazao uticaj opasnosti na postojeću opremu. Obezbeđeni su podaci u TTL formatu za potrebe simulacije kretanja vremenskih nepogoda.

Novonastala SPARQL Playground aplikacija osim osnovnih funkcija ima i mogućnost rada sa geoprostornim podacima, te je bilo potrebno postojeći model podataka elektrodistributivnog izvoda proširiti mogućnošću prikaza geoprostornih podataka.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Gavrić, docent.

2. TEORIJSKE OSNOVE**2.1 SPARQL**

SPARQL Protocol and RDF Query Language (SPARQL) je standardni jezik za upite i protokol za povezane podatke na web-u i Resource Description Framework (RDF) baze podataka [1]. SPARQL omogućava korisniku da čita podatke iz baze podataka ili iz nekog drugog izvora podataka koji se može mapirati na RDF graf. Upiti SPARQL-a se sastoje od skupa trojke gde svaki element (subjekat, predikat ili objekat) može biti promenljiva. Promenljive se mapiraju na trojke u bazama podataka. SPARQL pruža i niz funkcija za konstruisanje složenih upita, za navođenje dodatnih uslova za filtriranje i za formatiranje konačnog rezultata [3].

2.2 GeoSPARQL

GeoSPARQL je definisan kao OGC (Open Geospatial Consortium) standard i predstavlja proširenje SPARQL jezika mogućnostima upita nad geoprostornim podacima [2]. Standard podržava predstavljanje i ispitivanje geoprostornih podataka na semantičkom web-u. GeoSPARQL definiše i rečnik za predstavljanje geoprostornih podataka u RDF-u. Pored toga GeoSPARQL je dizajniran za podršku sistema zasnovanih na kvalitativnom prostornom zaključivanju i sistema zasnovanih na kvantitativnim prostornim proračunima [3].

2.2.1 Klase

GeoSPARQL opisuju dve glavne klase:

geo:SpatialObject i **geo:Feature**.

Klasa **geo:SpatialObject** predstavlja sve ono što može imati prostornu predstavu, dok **geo:Feature** predstavlja opšte karakteristike objekata.

GeoSPARQL podržava određen broj topoloških prostornih relacija. Topološke relacije definisane u familiji *Simple Features* koje su korišćene u radu kao rezultat vraćaju *true* ili *false*:

- 1. equals** – **geo:sfEquals** – u odnosu na jednakost dva geometrijska objekta;
- 2. disjoint** – **geo:sfDisjoint** – u odnosu na razdvojenost dva geometrijska objekta;
- 3. intersects** – **geo:sfIntersects** – u zavisnosti da li dva geometrijska objekta imaju presek;

4. **touches** – **geo:sfTouches** – u zavisnosti da li se dva geometrijska objekta dodiruju;

5. **within** – **geo:sfWithin** – u zavisnosti da li je jedan geometrijski objekat unutar granica drugog;

6. **contains** – **geo:sfContains** – u zavisnosti da li jedan geometrijski objekat sadrži drugi;

7. **overlaps** – **geo:sfOverlaps** – u zavisnosti da li jedan geometrijski objekat preklapa sa drugim;

8. **crosses** – **geo:sfCrosses** – u zavisnosti da li jedan geometrijski objekat preseca drugi.

2.3 RDF4J

Eclipse RDF4J je Java okruženje koje se koristi za rad nad *RDF* podacima. Uključuje kreiranje, parsiranje, skladištenje i analizu *RDF* upita. Nudi *API (Application Programming Interface)* koji je jednostavan za upotrebu i koji se može povezati sa svim vodećim rešenjima za *RDF* baze podataka i *RDF* repozitorijume. Omogućava kreiranje aplikacija koje koriste povezane podatke (*linked data*) i semantički *web*. Podržava dva skript jezika: *SPARQL* i *SeRQL* [4].

3. TEHNOLOGIJE I ALATI

3.1 Enterprise Architect i Visual Studio Code

Sparx Systems Enterprise Architect je korišćen za vizuelno modeliranje i dizajn zasnovan na *UML* jeziku [3, 5], dok je *Visual Studio Code* korišćen kao editor izvornog koda.

3.2 SPARQL Playground

SPARQL Playground je samostalna *web* aplikacija sa podrškom za rad na više platformi, koja se koristi za učenje *SPARQL*-a i *RDF* upitnog jezika [6]. Implementirana je u *Java Sprint Boot* i *AngularJS*. Verzija koja je korišćena se oslanja na *Sesame 2.8.6 framework*.

4. DIZAJN REŠENJA

4.1 Opis rešenja

Postojeća aplikacija *SPARQL Playground*, koja predstavlja osnovu ovog rada, ne ispunjava sve potrebne uslove za dostizanje cilja - rada sa geoprostornim i vremenskim podacima. Ona koristi *OpenRDF Sesame framework* kao *engine* koji ima mogućnost izvršavanja *SPARQL* upita, ali ne i *GeoSPARQL* upita. Tu nastaje potreba za migracijom *OpenRDF Sesame* na *RDF4J*, koji podržava rad sa geoprostornim podacima. Koristeći podatke date u *RDF* trojkama, kao i *RDF4J* i *Leaflet* tehnologije, predloženo rešenje implementira okruženje koje, koristeći *SPARQL* upitni jezik proširen sa *GeoSPARQL* upitima, prikazuje poligone koji predstavljaju mreže distributivnih izvoda, preseke linija izvoda, nadolazeće promene vremenskih uslova koje će uticati na rad opreme u polju, itd.

Rešenje koristi *Leaflet* biblioteku kako bi se prikazale pozicije na kojima su uz pomoć *SPARQL* upita iscertani izvodi, tačke preseka određenih izvoda, poligoni različitih promena vremenskih uslova (oluja, grom, požar, itd.), kao i koji distributivni izvodi mogu biti pogođeni tim promenama.

4.2 Mogućnosti predloženog rešenja

Urađena je migracija *Sesame* na *RDF4J* kako bi aplikacija imala mogućnost rada sa *GeoSPARQL* podacima i *GeoSPARQL* upitima. Radom sa *GeoSPARQL* podacima podržani su različiti upiti u vidu iscrtavanja izvoda, izvoda koji nisu pod nepogodama, kao i izvoda koji jesu pod nepogodama, te iscrtavanje samih opasnosti u vidu pravougaonika.

Koristeći *EnterpriseArhitect* alat koji se oslanja na rad sa *UML* jezikom, proširen je postojeći *UML* dijagram sa klasama za rad sa geoprostornim podacima. Takođe je kreiran i *CIM* profil koji definiše klase i njihova svojstva koje se koriste u aplikaciji.

Kako bi se omogućio rad sa poligonima vremenskih opasnosti, kao i simulacija istih, razvijeni su *WeatherService* i *GeoSparql* simulator za *WeatherService* koji pomeraju površine koje predstavljaju poligone opasnosti za definisani vremenski interval.

Upotrebom *RDF TTL* serijalizacije u vidu trojki (subjekat, predikat, objekat), uz oslonac na klase definisane *CIM* standardom, modelovani su geoprostorni i vremenski podaci koji su korišćeni za testiranje. Takođe je podržano i učitavanje *RDF* fajlova u okviru *SparqlPlayground* aplikacije kako bi se testirao rad *WeatherService*-a i *GeoSparql* simulatora u samoj aplikaciji.

5. IMPLEMENTACIJA

5.1 Proširenja potrebna za rad sa geoprostornim podacima

Definisane su dodatne klase i dijagram koji bi podržao rad sa klasama geoprostornih podataka. Sam klasni dijagram sa sobom donosi klase koje će služiti prilikom predstavljanja grafičkih podataka.

To su klase *CoordinateSystem*, *Location* i klasa *PositionPoint* koja u sebi sadrži informacije o koordinatama *x*, *y*, *z*.

5.2 Proširivanje CIMXML IEEE 37 Node Test Feeder modela sa geoprostornim podacima

S obzirom da *CIMXML IEEE 37 Node Test Feeder* model koji je korišćen u rešenju ne sadrži geoprostorne podatke, isti je proširen klasama koje omogućavaju rad sa geoprostornim podacima.

Dodate su klase *Location* i *PositionPoint* koje omogućavaju rad sa geoprostornim podacima. Klasa *Location* sadrži referencu na klasu *PowerSystemResource* preko koje je povezana sa klasama modela *IEEE 37 Node Test Feeder*. Takođe klasa *Location* sadrži referencu i na novododatu klasu *PositionPoint*. Klasa *PositionPoint* sadrži *xPosition* i *yPosition* atribut koja predstavljaju koordinate *x* i *y*. Na taj način bilo koji element može biti predstavljen tačkom u koordinatnom sistemu sa odgovarajućim *x* i *y* koordinatama.

5.3 Učitavanje RDF fajlova u aplikaciji SPARQL Playground

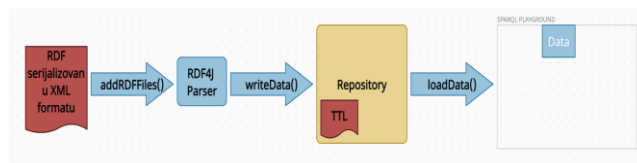
SPARQLPlayground aplikacija nudi mogućnost učitavanja podataka u vidu *RDF*-a, tj. neke od

serijalizacija RDF-a (konkretno TTL). Na samoj aplikaciji postoji tab sa imenom *Data* u koji se mogu smestiti podaci sa kojima se želi raditi.

Podaci se smeštaju u vidu trojki (subjekat-predikat-objekat), kako i nalaže sam RDF. Koristeći znanje iz prethodno kreiranog RDFS-a (*RDF Schema*), mogu se definisati trojke u aplikaciji *SPARQLPlayground*. Ugledajući se na podatke definisane u RDFS-u, na sličan način se mogu definisati podaci unutar *SparqlPlayground* aplikacije.

Prilikom pokretanja aplikacije iz *VisualStudioCode*-a poziva se metoda *addRDFFiles()* koja učitava podatke iz *rdf-data* foldera (koji su serijalizovani u XML format). Zatim se poziva i metoda *writeData()* koja na osnovu podataka koji su učitani iz *rdf-data* foldera, nakon parsiranja RDF podataka u TTL format uz pomoć predefinisanih *RDF4J* metoda, učitava podatke u *in-memory* repozitorijum. Proširenje u kodu predstavlja upravo *addRDFFiles()* metoda koja dozvoljava učitavanje podataka u RDF formatu.

Nakon učitavanja aplikacije *SPARQL Playground*, na predefinisanoj lokaciji, npr. <http://localhost:8080> sa *frontend*-a se poziva metoda *loadData()* uz pomoć koje se učitavaju podaci u *Data* prozor iz *in memory* repozitorijuma. Dijagram toka učitavanja podataka prikazan je na Slici 1..



Slika 1. Dijagram toka učitavanja podataka

5.4 Izrada WeatherService simulatora

WeatherService omogućava učitavanje podataka o opasnostima koji su definisani u formatu TTL, kao i simulaciju samog pomeraja opasnosti kroz definisan interval.

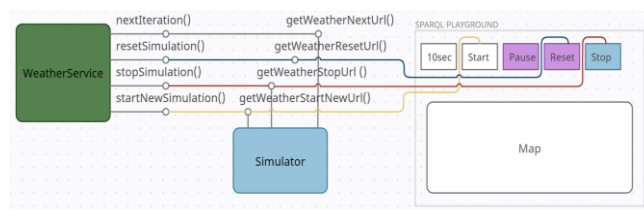
Servis ima sledeće opcije: *startNewSimulation*, gde se inicijalizuje model koji sadrži informacije o trojkama koje učestvuju u simulaciji, zatim *addNewStatements* koja služi za dodavanje novih trojki u repozitorijum kako bi *frontend* mogao da pročita nove vrednosti i, na kraju, *removeOldStatements*, gde se sa repozitorijuma brišu stare vrednosti trojki pre dodavanja novih.

Za potrebe simulacije, servis sadrži metode: *nextIteration*, gde se učitava sledeći fajl koji sadrži trojke koje su pomerene za određenu veličinu po y osi, zatim *resetSimulation*, koja služi kako bi simulacija ponovo počela i *stopSimulation*, gde se brišu svi izrazi iz repozitorijuma i radi se njegovo praznjenje.

GeoSparql simulator koji radi merenje vremena sadrži i metode *start*, *stop*, *reset* i *pause*, koje kontrolišu rad merenja vremena i rad *WeatherService*-a. Implementirane su i metode *getWeatherStartNewUrl*, *getWeatherNextUrl*, *getWeatherStopUrl* i *getWeatherResetUrl*, koje kontrolišu pozivanje metoda sa *WeatherService*-a. Metode korišćene pri simulaciji *WeatherService*-a prikazane su na slici 2.

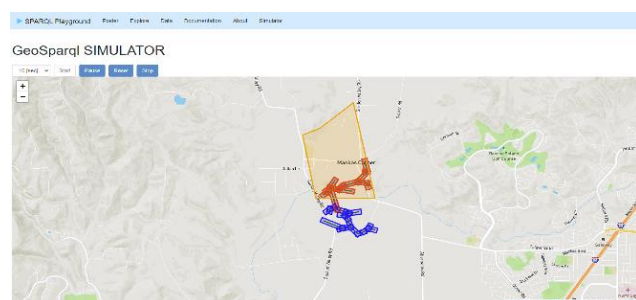
5.5 Izrada weather podataka

Do sada su definisani podaci za izvode, a definisan je i *WeatherService* koji će raditi sa podacima o opasnostima, kao i simulatori koji će omogućiti simulaciju rada sa podacima.



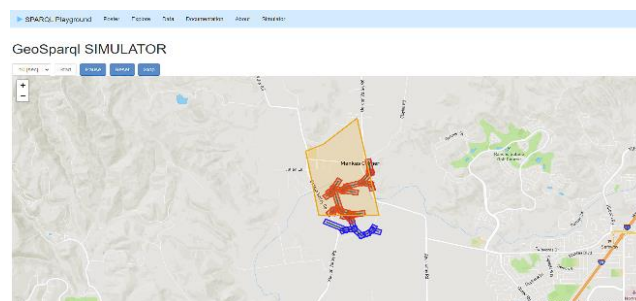
Slika 2. Metode korišćene pri simulaciji WeatherService-a

U ovom slučaju, za potrebe testiranja kreirana su 4 tipa dokumenta sa TTL podacima o opasnostima koji se kroz *WeatherService* smenjuju i tako daju utisak o pomeraju same opasnosti.



Slika 3. Slika sa početka simulacije

Nakon 10 sekundi, opasnost sa Slike 3. menja svoj položaj i na Slici 4. može se videti trenutno stanje.



Slika 4. Slika nakon 10 sekundi simulacije

5.6 Izrada alata za replikaciju CIMXML modela sa izmenjenim geoprostornim podacima

CIMXMLDuplicator je konzolna C# aplikacija koja vrši generisanje novih CIMXML datoteka na osnovu postojećih. Rad aplikacije se ogleda u nekoliko koraka. Pre svega se kopira postojeća datoteka, prođe se kroz sve koordinate i one se pomeraju za definisani *offset*. Generišu se i nove *mRID* i *Id* vrednosti, kako ne bi dolazilo do dupliranja podataka. O samom dizajnu aplikacije, kao i načinu rada, više informacija se može naći u radu [7].

5.7 Iscrtavanje elemenata zadatih u WKT formatu uz upotrebu Leaflet JavaScript biblioteke

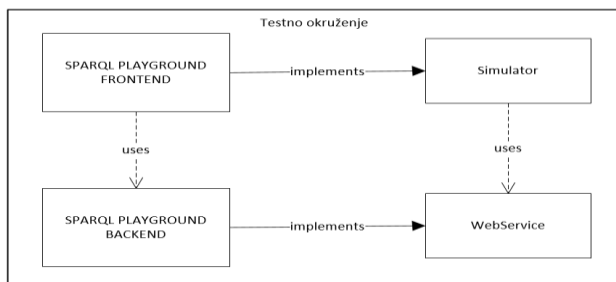
Izvode, preseke izvoda, opasnosti i sam pomeraj opasnosti potrebno je na neki način prikazati na mapama. Za to je korišćena biblioteka *Leaflet*. *Leaflet* podržava

različite funkcije mapiranja, te se podaci u WKT formatu lako prikazuju. *Well Known Text* (WKT) je OGC standard koji se koristi za predstavljanje prostornih podataka u tekstualnom formatu.

Više o samoj *Leaflet* biblioteci i njenom integriranju u rešenje više informacija se može naći u radu [7].

6. TESTNO OKRUŽENJE

Na **Slika 5.** prikazan je raspored komponenti testnog okruženja u rešenju koje je implementirano.



Slika 5. Raspored komponenti testnog okruženja

SPARQL Playground aplikacija se sastoji od SPARQL Playground frontend-a uz pomoć kojeg je prikazan izgled web aplikacije koji je dostupan korisniku (klijentska strana). SPARQL Playground frontend sadrži raspored, grafiku, navigaciju i prikaz osnovnog sadržaja aplikacije. Između ostalog sadrži (*implements* na **Slika 5.**) i Simulator WeatherService-a koji upravlja (*uses* na **Slika 5.**) radom WeatherService-a. Pored toga postoji i SPARQL Playground backend, koji je zadužen za serversku stranu (onu stranu koja nije dostupna krajnjem klijentu). SPARQL Playground backend se sastoji od koda aplikacije, upotrebe različitih framework-a i njihove zajedničke integracije. Deo backend-a aplikacije je (*implements* na **Slika 5.**) i sam WeatherService koji upravlja meteorološkim podacima.

SPARQL Playground frontend koristi (*uses* na **Slika 5.**) SPARQL Playground backend za rad aplikacije, servisa i isporuku podataka.

7. ZAKLJUČAK

Rešenje koje je prikazano donelo je mogućnost iscrtaavanja prostornih podataka, preseka distributivnih izvoda kao i vremenskih nepogoda na interaktivnoj mapi. Simulacija rada WeatherService-a daje osećaj pomeraja vremenskih nepogoda i njihovog uticaja na izvode u realnom vremenu.

Originalna SPARQL Playground aplikacija koja ne sadrži rešenja predstavljena u radu ima mogućnost rada sa SPARQL upitima, dok nova aplikacija sa razvijenim dopunama ima mogućnost rada sa GeoSPARQL upitima, mogućnost prikazivanja elemenata i simulaciju WeatherService-a.

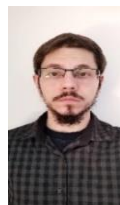
Eventualno unapređenje aplikacije bila bi integracija postojećeg WeatherService-a sa OpenWeather API-em kako bi se dobijali podaci vremenskih nepogoda i prognoza u realnom vremenu.

Potencijalno unapređenje bi takođe moglo da se ogleda kroz modelovanje tipova opasnosti u skladu sa CIM standardom.

8. LITERATURA

- [1] W3C, "SPARQL Query Language for RDF," [Online]. Available <https://www.w3.org/TR/rdf-sparql-query/>
- [2] Open Geospatial Consortium, "OGC GeoSPARQL - A Geographic Query Language for RDF Data" [Online], Available: <http://www.opengis.net/doc/IS/geosparql/1.0>, 2012
- [3] M. Gavrić, Standardi i modeliranje EES - beleške sa predavanja, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, 2019.
- [4] <https://rdf4j.org> (pristupljeno u martu 2022.)
- [5] Sparx Systems, "EnterpriseArchitect," <https://sparxsystems.com/products/ea/> (pristupljeno u martu 2022.).
- [6] SPARQL Playground, [Online] SIB Swiss Institute of Bioinformatics, sparql-playground.sib.swiss/help/doc/about (pristupljeno u martu 2022.).
- [7] P. Kaljević, Testno okruženje za grafički prikaz geoprostornih podataka na osnovu SPARQL upita, Novi Sad, 2021.

Kratka biografija:



Stefan Stojaković rođen je 01.09.1994. godine u Sremskoj Mitrovici. Osnovnu školu „Branko Radičević” u Šidu završio je 2009. godine. Završio je Gimnaziju „Sava Šumanović” u Šidu 2013. Fakultet tehničkih nauka je upisao 2014. godine, smer Elektroenergetski softverski inženjering. Godine 2019. završio je osnovne studije i upisao master akademske studije na Fakultetu tehničkih nauka, smer Primenjeno softversko inženjerstvo. Ispunio je sve obaveze i položio sve ispite predviđene studentskim programom.

APLIKACIJA ZA PRORAČUN ISPLATIVOSTI INVESTICIJE UGRADNJE SOLARNIH PANELA**APPLICATION FOR CALCULATING THE PROFITABILITY OF THE SOLAR PANELS INSTALLATION INVESTMENT**

Andrej Karavida, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PRIMENJENO SOFTVERSKO INŽENJERSTVO

Kratak sadržaj – U ovom radu predstavljena je aplikacija koja korisnicima pruža mogućnost da provere isplativost investicije ugradnje solarnih panela na određenoj lokaciji. Sistem pruža detaljnu kalkulaciju u odnosu na prosečnu mesečnu potrošnju električne energije uzimajući u obzir broj sunčanih sati na datoj lokaciji i cenu električne energije. Pored toga, omogućava praćenje proizvodnje solarnih panela u vremenskom intervalu od sat vremena. Implementirane su i funkcionalnosti autentifikacije pomoću Amazonovog Cognito servisa, prikaza svih solarnih panela na interaktivnoj mapi, pristup grafikonom istorijskih podataka, kao i stanje otplate u realnom vremenu.

Ključne reči: Solarni paneli, kalkulacija, investicija, Amazon, interaktivna mapa, grafikoni

Abstract – This paper describes and presents an application that can help users to check the profitability of making a solar panels investment on the specific location. An application provides a detailed calculation taking into consideration the electrical energy consumption and the number of sunny hours on that location. Moreover, it provides an ability to track the production by the one hour timeframe. Features that are also implemented include authentication using Amazon Cognito service, display of all solar panels on the interactive map, access to the history data charts as well as investment replayment status in real time.

Keywords: Solar panels, calculation, investment, Amazon, interactive map, charts

1. UVOD

Energija je svuda oko nas. Sve što nas okružuje zasnovano je na korišćenju neke vrste energije koja je potrebna ne samo nama ljudima, već svim živim bićima. Mašine dobijaju energiju sagorevanjem goriva (nafte, uglja, gasa). Međutim, na planeti Zemlji je sve manje iskoristive energije. Moderni stil života podrazumeva sve veću upotrebu energije, a trenutno se većina energetske potreba čovečanstva namiruje upotrebom vrlo štetnih fosilnih goriva. Njihov uticaj na životnu sredinu gotovo uvek je

negativan, od direktnih ekoloških katastrofa poput izlivanja nafte, kiselih kiša i radioaktivnih zračenja, do indirektnih posledica poput globalnog zagrevanja. Ovi lako iskoristivi oblici energije, po svojim kapacitetima predstavljaju veoma ograničene resurse, pa stoga postaju sve skuplji i teže dostupni. Činjenica da ovakav izvor energije u velikoj meri negativno utiče na životnu sredinu, u vrlo skoroj budućnosti bićemo prinuđeni da goriva zamenimo sa čistim izvorima energije – obnovljivim izvorima energije.

2. SOLARNA ENERGIJA

Solarna energija spada u obnovljive izvore energije, pod čijim pojmom se podrazumevaju svi izvori koji su prirodni i neiscrpni. Oni se nalaze svuda u prirodi, obnavljaju se u celosti ili delimično. Tehnologije koje se koriste za iskorišćavanje sunčeve energije mogu se podeliti u aktivne i pasivne. U aktivne tehnologije se svrstavaju solarni paneli i solarni kolektori, dok za orijentaciju objekta prilikom gradnje kažemo da spada u pasivne tehnologije [2]. Najjednostavniji način za iskorišćenje solarne energije je sakupljanje toplotne energije u vidu tople vode ili toplog vazduha, za čega se koriste solarni kolektori.

Ključni problem kod upotrebe solarne energije za grejanje objekata jeste što je najmanje ima zimi, kada je najpotrebnija, i obrnuto, najviše je ima leti kada su potrebe za toplotnom energijom najmanji. Savremeni sistemi za akumulaciju toplotne energije mogu uspešno da premoste ovaj izazov. Napredniji način jeste proizvodnja električne energije, za čiji proces su potrebni solarni paneli. Kako su solarni paneli napredovali, tako je i solarna energija postala pristupačna i dostupna gotovo svima. Opređenje za ugradnju solarnih panela donosi sa sobom mnoge benefite: ne zagađuju životnu sredinu, proces je veoma tih, omogućavaju visoku efikasnost čak i pri manje povoljnim vremenskim uslovima. Iako su inicijalni troškovi ugradnje visoki, jednom kada se solarni paneli instaliraju oni obezbeđuju energiju koja će isplatiti troškove tokom godina korišćenja iste. S obzirom na trajnu tendenciju porasta cene klasičnih energenata, iskorišćenje ovog najefikasnijeg obnovljivog izvora energije postaje apsolutni imperativ. Sunce nam svakoga sata pošalje dovoljnu količinu energije koliko celokupno stanovništvo Zemlje potroši u jednoj godini. Investiciju u ovakav izvor energije ne mogu poremetiti bilo kakva situacija u zemlji ili svetu, pa ni globalna ekonomska ili politička kriza.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Aleksandar Bošković.

2.1. Potencijal Republike Srbije

Na većem delu teritorije Republike Srbije broj časova sunčevog zračenja znatno je veći nego u mnogim evropskim zemljama (između 1500 i 2000 časova godišnje). Prosečan intenzitet sunčevog zračenja na teritoriji Republike Srbije se kreće od 1.1 kWh/m² po danu na severu do 1.7 kWh/m² po danu na jugu – tokom januara, a od 5.9 – 6.6 kWh/m² po danu – tokom jula.

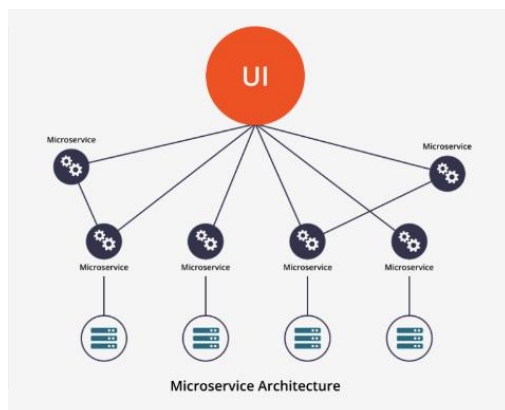
Tehnički iskoristiv energetski potencijal za konverziju energije Sunca u toplotnu energiju (za pripremu tople vode i druge namene) je procenjen na 0,194 miliona ten (tona ekvivalentne nafte) godišnje uz pretpostavku primene solarnih termalnih kolektora na 50% raspoloživih objekata u zemlji. Što se tiče proizvodnje električne energije, osnovno tehničko ograničenje, kao i u slučaju vetra, predstavlja mogućnost elektroenergetskog sistema da ovu energiju prihvati u letnjim mesecima, pošto je u pitanju varijabilna proizvodnja. Na osnovu trenutno raspoloživih kapaciteta elektroenergetskog sistema Republike Srbije za obezbeđenje tercijalne rezerve usvojeno je da je maksimalni tehnički iskoristiv kapacitet solarnih elektrana 450 MW [1].

Tehnički iskoristivi potencijal energije sunčevog zračenja za proizvodnju električne energije je promenljiva veličina koja će zavisiti od dinamike kojom se budu razvijale prenosna i distributivna mreža elektroenergetskog sistema Republike Srbije.

Izgradnja novih konvencionalnih elektroenergetskih kapaciteta (ugalj, prirodni gas, velike hidroelektrane), a posebno reverzibilnih hidroelektrana (RHE Bistrica i/ili Đerdap 3), značajno će povećati tehnički raspoloživ potencijal ovih intermitentnih izvora, zbog proširenja mogućnosti balansiranja snaga u sistemu [3]. Značajnu primenu solarnih elektrana dalo bi i zakonsko uređenje principa neto merenja za potrošnju električne energije za male potrošače

3.0 ARHITEKTURA SISTEMA

Softversko rešenje koje je tema ovog rada sastoji se od nekoliko servisa pisanih u različitim tehnologijama. Takav način dizajniranja predstavlja mikroservisnu arhitekturu.



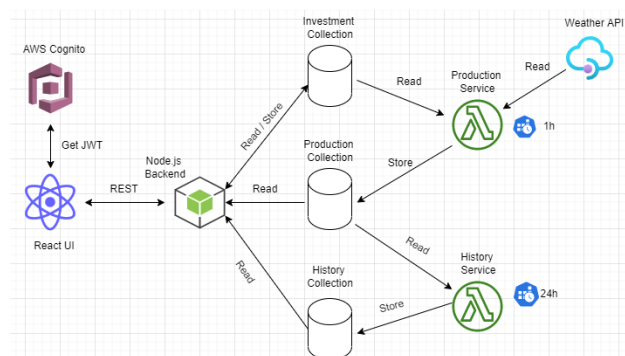
Slika 1 – Mikroservisna arhitektura

Mikroservisi omogućavaju da se aplikacija razdeli na mnoštvo manjih delova koji su mnogo lakši za razumevanje, održavanje i proširivanje, što sa sobom donosi mnogo drugih benefita kao što su slaba povezanost i visoka kohezija.

Slaba povezanost (eng. *loose coupling*) pruža mogućnost menjanja jednog servisa bez uticaja na drugi servis. Slabo povezani servisi ne znaju skoro ništa jedni o drugima iako ponekad neposredno sarađuju.

Sa druge strane, cilj visoke kohezije (eng. *high cohesion*) jeste da se grupiše logika aplikacije u povezane celine. Svaka celina sadrži logiku koja je usko vezana za njen kontekst, dok je sve ostalo u nekoj drugoj celini.

Ove celine se pretvaraju u servise. Ključne osobine mikroservisne arhitekture jesu: jedna funkcionalnost, decentralizacija, isporuka automatizovanim procesima, samostalan razvoj kao i nezavisna isporuka.



Slika 2 - Arhitektura sistema

4.0 KORIŠĆENE TEHNOLOGIJE I JEZICI

Za pisanje gore navedenih servisa, korišćeni su različiti alati, tehnologije i jezici. U nastavku sledi njihov prikaz.

4.1 JavaScript

JavaScript je objektni skriptni jezik čija je osnovna svrha korišćenja vezana za poboljšanje dinamike određene stranice. Kada se koristi samo *HTML* kod, tada je funkcionalnost web aplikacija veoma ograničena jer je njegova uloga samo da oblikuje određene elemente stranice (tekst, tabele, linkovi, forma...).

Kada se njemu pridoda *JavaScript*, tada se može kreirati ponašanje istih elemenata. Kombinacijom *HTML*-a, *CSS*-a i *JavaScript*-a nastaje takozvani dinamički *HTML* (*DHTML*). Kada se kaže da je neki skriptni jezik objektni, to zapravo znači da je tip podatka bitan.

Na samom početku korišćen je isključivo za pisanje aplikacija koji su deo web stranica i koje se izvršavaju u internet pregledaču. Međutim, *JavaScript* se danas koristi i za pisanje aplikacija za mobilne telefone, automatizaciju i sl. Postoji izreka koja prognozira da će svaka aplikacija koja može da bude napisana u *JavaScript*-u jednog dana i biti napisana u *JavaScript*-u.

4.2 React

React je *JavaScript* biblioteka koja omogućava opisivanje željenog korisničkog interfejsa (eng. *user interface*) jednostavnim i ponovno upotrebljivim komponentama koje se lako održavaju i testiraju. Komponente kao svoje elemente mogu sadržavati druge komponente i tako zajedno čine *React* aplikaciju. Zbog toga se upravo za komponente kaže da čine gradivne jedinice *React* aplikacije. *React* ima kontrolu nad *DOM* (*Domain Object Model*) elementima tako što pravi *JS* objekat za svaki *HTML* element. Taj *JS* objekat predstavlja element ili komponentu, koji u sebi mogu da sadrže druge elemente. Na taj način *React* pravi svoj virtualni *DOM* koji se kasnije koristi za upoređivanje i ažuriranje *HTML DOM*-a koji pregledač (eng. *browser*) koristi za renderovanje stranice. Zbog toga što *React* elementi opisuju *HTML* elemente, najprirodniji način za definisanje elemenata je *JSX* sintaksa (*JavaScript XHTML*).

4.3 Node.js

JavaScript je tokom godina svog postojanja skupio sve veći broj programera koji ga biraju kao prvi jezik pri izradi aplikacija. Relativno je lak za savladavanje, elegantan je i funkcionalan. Upravo iz tog razloga, mladi američki programer Rajan Dal (eng. *Ryan Dahl*) je 2009. godine kreirao *Node.js* – radno okruženje koje koristi program *V8*, *JavaScript* interpretator otvorenog koda iz *Google Chrome*-a i povezuje ga sa ulazno-izlaznim mogućnostima operativnog sistema na kojem je instaliran. U godinama koje su došle, *Node.js* je postao izuzetno popularan, pre svega kao serverski jezik (mogućnost pisanja *backend* aplikacija preko *JavaScript* sintakse), ali i kao svojevrsan ekosistem za kreiranje najraznovrsnijih desktop aplikacija, biblioteka, modula i drugih softverskih dodataka.

4.4 MongoDB

MongoDB je dokumentno orijentisana, otvorenog koda, multiplatformska *NoSQL* baza podataka koja nudi odlične performanse, veliko skladište podataka, sofisticirani jezik upita i automatsko skladištenje. Za čuvanje podataka koristi format sličan *JSON*-u. Spada u jednu od najpopularnijih baza podataka koja je pritom i besplatna. *MongoDB* dobio je ime od reči *humongous*, što znači veliko, enormno jer je cilj bila obrada velike količine podataka. *MongoDB* pruža velike mogućnosti za replikaciju podataka. Bezbednost je takođe na visokom nivou, a jedan od razloga leži u činjenici da *MongoDB* nije podložna *SQL* injektovanju (eng. *SQL injection*). Razlog je činjenica da se dokumenti skladište kao objekti, bez upotrebe *SQL* stringova.

4.5 AWS Cognito

Amazon Cognito je servis koji nudi usluge autentikacije i autorizacije korisnika za *web* i mobilne aplikacije. Korisnici se prijavljuju direktno koristeći korisničko ime i lozinku, ili putem trećeg lica (*Facebook*, *Google*, *Apple*...). Dve glavne komponente su *User Pools* i *Identity Pools*. *User Pools* nude mogućnost registracije i prijave na aplikacije, dok *Identity pools* nude mogućnost

dobijanja pristupa ostalim *AWS* servisima. Mogu se koristiti nezavisno, pojedinačno ili u kombinaciji.

5.0 DIZAN SISTEMA I FUNKCIONALNOST

Aplikacija za proračun isplativosti investicije ugradnje solarnih panela razlikuje tri tipa korisnika: gost, prijavljen korisnik i administrator sistema. Shodno tome, interfejs je podeljen u tri grupe, svaka namenjena jednom tipu korisnika. Na centralnom delu početne stranice nalazi se polje za unos grada za koji korisnik želi da izračuna solarni potencijal.

Aplikacija trenutno podržava samo gradove koji se nalaze u Republici Srbiji, dok će za nepostojeće gradove sistem izbaciti grešku sa odgovarajućom porukom. Stranica na koju će korisnik biti automatski prebačen predstavlja možda i najvažniju funkcionalnost sistema. Na njoj je prikazan celokupan proračun isplativosti ugradnje solarnih panela na unetoj lokaciji.

Korisnik može dodatno da prilagodi kalkulaciju i time dobije još preciznije rezultate. To radi tako što unosi prosečnu mesečnu potrošnju električne energije u kilovat-časovima. Za unetu potrošnju, sistem će proračunati i ispisati vrednost mesečnog računa za električnu energiju, po zvaničnom cenovniku Elektrodistribucije Srbije.

Osim toga, korisniku se pruža mogućnost da izabere tip solarnog panela od kojih želi da se investicija sastoji. Svaki solarni panel ima svoju snagu izraženu u vatima (W), kao i cenu izraženu u američkim dolarima (\$). Takođe, korisnik može modifikovati i broj sunčanih sati za grad za koji je investicija predviđena.

Gradovi u Republici Srbiji u proseku imaju oko 5 sunčanih sati dnevno (broj je znatno viši tokom letnjih meseci, odnosno znatno niži u zimskom periodu). Kada je i poslednji parametar unet, sistem će napraviti proračun tako da snaga investicije zadovolji mesečne potrebe korisnika. Biće izračunata snaga investicije, kao i broj panela koji je potrebno ugraditi. Na samom kraju, aplikacija ispisuje koliko bi uštedu korisnik ostvario ukoliko bi se odlučio da napravi ovakvu investiciju. Sistem će takođe pokazati koliko bi korisnik kao pojedinac doprineo očuvanju životne sredine.

Korisnik u svakom trenutku može da izvrši pregled svih napravljenih investicija. Za svaku investiciju prikazani su njeni detalji, istorija proizvodnje kao i upravo pomenuti uticaj na životnu sredinu. Što se detalja tiče, pored datuma kada je investicija napravljena i koliko je novca za nju izdvojeno, posvećena je pažnja i prikazu trenutne proizvodnje, kao i tome koliko je do sada investicija otplaćena, odnosno koliko novca je generisala. Kalendar daje korisniku mogućnost da odabere dane za koje želi da pogleda kako je tekla proizvodnja. To može da uradi odabirom početnog i krajnjeg datuma ili putem neke od predefinisanih prečica kao su: jučerašnji dan, trenutna nedelja, prošla nedelja, prošli mesec itd.

Grafik sa prikazom izabranog perioda će se pojaviti u modalnom dijalogu. Na samom grafiku, pored naslova koji pokazuje vremenski period za koji vrednosti odgovaraju, prokazane su minimalne, maksimalne i prosečne vrednosti proizvodnje (u kWh). Grafik ima

dodatne opcije kao što su pomeranje skale, zumiranje prikaza i sl.

Kako trenutno ulogovanog korisnika, tako i investicije svih ostalih korisnika u sistemu vidljive su na posebnoj stranici poređane na mapi Srbije. Svaka investicija prikazana je slikom solarnog panela, na čiji klik se pojavljuje tultip (eng. *tooltip*) koji sadrži ime korisnika, grad u kome je izgrađena, snagu, trenutnu proizvodnju i cenu. Na taj način korisnik može da vidi odnos između drugih korisnika, a svi *tooltip*-ovi su prikazani i na desnoj stranici u listi, obojeni zelenom bojom ukoliko pripadaju trenutno ulogovanom korisniku.

Da bi neregistrovan korisnik postao član, mora da prođe kroz registraciju na sistem. Korisnik popunjava obavezna polja (ime, prezime, *email* adresu i lozinku), a zatim šalje zahtev za registraciju. Kako bi se sprečile zloupotrebe, sistem će po prijemu zahteva za registraciju korisniku poslati *email* sa kodom za verifikaciju. Korisnik pristigli kod unosi u predviđeno polje i time je uspešno završio registraciju na sistem.

Nakon registracije, korisnik je u mogućnosti da se prijavi na sistem. To čini tako što u predviđenja polja unosi svoju *email* adresu sa kojom je izvršio registraciju, kao i lozinku. Sistem će unete podatke proslediti Amazonovom *Cognito* servisu koji će ih validirati. Ukoliko je kombinacija ispravna, korisnik će biti prijavljen i moći će da izvršava sve radnje namenjene prijavljenim korisnicima.

Administratori sistema imaju nešto više mogućnosti od prijavljenih korisnika. Oni su u mogućnosti da dodaju nove solarne panele koje korisnici mogu da biraju prilikom pravljenja investicije. Svi paneli dodati od strane administratora čuvaju se u posebnoj kolekciji u *MongoDB* klasteru (eng. *cluster*). Osim dodavanja solarnih panela, administratori mogu da vide i sve investicije koje su kreirane na platformi.

3. ZAKLJUČAK

Iz svega navedenog, lako je zaključiti da su obnovljivi izvori energije jedini pravi put koji treba slediti u budućnosti kada je vrsta energije u pitanju.

Pojavom solarnih panela pružila se mogućnost običnom čoveku da ne bude samo nemi posmatrač, već da pruži svoj doprinos kako očuvanju životne sredine, tako i podizanju svesti o važnosti ovog pitanja, ako uštedu za trenutak ostavimo po strani.

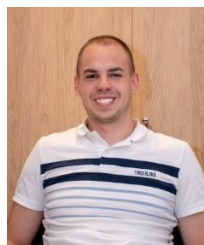
Nažalost, ekonomska situacija u našoj zemlji trenutno ne ide u prilog razvoju i investiranju u obnovljive izvore energije, što će se svakako u budućnosti odraziti povećanim uvozom energenata i sve većim zaduživanjem. Zbog ovog, a i mnogih drugih razloga, treba razmišljati o energiji dobijenoj iz obnovljivih, ekološki prihvatljivih izvora, koji ne uzrokuju ekološke probleme kao što su globalno zagrevanje, zagađenje vode, vazduha, odnosno životne sredine.

Ova aplikacija je doprinos ljudima da lakše uvide potencijal solarne energije i naprave svoj izbor. Prostora za unapređenja je mnogo, počevši od podrške van granica naše zemlje pa sve do unapređenja kalkulacije uzimajući u obzir realne faktore poput orijentacije i površine krova na kojem se instalacija vrši. Aplikacija trenutno ne uzima u obzir realna poskupljenja na tržištu električne energije, kao ni dodatne troškove instalacija.

4. LITERATURA

- [1] Ministarstvo nauke i zaštite životne sredine, studija NPEE, Beograd 2004
- [2] International Renewable Energy Agency, “*Renewable Power Generation Costs in 2017*”, Abu Dhabi, 2018.
- [3] Nacionalni akcioni plan za korišćenje električne energije, „*Službeni glasnik RS br. 53/2013*“, Beograd, 2013

Kratka biografija:



Andrej Karavida rođen je 1997. godine u Zrenjaninu. 2016. godine upisao se na Fakultet tehničkih nauka, odsek Primenjeno softversko inženjerstvo. Diplomski rad pod nazivom Migracija monolitnog sistema na distribuiranu mikroservisnu arhitekturu odbranio je 2020. godine, kada je i upisao master akademske studije. Tečno govori i piše na engleskom jeziku, poseduje B2 Cambridge sertifikat.

Kontakt: andrejkaravida@gmail.com

RAZVOJ APLIKACIJE ZA PRORAČUN ISPLATIVOSTI IZGRADNJE VETROGENERATORA U AZURE OKRUŽENJU

DEVELOPMENT OF AN APPLICATION FOR CALCULATING THE PROFITABILITY OF BUILDING A WIND GENERATOR IN AZURE ENVIRONMENT

Biljana Vukelić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PRIMENJENO SOFTVERSKO INŽENJERSTVO

Kratak sadržaj – U radu je opisan značaj obnovljivih izvora energije sa naglaskom na energiju vetra. Razvijena je aplikacija koja predstavlja sistem za proračun isplativosti investicija ugradnje vetrogeneratora na željenoj lokaciji. Sistem je implementiran u Azure okruženju.

Ključne reči: Obnovljivi izvori energije, snaga vetra, Azure, .NET Core, Angular

Abstract – The paper describes importance of renewable power sources, with the focus on the wind energy. An application has been developed that represents a system for calculating the profitability of investments in wind generators at the desired location. The system is implemented in Azure environment.

Keywords: Renewable power energy, wind power, Azure, .NET Core, Angular

1. UVOD

Sve zemlje sveta imaju težak zadatak: Kako smanjiti upotrebu fosilnih goriva i emisiju štetnih gasova koji izazivaju globalno zagrevanje, klimatske promene, prirodne katastrofe a istovremeno očuvati i dostignuti nivo tehnološkog i ekonomskog razvoja. Različiti obnovljivi izvori energije, zajedno sa svim merama unapređenja energetske efikasnosti i ušteda energije, predstavljaju jedno od idealnih rešenja za radikalno smanjenje upotrebe fosilnih goriva i emisije štetnih gasova.

Energija vetra jedna je od najjeftinijih tehnologija obnovljivih izvora energije koje danas postoje. Napaja ga prirodan vetar i stoga je obnovljiva i čista, ne zagađuje vazduh i ne emituje CO₂. Pri tom ne zrači i ne razara ozonski omotač, dostupna je praktično svuda na planeti. Prilikom planiranja novih kapaciteta, mnoge energetske kompanije se odlučuju za farme vetrogeneratora zbog toga što njihova primena ima ekonomskog i ekološkog smisla. Svaki kWh proizveden obnovljivim izvorima energije, zamenjuje isti koji bi s druge strane trebalo da bude proizveden u elektranama na fosilno gorivo, što ima za posledicu redukciju negativnih uticaja na životnu sredinu, a naročito emisiju CO₂ u atmosferu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Aleksandar Bošković.

Negativni uticaji vetrogeneratora na životnu sredinu postoje, ali su ti uticaji zanemarljivi u poređenju sa pozitivnim elementima. Glavni izazov pri korišćenju vetra jeste taj što je vetar povremen i ne duva uvek kada je potrebna struja.

Ne mogu svi vetrovi biti iskorišćeni da ispune vremenske zahteve za električnom energijom.

Često se dobre lokacije nalaze na udaljenim mestima, daleko od gradova u kojima je potrebna električna energija. Stoga bi potencijalni trošak povezivanja udaljenih vetrogeneratora na elektrodistributivnu mrežu bio previsok. Buka, vizualni efekat, ometanje radio lekomunikacija i uticaj na ptice samo su neke od negativnih karakteristika vetrogeneratora.

2. TEORIJSKE OSNOVE

2.1. OSNOVNI USLOVI ZA PROIZVODNJU ELEKTRIČNE ENERGIJE VETRA

Za proizvodnju električne energije iz vetra potrebni su vetrogeneratori čiji su osnovni delovi elise, prenosni mehanizmi, elektrogenerator, noseći stub i strujni transformator neophodan za priključivanje na elektrodistributivnu mrežu. Da bi se proverili da li postoje uslovi na određenoj lokaciji za postavljanje vetrogeneratora, prvo je neophodno izmeriti brzinu vetra na toj lokaciji. Ona zavisi od konfiguracije terena, objekta na tlu i njegove visine.

Ako vetar duva brzinom od oko 3 m/s pa sve do 20 m/s prvi uslov je zadovoljen. Ipak trebalo bi imati u vidu, da je potrebno da vetar duva najmanje 2.800 sati godišnje (jedna godina ima 8.760 sati) prosečnom brzinom od preko 6 m/s, da bi se određena lokacija uzela u obzir za izgradnju vetrogeneratora [1].

$$P = \begin{cases} 0 & V_w < 3 \text{ m/s} \\ 0.5 A \rho V_w^3 & 3 \text{ m/s} \leq V_w \leq 10 \text{ m/s} \\ P_n & 10 \text{ m/s} \leq V_w \leq 20 \text{ m/s} \\ 0 & 20 \text{ m/s} \leq V_w \end{cases}$$

$$A = 80 \text{ m}^3$$

$$\rho = 1.293 \text{ kg/m}^3$$

$$P_n = 103,44 \text{ kW}$$

Slika 1. Proračun trenutne snage vetrogeneratora

2.2. Tipovi vetrogeneratora

Kod vetrogeneratora, snaga izlazne energije dramatično raste sa porastom brzine vetra. Zbog toga je većina najisplativijih vetrogeneratora locirana na u vetrovitim oblastima. Kao što je rečeno, na brzinu vetra utiče i konfiguracija terena pa se zbog toga vetrogeneratori podižu na visokim tornjevima. Vetroturbina je mašina za pretvaranje kinetičke energije vetra u mehaničku energiju. Mašine koje pretvaraju mehaničku energiju u električnu energiju se nazivaju vetrogeneratori. Vetrogeneratori mogu imati **vertikalne i horizontalne ose radnih kola**. Osnovna podela vetrogeneratora je izvršena upravo prema položaju ose. Sem podele prema položaju ose vetrogeneratori se mogu podeliti i prema broju krilaca na **jednokrilne, dvokrilne, trokrilne i višekrilne**. Po tipu mogu biti obične **sa aerodinamičnim presekom krilaca** i obične **bez aerodinamičnog preseka krilaca** [2].

Prema aerodinamičnom efektu	Prema položaju ose rotacije	Prema brzini obrtanja
Rotori sa otpornim delovanjem	Rotori sa horizontalnom osom	Rotori sa promenljivom brzinom obrtanja
Rotori sa uzgonskim delovanjem	Rotori sa vertikalnom osom	Rotori sa konstantnom brzinom obrtanja

Tabela 1. Podela rotora vetrogeneratora

3. OPIS KORIŠĆENIH TEHNOLOGIJA I ALATA

C# - Moderan objektno orijentisan programski jezik višeg nivoa, razvijen od strane *Microsoft-a*

ANGULAR je radni okvir baziran na TypeScript jeziku koji omogućava razvoj klijentskih aplikacija za različite platforme. Arhitektura angular aplikacije oslanja se na neke od fundamentalnih koncepata kao što su NgModules, komponente, šabloni, direktive. Osnovni gradivni blokovi su NgModules, koji pružaju kontekst prevođenja (eng. compilation context) komponentama [3].

.NET CORE je platforma za razvoj softvera koju je Microsoft razvio za izradu i pokretanje aplikacija.

AZURE je platforma kompanije Microsoft koja nudi širok spektar cloud servisa. Jedan od ciljeva ove platforme je pomoćne raznim kompanijama da obezbede servise uvek dostupnim.

AZURE APP SERVICES je platforma kao usluga (eng. PAAS platform as service) koja omogućava objavljivanje (eng hosting) web aplikacije od strane Microsoft-a. Dizajnirana je za podršku i implementaciju aplikacije na skalabilnoj i pouzdanoj infrastrukturi.

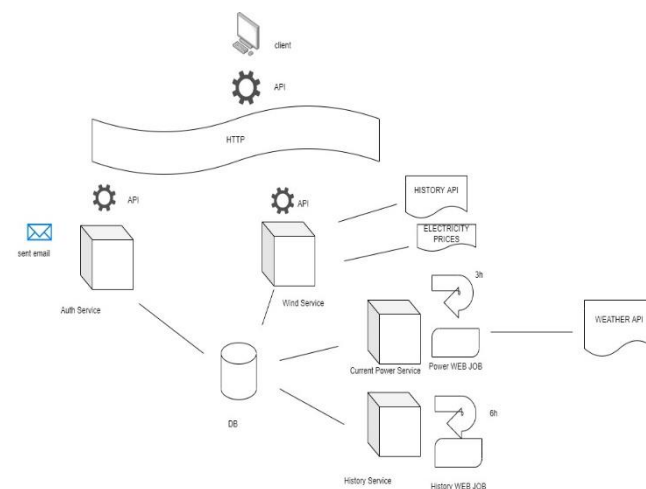
AZURE WEB JOBS je svojstvo AZURE APP SERVICE-a. Omogućava pokretanje programa ili skripte u istoj instanci kao i web aplikacija. Najčešće se koristi za

zadatke (eng. task) koji rade u pozadini. Mogu biti kontinuirani (eng. continuous) ili pokrenuti na događaj (eng. triggered).

AZURE SQL Database je baza podataka na AZURE koja deli svoje resurse sa svim klijentima koji koriste tu uslugu.

4. ARHITEKTURA SISTEMA

U ovom poglavlju biće opisana arhitektura sistema. Monolitske aplikacije su dobro rešenje samo za manje aplikacije koje nisu i od kojih se ne zahtevaju visoke performanske opsluživanja klijenata. Sa stanovišta ovih aplikacija uvek se javlja problem kako sistem održati stabilnim. Sa druge strane mikroservisna arhitektura omogućava skalabilnost, pouzdanost i proširivost sistema na brži i jednostavniji način. Podela aplikacije na manje celine odnosno na mikroservise pruža mogućnost da svaki servis obavlja samo svoju ulogu.



Slika 2. Arhitektura sistema

Sistem se sastoji od sledećih servisa:

- Klijentski deo aplikacije – pisan u Angular Frameworku;
- Auth servis – servis za autentifikaciju i autorizaciju korisnika. Koristi SMTP protokol za slanje šifre korisniku na email, nakon registracije;
- Wind servis – servis za dobavljanje podataka o vetroturbinama, tipovima vetroturbina kao i za proračun isplativosti;
- History servis – servis za dobavljanje istorijskih podataka o vetrogeneratoru;
- Current Power Servis – servis za proračun trenutne snage vetrogeneratora;
- History Web Job – poziva se jednom dnevno i vrši upis podataka o vetrogeneratoru za taj dan;
- Current Power Web Job – poziva se svaka 3h i vrši proračun trenutne snage vetrogeneratora;

5. IMPLEMENTACIJA

U ovom poglavlju biće objašnjena implementacija aplikacije za proračun isplativosti vetrogeneratora u Azure okruženju.

Implementirana aplikacija pruža mogućnost korisnicima da provere isplativost izgradnje vetrogeneratora na određenoj lokaciji. Pomenutu aplikaciju mogu da koriste samo registrovani korisnici. U sistemu postoje 2 tipa korisnika administrator i običan korisnik. Po default-u, postoji samo jedan administrator u sistemu. Neregistrovan korisnik ima mogućnost da se uloguje na sistem ukoliko već ima kreiran nalog ili da se registruje na isti. Proces registracije obuhvata unos polja e-mail koje je jedino obavezno, ime, prezime i broj telefona. Ukoliko je email koji unese validan, korisniku na taj email stiže poruka sa šifrom pomoću koje će moći da se uloguje na sistem. Ulogovanom običnom korisniku u sistemu su dostupne sledeće funkcionalnosti:

- Pružena je funkcionalnost prikaza korisniku da vidi koliko je isplativo da investira novac u vetrogeneratore na određenoj geo lokaciji;
- Svaki korisnik može da prati svoje investicije, koliko je do sada trenutno uložio, koliko mu se isplatilo i koliko je ostalo;
- Aspekt globalnog prikaza investicija svih korisnika u sistemu, kao i grafički prikaz trenutne snage svih vetrogeneratora.

Korisnik čija je uloga administrator, pored osnovnih funkcionalnosti gore navedenih ima mogućnost da:

- Dodaje/Ažurira/Briše sve korisnike u sistemu;
- Dodaje administratore sistema;
- Dodaje/Ažurira/Briše tipove vetrogeneratora.

Takođe svaki korisnik u sistemu ima mogućnost da promeni svoje lične podatke, kao što je lozinka. Neophodno je da unese staru lozinku a zatim novu. Ukoliko stara lozinka nije validna, korisnik neće moći da promeni na novu.

Kada se uloguje, korisniku se prikazuje mapa i desnim klikom na istu, ima mogućnost da selektuje željenu lokaciju. Takođe, korisnik ima mogućnost da klikne na dugme „investiraj“ (eng. invest) i unese tačne koordinate lokacije, naziv države i grada na kojoj želi da izgradi vetrogenerator.

Nakon toga, korisnik ima mogućnost da unese naziv investicije. Zatim je neophodno da izabere koji tip vetrogeneratora želi da izgradi. Prikazuju mu se svi tipovi vetrogeneratora u sistemu – koje administrator unosi, sa nazivom, cenom, troškovima investicije kao i maksimalnom snagom koji taj vetrogenerator može da proizvede. Sa desne strane tog prozora, korisnik se prikazuje dugme „Proveri profit“ (eng. check profit) i klikom na isti se vrši proračun da li se investicija isplati.

Sam proračun se vrši tako što *Current Power* servis, pristupa *Weather API*-ju, uzima koordinate željene lokacije i šalje zahtev za trenutnu brzinu vetra na toj lokaciji. Po formuli sa slike 1 se računa trenutna snaga vetrogeneratora. Kako bi izračunali profit, odnosno da li će se investicija isplatiti ili ne, *Wind* servis šalje zahtev *Weather History API*-ju i traži brzinu vetra na toj lokaciji u prethodnih godinu dana. Izabrani API vraća vrednosti prosečne brzine vetra, za svaki dan u prethodnih godinu dana. Te vrednosti se saberu i podele sa brojem dana koliko je godina imala i dobije se prosečna vrednost brzine vetra za prethodnih godinu dana. Kako bi se dobila

prosečna vrednost snage vetrogeneratora za isti period, ta prosečna brzina vetra se ubaci u formulu sa slike 2.

Cena struje je drugačija za svaku državu, a podaci koji su uzeti u proračunu isplativosti korišćeni su iz literature 4.



Slika 3. Globalne cene električne energije svih država na svetu po kWh iz 2021 godine

Podaci su smešteni u XML fajl pod nazivom *Electricity prices* i podignuti su na Azure servis u sklopu Wind servisa.

Proračun isplativosti investicije se računa po sledećoj formuli koja je prikazana na slici 4.

```
averagePowerOfTurbine = Calculate_WindPower(averageWindPowerForLastYear, powerOfTurbine);
var yearConsumptionOfTurbines = averagePowerOfTurbine * numberOfDaysLastYear;
var epsConsumption20Years = priceElectricity * yearConsumptionOfTurbines * 20;
var profit = epsConsumption20Years - globalPriceOfTurbine;
var index = (profit / (20 * 12));
```

Slika 4. Proračun isplativosti investicije

averagePowerOfTurbine – predstavlja prosečnu snagu turbine kWh;

yearConsumptionOfTurbines – predstavlja proizvodnju turbine za prethodnih godinu dana;

numberOfDaysLastYear – predstavlja broj dana za prethodnu godinu (365/366);

epsConsumption20Years – predstavlja cenu koju bi korisnik plaćao EPS-u u narednih 20 godina;

globalPriceOfTurbines – predstavlja ukupnu cenu vetrogeneratora sa investicionim troškovima kao što je cena ugradnje;

profit – predstavlja razliku između cene koju bi korisnik plaćao EPS-u i ugradnje vetrogeneratora;

index – predstavlja uštedu koju bi korisnik ostvario na mesečnom nivou.

Svaki korisnik u sistemu ima mogućnost pregleda svojih investicija. Omogućen mu je:

- grafički prikaz istorijskih podataka o snazi vetrogeneratora;
- prikaz investicije na mapi odnosno geo lokaciji na kojoj je izgrađen vetrogenerator;
- prikaz trenutne snage vetrogeneratora;
- prikaz trenutne isplativosti te investicije.

Svaki korisnik u sistemu ima mogućnost globalnog aspekta svih investicija u sistemu koji je prikazan je na mapi. Takođe, grafički je prikazana trenutna snaga svih vetrogeneratora u sistemu.

Korisnici imaju i tabelaran prikaz svih istorijskih podataka, kao i svih investicija u sistemu koje mogu da sortiraju i filtriraju po želji.

Administrator sistema ima mogućnost da dodaje/ažurira i briše i vidi tipove vetrogeneratora. Neophodno je da unese naziv, maksimalnu snagu i investicione troškove vetrogeneratora. Neobavezna polja su visina, širina, raspon krila, tip vetrogeneratora i slika.

Takođe, ima mogućnost dodavanja drugih administratora u sistem. Osim dodavanja administratora, ima mogućnost da vidi sve korisnike u sistemu, suspenduje, ažurira ili obriše iste.

6. ZAKLJUČAK

Obnovljivi izvori energije i nove tehnologije korišćenja ovih izvora postaju sve značajniji segment u svim poljima, a pogotovo utiču na životnu sredinu. Korišćenjem obnovljivih izvora energije se smanjuje trošenje neobnovljivih energetske resursa. Korišćenje ovih izvora je veoma značajno i sa aspekta zaštite životne sredine.

Čist primer je sama energetika koja je kao oblast privrede, na kojoj se temelji razvoj svih drugih oblasti privrede najveći zagađivač životne sredine, bez čije zaštite nije moguće ostvariti održivi razvoj ljudske vrste i živog sveta na planeti. Jedini način jeste smanjenje upotrebe fosilnih goriva i razvoj tehnologija korišćenja obnovljivih izvora energije. Zamena fosilnih goriva obnovljivim izvorima energije ima veliki uticaj na uspešno rešavanje globalnih ekoloških problema. Danas primena obnovljivih izvora dobija sve veći značaj, posebno iz razloga očuvanja životne sredine i korišćenja resursa koji su dostupni. Tehnologije iskorišćavanja obnovljivih izvora energije su svakim danom ekonomski sve isplativije i konkurentnije na tržištu.

Razlozi ulaganja u obnovljive izvore energije poslednjih godina prestaju da budu isključivo ideološki, već postaju racionalni i ekonomski. Stoga, aplikacija opisana u ovom radu ima za cilj da podigne svest da sistemi bazirani na energiji vetra postaju efikasniji i ekonomičniji, a benefiti se prepoznaju u budžetima pojedinaca, kompanija, različitih zajednica, ali i stanju životne sredine.

Ukoliko se saberu sve pozitivne strane upotrebe vetrogeneratora, kao i one negativne koje su navedene, a na to doda celokupan tržišni kontekst, ovaj izvor energije ima budućnost. On ima podršku naučne zajednice, tržište se adaptira u odnosu na njega kao budućnost i popularan je među građanima širom sveta. Ipak, da bi u potpunosti bio superioran na tržištu, mora biti predmet daljih istraživanja i tehnoloških iskoraka, kao i da postigne veći stepen ekonomije obima.

Rešenje prikazano u ovom radu, moglo bi se unaprediti u nekim aspektima. Prvo potencijalno unapređenje bi bilo to da se uzmu u obzir realna poskupljenja električne energije na tržištu kao i troškovi usled kvara vetrogeneratora. Drugo potencijalno unapređenje bi bilo da aplikacija bude napravljena u mobilnoj verziji, kako bi korisnici instaliranjem iste na svoj telefon još lakše i brže mogli da prate svoje investicije.

7. LITERATURA

- [1] <https://x-engineer.org/wind-turbine-energy/> (pristupljeno u septembru 2022.)
- [2] Turan Gonen, Third Edition "Electric Power Distribution Engineering", New York, 2014
- [3] <https://angular.io/guide/architecture> (pristupljeno u septembru 2022.)
- [4] https://www.globalpetrolprices.com/electricity_prices/ (pristupljeno u septembru 2022.)

Kratka biografija:



Biljana Vukelić rođena je u Novom Sadu 18.03.1997. god. Diplomirala je na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Primenjeno softversko inženjerstvo 2020.godine. Iste godine, upisuje master studije na istom odseku. kontakt: bilja.vukelic18@gmail.com

PRIMENA METRIČKOG UČENJA U POLJOPRIVREDI APPLICATION OF METRIC LEARNING IN AGRICULTURE

Tomislav Tumbas, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – ELEKTRONIKA ENERGETIKA I
TELEKOMUNIKACIJE**

Kratak sadržaj – U ovom radu prikazujemo primenu metričkog učenja i mogućnost lakog skaliranja na nove klase bez potrebe za ponovnim obučavanjem u oblasti poljoprivrede, na klasifikaciji vrste cveća, vrste lista biljaka i bolesti lista biljaka.

Ključne reči: Metričko učenje, mašinsko učenje

Abstract – In this paper, we show the application of metric learning and ability to easily scale to new classes without the need for retraining in the field of agriculture, on the classification of flower types, plant leaf types and plant leaf diseases.

Keywords: metric learning, machine learning

1. UVOD

Mašinsko učenje se uspešno primenjuje u različitim oblastima industrije. Prvi primeri mašinskog učenja bili su u obradi tabelarnih podataka u ekonomiji i bankarstvu. Napredovanjem grafičkih kartica ispunio se preduslov da se i računarska vizija primenom dubokog mašinskog učenja uspešno primeni na razne probleme u industriji.

Danas se duboko mašinsko učenje u računarskoj viziji uspešno primenjuje u mnogim granama industrije, od izrade automobila i autonomne vožnje do detekcije loše izrađenih komponenti u mašinskoj industriji.

Jedan od oblasti koja zaostaje za ostatkom industrije je poljoprivreda. U skladu sa time, tema ovog rada je primena metričkog dubokog učenja na problem klasifikovanja vrste cveća, klasifikovanje vrste lista biljaka i bolesti biljaka.

2. NADGLEDANO UČENJE

Nagledano učenje je pod grana mašinskog učenja gde su na jednom delu podataka poznati ulazi i izlazi datog problema. Ulazni i izlazni podaci se obično predstavljaju vektorski kao x i y . Potrebno je ustanoviti relaciju između ulaza i izlaza. Nakon što ustanovimo ovu relaciju ona se koristi za predviđanja izlaza y za ulazne podataka x za koji nam nije unapred poznat izlaz y .

2.2 Metričko učenje

Jedan od najneverovatnijih aspekata ljudskog vizuelnog sistema jeste sposobnost prepoznavanja sličnih objekata i

scena. Ne trebaju nam stotine fotografija istog lica da bismo mogli da ga razlikujemo od hiljada drugih lica koja smo videli. Da li je moguće dizajnirati duboku neuronsku mrežu sa sličnom sposobnošću da kaže koji su objekti vizuelno slični, a koji nisu? To je u suštini ono što duboko metričko učenje pokušava da reši, da kaže na osnovu neke metrike šta je slično a šta nije.

Metričko učenje pojavljuje se 2002 godine u radu Ksinga (eng. Ksing) [4].

Metričko učenje na daljinu (ili jednostavno, metričko učenje) ima za cilj automatsko konstruisanje metrike rastojanja (odnosno sličnosti) specifične za dati zadatak iz (slabo) nadziranih podataka, uz pomoć mašinskog učenja. Naučena metrika se tada može koristiti za obavljanje različitih zadataka (npr. K-NN klasifikacija, grupisanje, ekstrakcija informacija).

Metrički problemi učenja dele se u dve glavne kategorije u zavisnosti od vrste nadzora nad podacima korišćenim za obuku:

1. Nadgledano učenje: algoritam ima pristup skupu tačaka podataka, od kojih svaka pripada određenoj klasi (oznaci), kao u standardnom problemu klasifikacije. Uopšteno govoreći, cilj u ovakvoj postavci je naučiti metriku udaljenosti koja tačke sa istom oznakom približava zajedno, dok razdvaja tačke sa različitim oznakama.

2. Slabo nadgledano učenje: algoritam ima pristup skupu tačaka podataka sa nadzorom samo na nivou parova (tipično parovi, trojke ili četvorke tačaka podataka). Klasičan primer takvog slabijeg nadzora je skup pozitivnih i negativnih parova: u ovom slučaju cilj je naučiti metriku udaljenosti koja stavlja pozitivne parove blizu jedne, a negativne parove daleko.

Na osnovu gore navedenih (slabo) nadziranih podataka, problem metričkog učenja je generalno formulisano kao problem optimizacije gde se traže parametri funkcije udaljenosti koji optimizuju neku ciljnu funkciju mereći slaganje sa podacima u obuci.

Velika prednost metričkog učenja u odnosu na klasifikaciju se nalazi u tome što metričko učenje ima mogućnost da radi na klasama koje nisu viđene u procesu obučavanja duboke neuronske mreže. Metričko učenje može da razlikuje N klasa koje su viđene u toku obučavanja i dodatnih M klasa koje nisu viđene u procesu obučavanja.

3. OBUČAVANJE MODELA

Tok obučavanja modela obuhvata pripremu i podelu podataka, definisanje funkcije greške (funkcije cilja), definisanje modela i konačno obučavanje modela. Model

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Rastislav Struharik.

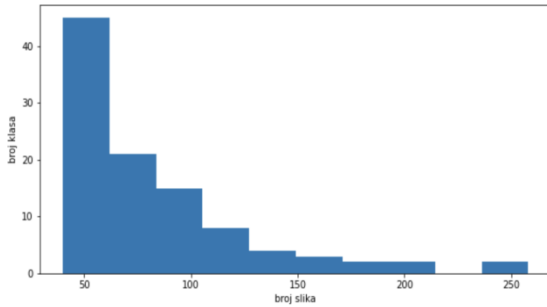
se obučava minimalizacijom greške između predviđanja modela i stvarnih vrednosti ciljne promenjive.

3.1. Podaci

3.1.1 Flower skup podataka

Flower skup podataka sadrži slike različitih tipova cveća.

Broj slika po klasi se kreće 40 do 258 sa srednjom vrednošću od 88,284 i standardnom devijacijom od 44,059.



Slika 1. Histogram slika po klasi

Podatke delimo u dve grupe, vrste cveća koje će biti viđene tokom treninga i vrste cveća koje neće biti viđene tokom treninga.

	Procent a klasa	Srednja vredno st broja slika po klasi	Standardn a devijacija broja slika po klasi	Minimaln a vrednost broja slika po klasi	Maksimaln a vrednost broja slika po klasi
Viđena grupa	78.43	79.63	40.47	40	251
Neviđena grupa	21.56	82.63	55.09	40	258

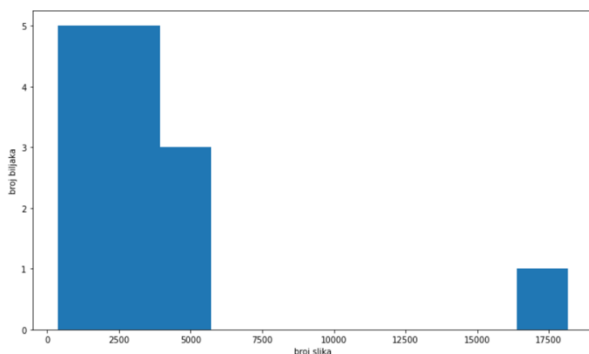
Tabela 1. podela Flower skupa podataka

Viđena grupa se deli po klasi na trening i validaciju u odnosu 80% i 20%.

3.1.2 PlantVillage skup podataka

PlantVillage skup podataka sadrži slike listova biljaka i od svake vrste biljaka par pod klasa koje predstavljaju bolest lista biljke.

Srednja vrednost broja slika po klasi je 3878,92 sa standardnom devijacijom od 4195,47.



Slika 2. Histogram slika po klasi

Podaci su podeljeni u viđenu i neviđenu grupu. U neviđenoj grupi nalaze se tri biljke sa svim svojim podgrupama.

	Broj biljaka	Broj podgrupa	Broj slika
Viđena grupa	3	8	8049
Neviđena grupa	11	30	46256

Tabela 2. podela PlantVillage skupa podataka

3.2. Funkcija cilja

Mnogi nedavni pristupi metričkom učenju zasnovani su parovima uzoraka. Njihove funkcije cilja mogu se izraziti preko kosinuse sličnosti u prostoru embeddinga. Ovu grupu metoda nazivamo dubokim metričkim učenjem zasnovanim na parovima. Tu ubrajamo: kontrasne funkcije cilja, funkcija cilja tripleta, funkciju cilja četvorke, funkciju cilja N parova, funkcije cilja binomske devijacije, funkcije cilja ugla, funkcije cilja hijerarhijske trojke itd.

U svim ovim pristupima, trening uzorci se slažu u parove, trojke, četvorke ili n-torke rezultujući postojanjem polinomijalno rastućeg broja n-torki za obučavanje od kojih su mnoge redundantne ili korelisane te manje informativne za obučavanje duboke neuronske mreže. Ovo dovodi do ključnog pitanja za metode zasnovane na parovima, gde obuka sa slučajnim uzorkovanjem može biti preplavljena redundantnim parovima, što dovodi do spore konvergencije i degeneracije modela sa lošijim performansama.

Jedna od pokušaja da se ovaj problem spore konvergencije reši jeste funkcija cilja više sličnosti (eng. Multisimilarity loss) (Wang, Han, Huang, Dong, Scot 2020 [3]) koji odabira teške uzorke u procesu obučavanja.

3.2.1 Biranje parova

Za jedan primer biramo pozitivan i negativan par. Primer za koji biramo pozitivan i negativan par nazivamo sidro.

Pozitivni parovi:

$$S_{ij}^- > \min_{y_k \neq y_i} S_{ik} - \epsilon \quad (4)$$

Samo oni negativni parovi treba da budu uključeni u obuku čija je sličnost sa sidrom veća od minimalne sličnosti pozitivna (pozitiva koja leži najdalje u embedding prostoru) treba da budu uključeni u obuku, a ostali se odbacuju.

Negativni parovi:

$$S_{ij}^+ < \min_{y_k \neq y_i} S_{ik} + \epsilon \quad (5)$$

Samo oni pozitivni parovi čija je sličnost sa sidrom manja od maksimalne sličnosti negativna (negativ leži najbliže sidru) treba da budu uključeni u obuku, dok se ostali odbacuju.

3.2.2 Funkcija cilja više sličnosti

$$L_{ms} = \frac{1}{m} \sum_{i=1}^m \left\{ \frac{1}{\alpha} \log \left[1 + \sum_{k \in P_i} e^{-\alpha(S_{ik}-\lambda)} \right] + \frac{1}{\beta} \log \left[1 + \sum_{k \in N_i} e^{-\beta(S_{ik}-\lambda)} \right] \right\} \quad (6)$$

Gde su:

- P_i - pozitivni parovi
- N_i - negativni parovi
- S_{ik} – kosinusna distanca između parova
- λ – margina sličnosti
- α, β – hiperparametri

3.3 Ciklični koeficijent obučavanja

Ciklični koeficijent obučavanja je tehnika podešavanja koeficijenta obučavanja tokom treninga obavljena u radu Smita (eng. Smith) 2017 godine [2]. Pre samog obučavanja neuronske mreže radi se pronalazak maksimalnog i minimalnog koeficijenta obučavanja.

Motivacija i ideja cikličnog rasporeda koeficijenta obučavanja je da povećanje koeficijenta obučavanja može kratkotrajno doneti negativne efekte ali dugotrajno pozitivne.

Kratkoročni negativan efekat predstavlja napuštanje lokalnog minimuma što zahteva pogoršanje kvaliteta mreže.

Nakon kratkoročnog pogoršanja mreža može da nastavi ka globalnom minimumu što predstavlja dugotrajno poboljšanje.

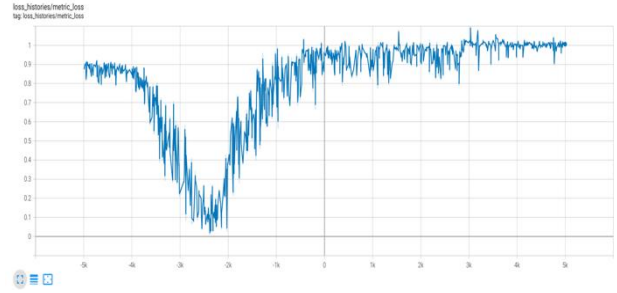
3.3.1. Pronalazak minimalnog i maksimalnog koeficijenta obučavanja

Koeficijent učenja se prvo postavi na veoma malu vrednost (na primer $1e-10$). Pokrene se obučavanje mreže, nakon svake iteracije slika koeficijent obučavanja se eksponencijalno povećava.

Za dati niz koeficijent obučavanja posmatramo izlaz funkcije cilja. Vrednost izlaza funkcije cilja će na početku da stagnira. U ovom trenutku koeficijent obučavanja još nije dovoljno porastao da bi neuronska mreža krenula da se obučava.

U nekom trenutku vrednost izlaza funkcije cilja kreće da opada. Koeficijent obučavanja je porastao dovoljno i proces obučavanja neuronske mreže je započeo. Neuronska mreža se obučava i vrednost izlaza funkcije cilja nastavlja da opada do trenutka kada kreće eksponencijalno da raste. Koeficijent obučavanja je previše porastao.

Koeficijent obučavanja u trenutku kada vrednost izlaza funkcije cilja kreće da opada predstavlja minimalni koeficijent obučavanja. Koeficijent obučavanja u trenutku kada vrednost izlaza funkcije cilja kreće da naglo raste predstavlja maksimalni koeficijent obučavanja.



Slika 3. Vrednost izlaza funkcije cilja tokom određivanja maksimuma i minimuma koeficijenta obučavanja

3.3.2. Ciklični raspored koeficijenta obučavanja

Logika cikličnog rasporeda koeficijenta obučavanja je sledeća:

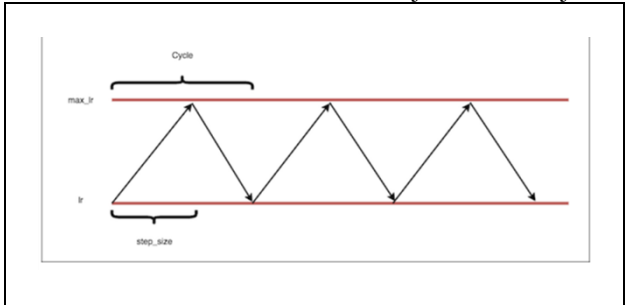
$$\text{ciklus} = \text{math.floor}(1 + \text{brojač-epohe} / (2 * \text{veličina-ciklusa})) \quad (1)$$

$$x = \text{math.abs}(\text{brojač-epohe} / \text{veličina-ciklusa} - 2 * \text{ciklus} + 1) \quad (2)$$

$$\text{koeficijent-obučavanja} = \text{minimalniKO} + (\text{maksimalniKO} - \text{minimalniKO}) * \text{math.max}(0, (1-x)) \quad (3)$$

Gde su:

- minimalniKO – minimalni koeficijent obučavanja
- maksimalniKO – maksimalni koeficijent obučavanja



Slika 4. Izgleda cikličnog koeficijenta obučavanja

3.4. Model

Za model neuronske mreže korišten je ResNet50 (eng. Residual Neural Networks) razvijenog od strane Majkrosoft istraživačkog departmana 2015 godine [1].

ResNet modeli uvode residualne skip konekcije koje sprečavaju efekat nestajućeg gradijenta (eng. Vanishing gradient) i omogućuju lakše učenje identifikacionih funkcija.

4. REZULTATI

4.1 Flower skup podataka

	Preciznost pri 1	Srednja prosečna preciznost
Viđeni primeri – trening	0.9957	0.9810
Viđeni primeri – validacija	0.9613	0.9162
Neviđeni primeri	0.9521	0.9048

Tabela 3. Rezultati na Flower skupu podataka

Prikazani rezultati su komparabilni sa najboljim klasifikacionim pristupima. Razlika između viđenog validacionog dela i neviđenog dela je veoma mala što nam govori da je ovaj model sposoban da radi na klasama koje nije video u treningu, što je jedna od glavnih prednosti metričkog učenja u odnosu na klasifikacione pristupe.

4.2 PlantVilage skup podataka

	Preciznost pri 1	Srednja prosečna preciznost
Viđeni primeri – trening	0.9998	0.9997
Viđeni primeri – validacija	0.9963	0.9950
Neviđeni primeri	0.9847	0.9431

Tabela 4. Rezultati na PlantVilage skupu podataka

Kao i kod prethodnog skupa podataka, rezultati su komparabilni sa najboljim klasifikacionim pristupima.

Nešto veća razlika je između validacije viđenog dela i neviđenog dela. Iz ovoga vidimo da ovaj model ne generalizuje dobro kao Flowers model. Ovo je moguće da se dešava iz više razloga:

1. Ceo skup podataka nema dovoljno primera da se napravi dovoljno dobra generalizacija
2. Nije urađena dobra podela na viđeni i neviđeni deo.
3. Regularizacija nije dovoljna ili je mreža prevelika i ipak dolazi do blagog pre-prilagođavanja na viđeni deo skupa podataka

Svakako, rezultati i dalje pokazuju da je model u stanju da radi i na klasama koje nije video u treningu.

5. ZAKLJUČAK

U mnogim realnim primenama veštačke inteligencije, broj klasa koje model treba da razlikuje i klasifikuje skoro

nikada se neće videti u procesu obučavanja. U slučaju poljoprivrede, vrste biljaka, bolesti biljaka, jako često će se pojavljivati nova klasa zbog koje bi se klasifikacioni model morao ponovo obučavati.

Ponovno obučavanje sa novom klasom zahteva skupljanje novih podataka, anotiranje i ponovno obučavanje modela. Anotiranje podataka je veoma dug i zahtevan proces. Anotacije nikada nisu u potpunosti tačne što znači da je potrebno primenjivati neke metode automatskih detekcija grešaka i ručno ispravljati greške. Ovo je veoma skup proces koji zahteva puno resursa i vremena. U mnogim slučajevima ovo nije ni moguće uraditi jer ne postoji dovoljno podataka niti resursa za anotiranje.

Bez dodatnog prikupljanja podataka, anotiranja i celog skupog procesa, metričko učenje može da uspešno prepozna nove klase. Novi podaci se automatski klasteruju pomoću postojećih algoritama za klasterovanje. Na primeru bolesti biljaka, klasterovanjem bi se mogli odrediti klasteri zasebnih biljaka i unutar njih manji klasteri bolesti tih biljaka.

Dobijeni rezultati na oba skupa podataka pokazuju da se ovaj pristup može koristiti kako na klasama koje su viđene u procesu obučavanja tako i na podacima koji pripadaju klasama koje nisu viđene u treningu.

6. LITERATURA

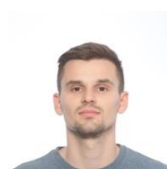
[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: Deep Residual Learning for Image Recognition, Microsoft Research, arXiv:1512.03385, 2015.

[2] Leslie N. Smith: Cyclical Learning Rates for Training Neural Networks, U.S. Naval Research Laboratory, arXiv: 1506.01186, 2017.

[3] Xun Wang, Xintong Han, Weilin Huang*, Dengke Dong, Matthew R. Scott: Multi-Similarity Loss with General Pair Weighting for Deep Metric Learning, Malong Technologies, Shenzhen, China, arXiv: 1904.06627, 2022

[4] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart J. Russell: Distance Metric Learning with Application to Clustering with Side-Information, Advances in Neural Information Processing Systems (NIPS) 15, pages 505–512, 2002.

Kratka biografija:



Tomislav Tumbas rođen u Subotici 1992 god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Embedded sistema i algoritama odbranio je 2022.god.

kontakt: tumbas.tomislav@gmail.com

**AUTOMATSKA REKONSTRUKCIJA DIJAKRITIČKIH ZNAKOVA U TEKSTOVIMA
NA SRPSKOM JEZIKU PRIMENOM MAŠINSKOG UČENJA****AUTOMATIC RECONSTRUCTION OF DIACRITICAL MARKS IN TEXTS IN THE
SERBIAN LANGUAGE USING MACHINE LEARNING**

Vuk Stanojev, *Fakultet tehničkih nauka, Novi Sad*

Oblast – OBRADA SIGNALA

Kratak sadržaj – Rad sadrži opis problema rekonstrukcije dijakritičkih znakova u tekstovima na srpskom jeziku. Problem je predstavljen kao klasifikacioni i predstavljene su tri metode mašinskog učenja pomoću kojih su dobijeni rezultati: neuronske mreže sa propagacijom unapred, mreže sa dugom kratkotrajnom memorijom i konvolucione neuronske mreže. Metode su poređene po merama klasifikatora i za najbolji od klasifikatora dat je prikaz rezultata, odnosno primeri kako je redijakritizacija izvršena.

Ključne reči: *Rekonstrukcija dijakritičkih znakova, Mašinsko učenje, Klasifikacija*

Abstract – This paper contains an overview of problems that follow reconstruction of diacritical marks in text in Serbian language. The problem is shown as a classification problem and three methods of machine learning are proposed with whom the results are obtained: Feed Forward Neural Networks, Short-Term Memory Neural Networks and Convolutional Neural Networks. Methods are compared based on classification metrics and for the best method, results of diacritic restoration is shown.

Keywords: *Reconstruction of diacritical marks, Machine learning, Classification*

1. UVOD

Učesnici u elektronskoj pisanoj komunikaciji na srpskom jeziku latiničnim pismom često zamenjuju slova sa dijakritičkim znakom odgovarajućim ekvivalentom bez dijakritičkog znaka. Oni prilikom čitanja ovih tekstova nemaju problem da razumeju šta je napisano, ali prilikom računarske analize teksta, računaru su reči kod kojih su izostavljeni dijakritički znaci nepoznate. Ukoliko postoje dve reči koje imaju ortografski zapis koji se razlikuje samo u pogledu prisustva ili odsustva dijakritičkih znakova, to za računar može predstavljati izvor dvosmislenosti [1]. Zbog toga je pre bilo koje obrade teksta na računaru neophodno izvršiti rekonstrukciju dijakritičkih znakova.

Iako danas postoje tastature sa slovima koja poseduju dijakritičke znake, ljudi ih i dalje veoma često izostavljaju. Neki razlozi zbog kojih smatramo da se to dešava su sledeći:

- Istorijski gledano, prve tastature koje su se koristile su bile prilagođene engleskom jeziku. Samim tim, nisu imale mogućnost upotrebe slova sa dijakritičkim znacima. Zbog toga su korisnici elektronskih uređaja kao što su računari i mobilni telefoni navikli da ne koriste deo tastature na kojem se danas nalaze slova sa dijakritičkim znacima.

- Nekada je kucanje reči sa dijakritičkim znacima znatno sporije nego da se ona izostave. Na primer, ukoliko je mobilni telefon podešen za rad na srpskom jeziku, tastatura na srpskom jeziku može a i ne mora u svom rasporedu da poseduje slova sa dijakritičkim znacima. Ukoliko ih ne poseduje, potrebno je zadržati prst na odgovarajućem ekvivalentu slova bez dijakritičkog znaka kako bi se pojavila opcija za kucanje slova sa dijakritičkim znakom. Ovakav dizajn tastature usporava kucanje.

- Ukoliko se šalje standardna SMS poruka u njoj je dozvoljeno 160 standardnih karaktera engleskog alfabeta, bez specijalnih karaktera i grafičkih elemenata. Ukoliko se u SMS poruci koriste latinična slova sa dijakritičkim elementom, broj raspoloživih karaktera u okviru jedne SMS poruke smanjuje se na 70 karaktera. Ljudi izbegavaju slova sa dijakritičkim elementima kako bi smanjili trošak telefonskih usluga. Mobilna komunikacija je danas jedan od najzastupljenijih vidova komunikacije. Količina teksta koja se generiše na mobilnim telefonima izuzetno je velika i može doprineti formiranju novih baza tekstova nad kojima je neophodno izvršiti dijakritizaciju.

1.1. Rekonstrukcija dijakritičkih znakova

Dijakritički znaci u srpskom jeziku koriste se za razlikovanje glasova i za razlikovanje slova kada homograf postoji u formi označenog i neoznačenog slova. U srpskom jeziku na latiničnom pismu postoji pet slova sa dijakritičkim znakom: Š, Č, Ć, Ž i Đ. Prilikom pisanja, ona se zamenjuju svojim odgovarajućim slovima bez dijakritičkog znaka, respektivno: S, C, C, Z i Dj. Svako slovo se, dakle, zamenjuje jednim slovom, osim u slučaju slova Đ, koje se zamenjuje sa dva slova. Iako i kombinovano slovo Dž poseduje dijakritički znak, te se u pisanju bez dijakritičkih znakova zamenjuje sa Dz, dijakritizaciju digrafa Dz nije potrebno posmatrati posebno, već se ovaj slučaj može podvesti pod dijakritizaciju slova Ž.

Problem rekonstrukcije dijakritičkih znakova prisutan je u raznim jezicima i veoma je jezički zavisna. Samim tim, za razne jezike su dati metodi koji rešavaju ovaj problem.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Sečujski, red. prof.

Jedna od najjednostavnijih metoda bi bila da se formira rečnik sa učestalostima pojavljivanja reči i da se svaka reč za koju postoji dvoumica oko toga da li u njoj postoje dijakritički znaci ili ne, zameni sa verovatnijom [2]. Za srpski, slovenački i hrvatski jezik predložen je model za rekonstrukciju dijakritičkih znakova koji posmatra najverovatniju reprezentaciju slova u datoj reči i kombinuje je sa verovatnoćom da se data reč nađe u datom kontekstu [1]. Najbolji rezultati za rekonstrukciju dijakritičkih elemenata u srpskom, slovenačkom i hrvatskom jeziku su dobijeni korišćenjem bidirekcionog rekurzivnog neuronske mreže sa tačnošću od 99.7% [2]. Za rekonstrukciju dijakritičkih elemenata na arapskom jeziku predloženi su modeli bazirani na neuronskim mrežama sa propagacijom unapred [3]. Za rumunski jezik korištene su bidirekcionog mreže sa dugom kratkotrajnom memorijom [4].

U nastavku rada biće prikazana baza podataka na kojoj će se obučavati klasifikatori formirani u okviru ovog istraživanja, porediće se performanse klasifikatora i biće prikazani rezultati rekonstrukcije dijakritičkih znakova.

2. BAZA PODATAKA

Baza podataka ukupno ima oko 2.4 miliona reči na srpskom jeziku, napisanih latiničnim pismom. Reči su uzimane iz tekstova koji su pisani različitim stilovima: novinarskom, naučnopopularnom, naučnom i literarnom stilu. Prilikom predobrade baze, svi nizovi znakova koji ne predstavljaju slova zamenjeni su po jednim razmakom. Takođe, sva velika slova su pretvorena u mala.

Rekonstrukcija dijakritičkih znakova je vršena na nivou slova. Posmatraju se isključivo slova koja mogu posedovati dijakritički znak Š, Č, Ć, Ž i Đ, kao i njihovi parovi bez dijakritičkog znaka S, C, C, Z i DJ. Za slova S, Z i DJ dovoljno je odrediti da li bi na mestu pojavljivanja datih slova trebalo da stoji slovo sa dijakritičkim znakom ili ne, dok za C treba odrediti i koji tačno znak treba upotrebiti. Za slova za koja se vrši redijakritizacija, nad delom baze određen je broj pojavljivanja datih slova (tabela 2.1). Iz tabele se vidi da postoje izuzetno velike razlike u pogledu zastupljenosti pojedinih slova u tekstovima na srpskom jeziku. Iako je u standardnim problemima klasifikacije uobičajeno da se u ovakvim slučajevima radi ujednačavanje zastupljenosti pojedinih klasa, odnosno slova, ovde to nije rađeno jer je namera bila da se očuvaju stvarne statistike u tekstu.

2.1. Modelovanje podataka

Baza podataka koja je opisana na početku ove glave nije prosleđivana klasifikatoru u originalnom obliku. Način na koji je baza podataka predobrađena je sledeći:

1. Identifikovani su svi indeksi slova iz tabele 2.1, odnosno, pozicije na kojima se tražena slova u bazi nalaze. Kako je naglašeno da dijakritizacija digrafa DZ nije neophodna, za par slova DZ i DŽ nisu identifikovani indeksi
2. U odnosu na pronađeni indeks, uzimano je deset karaktera pre i deset karaktera nakon datog karaktera. Na ovaj način formiran je prozor koji obuhvata 21 karakter. Dati niz od 21 karaktera predstavlja jedan uzorak koji se prosleđuje klasifikatoru. Kako je moguće da se slovo sa dijakritičkim znakom nađe u neposrednoj blizini samog

početka ili samog kraj teksta, tekst je dopunjen sa odgovarajućim brojem praznina.

3. Svakom nizu, u zavisnosti od centralnog karaktera, dodeljena je odgovarajuća klasna labela pomoću koje će se vršiti klasifikacija. Labele označavaju klasnu pripadnost tog karaktera, odnosno u našem slučaju, označavaju koje je zapravo slovo reprezentovano pomoću datog niza.

4. Nakon dodeljivanja klasnih labela uzorcima, sa slova Š, Č, Ć i Ž koja su se našla u nizu od 21 karaktera, izbrisani su dijakritički znaci, te su ona pretvorena u S, C, C i Z respektivno. Za slovo Đ uveden je obrnut pristup. Odnosno, svi parovi slova DJ pretvoreni su u slovo Đ (zapravo, mogli su biti pretvoreni u bilo koji specijalni karakter koji će čuvati informaciju da to može biti Đ ili DJ). Ovakvom reprezentacijom, klasifikacija slova Š i S, Ž i Z, Đ i DJ predstavlja binarnu klasifikaciju, a za slova C, Č i Ć imamo problem klasifikacije u tri klase.

5. Svaki niz od 21 karaktera koji predstavlja pojedinačni uzorak transformisan je u matricu veličine 21x28. Svaka vrsta odgovara jednom karakteru, dok kolone čine njegovu „one-hot vector“ reprezentaciju [5]. Alfabet za „one-hot vector“ kodovanje je: 'sczđdabefghijklmnopqrutvwxy '. Kako su slova Š, Č, Ć i Ž prethodno konvertovana u svoje parove bez dijakritičkih elemenata, ona nemaju svoju odgovarajuću „one-hot vector“ reprezentaciju. Iako srpski jezik u svojoj azbuci ne poseduje slova Q, W, X, Y, u tekstovima u bazi javljali su se i strani izrazi i nazivi koji su sadržali ta slova.

TABELA 2.1. BROJ POJAVLJIVANJA SLOVA ZA KOJE SE VRŠI REDIJAKRITIZACIJA

Slova	Broj pojavljivanja slova	Pojavljivanje slova [%]
S	75452	4.23340
Š	13900	0.75086
C	14297	0.79117
Č	10722	0.62450
Ć	12870	0.72110
Đ	4003	0.22296
DJ	12	0.00067
Z	26657	1.53183
Ž	7478	0.41480
DŽ	557	0.03135
DZ	58	0.00326

3. MODELI

Duboko učenje je klasa algoritama mašinskog učenja koji koristi slojeve za progresivno izdvajanje obeležja iz sirovih ulaznih podataka. Slojevi zapravo predstavljaju modele veštačkih neuronskih mreža. Veštačke neuronske mreže predstavljaju složene računarske sisteme nastale povezivanjem veštačkih neurona, inspirisane načinom funkcionisanja neurona u ljudskom mozgu [6].

Kao takve, imaju veliku moć generalizacije i primenjuju se u raznim problemima nadgledanog i nenadgledanog učenja. Modeli dubokog učenja koji su ispitivani u ovom radu su: neuronske mreže sa propagacijom unapred, mreže sa dugom kratkotrajnom memorijom i

konvolucione neuronske mreže. Ovi modeli su poređeni po merama klasifikatora i broju parametara koji su neophodni da bi se model obučio.

Neuronske mreže sa propagacijom unapred (eng. *Feed Forward Neural Network* – FFNN) su mreže kod kojih podaci putuju samo napred kroz mrežu i predstavljaju najjednostavniji model dubokog učenja. Arhitektura se sastoji od nekog broja slova koji sadrže određeni broj neurona koji su međusobno povezani sa svim neuronima iz prethodnog ili narednog sloja [7].

U zavisnosti od složenosti problema, broj slojeva, kao i broj neurona koji oni sadrže, može se menjati. Korištene su različite arhitekture neuronskih mreža sa propagacijom unapred prilikom klasifikacije. Od mogućih arhitektura koje su ispitivane, najbolji rezultati su dobijeni kada je postojao jedan skriveni sloj koji je sadržao 512 neurona.

Mreže sa dugom kratkotrajnom memorijom (eng. *Long-Short Term Memory* - LSTM) predstavljaju potklasu rekurentnih neuronskih mreža koje su sposobne da nauče dugoročne zavisnosti u podacima. Rekurentne neuronske mreže predstavljaju klasu veštačkih neuronskih mreža kod kojih postoje povratne veze.

Povratne veze omogućavaju da se prethodne informacije očuvaju i prenose iz jednog koraka u sledeći. Primenuju se u domenima prepoznavanja govora, prevođenja jezika, predviđanja akcija i mnogim drugima. Osnovna gradivna jedinica arhitektura mreži sa dugom kratkotrajnom memorijom jeste LSTM jedinica. Pomoću LSTM jedinice, mreža kombinuje trenutne vrednosti koje joj dolaze sa prethodnim i odlučuje o važnosti uticaja prethodnih vrednosti na trenutne [8, 9].

Prilikom konstrukcije klasifikatora pomoću mreža sa dugom kratkotrajnom memorijom, ispitivane su različite arhitekture u zavisnosti od broja LSTM jedinica koje sadrže. Najbolji rezultati su postignuti kada je korišteno 512 LSTM jedinica.

Konvolucione neuronske mreže (eng. *Convolutional Neural Network* - CNN) se najčešće primenjuju za probleme obrade slike, ali može da se primeni i u drugim oblastima, od kojih je jedna i oblast obrade prirodnog jezika. Konvolucione neuronske mreže tretiraju podatke kao prostorne. Umesto da su neuroni u jednom sloju povezani sa svakim neuronom u prethodnom sloju, oni su povezani samo sa neuronima koji su im bliski i svi neuroni imaju istu težinu.

Na ovaj način arhitektura mreže održava prostorni karakter skupa podataka [10]. Kako se crno-bele slike zapisuju u formi matrica na računaru, a podaci koji se prosleđuju mreži su matrice dimenzije 21x28, konvoluciona neuronska mreža je uspela da nauči prostorne karakteristike podataka i da pokaže dobre performanse.

Prilikom formiranja konvolucione neuronske mreže adaptirana je arhitektura “LeNet-5” sa malim izmenama. “LeNet-5” mreža je radila klasifikaciju slika ručno pisanih cifara koje su veličine 28x28 [11].

Motivacija za adaptaciju date arhitekture jeste dimenzionalnost slika, koja je približna dimenzijama ulaznih podataka u problemu dijakritizacije.

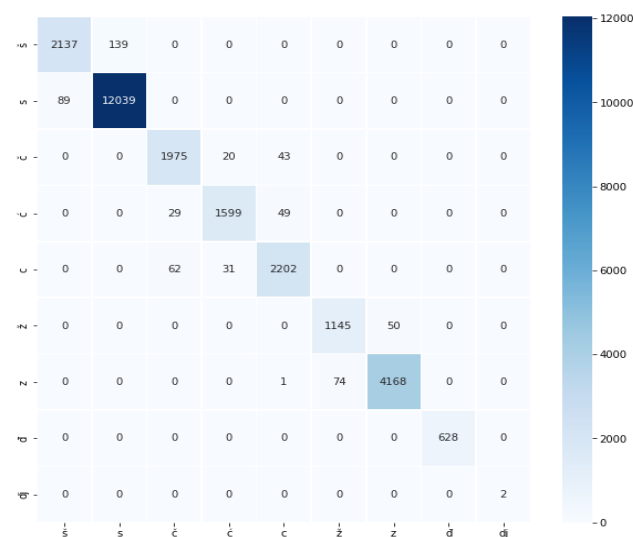
3.1. Performanse modela

Performanse modela koji su navedeni u prethodnoj glavi prikazane su u tabeli 3.1. Iz tabele se vidi da najbolje performanse ima LSTM model. Mada, LSTM model je znatno složeniji od modela formiranih na osnovu FFNN i CNN arhitektura. Iako je LSTM klasifikator znatno složeniji, zbog strukture podataka, klasifikator se veoma brzo obučava i brzo daje rezultate.

Zbog toga, konačni model klasifikatora za koji će biti prikazani rezultati jeste klasifikator baziran na LSTM neuronskoj mreži. Na slici 3.1. je prikazan matrica konfuzije klasifikatora. Iz nje možemo da vidimo da klasifikator uglavnom pogrešno klasifikuje slova koja imaju ortografski zapis koji se samo razlikuje u zavisnosti da li poseduje dijakritički znak ili ne, ali može da se desi i da skroz pogrešno klasifikuje slovo, kao u slučaju gde je jedno slovo Z klasifikovao kao C.

TABELA 3.1. PERFORMANSE RAZLIČITIH MODELA KLASIFIKATORA

Arhitektura	FFNN	LSTM	CNN
Broj parametara za obuku	306.185	1.098.249	274.521
Tačnost klasifikacije	0,966	0,977	0,967
Preciznost klasifikacije	0,850	0,973	0,850
Osetljivost klasifikacije	0,844	0,972	0,850
F1-mera	0.847	0,973	0,850



Slika 3.1. Matrica konfuzije za LSTM klasifikator

3.2. Rezultati

Za konačni model klasifikatora, u tabeli 3.2 prikazani su neki primeri slova koja su pogrešno klasifikovana. Iz tabele 3.2 vidimo da mreža ponekad greši kada postoje štamparske greške kao u slučaju „nCı“ umesto „noCı“. „Koloseum“ predstavlja retku reč koja je pogrešno

klasifikovana jer je model nije susreo u skupu za obuku. Za reč „znaCi“ klasifikator je predivdeo da treba da stoji Č umesto C, što je verovatno posledica činjenice da je reč „znači“ češća u srpskom jeziku od reči „znaci“. Postoje i neke veoma neobične greške, kao u primeru reči „saglaSnost“, gde je S prepoznato kao Š.

TABELA 3.2. PRIMERI POGREŠNO
KLASIFIKOVANIH SLOVA

Rečenica	Stvarni karakter	Prediktovani karakter
a u casopiSu spasic j	S	Š
rsina pod Secernom re	Š	S
ovlacenje Zandarmerij	Ž	Z
siran i u Samponskoj	Š	S
poput koloSeuma kapit	S	Š
lepotice nCi crveno i	Č	C
a je ne liSi cinjenic	Š	S
luposti i Carsijske i	Č	C
ranu saglaSnost zaint	S	Š
ubijenog Spijuna bri	Š	S
da preko Sahare goni	S	Š
da svi znaCi ukazuju	C	Č

4. ZAKLJUČAK

U datom radu predstavljene su različite arhitekture neuronskih mreža koje su vršile rekonstrukciju dijakritičkih znakova u tekstovima na srpskom jeziku. Najbolje performanse pokazala je LSTM mreža i prikazano je nekoliko primera kako ona radi.

Zbog ograničenih resursa koji su bili na raspolaganju, obuka mreže i testiranje njenih performansi rađeni su na relativno malom broju reči, odnosno nad delom baze koja je navedena u radu. Za praktične primene ovog modela njegovu obuku treba izvršiti na mnogo većem tekstu, za šta je moguće upotrebiti i tekstove dostupne na internetu. Velika količina podataka može znatno da poboljša performanse dubokih neuronskih mreža.

U radu su posmatrani uzorci koji su uzimali deset karaktera pre i posle karaktera za koji se određuje da li sadrži dijakritički znak ili ne. Ukoliko bismo proširili prozor, memorija koju zauzimaju ulazni podaci bi se znatno povećala. Zbog ograničenih računarskih resursa, ograničili smo se na fiksni prozor od 21 karaktera, iako bi trebalo ispitati i za performanse modela i za veće prozore, obzirom da je LSTM mreža pogodna za rad sa širim kontekstom, odnosno sa dužim vremenskim sekvencama.

Postoje metode koje pokazuju na kojim delovima podataka mreža uči, kao što su Grad-CAM [12], pomoću kojih bi mogla da se odredi optimalna širina prozora. Postoji verovatnoća da je prozor od 10 karaktera pre i 10 karaktera nakon traženog slova već i sam blizu optimuma. Međutim, ako bi se koristio prozor koji obuhvata na primer 100 karaktera pre i 100 karaktera posle karaktera za koji se vrši rekonstrukcija dijakritičkog znaka, mogla bi se tačno utvrditi optimalna veličina prozora za predstavljene tipove neuronskih mreža.

Problem koji je rešavan u ovom radu na neki način predstavlja uklanjanje šuma iz podataka. Pored modela koji su testirani u datom radu za rekonstrukciju dijakritičkih elemenata, trebalo bi testirati i neke druge, kao što je autoenkoder za uklanjanje šuma. Za rad sa sekvencijalnim podacima veoma dobro su se pokazale rekurentne neuronske mreže, tako da će u budućim istraživanjima pažnja biti posvećena i njima.

5. LITERATURA

- [1] Nikola Ljubešić, Tomaž Erjavec, Darja Fišer, „Corpus-Based Diacritic Restoration for South Slavic Languages“
- [2] Jakub Náplava, Milan Straka, Pavel Straňák, Jan Hajič, “Diacritics Restoration Using Neural Networks”
- [3] Ali Fadel, Ibraheem Tuffaha, Bara’ Al-Jawarneh, and Mahmoud Al-Ayyoub, “Neural Arabic Text Diacritization: State of the Art Results and a Novel Approach for Machine Translation”, Proceedings of the 6th Workshop on Asian Translation, pages 215–225 Hong Kong, China, November 4, 2019. Association for Computational Linguistics
- [4] Stefan Ruseti, Teodor-Mihai Cotet, Mihai Dascalu: *Romanian Diacritics Restoration Using Recurrent Neural Networks*. Septembar 2020.
- [5] <https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/>, 17.08.2022.
- [6] Tijana Nosek, Branko Brkljač, Danica Despotović, Milan Sečujski, Tatjana Lončar-Turukalo: *Praktikum iz mašinskog učenja*, Fakultet Tehničkih Nauka, Univerzitet u Novom Sadu
- [7] https://en.wikipedia.org/wiki/Feedforward_neural_network, 17.10.2022.
- [8] <https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9>, 17.10.2022
- [9] <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, 17.10.2022.
- [10] <https://colah.github.io/posts/2014-07-Conv-Nets-Modular/>, 18.10.2022.
- [11] Yann LeCun, Leon Bottou, Yoshua Bengio, Patrick Haffner, “Gradient-Based Learning Applied to Document Recognition”, “Proc. Of the IEEE”, Novembar 1998
- [12] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*, International Journal of Computer Vision, 2019.

Kratka biografija:

Vuk Stanojević rođen je 25.02.1998. godine u Novom Sadu. Završio je gimnaziju “Jovan Jovanović Zmaj” u Novom Sadu 2017. godine. Nakon toga upisuje Fakultet Tehničkih Nauka, smer Elektronika, energetika i telekomunikacije gde diplomira 2021. godine. Master akademske studije upisuje iste godine na Fakultetu tehničkih nauka, oblast obrada signala.

ПРЕДВИЂАЊЕ ТОКА КАРИЈЕРЕ ТЕНИСЕРА ПОМОЋУ ТЕХНИКА АНАЛИЗЕ ПОДАТАКА**PREDICTING THE FLOW OF TENNIS PLAYER'S CARRIER USING DATA MINING**Светлана Стојадинов, *Факултет техничких наука, Нови Сад***Област – РАЧУНАРСТВО И АУТОМАТИКА**

Кратак садржај – У овом раду описан је процес кластеризације методом к-средњих вредности на основу изабраних физичких карактеристика и стилова игре тенисера. За сваки кластер креирана је регресиона крива, која указује на зависност успеха играча кроз сезоне. Цео процес валидиран је *leave-one-out* методом. Сви кораци система су детаљно описани и визуализовани.

Кључне речи: тенис, анализа података, кластеровање, регресија

Abstract – This paper describes the process of clustering tennis players based on selected physical characteristics and playing styles. A regression curve was created for each cluster, which indicates the dependence of the players' success through the seasons. The entire process was validated using the *leave-one-out* method. All steps of the system are described and visualized in detail.

Keywords: tennis, data mining, clustering, regression

1. УВОД

У данашње време веома је упадљива човекова потреба за физичком активношћу и бављењем спортом, као начином за побољшање квалитета живота, елементом за одмор, забаву, рекреацију и дружење. Позитивно утиче на усвајање здравог начина живота и један је од кључних фактора за здрав живот. Од чисте љубави према неком спорту веома је лако препознати пут према професионалном бављењу одређеног спорта.

Спортови, попут тениса, захтевају посебан начин живота и максималну посвећеност и одрицање и то не само на тренинзима и у мечевима, него сваког дана. Представља стил живота, а не обавезу. Свакодневним улагањем себе у одређени спорт и одређена поља углавном доводи до нових амбиција и веома утиче на формирање такмичарског духа, као и тежње да се постане најуспешнији. Сразмерно популарности тениса као спорта и саме тежње тенисера да постану најбољи, расте и знатижеља љубитеља спорта који тенисер ће и успети да оствари успех у ближој будућности.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Александар Ковачевић, ред. проф.

Проблем којим се бави овај рад јесте груписање и анализа играча тениса на основу физичких особина и карактеристика стила игре ради предикције каријере играча. Само груписање играча врши се кластеризацијом, која служи за проналажење сличних каријера на основу особина које нису унапред специфициране. Подаци над којима се врши груписање су претходно процесирани и прилагођени за даљу обраду.

Кластеризација се врши методом к-средњих вредности, којом се унапред дефинише у колико кластера ће бити распоређене јединке. Поред тога, детаљном анализом група сличних играча добијена је и крива старења за сваку од њих, која указује на зависност успеха играча кроз сезоне. Издвајањем одређеног играча из кластера могуће је проверити колико одступа од криве. Рачунањем средње вредности одступања проверава се колико су одступања велика.

2. ПОДАЦИ

За сврхе овог пројекта коришћена су два скупа података [1, 2] из којих су селектовани само потребни атрибути. Након селекције потребних особина, скупови података су спојени на основу заједничког идентификатора.

2.1. Подаци о играчима

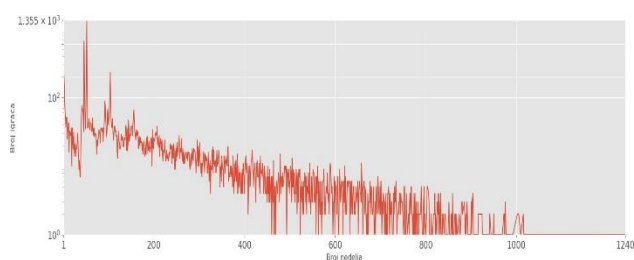
Први скуп [1] обухвата податке који садрже вредности о карактеристикама сваког играча појединачно. Међу атрибутима овог скупа налазе се лични подаци као што су име, презиме, информације везане за место и датум рођења, место становања и прва сезона када је одређени играч постао професионалац. Овај скуп података садржи такође и веома корисне информације о физичким особинама и податке који описују стил игре играча, што се сматра да има највећи утицај на каријеру коју ће он имати као професионалац. Подаци о играчима искоришћени су за сврху прављења кластера тенисера на основу одабраних особина.

Висина и тежина изражене су у две, односно три различите јединице мере (у фунтама и килограмима, односно стопама, инчима и сантиметрима) са прецизношћу једне децимале. Као две карактеристике које описују стил игре наведени су којом руком играч држи рекет и при томе су могуће вредности лево или десно. Међутим друга карактеристика стила игре је навођење доминантне руке играча у свакодневном животу (дакле, леваци или дешњаци), при чему постоје они који се изјашњавају да су им обе руке доминантне. Овај скуп бележи податке о преко 10.000 играча. Важно је напоменути да овај скуп садржи

недостајуће вредности као и очигледно нетачне вредности (на пример тежина од 0 килограма). Ови проблеми, али и уочавање и уклањање података који одступају ван граница прихватљивости решени су претпроцесирањем података и њиховом даљом припремом за примену у алгоритмима и визуализацији.

2.2. Подаци о недељном стању ранг листе

Други скуп [2] података садржи недељне прегледе стања АТП ранг листе. Скуп не садржи податке за сваку недељу - густина забележених недеља варира у сезонама. Најмања густина забележених недеља је у почетним годинама скупа података.



Слика 1. Зависност броја недеља и броја играча

На слици 1. приказан је граф који представља зависност евидентираних играча и броја забележених недеља у току каријере. Атрибути који су коришћени из овог скупа су датум почетка недеље чије стање анализира и позиција одређеног играча у тој недељи. Подаци тенисера о недељном положају на АТП ранг листи се бележе од 1973. године, док се подаци о недељним АТП поенима тенисера бележе тек од 1996. године.

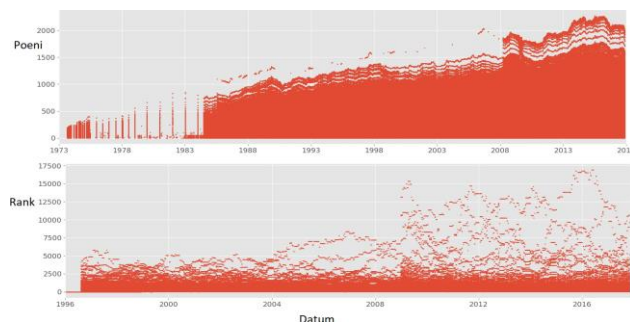
Од почетка бележења података о поенима, начин бодовања различитих категорија турнира се мењао. Ово представља велики фактор при анализи каријера играча у прелазним периодима (када престане примена једног система бодовања, а наступи употреба другог система бодовања), као и при упоређивању каријера играча из различитих периода. Разликују се системи бодовања из периода од 1996. до 1998. године, затим 1999. године, од 2000. до 2008. године и од 2009. године до данас.

Услед ових промена у начину бодовања различитих категорија турнира, графиконе података са положајем на АТП ранг листи и поенима играча карактеришу скокови на прелазима са једног система у други. Најочигледнији је скок између 2008 и 2009. године.

На слици 2. визуализовани су подаци о рангу (положају на АТП ранг листи) и АТП поени играча, како би се увидели скокови на прелазима система, као и густина забележених недеља кроз године.

3. ПРЕТХОДНА РЕШЕЊА

Рад се ослања на два претходна истраживања на тему анализе каријера тенисера и генерално спортиста. Један од циљева радова је описивање каријера спортиста путем регресионе криве, како би утврдиле законитости у каријерама спортиста.



Слика 2. Зависност ранг/поена и времена

Рад [3] се ослања на скуп података о положају тенисера на АТП ранг листи од 1973. до 2017. Рад тежи да утврди просечну криву перформанси професионалних тенисера из периода од 1974 до 2014, као и када ће тенисери достићи врхунац своје каријере. Такође описује рад алгоритма за предвиђање броја АТП поена у наредним годинама.

Утврђено је да већина тенисера достиже врхунац каријере у двадесет петој години живота. Такође, већина тенисера постиже највишу позицију на АТП ранг листи након 9 година каријере. Утврђена је крива којом се описује успех играча у току каријере.

Рад [4] се бави предвиђањем године живота спортиста када ће постићи врхунац каријере, за различите дисциплине. Анализирани су подаци са олимпијских игара, као и подаци о успеху из сваке дисциплине, за различите године. Посматрани су средња вредност година, стандардна девијација и модуло година за сваку спортску дисциплину, као и категорију у оквиру бејзбола.

Конкретно за тенис, утврђено је да тенисери у просеку постигну врхунац каријере са 24 године по модулу година, односно 25.43 по просеку година. Тенисерке достижу врхунац каријере са 23 године (модул година), односно 24.46 (просек година).

4. ОБРАДА ПОДАТАКА

Правилан одабир података, њихова обрада и доступност алата за њихову прераду и анализу је кључан фактор у радовима за анализу и обраду података. Рад са великом количином података доводи до великих организационих изазова. Подаци који се користе у обрадама, могу бити прикупљени на различите начине. Типови података могу бити представљени кроз различите типове. Радови, који се баве темама предлагања метода за прикупљање и обраду доступних података, су засновани на томе да објасне једноставне и ефикасне методе које служе за одабир информација од важности за даљу примену и њихово коришћење.

У овом раду велика пажња је посвећена детаљној и прецизној селекцији и прикупљању података, затим њиховој трансформацији, јер подаци представљају почетни корак за долазак до резултата. Битни кораци при припреми података су: одбацивање непотпуних података, одбацивање нерелевантних атрибута, трансформација типова података у други облик итд. Један од изазова у овом раду био је спајање података из два различита извора.

4. КЛАСТЕРИЗАЦИЈА

Већ припремљени подаци били су улазни параметар за почетак обраде и коришћење одговарајућих алгоритама.

Замисао овог сегмента рада је да се групишу и касније анализирају играчи са сличним карактеристикама са жељом да се покажу сличности и у њиховим каријерама без обзира на еру у којој су рођени и играју као професионалци. Узимајући ово у обзир, први корак је представљао процес кластеризације. За те потребе искоришћен је метод к-средњих вредности користећи библиотеку *scikit-learn* која је део програмског језика *python* у коме је имплементиран рад.

У зависности од вредности параметра k који се задаје, играчи су били класификовани у задат број кластера. У овом раду за број кластера изабрана је вредност 10. Корак доделе и ажурирања је потребно понављати док се не постигне апсолутна конвергенција или достигне максималан број итерација.

5. РЕГРЕСИЈА

Подаци о недељном положају играча на *АТП* ранг листи, као и подаци настали из кластеризације играча према њиховим физичким особинама, користе се у регресионој анализи тока каријере тенисера. Циљ овог дела рада је да се покажу потребни кораци за добијање довољно добре регресионе криве која описује ток каријере тенисера. Регресиона функција тежи да најбоље могуће опише све каријере играча једног кластера, при чему је издвојена каријера првог играча ради валидације.

5.1. Претпроцесирање за регресију

Један од проблема који је уочен у процесу припреме података за регресију, је тај што постоје играчи који се појављују у скупу података играча, али нису ни једном наведени у ранг листи неке од седмица. Као решење на настали проблем играчи из датотека кластера, за које не постоје подаци о положају на *АТП* ранг листи или *АТП* поенима, су уклоњени из скупа. Такође се одбацују и подаци играча за које није сигурно да ли су њихове каријере у целисти забележене у подацима.

То су играчи чији подаци се налазе у првој и последњој години података о недељном положају на *АТП* ранг листи, односно 1973. и 2017. године. Играчи за које је забележено мање од 10 недеља су изостављени из скупа јер се нису довољно дуго такмичили као професионалци и сматрају се нерелевантним.

5.2. Кораци

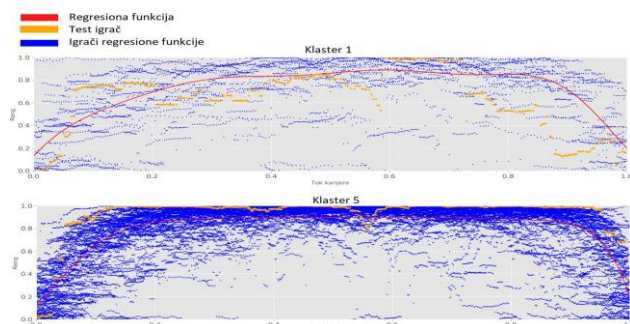
Потребно је да се криве тока каријере играча из одређеног кластера преклопе, независно од датума када су забележени подаци, када је започета или завршена каријера играча, као и колико дуго је трајала каријера. Тачније, потребно је да се криве преклапају по апсиси, која описује временски ток каријере. Слично, потребно је да се подаци преклапају и по ординати, односно независно од положаја на *АТП* ранг листи које је постигао играч у току каријере.

Нормализацијом се уклањају наведене зависности, односно свде да вредности од 0 до 1.

Нормализовани подаци о положају играча на *АТП* ранг листи играча у кластеру се издвајају у посебан скуп. Записи о положају на *АТП* ранг листи се затим сортирају по временској оси, у зависности од нормализованог тренутка у каријери када је податак забележен.

5.3. Валидација регресије

Тачност регресије се проверава тако што се подаци првог играча из кластера пореде са регресионом кривом насталом од података осталих играча. Због тога се за све играче (осим првог играча) у кластеру проналази регресиона функција. Одступање регресионе криве од података првог играча се посматра као средња вредност разлика положаја на *АТП* ранг листи валидационог играча и генерисаних положаја на *АТП* ранг листи. Ово одступање се користи као валидација.



Слика 3. Регресиона крива

6. РЕЗУЛТАТИ

Слика 3. приказује каријере и регресиону криву играча из два већа кластера. Наранџастим маркерима приказани су подаци првог играча у кластеру, плавим маркерима приказани су подаци свих осталих играча у кластеру, а црвеном линијом приказана је регресиона крива играча у кластеру (осим првог играча). Горњи графикон одговара првом кластеру, а средње одступање криве од података првог играча је мање од 18.13% висине каријере. Доњи графикон одговара петом кластеру, а за овај кластер је добијено средње одступање које је мање од 13.21% висине каријере.

Великом одступању података првог играча у кластеру, у односу на регресиону функцију кластера којем играч припада, доприноси неколико чинилаца. Због ових чиниоца, каријера играча се ретко кад може описати правилном кривом, јер је карактерише шум. Овај шум се испољава у скоковима и падовима положаја на *АТП* ранг листи, вишим положајима на почетку и крају каријере у односу на средину каријере, као и недостатком података.

Један од најчешћих чинилаца шума у криви тока каријере играча је што играчи често у току каријере направе паузу услед повреде или других здравствених разлога. Такве паузе трају од неколико месеци, па до неколико година. Услед паузе, положај на *АТП* ранг листи и поени доживљавају огроман пад. Каријера играча може и да се прекине у тренутку повреде или

другог личног догађаја. Такве каријере карактеришу високи положаји на *АТП* ранг листи на крају каријере, што битно утиче и на облик регресионе функције.

Други чинилац шума су промене у начину рангирања играча. Како је раније напоменуто, због ових промена система бодовања играча, каријере играча карактеришу степенице, односно континуални скокови или подови ранга.

Те степенице се јављају на прелазним годинама, где престаје употреба једног система бодовања и рангирања, а почиње употреба другог система. Регресиона крива не може да предвиди овакве поремећаје у току каријере играча.

Трећи чинилац великог одступања регресионе криве од каријере валидационог играча је чињеница да неке каријере уопште не прате криву. Каријере играча који су у току каријере постигли слабе резултате не карактерише раст ранга и поена у току почетка каријере, а ни пад ранга и поена на крају каријере. У току таквих каријера постоје чак и забележени падови на нижи ранг, од првог забележеног ранга.

Овакве каријере су валидне, услед тога што постоји довољни број убележених положаја на *АТП* ранг листи, као и дужина каријере није занемарљива. Због тога, овакви играчи се не смеју избацити из скупа играча којим се генерише регресиона функција. Последица узимања у обзир оваквих играча је што регресиона крива лошије описује играче са правилним каријерама.

Услед свих ових чинилаца, на слици 3. се примећује да постоје положаји на *АТП* ранг листи који су далеко од регресионе криве (плави маркери). Подаци који се налазе испод криве највероватније чине повреде и сличне паузе у току каријере.

Положаји изнад почетка регресионе криве највероватније чине играчи са неправилним каријерама. Положаји изнад краја криве чине играчи који су нагло завршили каријеру, или имају неправилне криве каријере. Овакав шум је битно померио врх каријере према нижим положајима, а почетак и крај каријере ка вишим положајима.

7. ЗАКЉУЧАК

Овим радом објашњени су поступци при реализацији пројекта који се бавио анализом тениских каријера почевши од 1973. године до 2017. године. Играчи су груписани према својим физичким особинама и стилу игре и према добијеним кластерима је одређена карактеристична крива за просечну каријеру сваког кластера. Ова крива представља такозвану крива старења и служи за предвиђање каријере новог играча који би припадао том кластеру. У добијеним резултатима примећен је шум који је оправдан ситуацијама као што су повреде или нагли завршетак каријере. Оваква анализа могла би да се искористи у сврхе одабира играча који ће бити заштитно лице одређених спонзора или при неком од облика предвиђања успешности одређеног играча у сезони.

8. ЛИТЕРАТУРА

- [1] Association of Tennis Professionals (ATP), Github <https://github.com/serve-and-volley/atp-world-tour-tennis-data.git>.
- [2] Association of Tennis Professionals (ATP), ATP World tour www.atpworldtour.com
- [4] Miha Mlakar, Tea Tušar, “Analyzing and Predicting Peak Performance Age of Professional Tennis Players”, Jožef Stefan Institute Jamova cesta 39 SI-1000 Ljubljana, Slovenia, pp 1-4, 2017.
- [3] “Peak Performance and Age Among Superathletes: Track and Field, Swimming, Baseball, Tennis, and Golf”, Richard Schulz, Christine Curnow, Jurnal of Gerontology: PSYCHOLOGICAL SCIENCES, Vol. 43, No. 5, P113 – 120, 1988.

Кратка биографија:



Светлана Стојадинов рођена је у Зрењанину 1994. године. Мастер рад на Факултету техничких наука из исте области одбранила је 2022. године.

контакт: stojadinov.svetlana@gmail.com

ФАЗ ТЕСТИРАЊЕ ВЕБ АПЛИКАЦИЈА**FUZZ TESTING OF WEB APPLICATIONS***Жељана Шиповац, Факултет техничких наука, Нови Сад***Област – РАЧУНАРСТВО И АУТОМАТИКА**

Кратак садржај – Тема рада јесте примјена фаз тестирања на веб апликацијама. У раду су анализирани алати за фаз тестирање, као и рањиве апликације са веба коришћене за потребе извођења фаз тестова, и демонстрације рада алата. Урађени су фаз тестови над овим апликацијама и анализирани су резултати тестирања.

Кључне речи: фаз тестирање, тестирање, безбедност

Abstract – The topic of the paper is the application of fuzz testing on web applications. Tools for fuzz testing are analysed, as well as vulnerable applications from the web used for the purposes of performing fuzz tests, and demonstrations of ways to use fuzzing tools. Fuzz tests were performed on these applications, using fuzz testing tools, and the test results were analysed.

Keywords: fuzz testing, testing, security

1. УВОД

Тестирање софтвера је метод провјеравања да ли стварни софтверски производ одговара очекиваним захтјевима и потврђивање да софтверски производ нема дефекте. То укључује извршавање софтверских/системских компоненти коришћењем ручних или аутоматизованих алата за процјену једног или више својстава од интереса.

Сврха тестирања софтвера јесте да се идентификују грешке, празнине или недостајући захтјеви у супротности са стварним захтјевима [1].

Постоје двије главне стратегије тестирања у тестирању софтвера: позитивно тестирање и негативно тестирање.

Позитивно тестирање утврђује да апликација ради како се очекује.

Негативно тестирање осигурава да апликација може елегантно да рукује неважећим уносом или неочекиваним понашањем корисника. Сврха негативног тестирања је да спречи пад софтверске апликације због негативних улаза и да побољша квалитет и стабилност [2, 3].

Извођењем само позитивног тестирања можемо само да се увјеримо да наш систем ради у нормалним

условима. Како би се направио систем који ради без грешака, морамо да будемо сигурни да може да се носи са неочекиваним условима [3].

Управо овим се бави негативно тестирање. Једна од врста негативног тестирања софтвера је и фаз тестирање које ће бити тема овог рада.

2. ФАЗ ТЕСТИРАЊЕ

Тестирање је саставни дио процеса развоја софтвера. Мотиви тестирања софтвера су, између осталог, потврда да имплементација одговара спецификацији, потврда робусности и отклањање програмских грешака. Поред правилног функционисања софтвера, потребно је обезбиједити и одређени степен безбједности за кориснике и системе који користе софтверски производ. Отклањање оваквих сигурносних пропуста у софтверу кроз тестирање и током развоја софтвера стога није практично [4].

2.1. Дефиниција фаз тестирања

Фаз тестирање или *fuzzing* је аутоматизована метода тестирања софтвера која убризгава неважеће, деформисане или неочекиване улазе у систем да би се открили софтверски недостаци и рањивости. Алат за фаз тестирање убризгава ове улазе у систем, а затим прати изузетке као што су падови или цурење информација. Једноставније речено, фаз тестирање уводи неочекиване улазе у систем и посматра да ли систем има било какве негативне реакције на улазе које указују на недостатке или проблеме у безбједности, перформансама или квалитету [5].

2.2. Развој фаз тестирања

Термин "fuzz" првобитно је креирао професор Бартон Милер 80-их година. Професор Милер је био са даљине пријављен у *Unix* систем преко *dial-up* мрежне везе током олује, што је изазивало много сметњи на *dial-up* линку и узроковало пад апликација које су користиле податке ван линије [6].

Професор Милер је након тога додијелио пројекат својој класи на Универзитету у Висконсину, у ком су студенти требали да развију основни фазер који ради преко командне линије, за тестирање поузданости *Unix* програма бомбардовањем са насумичним подацима и праћењем евентуалних падова програма. Професор Милер и његови ученици су открили улазе који изазивају пад програма за више од четвртине најчешће коришћених *Unix* услужних програма које

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Горан Сладић, ред. проф.

су испитали [5, 7]. Оригинални *fuzzing* пројекат наста- вио је да даје доприносе 1995, 2000, 2006. и недавно 2020. године.

2.3. Предности фаз тестирања

Фаз тестирање нуди широк спектар предности у пољу безбједности и квалитета софтверског производа [7, 8, 9]:

- Фаз тестирање даје добру општу слику квалитета циљног система и софтвера. Користећи фаз тестирање, може се лако процијенити робусност и безбједносни ризик система и софтвера који се тестирају.
- Фаз тестирање је примарна техника коју злонајерни хакери користе да пронађу рањивости софтвера.
- Фаз тестирање има ниске трошкове и за трошкове и за вријеме. Када се фазер покрене, може сам да почне да тражи грешке, без ручне/људске интервенције, и може да настави то да ради колико год је потребно.
- Фаз тестирање помаже у откривању грешака које не би биле откривене конвенционалним методама тестирања или ручним ревизијама.
- Сваки проналазак води до више проналазака.
- Фаз тестирање штити од неочекиваних ивичних случајева.
- Смањује трошкове развоја производа

3. АЛАТИ ЗА ФАЗ ТЕСТИРАЊЕ - ФАЗЕРИ

Фаз тестирање је, као што је већ наведено, метода аутоматског тестирања која убацује или инјектује невалидне, неочекиване, или малициозне податке у систем како би открила грешке и рањивости система. Алати који изводе фаз тестирање, или фазери инјектују ове податке у систем и након тога посматрају понашање система и прате да ли ће доћи до неких изузетака од нормалног рада система, на примјер заустављања рада или цурења информација. У овом поглављу, биће наведени фазери који су коришћени за фаз тестирање веб апликација.

3.1. Owasp ZAP

OWASP Zed Attack Proxy (ZAP) је један од најпопуларнијих бесплатних алата за тестирање безбједности софтвера, и активно је одржаван од стране интернационалног тима волонтера. Користи се за аутоматско откривање безбједносних рањивости у веб апликацијама. Такође може да се користи и за мануелно тестирање безбједности софтвера [10].

ZAP се одржава под окриљем *Open Web Application Security Project (OWASP)*. *ZAP* је дизајниран посебно за тестирање веб апликација и истовремено је флексибилан и проширив.

У својој основи, *ZAP* је оно што је познато као „прокси човек у средини“. Стоји између претраживача који користи тестер и веб апликације, тако да може пресрести и прегледати поруке послате између претраживача и веб апликације, измијенити садржај ако је потребно, а затим прослиједити те пакете на одредиште.

За област истраживања овог рада, и у нападима изведеним зарад испитивања безбједности веб апликација који ће бити објашњени у наредним поглављима, највише ће се користити „fuzzer“ опција коју нуди *OWASP ZAP*. Ова опција омогућава извршавање фаз тестова над пресретнутим захтјевима.

3.2. FFUF

FFUF, или „Fuzz Faster you Fool“ је алат за тестирање отвореног кода, који је намијењен откривању скривених елемената и садржаја веб апликација, или веб сервера [11]. Овај алат је намијењен брзом фаз тестирању, и написан је у *Go* језику. Ово је један од новијих, и до сада се показао као најбржи алат за фаз тестирање, отвореног кода.

FFUF је апликација вођена командном линијом, која се покреће у *Linux* терминалу или *Windows* командној линији, што значи да не садржи интерактивни GUI, већ се уместо тога покреће помоћу команда унесених преко командне линије.

Неки од примјера коришћења овог алата су:

- Откривање фајлова
- Фаз тестирање *GET* захтјева
- Фаз тестирање *POST* захтјева

3.3. Burp Suite

Burp Suite је интегрисана платформа/графички алат за извођење безбједносног тестирања веб апликација. Његови различити алати раде заједно како би подржали цијели процес тестирања, од почетног мапирања и анализе површине напада апликације, до проналажења и искоришћавања безбједносних пропуста.

Алат је написан на Јави и развијен од стране *PortSwigger Web Security*. Поред основних функционалности, као што су прокси сервер, скенер и уљез, алат садржи и напредније опције као што су паук, репетитор, декодер, екстендер и секвенцер [12].

Битан дио алата, који је неопходан за извођење фаз тестирања је прокси за пресретање. *Burp Suite* садржи прокси за пресретање који омогућава кориснику да види и измијени садржај захтјева и одговора док су у транзиту [13].

Burp Suite долази са интегрисаним *HTML Fuzzer*-ом, познатијим као *Burp Intruder*. Овај дио *Burp Suite*-а нам даје могућност за fuzz тестирање са високим степеном прилагођавања. Након што пресретнемо захтјев, користећи прокси, *Burp Suite* нам нуди опцију слања тог захтјева ка *Intruder*-у.

4. ВЕБ АПЛИКАЦИЈЕ СА РАЊИВОСТИМА

У овом поглављу биће описано пар рањивих веб апликација које ће бити кориштене за демонстрирање начина рада алата за фаз тестирање, и над којима ће бити извођени фаз напади. Апликације које ће бити описане направљене су тако да посједују одређене рањивости, те служе програмерима и људима задуже-ним за безбједност софтвера да побољшавају своје вјештине у домену безбједности.

4.1. OWASP Mutillidae II

OWASP Mutillidae II је бесплатна веб апликација отвореног кода, намјерно направљена као рањива веб апликација. Mutillidae се може инсталирати на Linux и Windows. Садржи десетине рањивости и наговјештаја за помоћ кориснику. Ово је окружење намијењено за извођење сигурносних напада.

Mutillidae је коришћен у дипломским курсевима безбедности, корпоративним веб курсевима обуке и као циљ за процјену за софтвер за процјене рањивости [14].

4.2. bWAPP

bWAPP, или *buggy web application*, је бесплатна и намјерно направљена као рањива веб апликација отвореног кода. Помаже заљубљеницима у безбједност, програмерима и студентима да открију и спријече пропусте на веб.

bWAPP садржи преко 100 рањивости. Покрива све главне познате веб несигурности, укључујући све ризике из OWASP top 10 пројекта. bWAPP је намијењен само за тестирање безбедности веб апликација, и за образовне сврхе [15].

4.3. DAMN VULNERABLE WEB APPLICATION (DVWA)

Damn Vulnerable Web Application (DVWA) је PHP/MySQL веб апликација која је циљано направљена рањива. Њен главни циљ јесте да буде помоћ професионалцима за безбједност да тестирају своје вјештине и алате у легалном окружењу, и да помогне веб програмерима да боље разумеју процесе обезбјеђења веб апликација. Овај софтвер има и документоване и недOCUMENTОВАНЕ рањивости с намјером да корисници покушају да открију што више проблема [16].

5. ИЗВОЂЕЊЕ ФАЗ ТЕСТИРАЊА

У раду је описано више изведених фаз тестова. Изведени су тестови са циљем откривања различитих врста рањивости: SQL Injection-а, XSS Injection-а, откривања осјетљивих информација, скривених фајлова, информација о корисницима...

Напади са циљем откривања SQL Injection-а су изведени над апликацијама bWAPP и OWASP

Mutillidae II. Коришћена су три различита алата OWASP ZAP, FFUF и Burp suite. Ови тестови су имали доста сличан ток, креће се од пресретања захтјева коришћењем проксија, затим се за алате OWASP ZAP и Burp suite захтјев даље шаље ка дијелу алата задуженом за фаз тестирање, док се код алата FFUF користе команде командне линије. У сва три случаја потребно је обезбједити улазне листе за фаз тестирање (у овом случају су то листе прилагођене за SQL Injection), и означити дијелове захтјева који ће бити мета фаз тестирања. Након тога могуће је покренути напад.

У свим тестирањима са циљем откривања SQL Injection-а детектована је рањивост на SQL Injection, а у неким је тест успио и да открије податке о разним табелама унутар базе података, као и податке о корисницима.

Изведени су и тестови са циљем откривања скривених фајлова. За ове тестове коришћен је FFUF алат који омогућава фаз тестирање у ову сврху. Тестови су изведени над апликацијама bWAPP и DVWA. За извршавање овог типа напада, прво је потребно открити ког типа су фајлови од којих је сачињена апликација. Да бисмо то открили, потребно је креирати фајл који садржи листу могућих екстензија. Потом се користи сљедећа команда, гдје након ознаке `-u` слиједи путања до почетног фајла апликације (`index`), а након ознаке `-w` слиједи путања до улазне листе могућих екстензија:

```
„ffuf -u http://localhost:8080/bwapp/indexFUZZ -w extension-discovery.txt“
```

Циљ команде је откривање екстензије улазног (`index`) фајла.

У оба случаја откривено је да је фајл типа `„php“`.

За други дио напада, тражење конкретних назива фајлова, користи се SecLists листа намијењена за `„directory-discovery“`. Овај пут траже се сви фајлови чији називи се налазе унутар улазне листе, а затим прецизирамо да се траже фајлови са екстензијом `„php“`. Формирана команда ће бити:

```
„ffuf -u http://localhost:8080/bwapp/FUZZ -w ../directory-discovery-seclists.txt -e .php“
```

Сљедећи корак је покретање команде и анализирање резултата.

У апликацији bWAPP је на овај начин успјешно откривен скривени фајл намијењен управо овој сврси. Откривени су и бројни други скривени фајлови (`secret.php`, `passwords`, `documents...`). У апликацији DVWA су такође откривени скривени фајлови. У оба случаја су као улазне листе коришћене листе са пројекта SecLists [17].

Сљедећи тестови који су извршени су били у циљу откривања осјетљивих информација. Први корак напада је било пресретање захтјева. Након тога се захтјев шаље дијелу алата за фаз тестирање. За први случај се тестирала форма за пријаву апликације bWAPP, и користио се тип напада `„cluster bomb“` алата Burp Suite, који се односи на два улаза. Унешене су листе за фаз тестирање оба улаза (у нашем случају корисничко име и лозинку), и напад је покренут.

Други напад је изведен помоћу алата OWASP ZAP над апликацијом *Mutillidae*. Мета напада је био дио захтјева за пријаву корисника на систем, који се односи на лозинку, а као улазна листа коришћена је листа најчешћих лозинки са интернета. Први напад је успио да открије креденцијале корисника, док су другим, у апликацији *Mutillidae*, такође откривени креденцијали корисника, као и неки други подаци о њему, на примјер потпис (*eng. signature*).

Последњи у низу тестова који су изведени су тестови на XSS рањивост. Поново су коришћени алати OWASP ZAP и *Burp Suite*, а за улазне листе су коришћене листе прилагођене за XSS нападе. Први корак напада је био пресретање захтјева коришћењем проксија. Затим се захтјев шаље дијелу алата за фаз тестирање, означе се дијелови захтјева који су мета напада и обезбиједи путања до улазних листи. У апликацијама *Mutillidae* и *bWapp* је откривена ова рањивост, и извршен је послати малициозни *JavaScript* код.

Из резултата тестирања закључено је да је за откривање скривених фајлова и фолдера погодније користити алате вођене командном линијом, као што је *FFUF*, док смо за рањивости као што је *SQL injection* и *XSS injection* више повратних информација добили од алата као што су OWASP ZAP и *Burp suite*.

6. ЗАКЉУЧАК

У овом раду је описано фаз тестирање и његови бенефити када је ријеч о квалитету израде софтвера.

У раду су приказане теоријске основе фаз тестирања, као и његов развој кроз вријеме. Описани су и неки од алата који се користе за фаз тестирање, и рањиве апликације доступне на вебу које су од помоћи за вјежбање вјештина фаз тестирања и откривања рањивости.

Такође, у раду су описани начини за откривање неких од рањивости веб апликација помоћу фаз тестирања, као и начини генерисања улазних листи за поменуте тестове. Откривене су рањивости као што је *SQL Injection*, *XSS Injection*, цурење информација и откривање скривених фајлова.

7. ЛИТЕРАТУРА

- [1] “*What is Software Testing? Definition*”, Thomas Hamilton
<https://www.guru99.com/software-testing-introduction-importance.html>
- [2] “*Negative Testing*”
<https://smarterbear.com/learn/automated-testing/negative-testing/>
- [3] “*What is Negative Testing? Test cases With Example*”, Thomas Hamilton

- [4] “*DETEKCIJA SIGURNOSNIH PROPUSTA FAZ TESTIRANJEM*”, Aleksandar Nikolić, Goran Sladić, Branko Milosavljević, Zora Konjović
<https://infom.fon.bg.ac.rs/index.php/infom/article/view/1551/1523>
- [5] “*Fuzz testing*”, Synopsys
<https://www.synopsys.com/glossary/what-is-fuzz-testing.html>
- [6] “*Fuzzing info – the art of unexpected input engineering*”
<https://fuzzinginfo.wordpress.com/history/>
- [7] “*Fuzz testing*”
https://fuchsia.dev/fuchsia-src/contribute/testing/fuzz_testing
- [8] “*Code intelligence, What Is Fuzz Testing?*”
<https://www.code-intelligence.com/what-is-fuzz-testing#FuzzTestingDefinition>
- [9] “*Fuzzing – what is it, and why bother?*”
<https://cytal.co.uk/fuzzing-what-is-it-and-why-bother/>
- [10] <https://github.com/zaproxy/zaproxy>
- [11] “*Everything you need to know about FFUF*”
<https://codingo.io/tools/ffuf/bounty/2020/09/17/everything-you-need-to-know-about-ffuf.html>
- [12] “*Web Security Testing with Burp Suite*”, Dr. Sunny Wear
<https://www.pluralsight.com/paths/web-security-testing-with-burp-suite>
- [13] “*What is Burp Suite?*”
<https://www.geeksforgeeks.org/what-is-burp-suite/>
- [14] <https://github.com/webpwnized/mutillidae>
- [15] “*bWapp*”
<http://www.itsecgames.com>
- [16] <https://github.com/digininja/DVWA>
- [17] <https://github.com/danielmiessler/SecLists>

Кратка биографија:



Жељана Шиповац рођена је 23.09.1997. године у Невесињу, БиХ. Завршила је основне академске студије 2020. године на Факултету техничких наука. Уписала је мастер студије исте године, студијски програм Електронско пословање.

контакт: sipovac.zeljana@gmail.com

PRIMENA NFT TOKENA U NAUČNIM PUBLIKACIJAMA**APPLICATION OF NFT TOKENS IN SCIENTIFIC PUBLICATIONS**Tamara Veličković, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSTVO I AUTOMATIKA**

Kratak sadržaj – U radu je prikazana implementacija tržišta za trgovinu NFT tokenima u vidu naučnih publikacija. Objašnjene su osnove Blockchain tehnologija, sa fokusom na NFT tokene. Opisani su model i arhitektura sistema, korišćene tehnologije i prikazana je implementacija.

Ključne reči: NFT, Blockchain, Ethereum, pametni ugovori

Abstract – The paper presents the implementation of a NFT marketplace for scientific publications. The basics of Blockchain technologies are explained, with a focus on NFT tokens. The model and architecture of the system are described in the paper and the project implementation is presented.

Keywords: Blockchain, NFT, Ethereum, smart contracts

1. UVOD

Nezamenljivi tokeni (eng. *non-fungible token*, *NFT*) su kriptografska sredstva na Blockchain sa jedinstvenim identifikacionim kodovima i metapodacima. NFT tokeni se mogu koristiti za predstavljanje vlasništva nad digitalnom svojinom pri čemu je vlasništvo zabeleženo u pametnim ugovorima na Blockchain. Trgovinu NFT tokena je moguće izvršiti putem velikog broja danas dostupnih tržišta kao što su OpenSea, Mintable, Nifty Gateway i Rarible. Kako bi se trgovalo tokenima neophodno je posedovanje sredstava u vidu kriptovaluta koje se skladište u softverskom novčaniku za interakciju sa Ethereum Blockchain.

2. BLOCKCHAIN

Blockchain je inovativna tehnologija baze podataka koja je osnova velikog broja kriptovaluta. Blockchain je distribuirana baza podataka, što znači da se ne nalazi na jednom mestu već je deljena među čvorovima računarske mreže. Cilj Blockchain je da omogući snimanje i distribuciju digitalnih informacija bez izmena, zbog čega predstavlja osnovu za nepromenljive zapise o transakcijama koji se ne mogu ažurirati ili brisati [1]. Zbog ovakvog pristupa, Blockchain se smatra DLT tehnologijom [2]. DLT (eng. *Distributed ledger technology*) predstavlja tehnološku infrastrukturu i protokole koji

NAPOMENA:

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

omogućavaju istovremeni pristup, validaciju i ažuriranje zapisa na nepromenljiv način širom mreže koja je raspoređena na više entiteta ili lokacija [3].

Ethereum je distribuirana softverska platforma otvorenog koda koja je osnovana na Blockchain tehnologiji. Ima sopstvenu izvornu kriptovalutu pod nazivom Ether (ETH) i programski jezik koji pod nazivom Solidity. Ethereum koristi Blockchain za sinhronizaciju i skladištenje promena stanja sistema. Solidity programski jezik se koristi za pisanje programa koji se nazivaju pametni ugovori, dok Ether kriptovaluta služi da se mere i organizuju troškovi resursa za izvršavanje [4].

Pametni ugovori (eng. *smart contracts*) predstavljaju programski kod koji se izvršava na Blockchain kako bi se primenili uslovi sporazuma definisani u ugovoru [5]. Pametni ugovori su računarski programi tipično napisani na jeziku visokog nivoa kao što je Solidity. Glavni cilj pametnog ugovora je da automatski izvrši uslove sporazuma kada se ispune navedeni uslovi [6].

Token predstavlja skup pravila kodiranih u pametnom ugovoru. Tokeni se mogu posmatrati kao digitalna imovina koja je bezbedno skladištena na Blockchain. Tokeni mogu biti zamenljivi (eng. *fungible*) ili nezamenljivi (eng. *non-fungible*). Karakteristike oba tipa tokena su vizuelno predstavljene u tabeli 1. [7].

Tabela 1: Karakteristike tokena [8]

Параметри	Замениливи	Незамениливи
Замениливість	ДА	НЕ
Униформність	ДА	НЕ
Дільність	ДА	НЕ

3. NFT TOKENI

NFT (eng. *non-fungible*) tokeni predstavljaju nezamenljive tokene zapisane na Blockchain koji se odnose na određenu digitalnu ili fizičku imovinu. Vlasništvo nad NFT tokenom se beleži na Blockchain i vlasniku je omogućeno da vlasništvo prenosi prodajom tokena. NFT tokeni obično sadrže reference na digitalne datoteke kao što su fotografije, video snimci, audio zapisi, datoteke, radovi ili publikacije koje može bilo ko da kreira [9]. Prodaja i kupovina NFT tokena obavlja se preko NFT tržišta (eng. *marketplace*) i obično zahteva kriptovalutu za trgovinu. Tržišta NFT tokena su onlajn platforme za sve što uključuje prodaju i kupvinu NFT tokena i zasnivaju se na Blockchain tehnologiji kako bi se potvrdilo vlasništvo nad tokenom. Transakcije za kupovinu, kreiranje ili prodaju NFT tokena se naplaćuju kroz određenu kriptovalutu koju izabrano NFT tržište zahteva. Kako bi se koristilo tržište potrebno je prvenstveno uspostaviti digitalni novčanik za

čuvanje kriptovalute i povezati ga sa željenim tržištem. Digitalni novčanici se mogu uspostaviti preko platformi kao što su Gemini, MetaMask, Binance i Coinbase. Danas se većina NFT tokena kupuje Ether (ETH) domaćom valutom Ethereum mreže, koja se može konvertovati iz američkih dolara na berzama kao što su Coinbase, Kraken i Gemini [10].

4. MODEL SISTEMA

U ovom radu je opisana implementacija platforme za NFT tokene korišćene za objavljivanje i trgovinu naučnih publikacija. Ova platforma pruža korisniku mogućnost kreiranja, kupovine i pretrage NFT tokena. Sistem NFT tržišta se bazira na javnom Blockchain odnosno Ethereum mreži, što podrazumeva da ne postoje ograničenja u pravima pristupa korisnika. NFT tržište može da koristi bilo koji korisnik kako bi pretraživao NFT tokene, dok za druge funkcionalnosti trgovine NFT tokenima korisnik mora biti povezan sa svojim novčanikom i mora posedovati dovoljno sredstava (ETH).

Korisnicima sistema su dostupne funkcionalnosti pretraživanja, kreiranja, prodaje i kupovine NFT tokena. Na NFT platformi je moguće objaviti NFT tokene koji sadrže medijski fajl (audio/video/slika) ili tekstualni dokument (knjige/naučne publikacije/tekstualni sadržaji). Na ovaj način autori publikacija mogu kreirati token od svojih radova i postaviti ih na platformu.

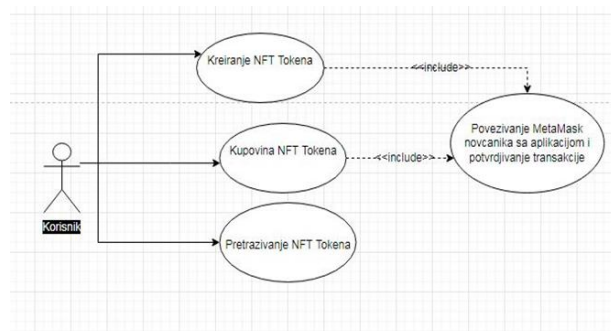
Postavljanje na platformu je moguće ukoliko korisnik ima dovoljno sredstava na svom računu za objavljivanje. Postavljene publikacije predstavljaju NFT tokene koje zatim drugi korisnici mogu pretraživati i kupovati. Kupovina NFT tokena moguća je ukoliko korisnik ima dovoljno sredstava na svom računu za izvršavanje transakcije.

Dijagram slučajeva korišćenja za NFT tržište prikazan je na slici 1. Arhitektura sistema NFT tržišta se sastoji od backend dela koji čine pametni ugovori za NFT tokene i NFT tržište, i frontend deo koji čini klijentska veb aplikacija NFT tržišta za rad sa tokenima. Arhitektura sistema NFT tržišta je prikazana na slici 2.

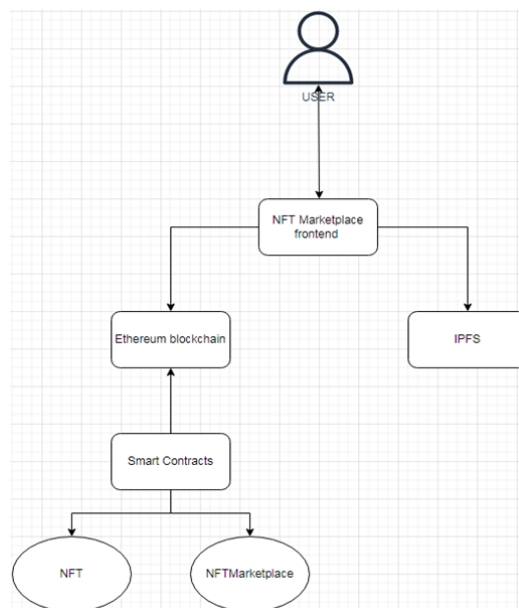
Korisnik NFT tržišta se prvo susreće sa frontend aplikacijom. Kada korisnik pokrene veb aplikaciju susreće se sa polaznom stranicom na kojoj se nalaze svi dostupni NFT tokeni koje platforma nabavlja iz IPFS skladišta. Korisnik ove tokene može da pregleda i pretražuje po autoru i nazivu tokena. Ukoliko korisnik želi da kupi, kreira ili pregleda svoje NFT tokene neophodno je da ostvari konekciju sa MetaMask novčanikom.

MetaMask novčanik predstavlja ekstenziju pretraživača preko koje se identifikuje trenutni korisnik. Korisnik ostvaruje interakciju sa Ethereum Blockchain pomoću web3.js biblioteke kroz MetaMask novčanik. Navedene funkcionalnosti su moguće korisniku ukoliko u svom novčaniku ima dovoljno sredstava (ETH) i ukoliko se izvrši uspešna transakcija prenosom sredstava.

Za potrebe sistema NFT platforme kreirana su dva pametna ugovora: ugovor za tokene napisan po ECR721 standardu i ugovor za NFT tržište. Pametni ugovori obavljaju kompletnu poslovnu logiku i postavljeni su (eng. *deployed*) na Ethereum Blockchain, odnosno na test mrežu.



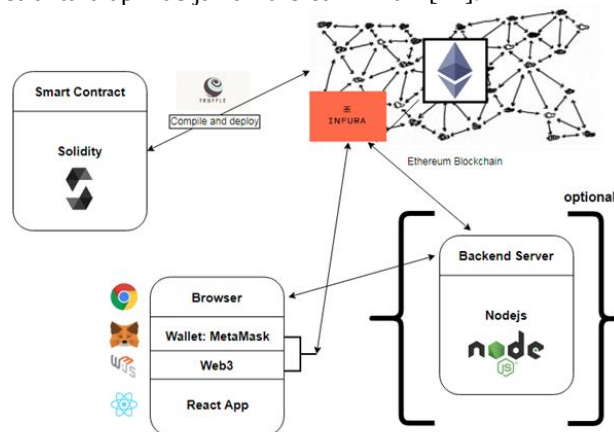
Slika 1: Dijagram slučajeva korišćenja



Slika 2: Arhitektura sistema NFT tržišta

5. IMPLEMENTACIJA SISTEMA

Za implementaciju rešenja NFT tržišta korišćeni su različiti alati i tehnologije kako bi se obezbedile funkcionalnosti kreiranja, pregleda i kupovine NFT tokena naučnih publikacija, rad sa pametnim ugovorima i razvoj frontend aplikacije. Takođe su korišćene tehnologije za isporuku i testiranje rešenja na različitim okruženjima. Na slici 3. prikazani su korišćeni alati i tehnologije kroz strukturu aplikacija na Ethereum mreži [11].



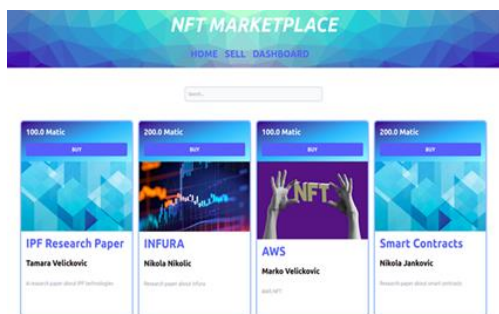
Slika 3: Struktura aplikacije na Ethereum mreži [11]

Bekend aplikaciju čine pametni ugovori napisani u Solidity programskom jeziku. Pametni ugovor za NFT token (NFT) predstavlja implementaciju ECR721 standarda. ECR721 standard čine određeni skup obaveznih funkcija i događaja koje svaki token mora da implementira. ECR721 standard se koristi za predstavljanje vlasništva jedinstvenih i nezamenljivih tokena (eng. *non-fungible*) [12]. Glavna funkcionalnost definisana u NFT pametnom ugovoru je kreiranje samog tokena. NFTMarket pametni ugovor predstavlja ugovor za samo tržište i u sebi definiše funkcije za rad sa tokenima, kreiranje tržišnog artikla i njihovo dobavljanje po različitim kriterijumima.

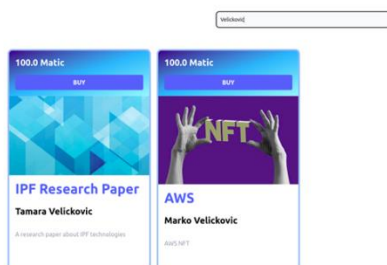
NFTMarket ugovor je takođe implementiran po standardu ECR721. ReentrancyGuard koji, NFTMarket ugovor nasleđuje, štiti transakcije tako što sprečava slanje više zahteva odjednom.

Frontend aplikacija je Next.js aplikacija koja pruža intuitivan korisnički interfejs NFT platforme za prodaju, kupovinu i pretraživanje NFT tokena. Za stil aplikacije je korišćena TailwindCSS biblioteka. Frontend aplikacija se sastoji od 3 glavne stranice a to su: Početna strana (Home), strana za kreiranje NFT tokena i njihovu prodaju (Sell) i korisnička tabla (Dashboard).

Početna stranica (Home) predstavlja prvi prozor koji korisnik vidi. Ukoliko u datom momentu ima dostupnih naučnih publikacija, odnosno NFT tokena za kupovinu, oni su prikazani na početnoj strani, kao što je prikazano na slici 4. Sa početne strane je naučnu publikaciju moguće kupiti ili pretražiti po autoru ili nazivu tokena. Pretraga po autoru je prikazana na slici 5. Za pregled početne stranice korisnik ne mora da bude ulogovan, dok će prilikom kupovine NFT tokena sa početne stranice korisnik biti upućen na MetaMask ekstenziju pretraživača kako bi se identifikovao.



Slika 4: Početna stranica (Home)

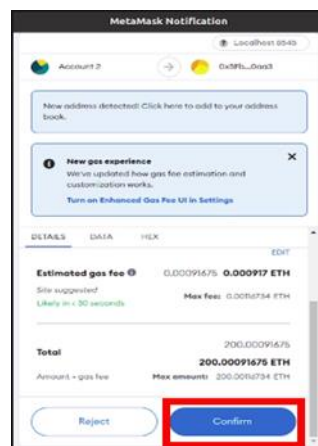


Slika 5: Pretraga po autoru

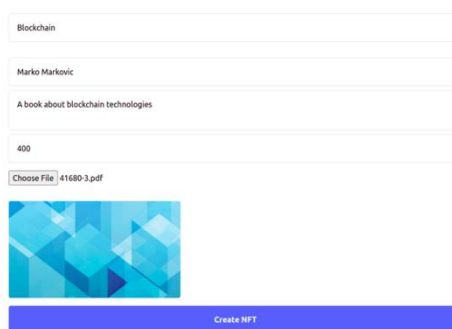
Kupovina NFT tokena započinje klikom na dugme za kupovinu tokena (BUY) sa početne stranice. Ulogovanom korisniku se zatim prikazuje zahtev za izvršenje transakcije u MetaMask ekstenziji za pretraživač (slika 6). Uko-

liko je transakcija uspešna, što podrazumeva da kupac ima dovoljno sredstava u svom novčaniku, vlasništvo NFT tokena se prebacuje na kupca i može se videti na njegovoj korisničkoj tabli u sekciji za kupljene tokene. Kod korisnika koji je prodao taj token, on se pojavljuje na njegovoj korisničkoj tabli u sekciji za prodane tokene.

Za kreiranje NFT tokena neophodno je da je korisnik ulogovan u svoj MetaMask novčanik. Kreiranje NFT tokena, odnosno kreiranje artikla na tržištu počinje na stranici za prodaju (Sell). Kako bi se kreirao token neophodno je da se navede naziv tokena, autor, opis, cena i da se izabere datoteka (slika 7). Datoteka može biti slika, dokument, audio ili video. Klikom za kreiranje tokena ulogovani korisnik dobija zahtev za dve transakcije. Prva transakcija predstavlja plaćanje benzina za kreiranje tokena, a druga transakcija predstavlja plaćanje za postavljanje artikla na tržište (eng. *listing fee*). Ukoliko je transakcija uspešna, što podrazumeva da kupac ima dovoljno sredstava u svom novčaniku za obe transakcije, NFT token se postavlja na tržište. Korisnik koji je kreirao ovaj token može ga videti na svojoj korisničkoj tabli u sekciji za kreirane tokene kao i na početnoj stranici dok se ne proda.



Slika 6: Izgled transakcije za kupovinu naučne publikacije



Slika 7: Izgled forme za prodaju tokena

6. ZAKLJUČAK

U ovom radu opisane su glavne karakteristike Blockchain tehnologija, Ethereum mreže, pametnih ugovora i promenljivih i nepromenljivih tokena. Navedene su karakteristike i benefiti korišćenja NFT tokena iz čega se može zaključiti da su nepromenljivi tokeni jedinstveni i da se mogu koristiti za predstavljanje vlasništva nad digitalnom svojinom koje je zabeleženo u pametnim ugovorima na Blockchain. Fokus rada je bio u primeni nepromenljivih

(NFT) tokena u naučnim publikacijama što je demonstrirano na projektnom rešenju NFT tržišta.

Implementirano NFT tržište nudi korisnicima, autorima radova, mogućnost za kreiranje, pretraživanje, kupovinu i prodaju naučnih publikacija preko kriptovaluta.

Autora naučne publikacije može biti više ali prilikom trgovine tokenom, korisnik koji je kreirao token jedino dobija naplatu za prodaju tog tokena, što potencijalno treba razmotriti u budućnosti kao proširenje ovog projektnog rešenja, kako bi svaki autor dobio nadoknadu. Implementacijom tržišta za NFT tokene zaključeno je da ovakav pristup objavljivanja naučnih publikacija nudi autorima mogućnost bržeg objavljivanja radova, priliku da dostignu veću publiku koja lako može pretražiti taj rad po autoru ili nazivu i garanciju da će njihov rad kao NFT token biti nepromenljiv i jedinstven.

Zaključak koji se takođe može izvesti iz ovog rada jeste da NFT osiguravaju vlasništva nad digitalnom svojinom ali vlasništvo nad NFT tokenom kako ga definiše Blockchain nema direktno pravno značenje i ne daje nužno autorska prava, prava intelektualne svojine ili druga zakonska prava nad njegovom povezanom digitalnom datotekom, zbog čega kao proširenje ovog sistema, potencijalno treba u budućnosti razmotriti i prodaju autorskih prava nad naučnim publikacijama.

7. LITERATURA

- [1] Michael Crosby, Nachiappan, Pradan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman, "Blockchain Technology: Beyond Bitcoin", Applied Innovation Review Issue No. 2, 2016.
- [2] Natarajan, Harish, Solvej Krause, Helen Gradstein. "Distributed Ledger Technology and Blockchain", World Bank, 2017.
- [3] Ali Sunyaev, "Distributed Ledger Technology. In: Internet Computing." Springer, Cham, 2020. [4] "Best Online Learning Platforms", Amanda Prah, <https://www.verywellfamily.com/best-online-learning-platforms-5073725>
- [4] Tikhomirov, Sergei, "Ethereum: state of knowledge and research perspectives", International Symposium on Foundations and Practice of Security, Springer, Cham, 2017.
- [5] Prološčić Nikola, "Blockchain - pametni ugovori", Sveučilište Josipa Jurja Strossmayera u Osijeku, Repozitorij Odjela za matematiku u Osijeku, 2022.
- [6] Maher Alharby, Aad van Moorsel, Blockchain-based smart contracts: A systematic mapping study." arXiv preprint, 2017.
- [7] A. M. Antonopoulos, Dr G. Wood, "Mastering Ethereum", O'Reilly Media, CA, USA, 2018.
- [8] Blockchain Simplified, "Understanding Fungible & Non-Fungible tokens ", <https://Blockchainsimplified.com/blog/understanding-fungible-non-fungible-tokens/>, 2020.
- [9] The Economist, "What is an NFT?", <https://www.economist.com/the-economist-explains/2021/10/12/what-is-an-nft>, pristupljeno: 21.09.2022.
- [10] OpenSea, "Building an open digital economy", <https://opensea.io/about>, 2022.
- [11] Solidity Developer, "The big picture of Solidity and Blockchain development in 2020", <https://soliditydeveloper.com/solidity-overview-2020>, 2020.
- [12] OpenZeppelin Documentation, "ERC721", <https://docs.openzeppelin.com/contracts/3.x/erc721>, pristupljeno: 25.09.2022.

Kratka biografija:



Tamara Veličković rođena je 19.11.1998. godine u Beogradu. Završila je osnovne akademske studije 2021. godine na Fakultetu tehničkih nauka. Upisala je master studije iste godine, studentski program Elektronsko poslovanje.

kontakt: tamara.velickovic98@gmail.com

ISPITIVANJE ALGORITAMA MAŠINSKOG UČENJA ZA PREDVIĐANJE NAPADA NA
SOFTVERSKI DEFINISANE MREŽEINVESTIGATING MACHINE LEARNING ALGORITHMS PERFORMANCE FOR
PREDICTING ATTACKS ON SOFTWARE DEFINED NETWORKS

Adedotun Olasunkanmi Ogundare, Faculty of Technical Sciences, Novi Sad

Field – SOFTWARE ENGINEERING AND
INFORMATION TECHNOLOGIES

Kratak sadržaj – Sa razvojem tehnologije obrade velikih podataka se razvijaju sistemi detekcije upada i anomalija. Malo istraživanja je sprovedeno u oblasti mreža definisanih softverom. Ovaj rad ispituje upotrebu tehnologija obrade velikih podataka za pronalazak sajber napada u softverski-definisanim mrežama. U radu su iskorišćeni algoritmi mašinskog učenja – stabla odlučivanja, nasumične šume i Naivni Bajes – za detekciju upada nad javno dostupnim skupom podataka. Postignuta preciznost je 96.7%.

Ključne reči: Veliki podaci, mašinsko učenje, mreže definisane softverom, sajber napadi, detekcija upada.

Abstract – As technology continues to evolve, the application of Big Data Technology is widening for intrusion and anomaly detection. However, not much has been done within the domain of Software Defined Networks (SDN). This paper explores how Big Data Technology can be used to predict cyber-attacks on SDNs by using historical dataset to build Machine Learning models. Using three widely-known supervised machine learning algorithms- Decision Tree, Random Forest and Naïve Bayes - results show average prediction accuracy of 96.7%. several other techniques were employed to evaluate the performance of the models.

Keywords: Big Data, Machine Learning, Software Defined Networks, Cyber Attacks, Intrusion Detection.

1. INTRODUCTION

Modern trends in application development, which require rapid service provisioning, have led to wide adoption of programmable networks like Software Defined Network (SDN) and Cloud Computing. These allow agile applications to be deployed on networks seamlessly without requiring infrastructure overhaul and the resulting capital expenditure.

However, these developments have also introduced new security concerns. Past works noted that despite the benefits brought by the modern computing developments, it has also expanded the frontiers of cyber-attacks [13]. Machine Learning can be used to uncover threats by using models trained of previous datasets to make assumptions, as demonstrated in this work.

REMARK:

This paper is result of the Master thesis supervised by dr. Nikola Luburić.

1.1. SDN Overview

Being the modern approach to networking, Software Defined Network (SDN) abstracts infrastructure from control layer to deliver a dynamic network architecture that supports quick deployment of agile applications. With a 19% CAGR, the global SDN market is growing as more enterprises transition from traditional network to improve productivity [10]. It has also led to an increase in attack footprint. The SDN controller is a potential attack surface that can give hacker a total control of network without the need to hack individual network devices.

1.2. Problem Statement

The daunting task of securing SDN controller lies in filtering malicious nodes from normal ones because they exhibit similar traffic patterns. The conventional approach for defending against attacks is to rely on data validation techniques and apply thresholds which is then used to define rules on SDN controllers [13]. However, this has limitations because it is often subjective and when a single deciding factor has been used, it is unable to detect more than one attack at a time.

1.3. Machine Learning Approach to Attack Detection

Machine Learning can be used to identify patterns in historical data in network traffic statistics. A large scale of network data can be collected in SDN such as security device logs, IP parsing packets, network traffic statistics, system protection logs, flow-tables in OpenFlow switch and so on. Many open-sourced network datasets have been established by global body of research. Using ML models trained from historical data on network traffic, predictions can be made on future traffic.

2. SELECTED RELATED WORKS

Recognizing that the rapid rise in cybercrime has placed software security issues on the spotlight, Rudolf, et al. [11] investigated the performance of machine learning techniques in predicting functions with possible security vulnerabilities in JavaScript programs. Using datasets from public databases, eight machine learning algorithms were used to build prediction models, and static source code metrics were used as predictors. The research found that the best performing algorithm was the KNN which created a model that predicted vulnerable functions with an F-measure of 0.76 (0.91 precision and 0.66 recall), thus providing a viable practical approach to applying machine learning in predicting web vulnerability [11].

Marwin, Florian, & Samuel [9] followed the same approach in building machine learning models for predicting Hard Disk drive Failure.

They contend in their work that Self-Monitoring Analysis and Reporting Technology (SMART) had been used hitherto to predict hard drive failures. The SMART approach can determine the current health state but it cannot predict failures. Their research presents four Machine Learning (ML) models for hard disk failure prediction and evaluates the performance of each machine learning model [9].

It must be noted that prior to the work of Marwin, Florian, & Samuel [9], there have been some proposed approaches for managing the shortcoming of SMART. Popular among these approaches is the work of Chaves [4] which attempt to directly apply ML algorithms on SMART to predict time-to-failure in the form of regression. Also, Botezatu, et al. [3] applied classification in determining whether HDDs will fail within a specific time window. While [3] and [4] provide some practical insights, they suffer the limitation of considering HDD failure prediction only as static task by building models on random subsets of available data [3][4][9].

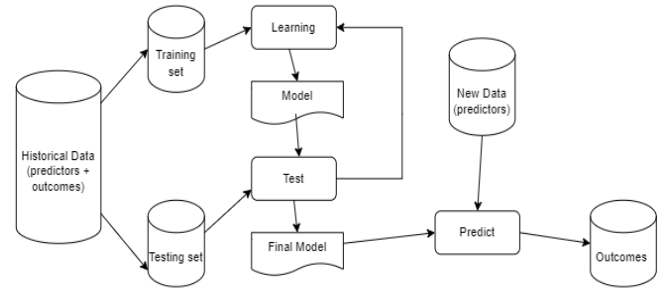
3. PREDICTION EXPERIMENT

A dataset representing historical data on SDN traffic was obtained and downloaded into a directory of the local machine used for this experiment.

Using Apache Spark technology implemented in JVM (Java Virtual Machine), the dataset was used to build classified ML models on Decision Tree, Random Forest and Naïve Bayes algorithm. A pipeline was developed to take the dataset through series of transformation based on Apache Spark Machine Learning workflow, as itemized below:

1. *Loading data from source into RDD (Resilient Distributed Dataset)* — For easy manipulation, dataset was intermittently transformed into data-frames using the Apache SQL library.
2. *Transforming RDD*—Data cleansing, float conversion, centering & scaling was performed to prepare the dataset for ML analysis.
3. *Creating ML Datatypes* —using the principles of dimensionality reduction, the Labeled Point was created as a function of the target variable and the dense vectors.
4. *Splitting training and testing data* —The split was varied at different ratios and performances metrics were recorded at each ratio.
5. *Creating a model*— ML models were built on the training data subset.
6. *Performing predictions* – Predictions were made on testing data subset.
7. *Testing accuracy* —Confusion matrix was generated to perform several accuracy and performance measures.

The workflow is repeated after scaling from a single machine and increasing partitions sequentially across two nodes, three nodes, and four nodes—running in parallel and measuring performance at each case.



3.1. Description of Dataset

The University of Nevada-Reno Intrusion Detection Dataset (UNR-IDD) has been chosen for the experiment. The dataset has 33 columns representing the feature variables (predictors) upon which the dense vectors are formed. It also has a binary label representing the target (outcome) variable which resolves to either of the two possible scenarios, ‘normal’ or ‘attack.’ The dataset has a total of 37,411 records out of which 33,638 portends an attack while 3,773 represents normal traffic [7].

In the description of the data collection technique, the developers of the UNR IDD dataset created custom application to collect and log port statistics captured at 5 seconds interval from OpenFlow (OF) switches [7]. The data features captured represents each column in the data set.

3.2. Dimensionality Reduction

In this experiment, three approaches of dimensionality reduction were used successively. At first, hypotheses were used in selecting predictors. Thereafter, correlation coefficient was used to reduce the number of predictors by discarding those that have low correlation with the target variable. Finally, Principal Component Analysis (PCA) was applied with varying constant K. Performance was measured at different values of constant K, and the best scenario was recorded. The same predictors were used to build models for the three selected supervised ML algorithms in order to provide a uniform benchmark for assessing and comparing their performance.

4. RESULTS AND ANALYSIS

The three ML models trained were used to predict attacks on the test data subset. After deriving the confusion matrix, performance is first measured by accuracy given by the equation 1 below:

EQ 1: Equation for Prediction Accuracy

$$= \frac{\text{No. of Correctly predicted attacks (TP)}}{\text{Total Predictions (TP + TN + FP + FN)}} \times 100$$

Further to that, the *sensitivity* is measured using hit rate and recall with the goal of determining how good the models are in predicting actual ‘attacks.’

EQ 2: Equation for Sensitivity

$$= \frac{TP}{TP + FN} \times 100$$

Understanding that accuracy alone does not tell the full story when evaluating the results of prediction, the f-measure was employed.

A threshold level β (Beta) percentage is chosen as minimum probability for varying precision. This helps the author to examine its effect on the prediction accuracy. The training & testing dataset is also split in different proportions in order to examine their effect on prediction accuracy. The Equation 3 below is used to determine the f-measure.

EQ 3: F Beta Measure Equation:

$$= \frac{1}{\beta X \frac{1}{precision} + (1 - \beta) X \frac{1}{recall}}$$

Table 1 shows the accuracy of the different Machine Learning (ML) algorithms used for this work. It also shows the f-measure with varying β threshold as well as varying training/testing split proportionality. The result reveals that Random Forest has the highest average prediction accuracy—98.404%.

Table 1. Accuracy of the different Machine Learning models on the attack prediction task

Decision Tree Algorithm				
Split	Acc	$B=.5$	$B=1$	$B=1.5$
70:30	96.70	99.72	99.99	100.25
60:40	96.19	99.91	99.95	99.99
50:50	96.33	99.85	99.85	99.86
40:60	96.27	99.80	99.98	100.16
30:70	95.98	99.84	99.88	99.91
AVG	96.29			
Random Forest Algorithm				
Split	Acc	$B=.5$	$B=1$	$B=1.5$
70:30	96.90	95.00	97.56	98.19
60:40	97.77	97.00	91.49	96.77
50:50	98.44	93.41	99.81	98.99
40:60	98.91	91.23	93.93	98.89
30:70	100.00	96.00	96.17	98.87
AVG	98.404			
Naive Bayes Algorithm				
Split	Acc	$B=.5$	$B=1$	$B=1.5$
70:30	91.00	87.20	94.24	94.57
60:40	94.20	94.13	94.96	94.78
50:50	97.30	93.87	93.01	94.99
40:60	97.19	91.16	93.19	95.01
30:70	97.56	91.19	93.19	95.01
AVG	95.45			

4.3. Findings

4.3.1 Effect of split ratio & probability threshold on Prediction Accuracy

The results reveal that in ML algorithm, the split ratio has no significant effect on prediction accuracy. This differs from the study of Rista et al. [12] who found tangible relationship between prediction accuracy and split ratio. The result however shows that threshold has notable effects on the prediction accuracy, thus agreeing with their findings.

4.3.2 Execution Performance of ML Algorithms

The performance of the various ML algorithm is further scrutinized by finding the average time it took each algorithm to perform prediction under different scale, as revealed in the Table 2 below.

Table 2 The execution performance of the tested Machine Learning algorithms

ML Algorithms	Execution Time in Milliseconds (MS)			
	Single JVM	Two Nodes	Three Nodes	Four Nodes
Decision Tree	32616	32121	32121	34119
Random Forest	31864	31985	31942	33191
Naïve Bayes	13652	12191	12666	13011
Clustering k = 2	27397	26291	26786	26789
Clustering K = 3	27397	23291	26645	26744
Clustering k = 3	27723	26264	26659	26759

The result shows Naïve Bayes has the best performance among the supervised ML algorithms. Measuring performance of ML at different scales extends on common approach of previous works, which mostly measured performance on a single scale—either as single machine or a specific number of clusters. For instance, Emre, et al. [8] measured performance of ML using two nodes which their work was premised upon. The result of performance evaluation done in this research compares results of different ML algorithm when run on single machine, two nodes, three nodes and four nodes.

4.3.3 Effects of Partitioning on Execution Performance

This research reveals that distribution of dataset across partitions and nodes can have effect on execution performance. More partitions (nodes) generally lead to better execution performance. However, this experiment reveals that if the dataset is not large enough to make sense of several partitions, then the performance goal of having multiple partitions may not be achieved due to the delay factors occasioned by cluster management. The principle of Lowest Usable Partitions (LUP), as coined by the author of this work, should be employed in view of the volume of data viz a viz the available computing resource of available nodes.

4.3.4 Effect of Dataset on Prediction Accuracy

Dataset is a factor that affects prediction accuracy. It was noted in [8] that higher variance in data will always increase the chances of false prediction. The intrusion detection dataset used for this work has relatively low variance, as reported by the developers of the dataset [7]. Hence, the predictions made from models built with the dataset shows reasonably high accuracy. An accurate prediction like that of this experiment can be leveraged by SDN controller to tighten the network security and block potential attacks.

5. CONCLUSION

This paper has leveraged the power of big data by building ML models to predict vulnerable hosts/traffic in SDN network and thereafter measuring accuracy and performance on different metrics. This has helped to identify factors that impact overall accuracy of prediction as well as execution performance. The models were built from historical dataset on network traffic. The research outcome can be used to define security rules for SDN controller to prevent malicious acts on a network. The result of the experiment has shown that ML algorithm can be used by SDN to define security rules based on the accurate predictions made.

LITERATURE

- [1] Ashraf, J., & Latif, S. (2014). Handling Intrusion and DDoS attacks in Software Defined Networks using Machine Learning Techniques. *Software Engineering Conference*, (pp. 55-60).
- [2] Ali, S., Sivaraman, V., Radford, A., & Jha, S. (2015). A survey of securing networks using software defined networking. *Reliability IEEE Transactions*, 1086-1097
- [3] Botezatu, M. (2016). Predicting disk replacement towards reliable data centers. *22nd ACM SIGKDD International Conference on Knowledge Discipline and Data Mining*.
- [4] Chaves, I. (2018). Hard disk drive failure prediction method based on bayesian network. *International Joint Conference on Neural Networks*. IEEE
- [5] Hu, F., Hao, Q., & Bao, K. (2014). A survey on software deined network and Openflow: From concepts to Implementation. *IEEE communications survey tutorials*, 2181 - 2206.
- [6] Kim, H., & Feamster, N. (2013). Improving network management with software defined networking. *IEEE Communications magazine*, 114-119
- [7] Das, T., Abu, H., Shukla, R., Sengupta, S., & Arslan, E. (2022). *University of Nevada - Reno Intrusion Detection Dataset (UNR-IDD)*. TechRxiv.
- [8] Emre, U., Sonali, S.-B., & Rattikorn, H. (2018). Towards Prediction of Security Attacks on Software Defined Networks. *Interntional Conference on Big Data*
- [9] Marwin, Z., Florian, E., & Samuel, K. (2021). Machine Learning Model Update Strategies for Hard Disk Drive Failure Prediction. *20th IEEE International Conference on Machine Learning and Applications*. IEEE
- [10] Nunes, B., Mendonca, M., Nguyen, N., Obraczka, K., & Turetletti, T. (2014). A survey of software-defined networking: past, present and future of programmable networks. *Communications Surveys and Tutorials, IEEE*, 1617-1634.
- [11] Rudolf, F., Peter, H., Peter, G., Gabor, A., Denes, B., & Tibor, G. (2019). Challenging Machine Learning algorithms in Predicting Vulnerable Javascript Functions. *7th International Workshop on Realizing Artificial Intelligence synergies in Software Engineering*. Szeged: University of Szeged.
- [12] Rista, A., Ajdari, J., & Zenuni, X. (2020). Predicting and analyzing Absenteeism at Workplace Using Machine Learning Algorithms. *43rd International Convention on Information, Communication and Electronic Technology* , 485-490
- [13] Saurav, N., Faheem, Z., Zasimer, D., Eric, W., & Baijian, Y. (2016). Predicting Network attack Patterns in SDN using Machine Learning Approach. *IEEE Conference on Network Function Virtualization and Software Defined Networks*. New York
- [14] Sommer, V. (2014). *Anomaly Detection in SDN Control Plane*. Munich: Master's thesis, Technical University of Munich
- [15] Ullman, S., Poggio, T., Harari, D., Zysman, D., & Seibert, D. (2020). Unsupervised learning - Clustering. *Center for Brains, Minds and Machines*

Short Biography:



Dotun Ogundare from Nigeria, earned his masters degree at the Faculty of Technical Sciences of the University of Novi Sad in October 2022 and subsequently enrolled in his PhD. His current interests include distributed computing, Software Defined Networking, Systems Dynamics, digital culture and technology determinism.

LED PAMETNA RASVETA ZA UZGOJ AKVARIJUMSKOG BILJA**LED SMART LIGHT FOR GROWING AQUARIUM PLANTS**Dušan Ilić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Ovaj rad prikazuje realizaciju projekta "LED pametna rasveta za uzgoj akvarijumskog bilja". Rast i razvoj bilja, kroz proces fotosinteze, se temelji na spektru, intenzitetu svetlosti i trajanju dužine osvetljenosti. Pomenuti faktori realizovani su pomoću adekvatnog dizajna hardvera i programskog koda.

Ključne reči: IoT, ESP-12, Mikrokontroler

Abstract – This paper features the actualization of the "LED smart lighting for growing aquarium plants" project. The growth and development of plants, through the process of the photosynthesis, is based on the spectrum, the intensity of the light and the illuminance length duration. The mentioned factors were realized by an adequate hardware design and program code.

Keywords: IoT, ESP-12, Microcontroller

1. UVOD

Sunce je nama najbliža i najbolje proučena zvezda koja emituje energiju u vidu sunčeve svetlosti i toplote, a time omogućava život na Zemlji putem procesa fotosinteze. Ideja LED pametne rasvete jeste da omogućava simulaciju Sunčevog ciklusa tokom 24h (izlazak i zalazak), a samim tim biće zadužena za rast i razvoj biljaka kroz proces fotosinteze koji se temelji na spektru, intenzitetu svetlosti i trajanju dužine osvetljenja. Spektar koji biljke apsorbuju, predstavlja vidljivi deo svetlosti čiji spektar zračenja uključuje talasne dužine od 400 nm do 700 nm [1]. Odabirom dveju LED lampi, odnosno *Full-spectrum* i hladno belih LED lampi, obezbeđen je adekvatan spektar zračenja za proces fotosinteze. Intenzitetom svetlosti dveju LED lampi upravlja se pomoću dva N kanalna MOSFET tranzistora. Dužina uključenosti i isključenosti tranzistora definisana je faktorom ispune impulsno širinske modulacije, PWM (engl. *Puls-width modulation*) signalom, koji se generiše na GPIO pinovima ESP8266 mikrokontrolera. Trajanje dužine osvetljenosti tokom 24h, biće podeljeno u 10 vremenskih intervala koje će korisnik moći definisati putem korisničkog interfejsa. Kako bi mikrokontroler imao informaciju o trenutnom vremenu, sistem sadrži sat relanog vremena, u daljem tekstu RTC (engl. *Real Time Clock*). Podešavanje funkcija LED pametne rasvete, odnosno podešavanje trenutnog vremena, 10 vremenskih intervala i intenziteta svetlosti dveju LED lampi za svaki vremenski interval, vršiće se pomoću dva korisnička interfejsa. Prvi korisnički interfejs

NAPOMENA:

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

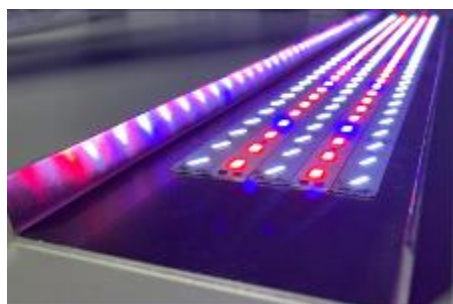
predstavljao četir tastera kojima korisnik ima mogućnost ručnog podešavanja funkcija LED pametne rasvete, a povratne informacije o podešavanju, biće prikazane na OLED ekranu radi vizualizacije aplikacije. Dakle u ovom projektu, mikrokontroler ESP8266 predstavlja *master*, dok su RTC i OLED ekran *slave* uređaji. Komunikacija između *master* i *slave* uređaja realizovana je I²C komunikacionim protokolom. Drugi korisnički interfejs predstavlja mobilna aplikacija za *real-time* podešavanje funkcija LED pametne rasvete, putem Wi-Fi mreže. Za programiranje ESP8266 mikrokontrolera koristiće se programski jezik C modifikovan od strane Arduino IDE programskog paketa za programiranje razvojnih ploča pomenutog proizvođača.

2. TEHNIČKO REŠENJE SISTEMA

Za realizaciju LED pametne rasvete, upotrebljene su dve vrste LED lampi:

1. *Full-Spectrum* LED biljne lampe (spektar plavih LED dioda iznosi 450 nm, dok za crvene iznosi 660 nm)
2. Hladno bele LED lampe (boja osvetljenosti iznosi 6500 K)

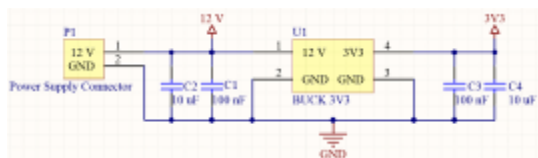
U cilju produženja životnog veka i zadržavanje karakteristika LED čipova, potrebno je obezbediti adekvatno hlađenje sistema. Za pasivno hlađenje, LED lampe se montiraju na aluminijumski hladnjak koji ujedno predstavlja i kućište LED rasvete. Sa unutrašnje strane montiraju se LED lampe u odnosu 3:2 i upravljačka elektronika kako je prikazano na narednoj slici.



Slika 1. Kućište i montirane LED lampe u odnosu 3:2

2.1. Napajanje sistema

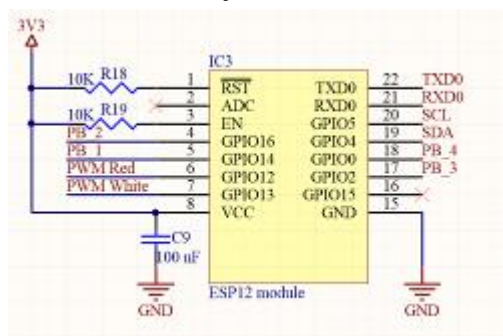
Napajanje LED pametne rasvete odabrano je na osnovu tehničkih specifikacija LED lampi čiji je ulazni napon 12 V jednosmernog napona. Za napajanje ostatka električnog kola, potrebno je smanjiti ulazni napon sa 12 V na 3.3 V što je rešeno pomoću DC-DC konvertora pod nazivom "Mini360". Na električnoj šemi DC-DC konvertor označen je sa "BUCK 3V3", a glavno napajanje sistema od 12 V dovodi na konektor P1.



Slika 2. Električna šema napajanja

2.2. ESP-12 Wi-Fi modul

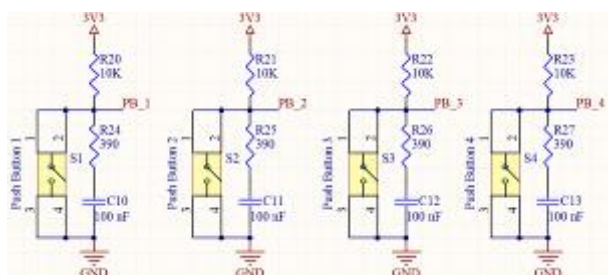
ESP-12 modul predstavlja tip čipa SoC (engl. System on a Chip) na kom su većina komponenti "računara" integrisana u jedan čip. Mikrokontroler ovog modula je ESP8266 sa ugrađenom Wi-Fi mrežom i antenom. Ovaj modul se nalazi u kućištu sa 22 nožice, dok su na raspolaganju samo 16 nožica ovog modula, a njihove funkcije prikazane su na električnoj šemi ESP-12 modula [2].



Slika 3. Električna šema ESP-12 modula

2.2.1. Treperenje kontakata tastera

Ručno podešavanje funkcija LED pametne rasvete vrši se putem četiri tastera. Svaki od krajeva tastera nosi svoje ime signala PB_1, PB_2, PB_3 i PB_4 (skraćeno od engl. Push Button) koji se dovode na GPIO pinove sa rednim brojevima 0, 2, 14 i 16 kako je prikazano na slici 3. Električna šema povezivanja tastera sa GPIO pinovima ESP-12 modula, prikazana je narednoj slici.



Slika 4. Električna šema povezivanja tastera

GPIO pinovi na svojim krajevima uvek vide visoki napon, a pritiskom tastera dovode se na masu, odnosno nizak napon što mikrokontroler detektuje kao trigger za izvršavanje određene funkcije. Pritiskom na taster dolazi do treperenja dirki tastera, a za eliminaciju ove pojave zadužen je kondenzator vrednosti 100 nF u električnom kolu [3]. Struja punjenja kondenzatora, kada je taster u otvorenom položaju, definisana je otpornicima vrednosti 10 kΩ i 390 Ω. Pritiskom tastera, zatvara se strujno kolo preko otpornika 390 Ω ka masi i vrši se pražnjenje kondenzatora. Prilikom pražnjenja sva energija kondenzatora se disipira na otporniku 390 Ω čime se štite dirke tastera od varničenja. Vremenska konstanta punjenja i pražnjenja kondenzatora na osnovu izraza [4]:

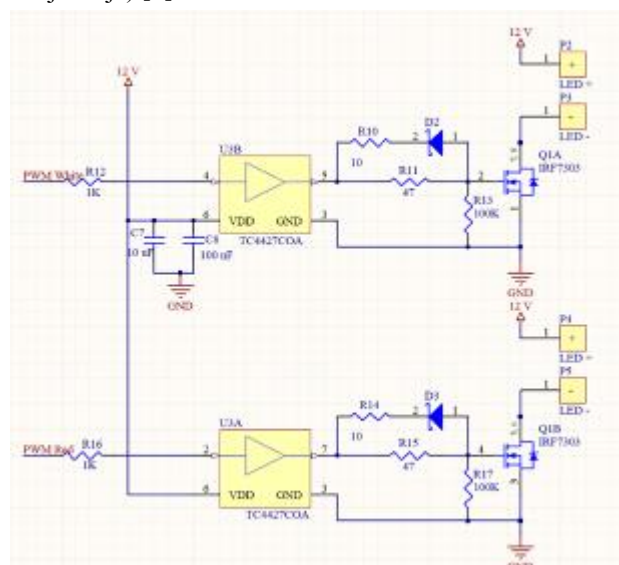
$$\tau = R \times C \quad (1)$$

za ovo kolo iznosi 1 ms. Dakle kondenzator od 100 nF se napuni na 100 % vrednosti napajanja (3V3) za vreme od 5ms, što iznosi 5 τ .

2.2.2. Impulsno širinska modulacija

Da bi se LED lampe dimovale nezavisno jedna od druge, GPIO pinovi 12 i 13 konfigurisu se kao izlazni. Na slici 3 prikazano je da se GPIO12 pinu dodeljuje signal PWM_RED pomoću kog se dimuju Full-Spectrum LED biljne lampe, dok se dimovanje hladno belih LED lampi vrši preko pina GPIO13 kome je dodeljeno ime signala PWM_White. Na pomenutim GPIO pinovima generisaće se PWM signal radi dimovanja LED lampi.

Dimovanje LED lampi vrši se sa dva N kanalna MOSFET tranzistora, čije se otvaranje i zatvaranje kontroliše PWM signalima. Kako je MOSFET naponski upravljana komponenta neophodno je dodati u kolo između ESP-12 modula i MOSFET-a gate driver, čija je uloga pojačavačka u smislu podizanja napona sa 3.3 V na 12 V koji je neophodan za pravilan rad tranzistora (uključenje i isključenje) [5].



Slika 5. Električno kolo za dimovanje LED lampi

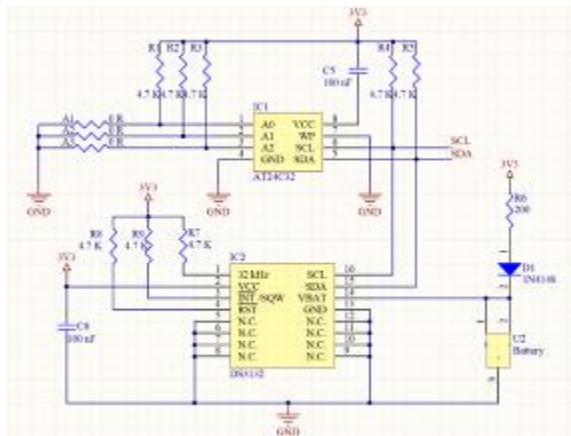
Signali PWM_White i PWM_RED dovode se na gate driver-e koji pojačavaju PWM signale sa 3.3 V na 12 V, a potom na gate pinove (2 i 4) MOSFET tranzistora. Na drain pinove (5, 6, 7 i 8) dovode se katodni (minus) krajevi LED lampi preko konektora P3 i P5 dok se anodni (pozitivan) kraj direktno veže na izvor napajanja 12 V preko konektora P2 i P4. Kako PWM signal predstavlja povorku pravougaonih impulsa, tranzistor se nalazi u uključenom stanju sve dok traje impuls PWM signala i tada struja kroz drain i source krajeve kreće da teče (I_D). Strujno kolo je sada zatvoreno od izvora napajanja 12 V kroz potrošač (LED lampe) kroz koje protiče struja I_D što dovodi do emitovanja svetlosti. Za vreme pauze PWM signala tranzistor se nalazi u isključenom stanju i predstavlja otvoreno kolo. Tada struja kroz LED lampe ne teče, a tom prilikom se ne emituje svetlost.

Intenzitetom svetlosti LED lampi upravlja se faktorom ispune PWM signala koji predstavlja odnos impuls/paza, a računa se prema sledećem izrazu [6]:

$$D = \frac{T_{ON}}{T} \times 100 [\%] \quad (2)$$

2.3. Sat realnog vremena

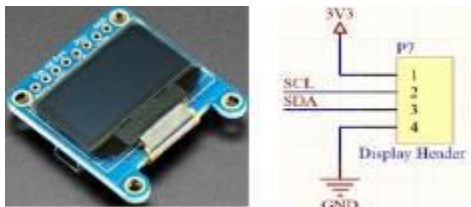
Trajanje dužine osvetljenosti tokom 24h (podeljeno u 10 vremenskih intervala) podešava se na osnovu vremena dobijenog od strane sata realnog vremena, odnosno RTC-a. Za RTC odabrano je DS3231 integrisano kolo koje razlikuje broj dana u mesecu kao i 24-časovni i 12-časovni format dana. U slučaju nestanka napajanja, RTC za čuvanje početnog vremena koristi nezavisno napajanje u vidu baterije CR2032. Takođe, uz RTC dolazi i EEPROM memorija AT23C32 za čuvanje dodatnih informacija [7]. Električna šema RTC-a prikazana je na narednoj slici.



Slika 6. Električno kolo DS3231 RTC-a

2.4. OLED ekran

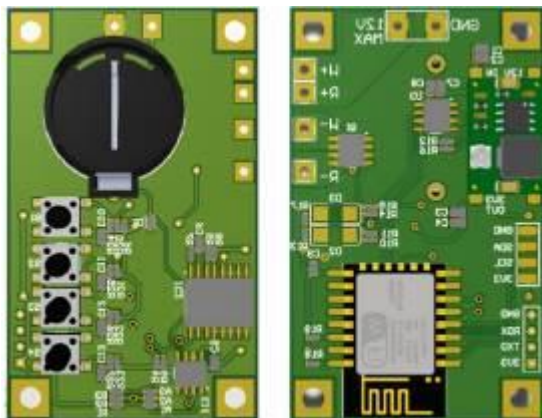
Prilikom ručnog podešavanja funkcija LED pametne rasvete, korisniku je omogućen vizualni kontakt na 1.3 inčnom OLED ekranu napravljenom od 128x64 individualnih piksela. OLED ekran se pomoću četiri žice preko konektora P7 povezuje sa štampanom pločom kako je prikazano na narednoj slici.



Slika 7. OLED ekran i konektor za povezivanje

2.5. Projektovanje štampane ploče

Na osnovu predstavljenih električnih šema projektovana je štampana ploča za rad LED pametne rasvete, čija 3D projekcija, je prikazana na narednoj slici.



Slika 8. 3D prikaz projektovane štampane ploče

3. KORISNIČKI INTERFEJS

3.1. Ručno podešavanje funkcija LED rasvete

Funkcije tastera za podešavanje funkcija LED pametne rasvete date su u narednoj tabeli.

Tabela 1. Funkcije tastera

Funkcija	Up	Down	Enter	Back
Switch	S4	S3	S2	S1
Push Button	PB 4	PB 3	PB 2	PB 1
GPIO	2	0	16	14

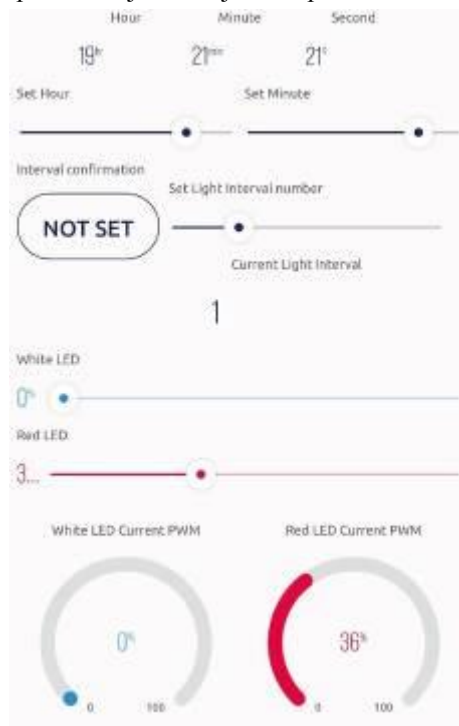
Podešavanje trenutnog vremena, vremenskih intervala i jačina intenziteta svetlosti za svaki interval, prikazano je na narednoj slici.



Slika 6. Meni aplikacije za ručno podešavanje funkcija

3.2. Mobilna aplikacija

Za mobilnu aplikaciju upotrebljena je Blynk IoT platforma, zamišljena kao ploča za izradu prototipa na mobilnom uređaju. Pomoću Blynk mobilne aplikacije hardware se može kontrolisati bežično u real-time [8]. Na narednoj slici prikazan je panel Blynk aplikacije sa svim kontrolama za podešavanje funkcija LED pametne rasvete.



Slika 7. Panel Blynk mobilne aplikacije

4. REZULTATI MERENJA

Merenje potrošnje struje LED lampi izvršeno je za ceo opseg podešavanja intenziteta svetlosti od 0 % do 100 % sa korakom povećanja od 5 %.

Tabela 2. Rezultati merenja potrošnje struje LED lampi

Duty Cycle [%]	U _{sr} [V]	Hladno bele LED lampe		Full-spectrum LED lampe	
		I [A]	P [W]	I [A]	P [W]
0	0.00	0.000	0.000	0.000	0.000
5	0.60	0.069	0.041	0.063	0.038
10	1.20	0.175	0.210	0.148	0.178
15	1.80	0.285	0.513	0.235	0.423
20	2.40	0.396	0.950	0.323	0.775
25	3.00	0.500	1.500	0.404	1.212
30	3.60	0.611	2.200	0.491	1.768
35	4.20	0.724	3.041	0.580	2.436
40	4.80	0.835	4.008	0.670	3.216
45	5.40	0.945	5.103	0.751	4.055
50	6.00	1.055	6.330	0.842	5.052
55	6.60	1.158	7.643	0.933	6.158
60	7.20	1.265	9.108	1.023	7.366
65	7.80	1.367	10.663	1.108	8.642
70	8.40	1.480	12.432	1.200	10.080
75	9.00	1.592	14.328	1.292	11.628
80	9.60	1.705	16.368	1.385	13.296
85	10.20	1.805	18.411	1.470	14.994
90	10.80	1.912	20.650	1.556	16.805
95	11.40	2.011	22.925	1.650	18.810
100	12.00	2.125	25.500	1.745	20.940

5. ZAKLJUČAK

Razvijena LED pametna rasveta omogućava simulaciju Sunčevog ciklusa tokom 24h, te je zadužena za rast i razvoj biljaka kroz proces fotosinteze koji se temelji na spektru, intenzitetu svetlosti i trajanju dužine osvetljenja. Spektar koji biljke apsorbuju, predstavlja vidljivi deo svetlosti čiji spektar zračenja uključuje talasne dužine od 400 nm do 700 nm što je realizovano sa *Full-spectrum* biljnim i hladno belim LED lampama. Na zahtev korisnika preko korisničkog interfejsa, pomoću mikrokontrolera (ESP8266) upravlja se intenzitetom svetlosti i trajanjem dužine osvetljenja tokom 24h.

Kontrola intenziteta svetlosti LED lampi, zasnovana je na dužini uključenosti i isključenosti dva N-kanalna MOSFET tranzistora. Dužinom uključenosti i isključenosti tranzistora upravlja se faktorom ispune PWM signala. Informacija o vremenu potrebnom za definisanje vremenskih intervala, dobija se sa RTC-a koji je u stalnoj komunikaciji sa mikrokontrolerom. Podešavanje jačine intenziteta svetlosti kao i određivanje vremenskih intervala tokom 24h, realizovano je uz pomoć dva korisnička interfejsa. Jedan od njih predstavlja četiri tastera za ručno podešavanje navedenih funkcija, dok se povratne informacije prikazuju na OLED ekranu. Drugi interfejs predstavlja *real-time* mobilna aplikacija.

Tokom razvoja oba korisnička interfejsa, vodilo se računa o jednostavnosti rasporeda funkcija za podešavanje, tako da je korisniku logično šta je prvo potrebno podesiti od pomenutih funkcija što predstavlja prednost ove rasvete. Takođe, tokom projektovanja električne šeme vodilo se računa o stabilnosti rada komponenti kao i čuvanje istih za dugotrajni radni vek.

Na stabilnost sistema, može da utiče visoka temperatura koja se razvija prilikom emitovanja svetlosti sa LED lampi, što predstavlja nedostatak ovog sistema. Budući da snaga koju razvijaju LED lampe može da dostigne i do 45 W, LED pametna rasveta spremna je da obezbedi

dovoljno jak intenzitet svetlosti za rast i razvoj najzahtevnijih akvarijumskih biljaka. Ukoliko korisnik želi da uzgaja manje zahtevne biljke kojima nije potreban tako jak intenzitet svetlosti, pomoću korisničkog interfejsa, korisnik može da prilagodi intenzitet svetlosti za svoje potrebe. Dakle još jedna od prednosti ovog sistema, jeste ta da je sistem totalno upravljiv i omogućava korisniku da podesi sistem prema svojim potrebama u bilo kom momentu na jednostavan način.

Razvijeni sistem moguće je unaprediti proširivanjem dizajna *hardware-a*. Dodavanjem ventilatora i rebrastog aluminijumskog kućišta u sistem, višak toplote bi se lakše odvodio čime bi se smanjila temperatura u sistemu. Ventilatori bi se pozicionirali na krajeve aluminijumskog kućišta, tako da jedan uvlači hladan vazduh u kućište sistema, dok bi drugi izvlačio višak toplote iz kućišta. Na ovaj način obezbedilo bi se strujanje vazduha kroz kućište sistema sa jednog kraja na drugi.

Paljenje i gašenje ventilatora vršilo bi se pomoću tranzistora na osnovu očitane temperature sa RTC-a. U električnu šemu može se dodati Holov senzor (ACS712) za merenje struje koju povlače LED lampe. Ovaj senzor na svom izlazu daje vrednost napona koja je srazmerna izmerenoj struji. Napon sa Holovog senzora moguće je dovesti na ADC mikrokontrolera. Podatak o trenutnoj snazi prikazivao bi se na OLED ekranu i mobilnoj aplikaciji. Pomenuti hardverski dodaci imali bi uticaj na razvijanje programskog koda, čime bi se ceo sistem učinio komplikovanijim ali ne i nemogućim, te se očekuje dalji razvoj LED pametne rasvete.

6. LITERATURA

- [1] Lopez, R & E. Rundle, Light Management in Controlled Environments, Ohio: USA, 2017.
- [2] https://docs.ai-thinker.com/_media/esp8266/docs/esp-12f_product_specification_en.pdf.
- [3] Milan Radenković, Aleksandar Stevanović, Merenje vremena treperenja tastera pomoću mikrokontrolera AT89S8253, Niš: Elektronski Fakultet, Katedra za elektroniku.
- [4] https://www.linquip.com/blog/what-is-rc-circuit/#What_is_RC_Circuit.
- [5] <https://ww1.microchip.com/downloads/aemDocuments/documents/APID/ProductDocuments/DataSheets/20001422G.pdf>.
- [6] Dogan Ibrahim, Designing Embedded Systems with 32-Bit PIC Microcontrollers and MikroC, 2014.
- [7] <https://www.farnell.com/datasheets/1904442.pdf>.
- [8] <https://docs.blynk.cc/>.

Kratka biografija:



Dušan Ilić rođen je u Somboru 1994. god. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu, na katedri za električna merenja 2019. godine.
kontakt: dušan.ilic165@gmail.com

СИСТЕМИ ЗА УПРАВЉАЊЕ СОФТВЕРСКИМ КОНТЕЈНЕРИМА**SYSTEMS FOR SOFTWARE CONTAINERS MANAGEMENT**

Радош Аћимовић, Факултет техничких наука, Нови Сад

Област - ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај - У овом раду истражени су системи за управљање софтверским контејнерима: историјат њиховог развоја, поређење разних алата уз навођење њихових предности и мана и начина на који се могу заједно користити.

Кључне речи: контејнери, оркестрација, Докер, Кубернетес, AWS

Abstract - This work explores systems for software containers management: history of their development, comparison of various tools - explaining their advantages and disadvantages as well as ways in which they can be used together.

Keywords: containers, orchestration, Docker, Kubernetes, AWS

1. УВОД

Софтверски контејнери [1] представљају виртуелно окружење за извршавање програма и процеса. Унутар контејнера је могуће покретати програме, од микросервиса до великих монолитних апликација. Контејнер садржи све непходне извршне фајлове, библиотеке и конфигурационе фајлове. За разлику од традиционалних виртуелних машина, контејнери не садрже у себи читав оперативни систем, већ користе кернел оперативног система машине на којој се налазе. Због тога они користе мање ресурса него виртуелне машине.

2. ВИРТУЕЛИЗАЦИЈА

Виртуелизација [2] је процес којим се рачунарски ресурси деле на независне логичке целине. Циљ је да се повећа искориштеност физичких ресурса, да би се на тај начин повећала ефикасност и смањили трошкови. Постоји неколико врста виртуелизационих технологија. Једна од најчешћих је хардверска виртуелизација. Она креира логичку апстракцију хардверске платформе у облику виртуелних машина, контејнера и сличних технологија.

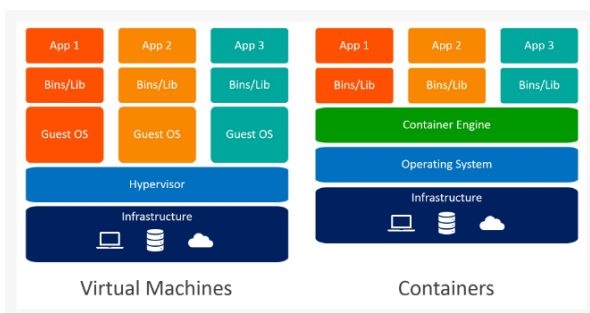
Виртуелне машине представљају виртуелну репрезентацију физичког рачунара и имају сопствени оперативни систем. Виртуелизација омогућава креирање више виртуелних машина на једној физичкој машини. Свака виртуелна машина има сопствени оперативни систем и апликације.

Контејнерске технологије користе карактеристике оперативног система физичке машине,

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Игор Дејановић, ред. проф.

да би направиле изоловано окружење. Контејнер представља посебан процес који је изолован од остатка система. Њему се ограничава приступ другим процесима, фајл систему; ограничава се количина процесорске моћи, оперативне меморије, диска итд. Добијено окружење из перспективе корисника изгледа као посебан одвојен оперативни систем. Зато контејнеризација представља подврсту хардверске виртуелизације која се називе виртуелизација на нивоу оперативног система. Предности контејнерских технологија су мања комплексност, брже перформансе и то што заузимају мање простора. На слици 1. налази се поређење виртуелних машина са контејнерима.



Слика 1. - Поређење виртуелних машина са контејнерима [3]

Осим хардверске виртуелизације, постоји још неколико врста: виртуелизација десктоп окружења, виртуелизација апликација, мрежна виртуелизација, меморијска виртуелизација, виртуелизација података и виртуелизација складишног простора (*storage-a*).

3. ИСТОРИЈАТ КОНТЕЈНЕРСКИХ ТЕХНОЛОГИЈА

Контејнерске технологије нису нови концепт. Оне су стекле широку употребу након појаве Докера, али већ дуги низ година постоје алати који нуде неки облик контејнеризације [4].

Chroot је алат који је направљен 1979. године за *Unix V7* оперативни систем. Овај алат користи системски позив који мења коренски фолдер, тако да произвољан фолдер може да постане коренски. Процес који је покренут у *chroot* окружењу, као и сви процеси који настану од њега, немају приступ ничему што је ван тог окружења. Због тога се то окружење често назива *jail* (срп. затвор).

FreeBSD jail се појавио 2000. године на *FreeBSD* оперативном систему. Осим трансформације именског простора фајл-система *FreeBSD jail* такође омогућава да виртуелно окружење тј. *jail* има

сопствене IP адресе. Процес покренут унутар окружења не може да приступи спољашњем окружењу нити може да утиче на спољашње процесе. Поврх тога, могуће је управљање корисницима на нивоу *jail*-а.

Након тога, појавили су се и алати као што су *Linux-VServer*, *Solaris* зоне и *Open VZ*, али један од битнијих помака се догађа 2006. године када компанија *Google* развија технологију **Process containers**. Циљ је да се омогући ограничење и изолација употребе системских ресурса у оквиру Линукс оперативног система. Годину дана касније је назив ове технологије промењен у „контролне групе“. Механизам контролних група користи већина модерних контејнерских алата, као што су *LXC* и *Докер*.

Још једна значајна година у развоју софтверских контејнера је 2013. година када се појавио **Докер**. Докер је понудио читав екосистем алата за управљање контејнерима и знатно поједноставио рад са контејнерима. Са појавом Докера креће масовна употреба контејнера.

4. ОСНОВНА ТЕРМИНОЛОГИЈА

Пре дискусије о конкретним алатима потребно је појаснити и усагласити терминологију [5].

Контејнерска слика у логичком смислу представља скуп података, метаподатака и конфигурација на основу којих се креира контејнер, а у физичком смислу представља обичну структуру фолдера и фајлова. Слика најчешће нема монолитну структуру, него се састоји од више **слојева**.

Репозиторијум је колекција сродних слика. То могу бити верзије слика, а могу бити и варијанте са различитим пропратним садржајем.

Container host је машина на којој се налази и извршава контејнер. Оперативни систем на тој машини се такође тако назива.

Container engine је алат који има API на кога корисници шаљу захтеве тј. команде. Команде се најчешће шаљу преко командне линије. Циљ је омогућити довољан ниво апстракције тако да вршење операција над контејнерима буде што једноставније. *Engine* у себи садржи и компоненту нижег нивоа која се назива **container runtime**. *Engine* велики број операција делегира тој компоненти, док он служи као „омотач“ који поједностављује рад са контејнерима.

Оркестрација контејнера представља технике обједињеног управљања са великим бројем контејнера.

5. КОНТРОЛНЕ ГРУПЕ И ОСТАЛИ МЕХАНИЗМИ КЕРНЕЛА

Постоји неколико механизма из Линуксовог кернела који омогућавају имплементацију контејнера.

Именски простори [6] су механизам кернела који омогућава да неки процес има сопствени поглед на систем: друге процесе, кориснике, фајл-систем, итд. На овај начин се добија изолација ресурса. Овај механизам омогућава да процеси у једном контејнеру не могу да утичу на процесе из другог контејнера. Постоји више типова именских простора, а сваки процес је у по једном од простора сваког типа. На пример, PID именски простор изолује идентификаторе процеса, мрежни именски простор

омогућава прављење мрежа изолованих од остатка система итд.

Контролне групе [7] су механизам кернела који омогућава ограничавање и мерење употребе неког ресурса. Није пожељно да један контејнер заузме читаву меморију, да заузме све CPU циклусе, заузме све уређаје, и тако даље. Због тога су потребне контролне групе. Контролне групе су организоване хијерархијски у облику стабла. Обично постоји више хијерархија за различите врсте ресурса и оне су независне једна од друге. Процес се налази у по једном чвору од сваке од тих хијерархија. Примери ресурса који могу бити ограничени контролним групама су: количина оперативне меморије, CPU циклуси итд.

Union File System (UFS) [8] нуди апстракцију над више фајл-система и омогућује да се они виде као један јединствен фајл-систем. Другим речима, могуће је имати фајлове и фолдере на различитим местима, а да они уз употребу UFS-а, са аспекта корисника буду на једном јединственом месту. Када су контејнерске технологије у питању, он омогућава слојевиту структуру контејнерске слике. Постоји више имплементација UFS-а, а једна од најпопуларнијих је *OverlayFS*.

Осим наведених механизма, кернел нуди и неке додатне механизме који служе као безбедносне контроле [9]. Пример таквог механизма су *seccomp* профили који се састоје из листа у којима се експлицитно наводи који системски позиви су дозвољени. Осим њих, користе се и механизми као што су *capabilities*, *SELinux* и *AppArmor*.

6. СТАНДАРДИ

Током развоја софтверских контејнера појавило се неколико стандарда који омогућавају интероперабилност разних класа алата за контејнере. Два најважније је креирала фондација *Open Containers initiative (OCI)* [10]. То су спецификације *image-spec* која дефинише стандардни формат контејнерске слике и *runtime-spec* која дефинише конфигурацију, извршно окружење и животни циклус контејнера.

7. CONTAINER RUNTIME

Container runtime [11] је алат преко кога се извршавају операције над контејнерима. *Container runtime*-ови се обично не користе самостално, већ се употребљавају као компоненте других контејнерских технологија. Представљају „срце“ сваког система за рад са контејнерима. Могу их је поделити на *runtime*-ове нижег нивоа, који раде директно са кернелом (покретање, заустављање контејнера итд.), и *runtime*-ове вишег нивоа, који омогућавају и основне операције са сликама, подешавања мреже итд. Најпознатији *runtime*-ови нижег нивоа су: *runc*, *railcar*, *crun*, *rkt*, *gVisor*, *Nabla* и *Kata Containers*. Најпознатији *runtime*-ови вишег нивоа су *containerd* и *cri-o*.

8. CONTAINER ENGINE

Container engine [5] представља класу алата којима се управља контејнерима. Ови алати су направљени тако да их лако могу користити и људи и програми. Раде

на високом нивоу апстракције и велики део посла делегирају *container runtime*-овима високог и ниског нивоа.

Докер нуди читаву палету алата за рад са контејнерима [12]. Између осталог, ту је и *Docker Engine* - најпопуларнији *container engine*. Он се састоји из сервера (*Docker Daemon*) и API-ја. Команде на *daemon* који управља објектима се шаљу преко *docker client* алата за командну линију. Објекат је заједнички назив за све ентитете којима рукује Докер - слике, контејнери, мреже, *volume*-и, *plugin*-и, итд. Неке од операција над сликама су: преузимање са репозиторијума, креирање нове слике, приказ информација, брисање итд. Операције над контејнерима су: креирање, покретање, заустављање, брисање, листање информација итд. Мреже и *volume*-и се могу креирати, листати, брисати и сл. Најкомплекснија операција је креирање слике. За креирање слике се користи *Dockerfile*. То је фајл који се састоји из низа инструкција преко којих се прави дефиниција слике. На пример: инструкција FROM дефинише слику на основу које се прави нова слика, инструкција COPY копира фајлове или фолдере у слику итд.

Buildah [13] је алат чији је фокус прављење слика: “од нуле”, на основу постојеће слике, помоћу *Dockerfile*-а и на основу постојећег контејнера. Осим тога, подржава и остале операције са сликама.

Skopeo [14] је алат чији је фокус рад са сликама и репозиторијумима. Најчешће се користи за конвертовање слика из једног формата у други, као и за преузимање информација о сликама без преузимања читавих слика.

Podman [15] је алат чији је фокус управљање контејнерима и *pod*-овима (група контејнера), а може и да врши операције са сликама. *Podman* управља читавим животним циклусом контејнера, од креирање до уништавања. Такође, може да управља мрежним подешавањима и подешавањима *storage*-а.

При избору *container engine*-а, постоје две опције. Прва је употреба Докера, а друга је употреба *Podman*-а заједно са *Buildah*. Боља опција је Докер, јер има већи број *feature*-а, додатне алате, а такође, има и већу заједницу корисника. *Skopeo* се може користити као помоћни алат у оба случаја.

9. КОНТЕЈНЕРСКИ РЕГИСТРИ

Контејнерски регистри [16] представљају скуп репозиторијума на којима се налазе слике. Они омогућавају складиштење слика, као и дељење тих слика са другим корисницима и системима. Осим тога, многи регистри омогућавају безбедносне контроле као што су аутентификација и ауторизација корисника и програма, скенирање слика итд. Регистри могу бити приватни и јавни. Осим тога, могу се поделити на *third-party* регистре и *self-hosted* регистре. Неки од најпознатијих регистара су: *Docker Hub*, *Docker Registry*, *Amazon ECR*, *Google Artifact Registry*, *IFrog Container Registry* итд.

10. ОРКЕСТРАЦИЈА КОНТЕЈНЕРА

Оркестрација контејнера [5] представља аутоматизовано управљање контејнерима. Алати који врше орке-

страцију се називају оркестратори контејнера. Операције као што су скалирање контејнера и машина, повезивање великог броја контејнера у мрежу и сл. би биле веома споре и неефикасне без оркестратора.

Docker Compose [17] је Докеров алат који служи за оркестрацију контејнера у оквиру једне *host* машине. Користи се углавном за прављење окружења за развој и тестирање софтвера. Може се користити и на продукцији ако се користи само једна *host* машина. Окружење се дефинише декларативно у *YAML* фајлу, који се најчешће назива *docker-compose.yml*. Све што може да се уради ручно преко Докерових команди и параметара - прављење контејнера, мрежа, дефинисање ресурса, безбедносних подешавања и тако даље - то исто може да се уради и преко овог алата, али на много бржи и једноставнији начин.

Podman Compose [18] је алат из истог пројекта као *Podman*. За креирање окружења користи фајлове формата као за *Docker Compose*. Команде су такође веома сличне. Битна разлика је у томе што се контејнери које дефинише *docker compose* фајл стављају у *pod*.

Swarm [19] је Докеров алат за оркестрацију контејнера на више *host* машина. Свака машина у том кластеру се назива чвор. Постоје два типа чворова. Радници су чворови на којима се покрећу контејнери. Менаџери управљају кластером - оркестрирају кластер тако што додељују послове радницима, управљају конфигурацијом итд. Основни објекат са којим ради *Swarm* назива се сервис. Он дефинише која слика треба да се користи за покретање контејнера, које портове треба отворити, на коју мрежу ставити контејнер, дефинише ограничења ресурса и тако даље. Веома често је потребно имати више инстанци (реплика) истог контејнера. У том случају ће сервис да има више задатака. Задаци представљају најмање јединице управљања у *Swarm*-у. Задаци се распоређују по расположивим радницима и могу имати максимално један контејнер.

Кубернетес је алат направљен од стране компаније *Google*. Кубернетес кластер [20] се састоји од контролне равни и чворова. Контролна равна обједињује управљачке компоненте кластера. Чворови су машине на којима се покрећу контејнери. Контролна равна се састоји од неколико независних компонента. Свака компонента је специјализована за одређени посао и налази се у контејнеру. Компоненте могу бити на једној машини, или на више њих. Једна компонента може да има више инстанци. Најважнији објекти у Кубернетесу су [21]:

- *Pod* је основна јединица управљања у Кубернетес кластеру. Може да садржи један или више контејнера.
- Сервис је објекат коме се додељују IP адресе и DNS имена. Захтеве који буду послати на те адресе и портове он прослеђује *pod*-овима, тачније, врши расподелу оптерећења и прослеђује саобраћај неком од *pod*-ова наведеним у дефиницији сервиса.
- *Replica set* је објекат који дефинише број реплика *pod*-ова и предузима акције да број *pod*-ова увек буде једнак том броју. То омогућава једноставно скалирање.

- *Deployment* је објекат који омогућава ажурирање верзије апликације без *downtime*-а. *Deployment* се користи искључиво за *stateless* апликације.
- *Stateful set* је објекат који управља *pod*-овима у којима се покрећу *stateful* апликације. Иако сви *pod*-ови у *stateful set*-у настају из истог дефинисаног шаблона, сваки од њих има свој посебан идентитет и стање.

Amazon ECS [22] је алат за оркестрацију контејнера који је директно интегрисан у инфраструктуру *AWS cloud*-а. Састоји се од контролне равни и равни података. ECS је сервис којим у потпуности управља *AWS*, тако да је контролна раван у потпуности „брига“ *AWS*-а. Раван података у којој се покрећу контејнери се дефинише при креирању кластера. Инфраструктура за раван података се обезбеђује из три врсте „извора“: *Fargate* (омогућава да ECS аутоматски добије инфраструктуру за покретање контејнера), *EC2* инстанци и спољних сервера који су ван *AWS Cloud*-а. Основна јединица управљања у ECS-у је задатак. Може да има један или више контејнера. Прво је потребно креирати дефиницију задатка. Дефиниција задатка је независна од кластера. У неком одређеном кластеру је затим потребно креирати задатак или сервис. Сервис може да има један или више реплика неког задатка који се креира на основу дефиниције задатка. Највећи број функционалности нуди Кубернетес. Истовремено, он је најкомплекснији алат. Будући да има велику заједницу корисника, постоји велики број готових решења за ствари опште намене, као што су *CI/CD*, мониторинг итд.

11. ЗАКЉУЧАК

Предмет истраживања овог мастер рада су били системи за управљање софтверским контејнерима. Први циљ овог рада је био да се да преглед свих релевантних технологија у овој области - њихових предности и мана, када које треба користити и како се међусобно допуњују. Други циљ овог рада је био да се опише како поједини алати функционишу интерно.

Употреба софтверских контејнера је и даље у успону и све више организација их користи. Постојећи алати се унапређују, а истовремено се појављују и многи нови алати.

12. ЛИТЕРАТУРА

- [1] <https://www.netapp.com/devops-solutions/what-are-containers/>, последњи приступ: октобар 2022.
- [2] <https://kompjuterar.com/pojam-virtuelizacije-i-osnovni-termini/>, последњи приступ: октобар 2022.
- [3] <https://www.weave.works/blog/a-practical-guide-to-choosing-between-docker-containers-and-vms>, последњи приступ: октобар 2022.
- [4] <https://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016>, последњи приступ: октобар 2022.
- [5] <https://developers.redhat.com/blog/2018/02/22/container-terminology-practical-introduction>, последњи приступ: октобар 2022.
- [6] *Mastering Linux Kernel Development*, Raghu Bharadwaj, 2017, стр. 99-106
- [7] <https://man7.org/linux/man-pages/man7/cgroups.7.html>, последњи приступ: октобар 2022.
- [8] <https://medium.com/@knoldus/unionfs-a-file-system-of-a-container-2136cd11a779>, последњи приступ: октобар 2022.
- [9] https://www.suse.com/c/rancher_blog/introduction-to-container-security/, последњи приступ: октобар 2022.
- [10] <https://opencontainers.org/about/overview/>, последњи приступ: октобар 2022.
- [11] <https://www.capitalone.com/tech/cloud/container-runtime/>, последњи приступ: октобар 2022.
- [12] <https://www.whizlabs.com/blog/docker-architecture-in-detail/>, последњи приступ: октобар 2022.
- [13] <https://github.com/buildah/blob/main/docs/tutorials/01-intro.md>, последњи приступ: октобар 2022.
- [14] <https://github.com/containers/skopeo>, последњи приступ: октобар 2022.
- [15] <https://github.com/containers/podman>, последњи приступ: октобар 2022.
- [16] <https://www.redhat.com/en/topics/cloud-native-apps/what-is-a-container-registry>, последњи приступ: октобар 2022.
- [17] <https://gabrieltanner.org/blog/docker-compose/>, последњи приступ: октобар 2022.
- [18] <https://phoenixnap.com/kb/podman-compose>, последњи приступ: септембар 2022.
- [19] <https://docs.docker.com/engine/swarm/key-concepts/>, последњи приступ: октобар 2022.
- [20] <https://kubernetes.io/docs/concepts/overview/components/>, последњи приступ: октобар 2022.
- [21] <https://medium.com/devops-moj/kubernetes-objects-resources-overview-introduction-understanding-kubernetes-objects-24d7b47bb018>, последњи приступ: октобар 2022.
- [22] <https://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html>, последњи приступ: октобар 2022.

Кратка биографија:

Радош Аћимовић рођен је 12. априла 1995. године у Суботици. На Факултет техничких наука у Новом Саду се уписао 2014. године - смер: Софтверско инжењерство и информационе технологије. Дипломирао је 2018. године и исте те године уписао мастер студије на истом факултету - смер: Софтверско инжењерство и информационе технологије, усмерење Електронско пословање. Положио је све испите предвиђене планом и програмом.

Контакт: radosacimovic@gmail.com

UPOREDNA ANALIZA MEĐUSPRATNIH SPREGNUTIH KONSTRUKCIJA: TIPOVI MOŽDANIK I ROBUSNOST**COMPARATIVE ANALYSIS OF THE COMPOSITE FLOOR STRUCTURES: TYPES OF SHEAR STUDS AND ROBUSTNESS**Slobodan Miletić, *Fakultet tehničkih nauka, Novi Sad***Oblast – GRAĐEVINARSTVO**

Kratak sadržaj – Rad se sastoji iz proračuna dva različita tipa međuspratnih konstrukcija, čiji je rezultat upoređivanje potrebnih količina materijala za njihovo izvođenje i odabir povoljnije varijante, i obezbeđivanje robusnosti konstrukcije kao osiguranje od progresivnog kolapsa.

Ključne reči: *Spregnute konstrukcije, moždanici, robusnost konstrukcija, progresivni kolaps.*

Abstract – The thesis consists calculation of two different types of composite floor structures, the result is comparison of the required quantities of materials for their execution and the selection of a more favorable variant, and ensuring the robustness of the construction as an insurance against progressive collapse.

Keywords: *Composite structures, shear studs, structural robustness, progressive collapse*

1. ROBUSNOST OBJEKATA OD ČELIČNIH OKVIRA

Robusnost je sposobnost konstrukcije da izdrži događaje kao što su požar, eksplozije, udari ili posledice ljudske greške, a da ne bude oštećena u meri koja je nesrazmerna prvobitnom uzroku. U čeličnim okvirnim konstrukcijama, robusnost je u velikoj meri obezbeđena projektovanjem odgovarajućih veza. Klasifikacija objekta se vrši prema: tipu objekta, popunjenosti i veličini objekta. Prema BS EN 1990 i BS EN 1991-1-7, uzimajući u obzir i aneks A, objekti se dele u četiri klase: 1, 2a, 2b i 3[1].

1.1. Objekti klase 1

Objekte koji spadaju u klasu 1, projektovane i izgrađene u skladu sa EN 1990- EN 1999, pri normalnoj upotrebi, nije potrebno osiguravati od progresivnog kolapsa. Ipak, preporuka je objekte izvedene od čelika u ramovskom sistemu osigurati minimalnim horizontalnim vezivanjem. Ovo se postiže projektovanjem veza stub-greda na normalnu zatežuću silu od 75 kN, koju ne treba kombinovati sa ostalim uticajima [1].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Đorđe Jovanović, dipl.grad.inž.

1.2. OBJEKTI KLASA 2A

Robusnost objekata, za klasu 2a, obezbeđuje se horizontalnim povezivanjem. Horizontalno povezivanje omogućava razvoj lančane reakcije i zadržava stubove u projektovanom položaju. Princip obezbeđivanja horizontalnog vezivanja dopušta gredama da prenesu opterećenja formiranjem mreže oko oštećenih delova konstrukcije.

Horizontalno vezivanje, projektovano prema Evrokodu, nema zahteva vezanih za duktilnost spojeva ili kapacitet rotacije spoja. Svaki vezni element, uključujući krajnje veze mora biti sposoban da izdrži projektovanu zatežuću silu T . Unutrašnji vezni elementi se projektuju na zatežuću silu T_i , dok se obodni vezni elementi projektuju na zatežuću silu T_p :

$$T_i = 0,8 \cdot (g_k + \psi q_k) \cdot s \cdot L \geq 75 \text{ kN} \dots (1)$$

$$T_p = 0,4 \cdot (g_k + \psi q_k) \cdot s \cdot L \geq 75 \text{ kN} \dots (2)$$

Zatežuće sile se računaju samo za elemente koji prenose opterećenje od poda međuspratne konstrukcije, ostali vezni elementi se projektuju na minimalnu zatežuću silu od 75 kN. Da bi stubovi bili zadržani u projektovanom položaju treba ih povezati gredama u oba pravca. Horizontalno povezivanje mora biti obezbeđeno u svim unutrašnjim gredama, obodnim spratnim gredama i obodnim krovnim gredama [1].

1.3. Objekti klase 2b

Za obezbeđivanje robusnosti objekata iz klase 2b, mogu se koristiti tri metode, ili njihova kombinacija. Metode su:

- povezivanje (horizontalno i vertikalno),
- zamišljeno uklanjanje elementa,
- ključni element.

Za objekte klase 2b, pri horizontalnom vezivanju, primenjuju se ista pravila kao i za objekte klase 2a. Vertikalno vezivanje omogućava redistribuciju sila kroz kompletnu strukturu objekta, tako da sile mogu da se prenose i kroz elemente koji su udaljeni od lokalnog oštećenja. Vertikalno povezivanje se obezbeđuje dovoljnom otpornošću na zatežuće sile u montažnim nastavcima stubova. Reakcije iz greda za proračunavanje potrebnih sila za vertikalno vezivanje se uzimaju iz normalnih kombinacija dejstava, ne koriste se incidentna dejstva.

Prednosti metode zamišljenog uklanjanja su u tome što umesto praćenja propisanih pravila, razmatraju se konkretniji scenariji štete, pri čemu se od projektanta zahteva procena površine oštećenja. Prednost ove metode je u tome što, ako konstrukcija ima razumno male raspone

greda i ako je dobro povezana, onda zamišljeno uklanjanje nudi projektantu priliku da zadovolji pravila robusnosti konstrukcije uz prihvatanje lokalnih oštećenja. Zamišljeno uklanjanje se sprovodi tako što se uklanja jedan po jedan noseći element. Za svaki pojedinačno uklonjeni element konstrukcija globalno mora ostati stabilna.

Noseći element, čije uklanjanje narušava globalnu stabilnost konstrukcije mora biti projektovan kao ključni element.

Razlika metode ključnog elementa sa prethodnim metodama, je u tome što su prethodne metode fokusirane na limitiranje širenja oštećenja ili kolapsa usled oštećenja nosećeg elementa, dok ova metoda sprečava oštećenje nosećeg elementa usled incidentnog dejstva, do mere da i dalje može da vrši noseću funkciju. Zahtevi za projektovanje ključnog elementa su dati u BS EN 1991-1-7, A.8, a to su:

- ključni element mora biti sposoban da izdrži incidentno dejstvo u horizontalnom i vertikalnom pravcu (ne oba pravca u isto vreme),
- preporučena vrednost A_d za zgrade je 34 kN/m^2 ,
- incidentno opterećenje treba da bude aplicirano na ključni element i svaku priključnu komponentu uzimajući u obzir čvrstoću spojnih komponenti i njihovu vezu,
- incidentno projektno opterećenje treba da bude primenjeno u skladu sa izrazom (6.11b) iz EN 1990.

Za neke ključne elemente prikladno je razmatrati još neka incidentna dejstva koja bi se mogla dogoditi, kao što je na primer udar vozila u spoljašnji stub [1].

1.4. OBJEKTI KLASSE 3

Razlika u strategijama robusnosti između objekata klase 2b i klase 3 je što se za klasu 3 radi sistematska procena rizika. Svrha procene rizika je da se utvrdi postoje li scenariji opasnosti koji imaju neprihvatljiv nivo rizika, i ako postoje, da se predlože postupci za ublažavanje istih. U BS EN 1991-1-7, B.4, navodi se da analiza rizika, za objekte klase 3, treba da se sastoji iz opisnog i numeričkog dela. Definicija procene rizika treba da obuhvati svrhu procene rizika, vremenski period koji treba uzeti u obzir i vrste opasnosti. U većini slučajeva, razumno je, iz procene rizika isključiti namerne ili zlonamerne opasne radnje.

Verovatnoća svakog hazarda treba da se proceni i da se dodeli unapred određenoj kategoriji verovatnoće. Za procenu posledica koristi se inženjersko prosuđivanje, iskustvo i približni proračuni, a na osnovu posledica, određuje se ozbiljnost hazarda. Procena rizika može se meriti u različitim jedinicama, često se koristi procena količine oštećenja konstrukcije ili procena broja žrtava. Završna faza sistematske procene rizika je prihvatanje preostalog rizika i izveštavanje o nalazima.

Izveštaj treba da obuhvati sve opasnosti i njihov pridruženi nivo rizika, sa objašnjenjima na osnovu kojih se rizici smatraju prihvatljivim. Takođe se mogu predložiti dodatne mere radi daljeg smanjenja rizika. Rezultati procene rizika će se vratiti u proces donošenja odluka za projektovanje i rad objekta.

Svi izvori podataka, pretpostavke i neizvesnosti u proceni treba da budu uključeni u izveštaj [1].

1.5. TRANSFERNE GREDE

Transferna greda je greda koja podupire jedan ili više stubova, jasno je da je transferna greda najkritičniji element u konstrukciji. Prilikom projektovanja konstrukcija za odupiranje incidentnim dejstvima, potrebno je prenosne grede, njihove spojeve i elemente koji ih podupiru, pažljivo razmotriti. Preporuka je da se za objekte iz klase 1, transfernim gredama obezbedi minimalno horizontalno vezivanje, tj. da greda i njene veze mogu da izdrže normalnu zatežuću silu od 75 kN . Za objekte iz klase 2a, transferne grede se za zadovoljenje robusnosti konstrukcije, projektuju na zatežuću silu koja se izračunava posebno za unutrašnje i obodne grede, sa tim što se u izraz dodaje i polovina reakcije stubova koje greda podupire. Reakcija stuba koja ulazi u proračun je iz incidentne kombinacije dejstva.

Kao što je napomenuto, u klasi 2b imamo tri metode za obezbeđivanje robusnosti konstrukcije: Horizontalno i vertikalno vezivanje, zamišljeno uklanjanje i ključni element. Transferna greda može biti uključena u sve tri metode, mada se najčešće koristi vezivanje ili ključni element. Zamišljeno uklanjanje često nije održivo rešenje za okvirne čelične konstrukcije. Ako se primenjuje metoda vezivanja, jednačine se moraju modifikovati kao i kod transfernih greda u klasi 2a i mora se obezbediti vertikalno vezivanje transferne grede sa stubovima koji se na nju oslanjaju.

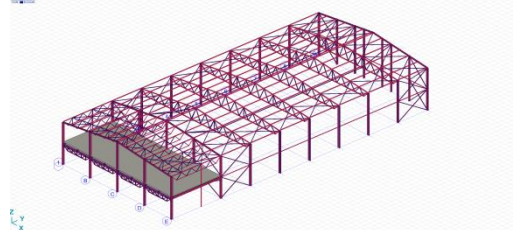
Umesto primene strategije vezivanja, transferna greda može biti dizajnirana i kao ključni element. U slučaju kada je transferna greda dizajnirana kao ključni element, kao ključni elementi moraju biti dizajnirani i stubovi na koje se transferna greda oslanja. Kao minimalan zahtev, transferna greda u objektima klase 3, treba da se projektuje u skladu sa smernicama iz klase 2b.

U nekim objektima iz klase 3, gubitak prenosne grede bi imao katastrofalne posledice, tako da bi strožije odredbe o robusnosti bile opravdane.

Proces procene rizika za objekte klase 3 će identifikovati da li su potrebne bilo kakve dodatne mere u vezi sa transfernom gredom [1].

2. OBJEKAT ZA KOJI SE PROVERAVAJU MEĐUSPRATNE KONSTRUKCIJE I ROBUSNOST

Objekat, čiji su delovi razmatrani u radu, je čelična hala dimenzija $24\text{m} \times 52\text{m}$, visina u slemenu $9,5\text{m}$. Lokacija objekta je Novi Sad. U prvom delu objekta (prednjih 10m) se nalazi magacin, iznad magacina je kancelarijski prostor. U drugom delu objekta (42m) je proizvodni pogon.



Slika 1. 3D model objekta.

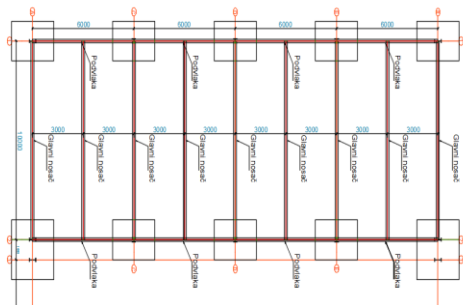
Za međuspratnu ploču iznad magacinskog prostora analizirana su dva tipa međuspratnih konstrukcija:

- spregnuta međuspratna konstrukcija sa moždanicima,

- međuspratna konstrukcija tipa “Slim Floor”.

Pored redovnih opterećenja, na ploči, zahtevano je dodatno stalno opterećenje od 1,41 kN/m² i korisno opterećenje od 3,8kN/m².

3. SPREGNUTA PLOČA SIDRENA NA KRAJEVIMA POMOĆU MOŽDANIKA

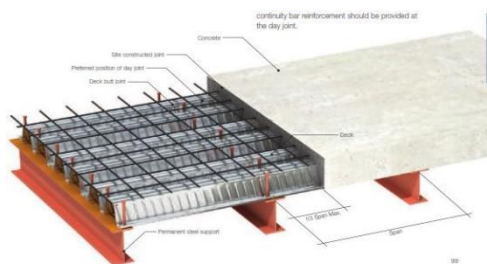


Slika 2. Dispozicija ploče sa moždanicima.

Betonska ploča je debljine d=130mm, sa trapezastim limom „ComFloor 51“, visine 51mm, debljine 0,86mm. Ploča je raspona 3m, u fazi izvođenja vrši se podupiranje u polovini raspona. Profilisani lim je kontinualan preko više raspona. Spregnuta ploča je kontinualna, prema EN1994-1-1 može biti razmatrana na dva različita statička sistema, kao jednostavno oslonjena ploča i kao alternativno kontinualna ploča. Ploča se proračunava za 2 faze: faza u toku izvođenja ploče i za spregnuto stanje. Profilisani lim preuzima sile zatezanja i služi kao izgubljena oplata prilikom betoniranja ploče.

Da bi se ostvarilo sprezanje čelika i betona, podužne smičuće sile koje se javljaju na spoju dva materijala moraju biti prihvaćene, tj. mora postojati smičući spoj. Sprezanje može biti potpuno i delimično. Podužna smičuća otpornost može se povećati sidrenjem na krajevima (pomoću moždanika) i uzimanjem u obzir armature izvedene u rebrima profilisanog lima (EN1994 - 1-1 9.7.3(10)). Računska otpornost moždanika $P_{pb,Rd}$, zavarenog kroz profilisani lim, jednaka je manjoj vrednosti od računске otpornosti moždanika ili računске otpornosti lima. Pri proračunu graničnog stanja upotrebljivosti računaju se ugibi u fazi izvođenja i u spregnutoj fazi. Za ugib u spregnutoj fazi moraju se u obzir uzeti i koncentrisane sile na mestima u kojima je ploča bila podupreta prilikom izvođenja.

4. SPREGNUTA GREDA SA MOŽDANICIMA



Slika 3. Spregnuta međuspratna konstrukcija tip 1 [3].

Spregnuti nosač je raspona 10m, na međusobnom rastojanju 3m. Analizirano je stanje u toku izvođenja konstrukcije i spregnuto stanje. Za gredu je usvojen

čelični profil IPE400 u kvalitetu S275, za sprezanje su usvojeni moždanici tipa „Nelson“, prečnika 19mm i visine 100mm. Raspored moždanika usvojen je prema potrebama za sidrenje ploče (jedan moždanik u svakom rebu profilisanog lima). Nosač u fazi montaže nije otporan na bočno torziona izvijanje, tako da se mora vršiti podupiranje nosača. Kako će se vršiti podupiranje čeličnog nosača sva opterećenja će primiti spregnuti presek. Za merodavnu nosivost moždanika se uzima manja vrednost od sledeće dve:

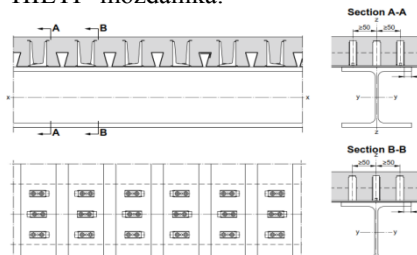
$$P_{Rd}^{(1)} = \frac{0,8 * f_u * (\pi * d^2 / 4)}{\gamma_v} \quad (3)$$

$$P_{Rd}^{(2)} = \frac{0,29 * \alpha * d^2 * \sqrt{f_{ck} * E_{cm}}}{\gamma_v} \quad (4)$$

Prva jednačina predstavlja graničnu nosivost prilikom otkazivanja tela moždanika, dok je druga jednačina granična nosivost pri otkazivanju betona. Za usvojeni broj moždanika osigurana je potpuna smičuća veza, tako da se otpornost na savijanje ne mora redukovati. Nosač je statičkog sistema proste grede, za predviđeno opterećenje pritisnuta je gornja nožica koja je pridržana, tako da nije potrebno dokazivati otpornost na bočno torziona izvijanje u spregnutoj fazi. Zbog podupiranja nosača u fazi izrade spregnute međuspratne konstrukcije, nosač u ovoj fazi neće imati ugib. Ukupan ugib zadovoljava uslov, tj. manji je od L/250. Za statički sistem proste grede ne zahteva se kontrola širine prslina.

4.1. VARIJANTA SA “HILTI” MOŽDANICIMA

Hilti moždanici se sa nosačem spajaju ekserima, koji se u čelični nosač upucavaju specijalnim pištoljem sa kapislama. Tip “HILTI” moždanika se usvaja na osnovu visine profilisanog lima i debljine betonske ploče. Pošto se moždanici apliciraju preko profilisanog lima, nosivost moždanika se mora redukovati. Faktor redukcije zavisi od geometrije i orijentacije profilisanog lima. Od debljine materijala u koji se moždanik upucava, zavisi tip kapisle koja se mora koristiti. Kako je nosivost HILTI moždanika skoro tri puta manja od nosivosti moždanika sa glavom, u jedno rebro profilisanog lima moraju se postaviti tri “HILTI” moždanika.

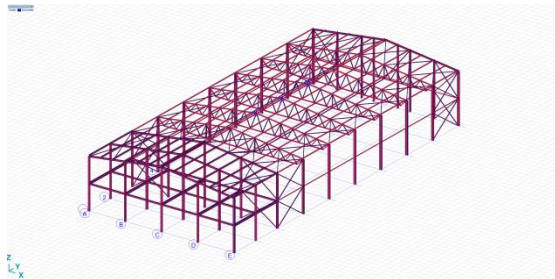


Slika 4. Dispozicija HILTI moždanika [4].

5. ČELIČNA GREDA ZA SLIMFLOOR KONSTRUKCIJU

U prvoj varijanti “Slim Floor” konstrukcije, za čelični gredni nosač pretpostavljen je profil 300ASB196, u kvalitetu S355. Nosač je raspona 10m, na međusobnom rastojanju od 6m. Nakon proračuna, ustanovljeno je da nosač ne zadovoljava granična stanja upotrebljivosti, za zadata opterećenja. Nakon toga, predložena je druga varijanta sa stubovima u sredini raspona (međuspratne

grede raspona 5m) uz osiguranje konstrukcije od progresivnog kolapsa.



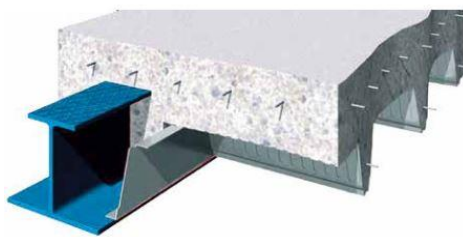
Slika 5. 3D Model objekta za "Slim Floor".

Usvojen je profil 280ASB74 u kvalitetu S275, na međusobnom rastojanju od 6m. Prilikom izvođenja međuspratne konstrukcije pojaviće se situacija u kojoj je ploča sa jedne strane ASB profila izvedena, a sa druge nije, za tu situaciju se mora proveriti poprečno savijanje donje flanše. Pored poprečnog savijanja donje flanše, u fazi izvođenja, konstrukcija se proverava još i na izvijanje izazvano torzijom, savijanje nosača za uravnoteženo opterećenje (ploča izbetonirana sa obe strane ASB profila). Za spregnutu fazu, proverava se moment otpora spregnutog preseka i uzdužna otpornost nosača na smicanje. Ugibi se proveravaju posebno za dve različite faze. Kako je u ovom slučaju ostvarena samo delimična smičuća veza ($F_{sb} < R_c$), otpornost na savijanje se mora redukovati[5].

6. SPREGNUTA PLOČA ZA "SLIM FLOOR" KONSTRUKCIJU

Za spregnutu ploču "Slim floor" konstrukcije usvojen je duboko profilisani lim „ComFlor 210“. U svako rebro duboko profilisanog lima postavlja se armaturna šipka prečnika 10mm, koja se koristi i za sidrenje ploče na krajevima kako bi se poboljšala podužna smičuća otpornost.

Prema članu 9.8.2(4), EN 1994-1-1 ugibe nije potrebno proveravati, zato što su uslovi ograničenja vitkosti ploče zadovoljeni ($L/d < 30$ za unutrašnja polja kontinualne ploče i $L/d < 26$ za spoljašnja polja kontinualne ploče). U obe ploče iznad trapezastog čeličnog lima se postavljaju armature mreže Q335, zbog uslova iz EN1994-1-1 za pukotine u betonu. Prema EN1994-1-1 minimalna površina poprečnog preseka armature iznad rebra profilisanog čeličnog lima ne sme biti manja od 0,4% površine betona iznad rebra, za izvođenje sa podupiranjem.



Asymmetric Slimflor® beam (ASB), which is a hot rolled section with a narrower top flange than bottom flange.

Slika 7. "Slim floor" konstrukcija [6].

7. ZAKLJUČAK

Sa aspekta utrošenog materijala, povoljnija je "Slim floor" konstrukcija.

Tabela 1. Količine materijala za razmatrane međuspratne konstrukcije:

POZICIJA	Spregnuta međuspratna konstrukcija sa moždanicima	"Slim floor" međuspratna konstrukcija sa gredama raspona 5m
ČELIČNA KONSTRUKCIJA	23 387.29 kg	18 114.04 kg
BETON U MEĐUSPRATNOJ KONSTRUKCIJI	28.8 m3	28.8m3
ČELIČNI TRAPEZASTI LIM	3120 kg	3840 kg
ARMATurna MREŽA U PLOČI	858 kg	858 kg
ARMATURNE ŠIPKE U REBRIMA	708.67 kg	284.09 kg
MOŽDANICI	576 "Nelson" ili 1728 "Hilti"	/

Ako poredimo ove dve varijante po brzini izvođenja, ponovo je povoljnija varijanta sa "Slim Floor" međuspratnom konstrukcijom. Za nju je potrebno montirati manju količinu čelika i nema upucavanja moždanika.

8. LITERATURA

- [1] A G J Way, Structural robustness of steel framed buildings, SCI Publication P391
- [2] Boris Androić, Darko Dujmović, Ivan Lukačević: Projektiranje spregnutih konstrukcija prema Eurocode 4, „I.A. PROJEKTIRANJE“ Zagreb
- [3] Corus International, ComFlor decking system, www.comflor.com
- [4] HILTI, Hilti X-HVB system Solutions for shear connections, www.hilti.com
- [5] Dennis Lam, Xianghe Dai, Ulrike Kuhlmann, Jochen Raichle, Matthias Braun: Slim-floor construction – design for ultimate limit state
- [6] Mark Lawson, Philippe Beguin, Renata Obiala, Matthias Braun: Slim-floor construction using hollow-core and composite decking systems
- [7] R M Lawson, D L Mullett, J W Rackham: Design of Asymmetric Slimflor Beams using Deep Composite Decking, SCI PUBLICATION P175
- [8] EVROKOD 4, PRORAČUN SPREGNUTIH KONSTRUKCIJA OD ČELIKA I BETONA

Kratka biografija:

Slobodan Miletić rođen je u Novom Sadu 1990. god. Master rad na Fakultetu tehničkih nauka iz oblasti Građevinarstvo – Spregnute konstrukcije odbranio je 2022.god. kontakt: slomil@hotmail.com

**ПРОЈЕКАТ МОСТА ДУШАНОВАЦ-ПРАХОВО НА СТАЦИОНАЖИ КМ 6+734.824
ЛЕВО И КМ 6+801.266 ДЕСНО****DESIGN OF BRIDGE DUŠANOVAC-PRAHOVO AT THE STATION 6+734.824 LEFT AND
6+801.266 RIGHT**

Маја Драгичевић, Игор Цолев, Факултет техничких наука, Нови Сад

Oblast – ГРАЂЕВИНАРСТВО

Kratak sadržaj – У овом раду приказана је анализа армиранобетонске-преднапрегнуте конструкције са монтажним начином градње распонске конструкције, према важећим европским стандардима. Усвојеним статичким системом остварена је монолитна веза средњих стубова и распонске конструкције, чиме је избегнут значајан број лежишта. Урађен је статички прорачун, димензионисање и усвајање арматуре. Поузданост прорачуна је контролисана упоредним креирањем нумеричког рачунског модела у два различита програмска пакета.

Кључне речи: армиранобетонска-преднапрегнута конструкција, монтажни „Т” носачи, европски стандарди.

Abstract – In this paper, a reinforced concrete prestressed structure with the assembly method of building a span structure, according to European standards, is analysed. The static structural system adopted a monolithic connection of the middle columns and the main girder, which avoided a significant number of bearings. Static calculation, dimensioning and design of reinforcement were done. Reliability of the calculation was controlled by the comparison of the numerical calculation models, which were developed in two different software packages.

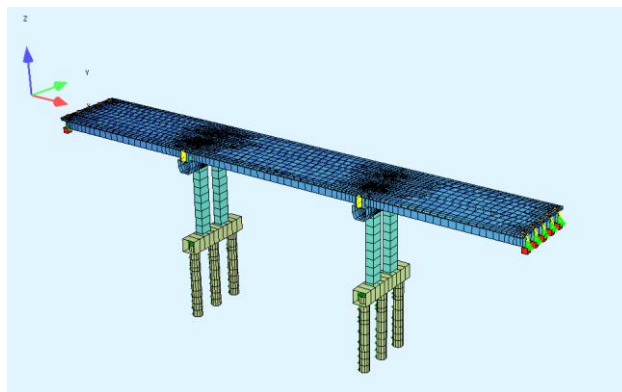
Keywords: reinforced concrete-prestressed structure, prefabricated “T” beams, European standards.

1. УВОД

Реконструкција и изградња постојеће путне мреже (државни пут II реда број 168 и државни пут II реда број 400) у државни пут I реда, од државног пута I реда број 35 од чвора Душановац до краја ко Прахово. На овој траси Душановац-Прахово пројектованој за брзину од 50 km/h, уместо постојећег путног прелаза у нивоу, пројектован је нови друмски надвожњак на стационажи км 6+734.824 лево и км 6+801.266 преко железничке пруге.

2. ТЕХНИЧКИ ОПИС**2.1. Саобраћајни услови**

Друмски надвожњак је на стационажи км 6+734.824 лево и км 801.266 десно преко изливне железничке пруге под углом 90°. Попречни нагиб моста је 2,5%. На тој деоници усвојене су следеће димензије попереч-



Слика 1. 3Д модел конструкције ног профила пута. Профил чине две коловозне траке са по две саобраћајне траке у сваком смеру раздвојене разделним острвом:

- ширина коловоза 2x6,50 m
- ширина разделног острва 1,50 m
- ширина саобраћајне траке 3,25 m
- ширина возне траке 3,0 m
- ширина ивичне траке 0,25 m
- ширина банке 2x2,75 m

2.2. Диспозиција

Диспозиционим решењем пројектован је мост рамовског система. Укупна дужина моста је 65 m и састоји се од три распона чија дужина износи 20 + 25 + 20 m. Карактеристичан попречни пресек распонске конструкције садржи пешачку стазу ширине 150 cm, одбојну ограду са сигурносним простором укупне ширине 2,75 m, возне траке, зауставне траке, ивичне траке и појас између одбојних ограда у зони разделног појаса. С десне стране простор за смештај ивичњака и одбојне оgrade је укупне ширине 65 cm. Ширина моста једне траке саобраћајнице је усклађена са ширином коловоза и захтевима пројектног задатка и износи:

- ширина коловоза: 6,50 m
- ширина пешачких стаза: 2,75 m
- ширина ревизионих стаза: 0,65 m
- укупна ширина моста једне траке: 9,00 m
- укупна ширина моста обе траке: 20,0 m

Горњи строј конструкције састоји се од 5 монтажних армиранобетонских „Т” носача, међусобно повезаних коловозном плочом дебљине $d = 20$ cm. У фази монтаже, носачи су статичког система просте греде. Плоча се над средњим стубовима лије заједно са поперечним носачима и везује круто са главним

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Игор Цолев, доцент.

носачима, чиме се у фази експлоатације добија интегрални рамовски систем и на тај начин избегавају слаба места (дилатације, лежишта, продор воде), што олакшава одржавање моста и продужава трајност [1]. На обалним стубовима распонска конструкција се ослања на лежишта. Попречни пресек носача је „Г” облика, висине $d = 1,35$ m, монтажни носач $+ 0,20$ m плоча и ширина ребра 40 cm. Монтажни носачи распона 20,0 m утежу се са по укупно 3 кабла $7\text{Ø}15,2$, док код распона 25,0 m, први кабел је $12\text{Ø}15,2$, а други и трећи $7\text{Ø}15,2$.

Средњи стубови моста састоје се од по два стуба, димензија $1,50 \times 1,00$ m, висине 7,11 m. Изнад средишњих стубова лије се лежишна греда, димензија $1,80 \times 1,55$ m која са попречним носачима горњег строја, димензија $0,80 \times 1,65$ m, чини круту везу конструкције. Дужина лежишних греда и поречних носача је 9,20 m. Средњи стубови су фундирани на три шипа $\text{Ø}120$ cm, дужине 8,90 m који су међусобно повезани наглавном гредом димензија $1,70 \times 1,80$ m, дужине 9,60 m. Крајње стубове моста чини ослоначка греда са попречним носачима, крилним зидовима и парапетом испред шипова. Ослоначка греда са попречним носачима се изводи из две фазе бетонирања. Прва фаза обухвата бетонирање до кота ослањана главних носача, а у другој фази се бетонира попречни носач заједно са коловозном плочом и главним носачима ради остваривања круте везе. Ослоначка греда крајњег стуба, димензија $2,00 \times 1,90$ m, се ослања на три шипа $\text{Ø}120$ cm дужине 16,55 m [2].

За прелаз са моста на труп пута, предвиђене су прелазне плоче, дужине 5,70 m и дебљине 20 cm, ослоњене на ослоначку греду преко кратког елемента. Испред и иза моста, а због разлике крутости насипа пута и објекта, предвиђена је израда шљунчаног клина између крилних зидова [3]. Да би се елиминисала појава хидростатичког притиска на зид и крила крајњих стубова, те да би се спречило продирање површинске воде у насип иза крајњих стубова, испред слоја шљунчаног клина, дуж читаве висине зида крајњих стубова, уграђује се филтерски слој од чистог крупнозрног шљунка дебљине 1 m умотаног у геотекстил [4]. Испод шљунчаног клина и филтерског слоја уз наглавну греду се уграђује шљунчано-пешчано цементна стабилизација у паду од 10%. Начин и дубина фундирања су одређени на бази геолошко-геотехничких испитивања и података срачунатих и интерпретираних у геолошко-геотехничком елаборату.

3. АНАЛИЗА ОПТЕРЕЂЕЊА

Стално оптерећење чине сопствена тежина конструкције (распонска конструкција, лежишна греда, ивични венац, пешачке стазе крила, попречни носачи, наглавна греда, шипови, прелазне плоче, стубови) и тежина неносећих елемената. Сопствена тежина носећих елемената се прорачунава софтвером, а тежина неносећих елемената на основу запреминске тежине и запремине елемента, дефинисане према EN 1991-1-1:2012 [5].

Оптерећење снегом се рачуна према европском стандарду EN 1991-1-3:2017 [6]. Како на предметној

површини не може истовремено бити већа количина снега и возила, а саобраћајно оптерећење је својим износом веће и од најнеповољнијег снега, оптерећење снегом не узима се у обзир при прорачуну у комбинацији са саобраћајним оптерећењем.

Оптерећење ветром је срачунато према европском стандарду EN 1991-1-4:2012 [7], као дејство ветра на распонску конструкцију са саобраћајним оптерећењем, срачунато као површинско оптерећење и затим је конвертовано и задато као линијско оптерећење.

Оптерећење температуром је срачунато према европском стандарду EN 1991-1-5:2012 [8], на основу улазних параметара: надморске висине, најниже и највише температуре ваздуха.

Корисно оптерећење је срачунато према европском стандарду EN 1991-2:2012 [9], за возила модела LM1 са комбинацијом концентрисаних сила и одговарајућим пратећим расподељеним оптерећењем. Оптерећују се две рачунске траке, са силама 300 kN, 200 kN, и силама од 100 kN у свим комбинацијама возила. Софтвер аутоматски узима у обзир најнеповољнији положај возила, с обзиром на положај у попречном смислу који задаје корисник.

Скупљање бетона обухвата дуготрајне процесе скупљања и течења бетона те релаксацију каблова за преднапрезање. Коефицијент течења $\phi(t, t_0)$ и релативне деформације скупљања ϵ_{cs} су узети у складу са прописима за релативну влажност околине 70%.

Сеизмичко оптерећење се прорачунава помоћу софтвера, Tower 8.0 [10], који нуди опцију сеизмичког прорачуна према еврокод стандарду EN 1998-2:2012 [11] мултимодалном спектралном анализом и EN 1998-5:2012 [12] уз обухватање геотехничких аспеката прорачуна.

Оптерећење од насипа се израчунава према еврокод стандарду EN 1997-1:2017 [13], уз прорачун динамичког коефицијента притиска земље према Мононобе-Окабе методи.

3.1. Преднапрезање

Детаљније ће бити приказано накнадно претходно напрезања из разлога што је за мост који је предмет самог рада усвојен систем од накнадно преднапрегнутих носача [14]. Усвајају се ужади система SPB SUPER класе B, затезне чврстоће 1860 N/mm^2 , састављена од 7 глатких жица, номиналног пречника 15,2. Монтажни носачи распона 20,0 m утежу се са по укупно 3 кабла $7\text{Ø}15,2$, док код распона 25,0 m, први кабел је $12\text{Ø}15,2$, а други и трећи $7\text{Ø}15,2$.

Највећа сила затезања у ужету током преднапрезања (пре тренутних губитака и пре губитака који зависе од времена):

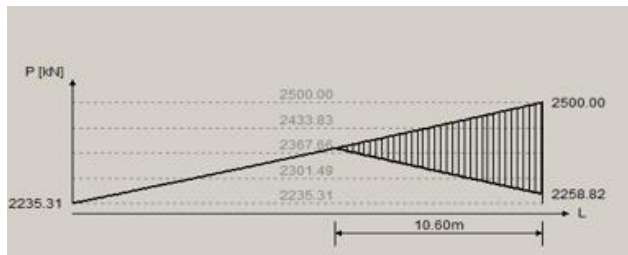
$$P_{\max} = A_p \times \min(k_1 \times f_{pk}; k_2 \times f_{p0,1k}) = \min(209; 211) = 209 \text{ kN} \quad (1)$$

Почетна сила преднапрезања у ужету (после тренутних губитака, а пре губитака који зависе од времена):

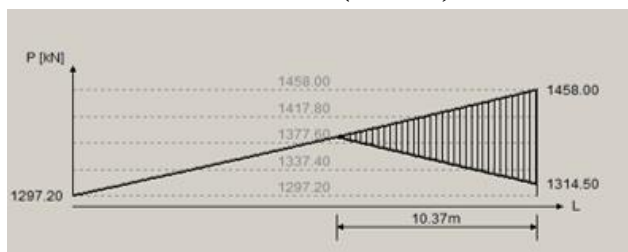
$$P_{to} \leq A_p \times \min(k7 \times f_{pk}; k8 \times f_{p0,1k}) = \min(195; 199) = 195 \text{ kN} \quad (2)$$

Ограничење напона у ужету за карактеристичну комбинацију оптерећења (гранична стања употребљивости) са силом преднапрезања након свих губитака:

$$k5 \times f_{pk} = 0,75 \times f_{yk} = 1252 \text{ kN} \quad (3)$$



Слика 2. Губици силе преднапрезања услед трења и заклињавања (12Ø15,2)



Слика 3. Губици силе преднапрезања услед трења и заклињавања (7Ø15,2)

3.2. Статички прорачун

Формиран је прорачунски модел у програму Tower 8.0, који се састоји из два прорачунска модела. Модел 1 садржи леву половину моста са распонском конструкцијом ослоњеном на тачкасте ослонце, односно горњи и доњи строј се засебно моделирају. Модел 2 садржи леву половину моста са распонском конструкцијом и средњим стубовима (али не и крајњим стубовима). Модел 3 садржи и крајње стубове, односно лежишну греду са шиповима.

Монтажни мост се моделира са линијским елементима, при чему се свака греда моделира засебно. Коловозна плоча се моделира помоћу површинских елемената, али јој се дефинише својство ортотропије (носивост само у попречном правцу). На тај начин се реално моделира прерасподела између гредних носача.

Рачунски модели у Tower-у служе првенствено за одређивање статичких утицаја за релевантне случајеве оптерећења (и комбинације), за контролу одређивања почетних губитака силе претходног напрезања, као и за приближну контролу добијених резултата контроле напона.

3.3. Контрола напона

Ограничење напона притиска у бетону прорачунава се према EN 1992-2:2014 [15] за мостове (гранично стање употребљивости). Ограничење напона притиска у бетону ради избегавања појаве подужних прслина и микропрслина износи:

$$- \text{ за карактеристичну (ретку) комбинацију оптерећења } 0,6 \times f_{ck,cyl} = 0,60 \times 35 \text{ MPa} = 21 \text{ MPa} \quad (4)$$

- при наношењу силе преднапрезања (пре губитака), према SRPS EN 1992-1-1/NA:2015 [16]

$$0,7 \times (0,75 \times f_{c,cyl}) = 0,70 \times (0,75 \times 35) = 17,15 \text{ MPa} \quad (5)$$

Ограничење напона затезања у бетону према EN 1992-1-1:2015 [17]

$$- \text{ за карактеристичну (ретку) комбинацију оптерећења } f_{ctm} = 3,21 \text{ MPa} \quad (6)$$

Извршене су све прорачунске контроле и на основу њих се може закључити да су сви напони у оквиру допуштених.

3.4. Динамички прорачун

Након спроведене модалне анализе конструкције, приступа се дефинисању параметара за прорачун сеизмичких сила. Програмски пакет нуди опцију сеизмичког прорачуна према европским стандардима, преко мултимодалне спектралне анализе која се сврстава у групу линеарно-еластичних анализа.

Табела 1. Периоди осциловања конструкције

No	T [s]	f [Hz]
1	1,2240	0,8170
2	0,8367	1,1951
3	0,6973	1,4340
4	0,2339	4,2757
5	0,2249	4,4459
6	0,2212	4,5212
7	0,2057	4,8814
8	0,1808	5,5318
9	0,1730	5,7790
10	0,1723	5,8038

Сеизмичка анализа се спроводи у складу са правилима и препорукама европског стандарда EN 1998-2:2012 који садржи правила, сеизмичка дејства и правила за мостове.

Табела 2. Параметри за сеизмички прорачун

Категорија тла:	C
Категорија значаја:	III ($\gamma=1,2$)
Однос a_g/g :	0,15
Фактор понашања:	1,00
Коефицијент пригушења:	0,05
S:	1,05
Tb:	0,07
Tc:	0,70
Td:	3,00

3.5. Димензионисање

Елементи моста се димензионишу према граничном стању носивости, према максималним статичким утицајима добијеним прорачуном у софтверском пакету коришћеном за овај пројекат. Према европским стандардима, носивост на савијање и

смицање се израчунавају према стандарду EN 1992-1-1:2015.

3.6. Контрола деформација

Извршена је контрола угиба. Проверено је колико је одизање монтажног носача након утезања ужади, очвршћавања бетона носача, обављеног скупљања и течења бетона и релаксације челика, додатног оптерећења и температурне разлике.

Укупан коначан угиб за распон $L = 25,0$ m износи 58,4 mm.

Укупан коначан угиб за распон $L = 20,0$ m износи 31,80 mm.

3.7. Лежишта и дилатационе спојнице

На основу реакција добијених из комбинација за гранична стања носивости и употребљивости, усвајају се лежишта са одговарајућим крутостима из каталога Fixed Elastofip.

ознака	code	ЕФ125-13
номинална вертикална сила	F_{vd}	1250 kN
максимална хоризонтална сила	F_{vy}	130 kN
деформација којој одговара макс. хор. сили		40,4 mm
хоризонтална крутост	K_0	3,22 kN/mm
вертикална крутост	K_v	993 kN/mm

Слика 4. Усвојени тип лежишта

4. ЗАКЉУЧАК

На основу претходно изложених прорачуна на примеру моста, може се закључити да су резултати прорачуна напона, добијени у оквиру табеларног прорачуна, у оквиру допуштених. Прорачун напона добијен софтверским пакетом Sofistik [18], који је такође приказан у оквиру мастер рада, не разликује се значајно од напона добијених табеларним прорачуном (напони су проверени за поједине случајеве оптерећења). Мала разлика у напонима може да буде последица реологије која обухвата дуготрајне процесе скупљања и течења, с обзиром да софтверски пакет Sofistik даје прецизније резултате.

Усвојени број ужади за претходно напрезање на стази у наведеном примеру је задовољавајући, те није потребно вршити корекцију.

Одизање монтажног носача након утезања ужати, очвршћавања бетона носача, обављеног скупљања и течења бетона и релаксације челика, додатног оптерећења и температуре је испод границе, чиме је и контрола угиба задовољена.

Максимално померање на месту усвојених лежишта из каталога је $h_r = 32,08$ mm, што је у оквиру допуштених.

5. ЛИТЕРАТУРА

- [1] Мишко Гуњача: Приручник за пројектовање, надзор код изградње и изградњу мостовских конструкција према SRPS EN 1990-1998/NA, 2021.
- [2] SRPS EN 1993-5:2012 Еврокод 3 - Пројектовање челичних конструкција – Део 5: Шипови.
- [3] Андрија Рашета, Игор Џолев: Пројектовање бетонских и металних мостова, део Бетонски мостови, Радни материјал за предавања и вежбе, 2022.

- [4] O Brooker, P A Jackson, S W Salim: Concise Eurocode 2 for bridges, 2009.
- [5] SRPS EN 1991-1-1:2012 Еврокод 1 - Дејства на конструкције - Део 1-1: Општа дејства - Запреминске тежине, сопствена тежина, корисна оптерећења за зграде.
- [6] SRPS EN 1991-1-3:2017 Еврокод 1 - Дејства на конструкције - Део 1-3: Општа дејства - Оптерећења снегом.
- [7] SRPS EN 1991-1-4:2012 Еврокод 1 - Дејства на конструкције - Део 1-4: Општа дејства - Дејства ветра.
- [8] SRPS EN 1991-1-5:2012 Еврокод 1 - Дејства на конструкције - Део 1-5: Општа дејства - Топлотна дејства.
- [9] SRPS EN 1991-2:2012 Еврокод 1 - Дејства на конструкције - Део 2: Саобраћајно оптерећење на мостовима.
- [10] Tower 8.0, 2022, <https://www.radimpex.rs/sr/tower>
- [11] SRPS EN 1998-2:2012 - Еврокод 8 - Пројектовање сеизмички отпорних конструкција - Део 2: Мостови.
- [12] SRPS EN 1998-5:2012 - Еврокод 8 - Пројектовање сеизмички отпорних конструкција – Део 5: Темели, потпорне конструкције и геотехнички аспекти.
- [13] SRPS EN 1997-1:2017 - Еврокод 7 - Геотехничко пројектовање - Део 1: Општа правила.
- [14] Вања Алендер: Претходно-напрегнути бетон, 2003.
- [15] SRPS EN 1992-2:2014 Еврокод 2 –Пројектовање бетонских конструкција - Бетонски мостови - Правила пројектовања и конструкисања.
- [16] SRPS EN 1992-1-1/NA:2015 Еврокод 2 – Пројектовање бетонских конструкција - Део 1-1: Општа правила и правила за зграде - Национални прилог.
- [17] SRPS EN 1992-1-1:2015 Еврокод 2 –Пројектовање бетонских конструкција - Део 1-1: Општа правила и правила за зграде.
- [18] Sofistik, 2022, <https://www.sofistik.de/>

Кратка биографија:



Маја Драгичевић рођена је у Власеници 1997. год. Октобра 2016. год. уписује студијски програм Грађевинарство на Факултету техничких наука у Новом Саду. Октобра 2021. год. стиче звање дипломираног инжењера грађевинарства. Мастер рад на Факултету техничких наука из области Пројектовања бетонских мостова одбранила је 2022. године.

контакт:
maja.dragicevic015@gmail.com

**IDEJNO REŠENJE VODOSNABDEVANJA I ODVOĐENJA ATMOSFERSKIH VODA U
NASELJU SEFKERIN****CONCEPTUAL DESIGN OF WATER SUPPLY AND ATMOSPHERIC WATER
SEWERAGE IN SEFKERIN SETTLEMENT**

Boris Zastranović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – GRAĐEVINARSTVO - HIDROTEHNIKA**2. PROJEKTOVANJE VODOVODA**

Kratak sadržaj – U radu su izrađeni modeli vodovodne i atmosferske kanalizacione mreže u naselju Sefkerin. Na osnovu potreba korisnika, hidrauličke analize formirana je vodovodna mreža. Proračun i hidraulička analiza su sprovedeni u okviru programskog paketa EPANET. Kanalizaciona mreža je trasirana u zavisnosti od topografskih uslova, dok je dimenzionisanje izvršeno na osnovu padavina dvogodišnjeg povratnog perioda. Hidraulički proračun je proveren pomoću programskog paketa EPA SWMM. Na kraju je sistem proveren za povratni period dvadesetogodišnje padavina pomoću koga su dobijene nepovoljne tačke sistema, odnosno mogućnost pojave izlivanja.

Ključne reči: Vodovodna mreža, kanalizaciona mreža, hidraulički proračun, idejno rešenje

Abstract – In the paper, models of the water supply and atmospheric sewage network in the Sefkerin settlement were created. Based on the needs of users, hydraulic analysis, a water supply network was formed. Calculation and hydraulic analysis were carried out within the EPANET software package. The sewerage network was laid out depending on the topographical conditions, while the dimensioning was done based on the rainfall of the two-year return period. The hydraulic calculation was verified using the EPA SWMM software package. At the end, the system was checked for the return period of twenty years of precipitation, which was used to obtain the unfavorable points of the system, i.e. the possibility of spills.

Keywords: water supply, atmospheric sewerage, hydraulic calculation, conceptual design

1. UVOD

U Južnobatanskom okrugu, na levoj obali reke Tamiš nalazi se naselje Sefkerin. Naselje je ravničarskog tipa, sa individualnim domaćinstvima seoskog tipa relativno gusto naseljeno na preko 3500 ha površine. Stanovništvo se pretežno bavi poljoprivredom i na severu naselja je predviđena industrija tercijalne delatnosti na 50 ha. Nadmorska visina je u malom opsegu od 70 do 76 m.n.m. Klima je umereno-kontinentalna sa prosečnom količinom godišnjih padavina oko 600mm.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Matija Stipić.

Za potrebe snabdevanja naselja Sefkerin primenjena je prstenasta mreža. Njena prednost u odnosu na granatu mržu je mogućnost napajnja sa dve ili više strana. Proračun se vrši postepenim približavanjem i rešavanjem sistema jednačine kontinuiteta i Bernulijeve jednačine za prstenove. Mreža je podeljena na više grana, deonice koje su podeljene čvorovima. Moraju biti zadovoljena dva zakona. algebarski zbir protoka u jednom čvoru mora biti jednak nuli, odnosno protok koji ulazi treba biti jednak protoku koji izlazi iz čvora. Drugi je da se pri konstantnom strujanju, razlika pritisaka između bilo koja dva preseka u mreži troši se za savladavanje otpora i visinske razlike između ta dva preseka.

U proračunu se takođe obrađuju gubici na mreži, spoljni i unutrašnji. Bitan faktor prilikom projektovanja vodovodne mreže je predviđanje dovoljne količine vode za protivpožarne potrebe. U zavisnosti od broja stanovnika razlikujemo broj istovremenih požara i protok u 2 časa za šta se smatra dovoljno vremena za gašenje požara. Potreban pritisak u mreži mora biti minimalan 2,5 bara i pretpostaviti položaj hidranata na najnepovoljnijim tačkama. Hidrantima se snabdevaju cisterne vatrogasnih vozila i razmak između 2 hidranta ne sme da prelazi 150m.

Na ukupnu potrošnju vode u jednom naselju utiču podaci o svim korisnicima, odnosno potrebama stanovništva, domaćih životinja, javnih ustanova, industrije, sportskih centara, kao i snabdevanje dovoljne količine vode za hidrante, odnosno gašenje požara. Potreban kapacitet određen je maksimalnom očekivanom potrošnjom konzumnog područja za koje se sistem vodosnabdevanja projektuje. Normativne potrošnje se definišu u skladu s preporukama tehničkih normi DVGW W 410 [1].

Svaka grupa ovih korisnika ima svoju specifičnu potrošnju koji zavisi od određenog područja, navika, klime i predstavlja prosečnu dnevnu potrošnju vode određenog područja svedenu na jednog stanovnika.

Vodovodna mreža je projektovana kao gravitaciona. Voda se od izvorišta, pumpama dovodi do rezervoara za izravnanje dnevne potrošnje, smeštenog na vodotoranju odakle se gravitaciono prstenastom mrežom distribuira do korisnika. Ukupna dužina je 17.235 m i usvojen je minimalni prečnik DN125mm. Na deonicama neposredno uz vodotoranj predviđeni su prečnici DN200mm, odnosno 2 deonice od DN160mm. Dubina ukopavanja je 1.2m.

Izvorište vode je podzemna vode u zoni reke Tamiš gde su planirani bunari kapaciteta 10 L/s. Predviđaju se 6

vertikalnih cevastih bunara prečnika 1m. Izrađuju se sa zaštitnim rešetkastim omotačem u zrnasto poroznoj sredini gde treba da obezbede kapacitet od 55 L/s. Brzina u cevi je manja od 1.5 m/s.

Sanitarna zaštita izvorišta se definiše kroz 3 zone zaštite na prostoru oko vodozahvatnog objekta na kom se prati izgradnja i delatnost izgrađenih objekata i vršnj drugih aktivnosti koje mogu da izazovu promenu prirodnog sastava vode unošenjem patogenih mikroorganizama ili promenu fizičko-hemijskih svojstava vode.

Za održavanje potrebnog pritiska, rezervu protivpožarne potrošnje i za potrebe poravnanja dnevne potrošnje predviđa se rezervoar u formi vodotornja. Primarna funkcija vodotornja je izravnanje potrošnje i zbog dobijanja dovoljnog hidrostatskog pritiska podizanjem II kote na određenu visinu i zavisi od rada pumpne stanice. Zapremin za izravnanje ΔV određuje se iz sumarnih krivih linija doticanja i potrošnje.

Predviđa se rezervoar sfernog oblika izgrađen od čelika, zaštićen od korozije i postavljenom toplotnom izolacijom. Usvojen je prečnik 11m, sa debljinom zidova od 50cm. Treba osigurati brzinu dovoda veću od 0.1 m/s kao preduslov za optimalni kvalitet vode i kako bi osiguralo potrebno mešanje.

Sadrži zatvaračnicu u kome je smštena oprema kojom se kontroliše dovod, odvod i sadrži neophodnu armaturu i fazonske komade (regulacioni i nepovratni ventil). Potrebno je pre puštanja u rad ispitati punjenje rezervoara i voditi zapisnik.

Cevi vodovodne mreže su predviđene od polietilna visoke gustine HPDE-100 prečnika 125-200 mm. Ovaj materijal je apsolutno netoksičan i potpuno inertan u kontaktu sa vodom. Cevi su lake za transport i rukovanje i pogodne su za nastavljajanje i spojnice. Dugog su životnog veka i nemaju uticaj na miris i ukus vode. Ne hvata se kamenac i shodno tome ne dolazi do smanjivanja protoka tokom dugoročne upotrebe. Fleksibilne i izuzetno otporne na vibracije. Pogodne su za praćenje konfiguracije terena i nema potrebe mnogo za fazonskim komadima. Radijus savijanja je 20d [2].

Preporučuje se elektrofuziono zavarivanje i spajanje sa tuljcima i prirubnicama sa drugim tipovima materijala.

3. REZULTATI SIMULACIJE VODOVODNE MREŽE

Na osnovu simulacije urađene u programskom paketu EPANET, dobijeni rezultati su u skladu s preporukama projektovanja vodovodne mreže. Karakteristična stanja mreže prikazani su kroz časove minimalne, maksimalne potrošnje i potrošnje u času sa koeficijentom 1.3 časovne neravnomernosti zajedno sa protivpožarnom potrošnjom. Dodatni protok od 15 L/s merodovan za naselje od 5000 do 10000 stanovnika za požar uvršćen je u najnepovoljnijim tačkama mreže (čvorovi 8, 18 i 19) po 5 L/s.

Pritisak u mreži, u najnepovoljnijoj tački sistema (čvor 19), ide od minimalnog 2,65 bara do 4 bara (slika 1), dok je II kota u rasponu od 110,48 do 113,94m. Brzine u cevi koja vodi od rezervoara do prvog narednog čvora (čvor 41) je u rasponu od minimalne 0,51 m/s do maksimalne 2,13 m/s.



Slika 1. Promena pritiska u čvoru 19 za 24 sata

Protok u sistemu prikazan je dijagramom kroz cev koja spaja rezervoar (vodotoranj) sa tačkom 41. Kreće se u rasponu od 10,17 L/s do 42,15 L/s u času maksimalne potrošnje 1.3 $Q_{\max,h}$ i 15 L/s za požar, dato na slici 2.



Slika 2. Promena protoka u deonici od rezervoara do susednog čvora

4. PROJEKTOVANJE ATMOSFERSKE KANALIZACIJE

4.1. Padavine

Za potrebe projektovanja atmosferske kanalizacije potrebno je obraditi prvi ulazni podatak, a to su padavine. Pod padavinama se podrazumeva taloženje vode iz atmosfere na površinu zemlje. One obuhvataju kišu, sneg i druge oblike u kojima voda dospeva do površine zemlje, kao što su grad ili ledena kiša. Pojave kao što su rosa, magla ili inje takođe predstavljaju padavine, ali do njih dolazi kondenzacijom zasićenog vazduha u dodiru sa hladnijim površinama na zemlji [3].

Proticaji koji se stvaraju od padavina u kanalima su vrlo promenljivi. Kada nema oni su jednaki nula, dok za vreme kiša količina vode se povećava naglo u vrlo kratkom periodu.

Količina godišnje kišnice predstavljaju 20 do 40 % ukupne količine upotrebljene vode u jednom naselju. Kiše se karakterišu trajanje, intezitetom, učestalošću, a pored toga još i rasporedom inteziteta u toku trajanja i po površini koju zahvataju, pravcem kretanja i dobom godine u kom se javljaju. Budući da gotovo da ne postoji dve identične, potrebno je pratiti intezitet i trajanje svake kiše jer ne postoji direktna funkcionalna zavisnost između njih. Dve kiše istog trajanja nemaju isti intezitet. Međutim za određena područja moguće statistički doći do povezanosti, za kiše manjeg inteziteta.

Za vreme padavina formiraju se atmosferske vode. Stepem zagađenja ovih otpadnih voda je veći u urbanim sredinama spiranjem nečistoća sa nepropusnih površina. U naseljenim mestima mogu da predstavljaju velike probleme izazivanjem poplava na saobraćajnicama i eventualnim ulaskom u podrum. Čistija je od voda iz domaćinstva, ali u kontaktu s površinom, odnosno, s suspendovanim i organskim materijama, pa je potrebno na nekim mestima je prečistiti pre uvođenja u kanalizaciju (industrija, parkinzi...).

4.2. Kanalizacioni sistem

Otpadne vode se mogu odvoditi mrežom kanala po opštem, separacionom ili poluseparacionom sistemu. Opšti sistem podrazumeva odvođenje svih voda jednom jedinstvenom kanalskom mrežom iz svih delova naselja. Ako se svaka vrsta otpadne vode odvoji posebno onda je to separacioni sistem.

Poluseparacioni sistemi se predviđaju kada je potrebno izbegavanje izlivanja vode od pranja ulica i od slabih kiša, kao i vode od jakih kiša. Tada se predviđaju prelivne građevine kojima se voda iz kišne kanalizacije pri malim proticajima izliva u mržu za upotrebljenu vodu [2].

Delovi jednog sistema su objekti za sakupljanje otpadnih voda, za prečišćavanje, za ispuštanje vode u recipijente.

Kanalizaciona mreža se uglavnom projektuje kao gravitaciona, što podrazumeva da voda teče sa slobodnim vodenim ogledalom. U posebnim topografskim uslovima može biti pod pritiskom, odnosno vakuumom.

Odvođenje atmosferske vode se ne mora odvijati u svim delovima naselja putem kanalske mreže. Ako se kišnica ne zadržava tako da pravi štetu ili ako teren ima dovoljno veliki pad ka nekom prirodnom recipijentu može se odvoditi ulicama. U ravničarskim naseljima moguće je vodu skupljati u neobloženim kanalima i rigolama iz kojih će isparavati i ponirati [1].

4.3. Kanalizaciona mreža - Sefkerin

Za projektovanje atmosferske kanalizacije urađen je proračun za povratni period padavina 1 u 2 godine trajanja kiše 20 minuta, kao primena standarda EN 752-2 za proveru učestalosti preopterećenja kod planiranja novog sistema, kao i za povratni period 1 u 20 godina trajanja kiše 20 minuta za zaštitu od plavljenja, odnosno do izlivanja sme doći kada maksimalna visina vode pređe određeni referentni nivo.

Kao prvi ulazni podatak uvodi se ITP kriva za posmatrano područje koja je data u tabeli 1.

Tabela 1. Računska kiša za povratni period 1 u 20 godina trajanja 20 minuta

Vreme (h:min)	Visina padavina (mm)
15:00	0
15:05	7
15:10	10
15:15	14
15:20	14.2

projektovana kanalizaciona mreža je projektovana kao gravitaciona, separaciona, kombinacijom otvorenih kanala i zatvorenih kolektora. Sva voda se odvodi ka jugu naselja, čvoru „izliv“ odnosno recipijent - reka Tamiš. Podeljena je na podslivove ukupne površine 3339 ha, 75 čvorova i 76 deonica ukupne dužine 15995m sa prosečnim nagibom terena od 0,1%.

Slivne površine su formirane na bazi Tisenovih poligona prema liniji razdvajanja površinskog toka. Procenat nepropusnih površina varira od 5% po obodu naselja do 8% u centralnim delovima naselja, gde je gustina naseljenosti gušća.

Kako je već pomenuto, u severnom delu naselja je predviđena industrija za koju je potrebno izgraditi separatore ulja kako se atmosferska voda ne bi dodatno zagadila.

Zbog topografije terena, tipičnog ravničarskog naselja, uglavnom se poseže za otvorenim kanalima dok god je to moguće. Oni se postavljaju dok dubina ukopavanja ne prelazi 1.3 m, duž puta i/ili unutar razdelnog pojasa kod puteva i ulica. Karakterističan poprečni presek kanala koji se projektuje je trapznog oblika, čija je geometrija definisana širinom u dnu B i nagibom kosina. Širina kanala se kreće u rasponu od 0,5 do 0,7m, dok se za nagib svuda usvaja 1:1:5. Dubina kanala je uslovljena tako da je moguće ulivanje vode slobodno koja se prikuplja drenažnim sistemom kolovozne konstrukcije. Propusna sposobnost kanala se računa primenom Chezy-Manning-ove jednačine za uniformno tečenje (1):

$$Q = \frac{1}{n} h^{\frac{8}{3}} F(h) \sqrt{S} \quad (1)$$

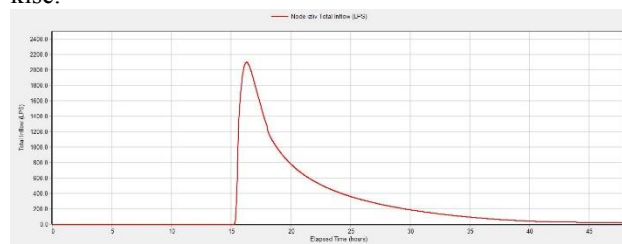
Ovakav vid odvođenja atmosferskih voda je pogodan u izgradnji zbog malih troškova, ali je dosta nepovoljan u eksploataciji. Takođe, na mestima ukrštanja sa prilaznim putevima i stazama i ulicama potrebno je izgraditi propuste.

Na mestima gde dubina ukopavanja prelazi 1,3m, kao i gde nije moguće graditi otvorene kanale, planiraju se zatvoreni kanali. Dimenzionišu se na osnovu merodavnih protoka deonica, tako da voda teče sa slobodnim vodenim ogledalom. Najveća dubina ukopavanja ne bi trebalo da prelazi 4m, izuzetno do 5m. Usvojeni minimalni prečnik mreže je 400mm, a najveći 1200mm (glavni kolektor koji vodi ka recipijentu).

Predviđene su PVC kanalizacione cevi kao najsavremeniji vid materijala pogodnih za izvođenje kanalizacione mreže. Karakterišu ih lako postavljanje i spajanje, lak prenos i rukovanje. Otporne su na koroziju i dobar su električni izolator. Dugog su veka trajanja, duži od 50 godina.

5. REZULTATI SIMULACIJE KANALIZACIONE MREŽE

Projektovani sistem atmosferske kanalizacije zadovoljava osnovne uslove da nema preopterećenja mreže, odnosno izlivanja, kao i da nema tečenja pod pritiskom. Na izlivu je dobijen hidrogram proticaja koji odgovara gorenavedenim uslovima, gde se maksimalni protok od 2098,16 L/s dešava u 16:17 h, sat i 17 minuta posle početka zadate kiše.

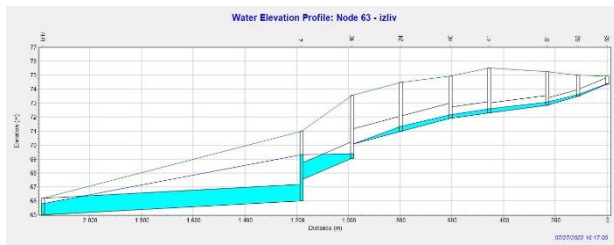


Slika 3. Hijetogram čvora „izliv“ – promena protoka kroz vreme

Najduža deonica od čvora 71 do izliva je prikazana na slici 20. Deonica počinje otvorenim kanalom do čvora 13 odakle se zatvorenim kanalima prečnika 1000 i 1100mm

odvodi ka izlivu. Primećuje se da je poslednja deonica poprilično opterećena gde je maksimalni stepen ispunjenosti 85%. Tečenje je sa slobodnim vodenim ogledalom i nema izlivanja.

Najopterećeniji kolektori u mreži su na deonici od čvora 63 do izliva zbog najvećeg broja uliva. Zbog izbegavanja izlivanja predviđen je kolektor na deonici 10-4 prečnika 1200mm.



Slika 4. Podužni profil sa maksimalnim nivoom vode na deonici od čvora 63 do izliva, 1 u 2 povratni period

5.1. Proračun za projektovani dvadesetogodišnji period

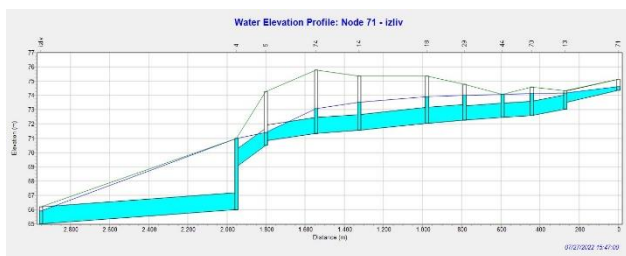
Pri proračunu dvadesetogodišnjeg povratnog perioda dozvoljeno je plavljenje. U obrađenoj simulaciji dolazi se do podatka da se u projektovanoj mreži pojavljuju dva mesta na kojima dolazi do izlivanja. Računska kiša sa ITP krive počinje u 15:00 h i traje 20 minuta i nakon 32 minuta dolazi do ispunjenja u čvoru 4 neposredno pred uliv vode u glavni odvodni kolektor ka izlivu. Trajanje izliva je 2 sata i 5 minuta. Protok kroz izliv je 2592.22 L/s.

Drugo značajno mesto je čvor 44. Trenutak kada počinje plavljenje je 15:47 h i traje 40 minuta. Primećuje se da je sistem preopterećen u južnom i istočnom delu naselja za dvadesetogodišnji povratni period.

olektori D30 (24-10), D31 (10-4) i D41 (61-41) su preopterećeni gde dolazi do potpunog ispunjenja profila što za posledicu ima izlivanje u čvoru 4

Na podužnom profilu od čvora 71 do izliva primećuje se opterećenje na većem broju kolektora, dok se izliv dešava u čvoru 44.

Primećuje se da u deonicama D63 i D64 dolazi do naglih skokova protoka u vremenu. Dolazi do naglog porasta i pada da bi se opet ponovo naglo podigao protok. Do izlivanja dolazi za svega 44 minuta od početka padanja kiše povratnog perioda 20 godina.



Slika 5. Podužni profil sa maksimalnim nivoom vode na deonici od čvora 71 do izliva, 1 u 20 povratni period

6. ZAKLJUČAK

Ovim radom su obuhvaćene dve oblasti komunalne hidrotehnike, vodosnabdevanje i odvođenje atmosferskih voda u naselju Sefkerin. Urađeno je idejno rešenje vodovodne i

kanalizacione mreže sa detaljnom analizom proračuna i opisom objekata koji su deo ovog projekta.

Vodovodna mreža je predviđena sa zadatkom da svi korisnici naselja dobiju čistu vodu i da bude obezbeđen dovoljan pritisak u mreži zajedno sa protivpožarnom zaštitom. Na osnovu topografije terena, podataka o naselju, stanovnicima i ostalim korisnicima posegnuto je za simulacijom u programskom paketu EPANET kojim je izvršen hidraulički proračun i dimenzionisana mreža.

Minimalni prečnik u sistemu je 125mm PEHD cevi koji zadovoljava minimalni pritisak od 2.5 bara. Mreža je koncipirana tako da se od izvora, odakle se voda crpi iz bunara, pumpama voda dovodi do rezervoara za poravnanje dnevne potrošnje.

Sva otpadna voda nastala od padavina se uvodi u novo-projektovanu kanalizacionu mrežu otvorenih i zatvorenih kanala. Na osnovu ravničarskog terena i računske kiše, odnosno dobijenih protoka dimenzionisana je mreža. Hidraulički proračun je obrađen u programu EPA SWMM. Kao recipijent je odabrana reka Tamiš.

Proračun je izvršen na osnovu dvogodišnje kiše za trajanje od 20 minuta gde je pokazano da projektovani sistem zadovoljava uslove ustaljenog tečenja sa slobodnim vodenim ogledalom i da se ni u jednom trenutku za vreme trajanje simulacije ne događa izlivanje.

Na kraju je izvršen proračun za kišu sa povratnim periodom 1 u 20 godina za 20 minuta padavina. U tim uslovima je dozvoljeno izlivanje i ono se javlja na 2 mesta u sistemu. Najveće i najduže izlivanje se dešava u zbirnom čvoru mreže, neposredno u ulasku glavni odvodni kolektor ka izlivu.

7. LITERATURA

- [1] Dr Miloje Milojević, Snabdevanje vodom I kanisanje naselja. Građevinski fakultet, 1976;
- [2] Stipić Matija, Skripta iz predmeta Komunalna hidrotehnika za studente IV godine odseka za hidrotehniku. Fakultet tehničkih nauka u Novom Sadu, Novi sad, 2017;
- [3] Jasna Petrović. Skripta iz predmeta Uvod u hidrologiju - Građevinski fakultet u Beogradu, 2001;
- [4] Hajdin Georgije, Mehanika fluida, knjiga druga, uvođenje u hidrauliku. Građevinski fakultet u Beogradu, 2001;
- [5] Hajdin Georgije, Mehanika fluida, knjiga treća, uvođenje u hidrauliku. Građevinski fakultet u Beogradu, 2001;
- [6] Marko V. Ivetić, Računska hidraulika, tečenje u cevima Beograd, 1996.

Kratka biografija:



Boris Zastranović rođen je u Zaječaru 1994. godine. Diplomirao je na Fakultetu za graditeljski menadžment gde je odbranio diplomski rad sa temom „Projektovanje AB konstrukcije objekta Po+Pr+3+Pot”. Master studije završava na Fakultetu tehničkih nauka gde izrađuje Master rad iz oblasti Komunalna hidrotehnika.

**АНАЛИТИЧКА МЕТОДА ЗА КОНТРОЛУ ПРОХОДНОСТИ И ПРОРАЧУН
ЕЛЕМЕНАТА СИТУАЦИОНОГ ПЛАНА ПОВРШИНСКИХ РАСКРСНИЦА****ANALYTICAL METHOD FOR VEHICLE SWEEP PATH TESTING AND
CALCULATION OF AT GRADE INTERSECTIONS LAYOUT ELEMENTS**

Марија Долић, Факултет техничких наука, Нови Сад

Област – ГРАЂЕВИНАРСТВО

Кратак садржај – У оквиру рада приказана је анализа метода за прорачун проходности и прорачун проходности елемената ситуационог плана површинских раскрсница као и експериментална метода одређивања граничних трајекторија.

Кључне речи: Пројектовање путева и раскрсница, анализа проходности возила у кружним кривинама, offtracking

Abstract – The paper presents the analysis of methods for the calculation of the passability and the calculation of the passability elements of the horizontal alignment of at grade intersections, as well as the experimental method of determining the boundary trajectories.

Keywords: Design of roads and intersections, analysis of vehicle passability in roundabouts, offtracking

1. УВОД

При скретању возила малим брзинама задњи точкови возила не прате трајекторије кретања предњих, него се све више заносе ка унутрашњости кривине како возило напредује према крају кривине. Ово постепено заносење задњих точкова ка унутрашњости кривине начешће се дефинише у литератури као “Offtracking” феномен. Највећа “слабост” постојећих математичких модела за прорачун offtracking-а јесте непрецизност при одређивању тачног положаја возила у којем се достиже највеће растојање између граничних трајекторија возила, или на енглеском “Maximum offtracking problem”.

У најширем случају силе отпора које дејствују на возило у кретању могу се поделити на унутрашње и спољашње силе отпора. Под унутрашњим силама отпора подразумевају се све силе које дејствују при преносу снаге од мотора до точка, како инерционе тако и силе трења елемената трансмисије. Стога се ове силе отпора и зову унутрашњим силама. Њихово дејство се може са довољном тачношћу апроксимирати степеном корисности трансмисије, тако да ће се у даљем разматрању узимати као ефективна сила вуче, она која се добија погонским точковима возила.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био доц. др Милош Шешлија.

Уколико се као општи случај претпостави кретање возила устаљеном брзином, на успону уколико то возило вуче приколицу, онда ће се његовом кретању супростављати следеће силе отпора:

- Силе отпора при котрљању R_f
- Силе отпора ваздуха R_v
- Силе отпора при успону R_α ,
- Отпор инерцијалних сила R_i ,
- Сила отпора вуче приколице R_p .

2. СТАБИЛНОСТ ВОЗИЛА

Под стабилношћу возила, у суштини се подразумева способност да се креће задржавајући свој смер кретања без обзира на дејство спољних сила. У том смислу може да се говори о стабилности са аспекта:

- превртање
- проклизавања (попречна)
- дејства центрифугалне силе при вожњи у кривини
- утицаја силе бочног ветра.

**2.2. Превртање возила на равном хоризонталном
оуту у кривини**

Величина центрифугалне силе сразмерна је маси возила и квадрату брзине, а обрнуто пропорционална полупречнику кривине (R), дакле:

$$F_c = \frac{G \cdot v^2}{g \cdot R} \quad (1)$$

а повећањем угла нагиба коловоза, повећава се и брзина кретања возила.

Да ни при којој брзини кретања не би дошло до превртања, потребно је да израз буде бесконачан, односно да је $h_T - s \cdot \tan \beta = 0$, то јест да угао бочног нагиба коловоза буде:

$$\tan \beta \geq \frac{h_T}{s} \quad (2)$$

У случају коловоза без нагиба ($\beta = 0$), да не би дошло до превртања, потребно је да брзина буде мања од:

$$\frac{v^2}{R} \leq \frac{s \cdot g}{h_T} \rightarrow v_p \leq \sqrt{\frac{g \cdot s \cdot R}{h_T}} \quad (3)$$

2.3. Проклизавање возила на равном хоризонталном путу у кривини

Из горње анализе јасно произилази и закључак да је проклизавање на равном хоризонталном коловозу критичнији случај од случаја када је коловоз са нагибом. Да би дошло до проклизавања точкова, потребно је да збир хоризонталних реакција тла:

$$Y' + Y'' = \frac{G \cdot v^2}{g \cdot R} \cdot \cos \beta - G \cdot \sin \beta \quad (4)$$

Бочно проклизавање ће наступити када је:

$$\frac{v^2}{R} \geq \frac{g \cdot (\varphi + \tan \beta)}{1 - \varphi \cdot \tan \beta} \rightarrow v_k \geq \sqrt{\frac{g \cdot R \cdot (\varphi + \tan \beta)}{1 - \varphi \cdot \tan \beta}} \quad (5)$$

$$t = \tan\left(\frac{\beta}{2}\right) = \frac{1 - e^{K\theta_s}}{X - K - X - X \cdot e^{K\theta_s} - K \cdot e^{K\theta_s}} \quad (6)$$

Једначина (6) је најважнија за први сегмент ИСАО-овог математичког модела, јер се из ње рачуна тренутни скретни угао β између подужне осе авиона и тангенте на лук у зависности од угловне координате положаја кабине авиона θ_s над кружном путањом.

$$\ln\left(\tan\left(\frac{\beta}{2}\right)\right) = \ln\left(\tan\left(\frac{\beta_{max}}{2}\right)\right) - \frac{F}{d} \quad (7)$$

Једначина (7) најважнија је за други сегмент ИСАО-овог математичког модела. У њој је дефинисано како се, у зависности од тренутног скретног угла β_{max} на узлазној тачки кружног лука, може срачунати тренутни скретни угао β у моменту када се кабина авиона налази на одстојању F од ове излазне тачке.

3. ДЕФИНИЦИЈА ФЕНОМЕНА „OFFTRACKING-A“ ПРИ СКРЕТАЊУ ВОЗИЛА

Током скретања друмских возила при малим брзинама предњи точкви настоје да „вуку“ задње точкове директно према њима, секући на тај начин унутрашњост кривине у односу на предње при маневрима скретања, дефинише се у литератури на енглеском језику као „offtracking“ феномен. Код дугачких возила као што су аутобуси, камиони и друга тешка теретна возила, величине offtracking-а знатно су веће и представљају значајан фактор при пројектовању површинских раскрсница, паркинга, прилазних путева, рампи и других саобраћајних површина.

У зависности од брзине кретања возила могу се јавити два типа offtracking-a:

- Појава offtracking-a при кретању возила малим брзинама ($V = 15.00 \text{ km/h}$) (у литератури се најчешће дефинише као „Low-speed offtracking“);
- Појава offtracking-a при кретању возила већим брзинама ($V = 80.0 \text{ km/h}$) (High-speed offtracking).

3.1. ИСАО математички модел за прорачун offtracking-a

Према начину извођења математичких израза за дефинисање положаја авиона (возила), ИСАО математички модел може се јасно поделити на два одвојена сегмента или суб-модела:

- Први, који се бави прорачуном положаја возила (авиона) када оно, крећући се из почетног положаја у којем је осовина возила идеално поравната са улазном тангентом, базном (почетном) тачком круте базе прати кружни лук.
- Други, када базна тачка круте базе возила (авиона) прати праву линију у виду излазне тангенте на крају просте кружне кривине.

4. АНАЛИЗА НАЈПОПУЛАРНИЈИХ КОМЕРЦИЈАЛНИХ СОФТВЕРСКИХ РЕШЕЊА ЗА СИМУЛАЦИЈУ КРЕТАЊА ВОЗИЛА У САД ОКРУЖЕЊУ

У конкретном примеру тестирана су 4 најзаступљенија комерцијална софтверска решења за симулацију кретања и то: AutoTurn, Vehicle Tracking, Autopath PRO I Gavran Civil Modeller-GCM. Као огледно возило изабран је тегљач са полуприколицом, за меродавна возила. Потребне ширине саобраћајне траке за безбедан маневар возила, односно, растојања између граничних трајекторија мерена су на сваком од 9 попречних пресека постављених управо на исцртане водеће путање. Мерења су обављена редом за све изведене симулације кретања.

Резултати мерења сортирани су по попречним пресецима. Детаљнијом анализом резултата мерења може се видети да се највећа одступања у растојањима између граничних трајекторија јављају на месту попречних пресека 6 и 7 када возило почиње да се исправља после изласка из кружне кривине. Анализом свих мерених вредности, може се запазити да се најужа растојања између граничних трајекторија добијају после симулације кретања шлепера са софтверским пакетом GCM++, а затим, редом, са Vehicle Tracking-ом, AutoTurn-ом и Autopath-ом.

5. ЕКСПЕРИМЕНТАЛНЕ МЕТОДЕ ЗА ОДРЕЂИВАЊЕ ГРАНИЧНИХ

Генерално посматрано, постоје два различита технолошка приступа за прецизно мерење простора обухваћеног граничним трајекторијама возила при извођењу реалних маневра:

- технологије или системи који користе возила опремљена специјалним уређајима за детектовање њиховог положаја (GNSS мерни систем);
- системи који не захтевају никакве унапред инсталиране инструменте у возилу (класичне техиметријске и методе базиране на обради видео и фото снимака).

6. АНАЛИЗА ПРИМЕНЕ КОНЦЕПТА ЕКВИВАЛЕНТНЕ БАЗЕ ЗА ПРОРАЧУН ТРАЈЕКТОРИЈЕ ПОВЛАЧЕЊА КОМБИНОВАНИХ ВОЗИЛА

Тек када су познати положај базне тачке и скретни угао водећег возила, као и одговарајући углови артикулације прикључних возила, могуће је у потпуности дефинисати положај свих слогова комбинованог возила у односу на водећу путању. На основу међусобног поређења трајекторија повлачења од дугачких комбинованих и њихових заменских возила, закључено је следеће:

- Трајекторије повлачења комбинованих и одговарајућих заменских возила у оба случаја не поклапају се идеално;
- Све док се возила крећу дуж кружног одсека водеће путање и почну да се исправљају дуж излазне тангенте, након достизања положаја са максималном вредношћу offtracking-a, трајекторија повлачења заменских возила нису на „страни сигурости“ у односу на трајекторије повлачења комбинованих возила;
- Након што базне тачке заменских возила, мерено од краја кружног лука, пређу растојање приближно једнако дужини тих возила, и даље наставе да прате излазну тангенту све до краја водеће путање, трајекторија повлачења заменских прелаза на „страну сигурности“ у односу на трајекторије повлачења комбинованих возила;
- Код комбинованог возила састављеног од три слога, одступања између трајекторија повлачења комбинованог и заменског возила мања су него између трајекторија повлачења за комбиновано возило са два слога и њему одговарајуће заменско возило;
- Што се комбиновано возило састоји од више слогова приближно једнаке дужине, то ће и одступање између трајекторија повлачења тог комбинованог и њему одговарајућег заменског возила бити мања;
- Комбиновано возило, било са два или три слога, достиже максималну вредност offtracking-a у ситуационо разичитом положају у односу на њему одговарајуће заменско возило.

7. АНАЛИЗА ПОДОБНОСТИ ПРИМЕНЕ ПОСТОЈЕЋИХ ГЕОМЕТРИЈСКИХ ФОРМИ ТРОЦЕНТРИЧНЕ КРИВИНЕ ЗА АПРОКСИМАЦИЈУ ТРАЈЕКТОРИЈА ПОВЛАЧЕЊА И ПРЕДЛОГ НОВЕ

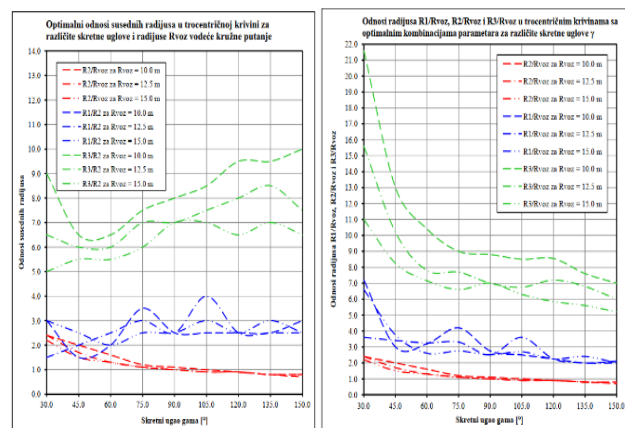
7.1. Геометријске форме троцентричних кривина за апроксимацију трајекторија повлачења соло аутобуса

Прво су одређене комбинације геометријских параметара троцентричних кривина које најмање одступају на „страну сигурности“ од трајекторија повлачења сваког од изабраних возила (соло аутобус,

тегљач са полуприколицом и камион са приколицом) за различите радијусе и скретне углове водећих кружних путања. Потом су срачунате оптималне комбинације геометријских параметара троцентричних кривина које најбоље апроксимирају све трајекторије повлачења појединих типова возила, као и оптимална комбинација параметара „јединствене“ троцентричне кривине за апроксимацију свих трајекторија повлачења од свих возила. Оптималне комбинације геометријских параметара троцентричних кривина срачунате помоћу „3R_OPTIMA“ рутине, које најбоље апроксимирају трајекторије повлачења соло аутобуса за различите радијусе и скретне углове водећих путања, приказане су у табели (1).

Табела 1. Оптималне комбинације геометријских параметара

Rvoz [m]	gama [°]	R1 [m]	R2 [m]	R3 [m]	R1/R2	R2/Rvoz	R3/R2	alpha [°]	BETA [°]	DELTA [°]	R1/Rvoz	R3/Rvoz
10.00	30.00	72.000	24.000	216.000	3.00	2.40	9.00	1.40	4.70	1.40	7.20	21.60
10.00	45.00	10.000	30.000	130.000	1.50	2.90	6.50	1.40	4.70	1.40	3.00	13.00
10.00	60.00	42.000	16.000	164.000	2.00	1.60	6.50	1.00	5.10	1.40	4.20	16.40
10.00	75.00	42.000	12.000	96.000	3.50	1.20	7.50	0.80	5.40	1.30	4.20	9.60
10.00	90.00	27.500	11.000	88.000	2.50	1.10	8.00	1.00	5.40	1.10	2.75	8.80
10.00	105.00	25.000	10.000	85.000	2.50	1.00	8.50	0.80	5.70	1.00	2.50	8.50
10.00	120.00	27.500	9.000	85.500	3.10	0.90	9.50	0.80	5.10	0.90	3.15	8.55
10.00	135.00	29.000	8.000	76.000	3.60	0.80	9.50	1.00	5.60	0.90	2.90	7.60
10.00	150.00	21.000	7.000	70.000	3.00	0.70	10.00	1.10	5.50	0.90	2.10	7.00
12.50	30.00	45.000	30.000	195.000	1.50	2.40	6.50	0.80	5.30	1.40	3.80	15.80
12.50	45.00	43.500	21.250	177.500	2.00	1.70	6.00	0.80	5.30	1.40	3.40	16.20
12.50	60.00	40.625	16.250	97.500	2.50	1.30	6.00	0.80	5.20	1.40	3.25	7.50
12.50	75.00	41.375	13.750	96.375	3.00	1.10	7.00	0.80	5.40	1.30	3.80	7.70
12.50	90.00	31.250	12.500	87.500	2.50	1.00	7.00	1.00	5.40	1.10	2.50	7.00
12.50	105.00	45.000	11.250	84.375	4.00	0.90	7.50	0.80	5.70	1.00	3.60	6.75
12.50	120.00	38.125	11.250	96.000	3.40	0.80	8.00	1.00	5.70	0.80	3.75	7.50
12.50	135.00	25.000	10.000	85.000	2.50	0.80	8.50	1.20	5.50	0.80	2.80	6.80
12.50	150.00	25.000	9.000	75.000	2.50	0.80	7.50	0.90	5.80	0.80	2.80	6.00
15.00	30.00	99.000	33.000	165.000	3.00	2.20	5.00	0.80	5.30	1.40	6.60	11.00
15.00	45.00	36.250	22.500	123.750	2.50	1.30	3.50	0.80	3.30	1.40	3.75	8.25
15.00	60.00	39.000	19.500	167.500	2.00	1.30	5.50	0.80	5.50	1.30	3.90	7.15
15.00	75.00	41.250	16.500	99.000	2.50	1.10	6.00	0.80	5.60	1.10	2.75	6.60
15.00	90.00	37.500	15.000	105.000	2.50	1.00	7.00	1.00	5.60	0.90	3.50	7.00
15.00	105.00	40.500	13.500	94.500	3.00	0.90	7.00	0.80	5.70	0.90	2.70	6.30
15.00	120.00	33.750	13.500	87.750	2.50	0.90	6.50	0.90	5.80	0.80	2.25	5.85
15.00	135.00	36.000	12.000	84.000	3.00	0.80	7.00	1.10	5.60	0.80	2.40	5.60
15.00	150.00	40.000	12.000	76.000	3.30	0.80	6.50	1.00	5.70	0.80	2.90	5.20



Слика 1. Збирни дијаграм са оптималним односима суседних радијуса (а); Дијаграм са односима R1/R2, R2/Rvoz, R3/R2 у троцентричним кривинама за апроксимацију трајекторија повлачења соло аутобуса

У посебним сиво обојеним колонама на крају табеле 1 израчунати су односи и радијуса R1 и R3 из троцентричних кривина са оптималним комбинацијама параметара и радијуса Rvoz водећих кружних путања соло аутобуса. Затим су тако срачунати односи радијуса R1/R2, R2/Rvoz, R3/R2

приказани у виду збирног дијаграма на слици 1 (а). На дијаграму са слике 1 (б) такође се види да се односи оптималних радијуса $R2/R_{voz}$ за разлику од односа $R1/R_{voz}$ и $R3/R_{voz}$ мењају по сличној законитости за све вожене радијусе R_{voz} соло аутобуса, а нарочито за скретне углове γ веће од 75° . Исти поступак је примењен и са тегљач са полуприколицом као и за камион са приколцом.

7.2. Геометријске форме троцентричних кривина за апроксимацију трајекторија повлачења свих изабраних возила

Трајекторије повлачења из .txt фајлова, претходно генерисаних за свако од изабраних возила, груписане су по изведеним маневрима за различите вожене радијусе R_{voz} и скретне углове γ водећих путања.

На крају, све трајекторије повлачења од свих изабраних возила груписане су у један .txt фајл, и за њих је срачуната једноставна оптимална троцентрична кривина. Посматрајући срачунате комбинације долази се до закључка да је јединствена (оптимална) комбинација геометријских параметара троцентричних кривина за апроксимацију свих трајекторија повлачења од свих изабраних возила потпуно је иста као оптимална комбинација параметара у троцентричној кривини за апроксимацију свих трајекторија повлачења тегљача са полуприколицом.

Врло је важно нагласити да су у домаћим смерницама, препоруке за избор централног радијуса $R2$ дата за троцентричне кривине конструисане између тангенти које су паралелно померене тачно за половину ширине возила $b/2$ ка центру кривине у односу на тангенте водеће путање. Стога, у домаћим смерницама није, као у анализи спроведеној у овом мастеру, излазна тангента троцентричне кривине паралелно померена за $0,25m$ ка центру кривине у односу на изворну тангенту.

8. ЗАКЉУЧАК

Применом развијеног софтверског алата, односно „3R_OPTIMA” AutoLISP рутине, за прорачун оптималних комбинација геометријских параметара троцентричних кривина за апроксимацију граничних трајекторија повлачења возила дошло се до следећих закључака:

- На однос суседних радијуса и удела углова и троцентричним кривинама са оптималним комбинацијама параметара много више утичу вредности скретних углова γ од величине вожених радијуса R_{voz} водећих кружних путања.
- Ако се тражи оптимална комбинација параметара у троцентричним кривинама за апроксимацију трајекторија повлачења више различитих типова која се крећу дуж истих водећих путања, резултат прорачуна увек ће бити она комбинација параметара троцентричне кривине која најмање одступа на „страну сигурности“ од трајекторија

повлачења возила са највећим вредностима offtracking-a.

- За сва изабрана (тестна) возила односи радијуса $R2/R_{voz}$ у троцентричним кривинама са оптималним комбинацијама параметара за апроксимацију трајекторија повлачења свих возила, знатно одступају од традиционалних геометријских форми троцентричних кривина дефинисаних у домаћим смерницама.
- Постојећи дијаграм са кривама за избор централног радијуса $R2$ из домаћих смерница није адекватан за конструкцију троцентричних кривина.

Развијени софтверски алат може се искористити за израду нових дијаграма за избор централног радијуса $R2$ троцентричне кривине у зависности од најмањег полупречника окретања (R_s) изабраног возила и скретног угла γ водеће путање.

9. ЛИТЕРАТУРА

- [1] „Закон о јавним путевима“, Службени гласник РС бр.-101/2005;
- [2] Ј. Катанић, В. Катанић, М. Малетин, „Приручник за пројектовање путева“, Београд 1983.
- [3] С. Фриц, Д. Гавран, В. Илић, Ф. Трпчевски, С. Врањевац, „Анализа главних утицајних фактора на брзину и трајекторију кретања возила у путним кривинама“
- [4] С. Фриц, Д. Гавран, В. Илић, Ф. Трпчевски, С. Врањевац, „Експерименталне методе за одређивање граничних кривина проходности возила“
- [5] Ж. Корлает, В. Драгчевић, „Трајекторија повлачења у кинематичком моделу сложеног возила“, часопис Хрватског савеза грађевинских инжењера, 1999;
- [6] „Техничка упутства за пројектовање површинских раскрсница“, Приручник за пројектовање путева у Републици Србији;
- [7] Б. Илић, „Аналитичка метода за контролу проходности и прорачун елемента ситуационог плана површинских раскрсница“
- [8] З. Лалић, „Експлоатација и одржавање моторних возила“;
- [9] М. Ђорђевић, „Друмска моторна возила“;

Кратка биографија:



Марија Долић рођена је у Крушевцу 1994. год. Основне академске студије грађевинарства завршила је 2020. год на Факултету техничких наука. Мастер рад из области Грађевинарства – Саобраћајнице одбранила је 2022. године.

Контакт: marijadolic@gmail.com

PROCJENA STANJA I SANACIJA PROIZVODNE HALE U SKLOPU FABRIKE ŠEĆERA U CRVENKI SA ASPEKTA TRAJNOSTI I PROMENE NAMENE**ASSESSMENT AND REPAIR OF PRODUCTION HALL IN THE SUGAR FACTORY IN CRVENKA FROM THE ASPECT OF DURABILITY AND CHANGE OF PURPOSE**

Dejan Šipovac, *Fakultet tehničkih nauka, Novi Sad*

Oblast – GRAĐEVINARSTVO

Kratak sadržaj – Rad se sastoji iz teorijskog i praktičnog dijela. U teorijskom dijelu su opisani požari u industrijskim objektima: uzroci požara, neki od najvećih požara, zakonska regulativa za zaštitu od požara. U praktičnom dijelu urađena je procjena stanja konstrukcije hale u sklopu fabrike „Crvenka“. Kako bi se utvrdio stepen oštećenja i dao prijedlog sanacionih radova, sproveden je detaljan vizuelni pregled svih dostupnih elemenata konstrukcije. Zaključeno je da na AB i zidanim elementima postoje brojni defekti i oštećenja. Na osnovu analize registrovanih defekata i oštećenja dat je prijedlog sanacionih radova u cilju povećanja trajnosti konstrukcije hale.

Ključne reči: procjena stanja, defekti, oštećenja, sanacija, požari u industrijskim objektima

Abstract – The paper consist of a theoretical and a practical part. The theoretical part describes fire hazards in industrial buildings: causes of fire hazards, some of the largest fire hazards, legal regulations for fire protection in industrial buildings. The assessment of the construction of the hall withing the sugar factory „Crvenka“ A.D. was carried out in the practical part. In order to determine the level and cause of demages, as well as the type of repair measures, a detail visual inspection of structural elements was undertaken. Large number of defects and demages has been detected. Based on the analysis of those demages and defects, repair measures were suggested in order to improve the durability of the structure.

Keywords: assessment, defects, demages, repair, fire hazards in industrial buildings

1. POŽARI U INDUSTRIJSKIM OBJETIMA

Nagli razvoj industrije uticao je na povećanja broja različitih havarija i rizika, ali istovremeno je doveo i do razvoja nauke o održavanju i pouzdanosti tehničkih sistema, kao najefikasnijeg načina zaštite od različitih hazardnih situacija. Pojavom i razvojem opreme, računara i različitih tehnologija javljaju se i novi načini borbe protiv neželjenih događaja, čiji broj raste proporcionalno tehničko-tehnološkom razvoju same industrije.

Nastanak i razvoj požara je složeni proces, a posljedica je

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Mirjana Malešev, red. prof.

hemijskih i fizičkih procesa sa karakteristikama tehnološkog procesa, građevinskog materijala koji čini konstrukciju objekta, prisustvo zapaljivih materijala, kao i pouzdanosti tehničkih sistema za zaštitu od požara u sklopu objekta.

1.1. Uzroci požara u industrijskim objektima

Najčešći uzroci pojave požara u industrijskim objektima:

- Zapaljiva prašina,
- Toplotni radovi,
- Zapaljivi proizvodi,
- Neispravna oprema i mašine,
- Električne opasnosti.

Često dolazi do multiplikacije opasnosti (multirizika), pa na primjer u kombinaciji poput toplotnih radova i zapaljive prašine ti rizici postaju ozbiljni. U slučaju raspršivanja prašine u zatvorenom prostoru moguća je pojava i eksplozija.

1.2. Zakonska regulativa u zaštiti od požara industrijskih objekata u Republici Srbiji

Požare je vrlo teško spriječiti, pa se pitanjem ovoga rizika intenzivno bave državne vlade koje svoje strategije zaštite od požara kontinuirano unaprijeđuju.

Pravilnikom o tehničkim normativima za zaštitu od požara industrijskih objekata u Republici Srbiji propisuju se tehnički zahtjevi u pogledu zaštite od požara industrijskih objekata, odnosno njihovu izgradnju, dogradnju i rekonstrukciju.

Pravilnik se sastoji od 50 članova, u kojima se govori o uvodnim odredbama, pristupu za vatrogasna vozila, građevinskim mjerama za spriječavanje prenosa požara, evakuacionim putevima, tehnološkim mjerama bezbjednosti, sistemima ventilacije i grijanja, električnim instalacijama, liftovima, posebnim sistemima i mjerama, te se daju odredbe za proračunsko određivanje potrebne otpornosti na požar elemenata konstrukcije.

1.2.1. Građevinske mjere za spriječavanje prenosa požara

U sklopu pravilnika za zaštitu od požara industrijskim objekata daju se sledeća pravila, kao građevinske mjere, za spriječavanje prenosa požara:

- Bezbjedna udaljenost objekata (član 8), koja se ispunjava postavljanjem objekta na bezbjedonosno rastojanje ili izgradnjom posebnog zida prema susjednom objektu koji

može spriječiti prenos eksplozivne smješe požara i efekata eksplozije.

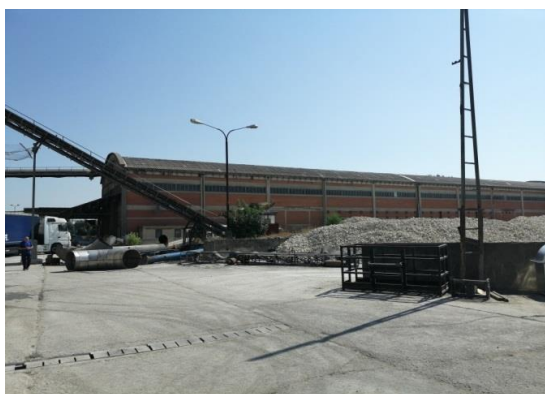
- Skladištenje zapaljivih materija (član 9) na udaljenosti manjoj od 6 metara od zida objekta.
- Podjela na požarne sektore (član 10) u zavisnosti od visine na kojoj se sektor nalazi u odnosu na kotu terena.
- Fasada objekta (član 11) – definisanje položaja zidova i požarne otpornosti istih na granicama požarnih sektora.

2. PROCJENA STANJA OBJEKTA

2.1. Opis konstrukcije

Predmetni objekat se nalazi u sklopu fabrike šećera „Crvenka“ A.D. u Ulici Masarikova broj 7, opština Kula. Objekat je po namjeni projektovan kao proizvodna hala, ali se proizvodnja u međuvremenu proširila i preselila u nove objekte, dok je predmetna hala pretvorena u podno skladište – magacin za skladištenje rezanca (Slika 1).

Objekat je prizemne spratnosti sa visinom sljemena krova na koti +13.45m od kote okolnog terena. Ukupna dužina objekta iznosi 101.10m, dok mu je širina 20.60m – zbog veličine objekta fizički je podijeljen (dilatan) na dva identična dijela.



Slika 1 – Izgled hale (južna fasada)

Dispoziciono, hala se sastoji od ukupno 10 polja (2 dijela po 5 polja), gdje je svako polje dužine 10m. Kalkanski zidovi, u poprečnom pravcu, su podijeljeni na 3 polja, od kojih su dva dužine 6.50m, dok je jedno, srednje, polje dužine 7.0m.

Konstrukciju objekta čine armiranobetonski elementi i zidani zidovi (ispuna). Kao glavni vertikalni elementi predviđeni su AB stubovi pravougaonog i „I“ poprečnog presjeka, te vertikalni serklaži kao ukrute zidanih zidova. Kao elementi koji povezuju sve vertikalne elemente konstrukcije predviđene su AB grede i AB horizontalni serklaži pravougaonog poprečnog presjeka.

Krovna konstrukcija objekta je armiranobetonska i sastoji se od glavnih, lučnih, nosača obrnutog „T“ poprečnog presjeka i sekundarnih nosača, armiranobetonskih rebara „T“ poprečnog presjeka. Fundiranje objekta obavljeno je na temeljima samcima međusobno povezanih temeljnim gredama.

Zidani zidovi izvedeni su po cijelom obimu objekta (između AB stubova) od pune opeke u dva reda, odnosno u debljini od 25cm.

2.2. Procjena stanja – vizuelni pregled objekta

Detaljnim vizuelnim pregledom obuhvaćeni su svi dostupni elementi konstrukcije:

- Armiranobetonski stubovi,
- Armiranobetonske grede,
- Armiranobetonski vertikalni serklaži,
- Armiranobetonski horizontalni serklaži,
- Armiranobetonska krovna konstrukcija,
- Zidani zidovi (zidana ispuna).

Vizuelnim pregledom detektovan je veliki broj defekata na AB elementima konstrukcije, kao što su: vertikalna i horizontalna linijska segregacija, geometrijske imperfekcije, mala debljina zaštitnog sloja, rupičasta/šupljikava površina betona, betonska jezgra i vidljivi prekidi betoniranja. Na zidanim elementima (zidana ispuna) registrovan je samo jedan tip defekata u vidu neadekvatnog izvođenja, odnosno nepostojanju adekvatne veze između zidanih zidova i AB vertikalnih serklaža objekta.

Vizuelnim pregledom registrovana su oštećenja na AB konstruktivnim elementima objekta, kao što su: mehanička oštećenja, mrežaste prsline na površini betona, vertikalne prsline i pukotine, vidljiva armatura sa korozijom, oštećenja izazvana korozijom armature, klobučenje, pucanje i odvajanje zaštitnog sloja betona i neadekvatni sanacioni radovi, te oštećenja na zidanim zidovima (zidana ispuna objekta): mehanička oštećenja, oštećenja usled neadekvatnih sanacionih radova, oštećenja usled horizontalnog opterećenja od uskladištenog materijala i oštećenja od mraza na opekama zidova.

Na narednim slikama (slike 2-9) ilustrovani su opisani karakteristični defekti i oštećenja elemenata konstrukcije.



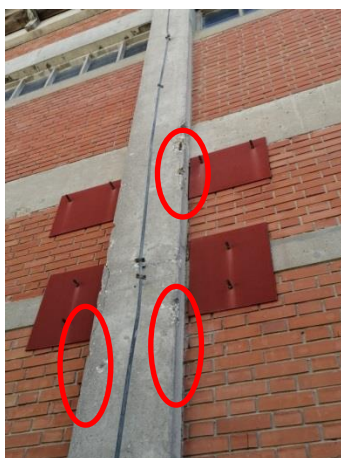
Slika 2 – Vertikalna suva linijska segregacija na AB stubu



Slika 4 – Mala debljina zaštitnog sloja na AB gredama i AB horizontalnim serklažima



Slika 3 – Geometrijska imperfekcija AB stuba



Slika 5 – Mehanička oštećenja na AB stubu



Slika 6 – Vidljiva armatura na mjestu mehaničkog oštećenja



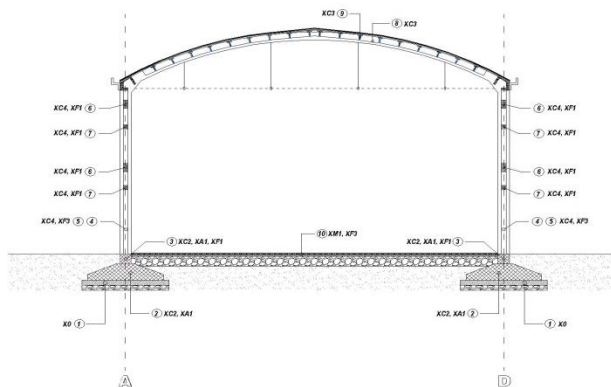
Slika 7 – Vidljiva armatura na AB gredi



Slika 9 – Oštećenje zidanog zida usled hor. opterećenja

3. KLASSE IZLOŽENOSTI

Bazirano na evropskim normama EN 206 i EN 1990, elementi AB konstrukcije predmetne hale su svrstani u odgovarajuće klase izloženosti – Slika 10.



Slika 10 – Klase izloženosti za elemente hale

4. SANACIJA OBJEKTA

Prema registrovanim defektima i oštećenjima na konstrukciji daje se prijedlog sanacionog rješenja (konstrukcijska i nekonstrukcijska sanacija). Prijedlog sanacionog rješenja prikazan je za sve elemente konstrukcije predmetnog objekta u tabelama 1 i 2.

Tabela 1 – Vrste sanacionih radova za pojedinačne elemente konstrukcije

Element konstrukcije	Konstrukcijska sanacija	Nekonstrukcijska sanacija
AB stubovi	Ojačanje AB stubova kontraforima	Čišćenje i zaštita korodiranih šipki i lokalna reprofiliacija oštećenog betonskog presjeka. Reprofiliacija mehaničkih oštećenja.
		Zaštita sa unutrašnje strane od hemijske agresije (od rezanca) i sa unutrašnje i spoljašnje strane od korozije armature izazvane karbonatizacijom.
AB vertikalni serklaži	Ojačanje AB vertikalnih serklaža kontraforima	Sanacija korodirane armature i lokalna reprofiliacija oštećenog betonskog presjeka. Reprofiliacija mehaničkih oštećenja.
		Zaštita sa unutrašnje strane od hemijske agresije (od rezanca) i sa unutrašnje i spoljašnje strane od korozije armature izazvane karbonatizacijom.
AB grede		Čišćenje i zaštita korodiranih šipki i lokalna reprofiliacija oštećenog betonskog presjeka.
		Lokalna reprofiliacija oštećenog zaštitnog sloja betona na zonama neadekvatno izvedenih sanacionih radova.
		Sanacija površinski korodiranih šipki na na bočnim stranama betonskih elemenata. Zaštita od korozije armature izazvane karbonatizacijom.

Tabela 2 – Vrste sanacionih radova za pojedinačne elemente konstrukcije

Element konstrukcije	Konstrukcijska sanacija	Nekonstrukcijska sanacija
AB horizontalni serklaži	/	Zaštita korodirane armature (na mjestima vidljive armature) i lokalna reprofiliacija oštećenog betonskog presjeka.
		Lokalna reprofiliacija oštećenog zaštitnog sloja betona na zonama neadekvatno izvedenih sanacionih radova, kao i u zonama pojave horizontalnih prslina na mjestima vidljivog prekida betoniranja.
		Zaštita od korozije armature izazvane karbonatizacijom
AB krovna konstrukcija	/	/
Zidani zidovi	Ojačanje zidanih zidova novim elementima u vidu AB zidova i kontrafora po cijelom obimu objekta.	/

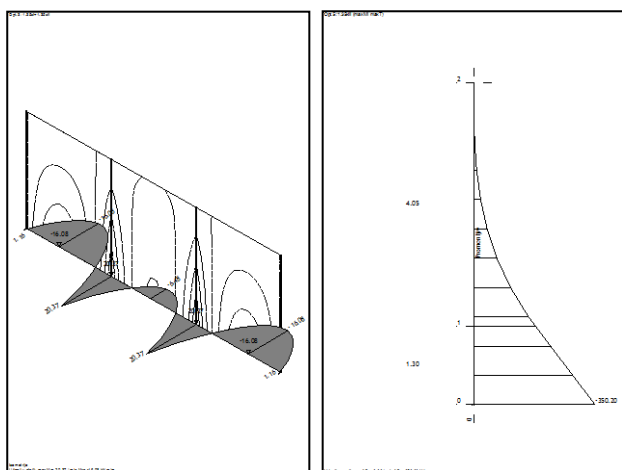
4.1. Opis sanacionih radova

Trošni, ispucali ili porozni dijelovi betona uklanjaju se primjenom pikhamer. Na isti način uklanjaju se i površinski slojevi betona, radi „otkrivanja“ armature, a u cilju ostvarivanja bolje athezije sa novim zaštitnim slojem betona. Dubina odštemovanog sloja betona zavisi od očuvanosti athezije između armature i betona i stepena korozije armature.

Čišćenje armature žičanim četkama, ručno ili mehanički primjenjuje se radi uklanjanja produkata korozije čelika i ostataka cementnog kamena – čišćenje armature do „metalnog sjaja“.

4.2. Konstrukcijska sanacija zidanih zidova, AB stubova i vertikalnih serklaža

Statički proračun novih elemenata za ojačanje sprovedena je u softveru Radimpex – Tower, pri čemu se nova AB dijafragma (zid) posmatrao površinski element, koji nosi u jednom pravcu, statičkog sistema kontinualnog nosača sa tri polja (Slika 11). Kontrafor se, statički, posmatrao kao konzolni nosač, uklješten na kontaktu sa temeljnom stopom, dok se sama stopa posmatrala kao pločasti element oslonjen na postojeće tlo.



Slika 11 – Statički proračun novih elemenata za ojačanje postojeće konstrukcije

Metoda konstrukcijske sanacije obuhvata:

- priprema površina starog betona na AB stubovima (hrapljavljenje) u cilju ostaviravanja bolje veze između postojećeg i novog betona,

- postavljanje ankera (armaturnih šipki) u postojeće stubove za vezu postojećih stubova i novih elemenata konstrukcije za ojačanje (AB zidova i AB kontrafora),
- postavljanje oplata,
- postavljanje armature novih elemenata za ojačanje postojeće konstrukcije, u svemu prema statičkom proračunu AB temeljnih stopa, zidova i kontrafora,
- betoniranje novih elemenata, u svemu prema prijedlogu konstrukcijske sanacije, trofrakcijskim betonom, klase C30/37 sa dodatnim svojstvima.

5. ZAKLJUČAK

U radu je data procjena stanja i prijedlog sanacionog rješenje za postojeću halu u okviru fabrike šećera „Crvenka“ A.D. Na osnovu svih prikupljenih podataka, dobijenih vizuelnim pregledom, mogu se izvesti sljedeći zaključci:

- Stabilnost zidanih zidova objekta je ugrožena,
- Trajnost objekta je smanjena. Postojanje velikog broja defekata, nastalih u fazama građenja i izvođenja konstrukcije, doveli su do progressa oštećenja tokom vremena,
- Nosivost konstrukcije, kao cjeline, je ugrožena na horizontalno dejstvo od uskladištenog materijala, na koje postojeća konstrukcija nije projektovana,

Predloženim sanacionim rješenjem se predmetna hala može dovesti u stanje funkcionalnosti uz obezbjeđenje trajnosti AB elemenata konstrukcije.

6. LITERATURA

- [1] Graditeljstvo i životna sredina, materijal sa predavanja, Doc. Dr Slobodan Šupić,
- [2] Pravilnik o tehničkim normativima za zaštitu industrijskih objekata od požara (“Sl. Glasnik RS”, broj 1/2018),
- [3] Malešev M., Radonjanin V.: “Trajnost i procena stanja betonskih konstrukcija”, skripta sa predavanja, Fakultet tehničkih nauka, Novi Sad
- [4] SRPS EN 206 i serija standarda SRPS EN 1504

Kratka biografija:



Dejan Šipovac rođen je u Trebinju, RS/BiH 04.02.1995. god. Master rada na Fakultetu tehničkih nauka iz oblasti Građevinarstvo – Konstrukcije, procena stanja i sanacija betonskih konstrukcija odbranio je 2022. godine.
Kontakt:
dejan.sipovac.351@gmail.com

**ПРИМЈЕНА СИСТЕМА ПОСЛОВНЕ ИНТЕЛИГЕНЦИЈЕ У БЕЗБЕДНОСТИ
САОБРАЋАЈА****APPLICATION OF BUSINESS INTELLIGENCE SYSTEMS IN TRAFFIC SAFETY**

Љубинка Видовић, Факултет техничких наука, Нови Сад

Област – САОБРАЋАЈ

Кратак садржај – Напретком информатичке технологије могуће је прикупити све веће количине релевантних података, као што су свеобухватне базе података о незгодама, као и податке о вожњи у реалном времену. Иновације у информационим технологијама омогућиле су развој унапријеђеног система, па се тако појавио и почео користити концепт пословне интелигенције у различитим областима, те и у области анализе саобраћајних незгода и безбедности саобраћаја.

Кључне речи: Безбедност саобраћаја, систем пословне интелигенције.

Abstract – Advances in information technology make it possible to collect ever-increasing amounts of relevant data, such as comprehensive accident databases as well as real-time driving data. Innovations in information technologies enabled the development of an improved system, and thus the concept of business intelligence appeared and began to be used in various areas, including in the area of traffic accident analysis and traffic safety..

Keywords: Traffic safety, Business Intelligence Systems.

1. УВОД

Саобраћајне незгоде представљају један од водећих узрока смртних случајева и тешких тјелесних повреда у свијету. Упркос значајном технолошком напретку у области безбедности саобраћаја, саобраћајне незгоде су и даље горући глобални здравствени проблем. Свјетска здравствена организација процјенила је да су повреде на путевима 8. водећи узрок смрти широм свијета, што доводи до 1,4 милиона смртних случајева годишње [11].

Напретком информатичке технологије могуће је прикупити све веће количине релевантних података, као што су свеобухватне базе података о незгодама и вожњи у реалном времену.

Надаље, дошло је до огромног побољшања у разноликости и могућностима алата и метода за анализу података који се могу примјенити на све кораке моделирања [1].

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Драган Јовановић, ред. проф.

2. СИСТЕМ ПОСЛОВНЕ ИНТЕЛИГЕНЦИЈЕ

Ефикасна комуникација је кључ напретка на свим пољима људских настојања. Последњих година постало је евидентно да су садашње комуникацијске методе потпуно неадекватне за будуће захтјеве. Информације се сада стварају и користе све бржим темпом због убрзаног развоја и обима људских активности и сталног пораста просјечног нивоа образовања. У исто вријеме, раст организација и повећана специјализација и поdjела створили су нове препреке за проток информација.

Такође постоји све већа потреба за бржим одлукама на нивоима одговорности далеко испод оних уобичајених у прошлости. Несумњиво највећи проблем комуникације је огромна количина информација које се морају детаљније и боље обрађивати. С обзиром на тренутне трендове раста, чини се да аутоматизација нуди најефикасније методе за проналажење и ширење информација.

2.1 Појам

Кад је ријеч о термину пословне интелигенције (БИ), он укључује технологије, апликације и начине (практике) прикупљања, интеграције, анализе и представљања пословних информација, са сврхом унапређења доношења пословних одлука. Термин пословна интелигенција је, релативно гледано, доста нов, те је његово увођење замијенило изразе: подршка одлукама и извршни информациони системи [2].

Према Reinschmidt and Francoise [6], БИ систем је „интегрисани скуп алата, технологија и програмираних производа који се користе за прикупљање, интеграцију, анализу и стављање података на располагање”. Једноставно речено, главни задаци БИ система укључују „интелигентно истраживање, интеграцију, агрегацију и вишедимензионалну анализу података који потичу из различитих информационих извора”. Имплицитно у овој дефиницији, подаци се третирају као високо вриједан корпоративни ресурс и трансформишу се из квантитета у квалитет [10].

БИ системи комбинују прикупљање података, складиштење података и управљање знањем са аналитичким алатима за представљање сложених интерних и конкурентних информација планерима и доносиоцима одлука. У овој дефиницији имплицитна је идеја (можда идеална) да системи пословне интелигенције пружају радње које се достављају у право вријеме, на правој локацији и у правом облику за помоћ доносиоцима одлука. Циљ је побољшати

правовременост и квалитет инпута у процесу доношења одлука, чиме се олакшава менаџерски посао [5].

2.2 Концепт

Концепт пословне интелигенције се све више користи посљедњих година у различитим подручјима.

Пословна интелигенција представља широко подручје апликација и технологија за прикупљање, складиштење, анализу и пружање приступа информацијама за побољшање квалитета моделирања пословних процеса [4].

Концепти пословне интелигенције односе се на употребу дигиталних рачунарских технологија у облику складишта података, аналитике и визуелизације са циљем да се идентификују и анализирају есенцијални подаци засновани на пословању како би се генерисали нови, ефикасни корпоративни увиди. БИ технологије нуде садашње (у реалном времену), историјске и предиктивне погледе на интерно структуриране податке који се односе на сва одјељења унутар организације, што експоненцијално унапређује оперативни увид и унапређује процес доношења одлука.

Једноставно речено: пословна интелигенција је процес откривања вриједних трендова или образаца у подацима ради доношења ефикаснијих и тачнијих одлука у вези са пословним циљевима и стратегијама.

2.3 Значај

Пословна интелигенција може помоћи компанијама да донесу боље одлуке приказујући садашње и историјске податке у оквиру њиховог пословног контекста. Аналитичари могу да искористе БИ да обезбиједи перформансе и референтне вриједности конкурената како би организација радила без проблема и ефикасније.

Аналитичари такође могу лакше уочити тржишне трендове како би повећали продају или приход. Када се ефикасно користе, прави подаци могу помоћи у било чему, од усклађености до запошљавања.

Значај БИ је у томе што помаже у доношењу брзих одлука. То је једна од могућности које омогућавају савремени БИ алати и праксе: Доношење одлука које иду у корак (или довољно близу томе) са брзином којом се подаци производе.

Подаци се сада производе брзо и у великим количинама да их је немогуће анализирати и ефикасно користити када се користе традиционалне, ручне методе као што су табеле, које су подложне људској грешци. Предност БИ је у томе што аутоматски анализира податке из различитих извора, који су тачно представљени на једној контролној табли која се лако чита.

Често постоји превише информација које су несхватљиве у свом сировом облику или би захтевале неvjероватну количину времена да се вриједност у њима утврди. БИ ту велику количину информација трансформише у употребљив облик, омогућавајући крајњим корисницима да заиста искористе податке који су им представљени. Због тога многи алати и платформе стављају велики нагласак на контролне табле и визуелизацију као примарни.

3. ОКВИР ПОСЛОВНЕ ИНТЕЛИГЕНЦИЈЕ ЗА ПРЕГЛЕД БЕЗБЈЕДНОСТИ МРЕЖЕ ПУТЕВА

Један од критичних програма ХСИП-а (The Highway Safety Improvement Program) јесте процес управљања безбједношћу саобраћаја, који укључује годишње извјештавање о локацијама путева које показују најоштрије потребе за унапређење безбједности саобраћаја. Идентификовањем најопаснијих локација на путу, могу се примјенити посебне противмере које би унаприједиле безбједносне услове.

Препреке које спречавају употребу теоријски исправних метода укључују:

- 1) значајне потребе за подацима и интеграцију,
- 2) потребна је посебна шема да би се омогућила анализа помоћу специјализованог софтвера,
- 3) потребни су дуготрајни и интензивни процеси,
- 4) недостаје релевантно техничко знање,
- 5) недостају способности визуелизације,
- 6) потребна је координација између различитих власника података.

Да би се ријешиле ове баријере инжењери су дошли до новог метода. Оквир је развијен коришћењем приступа пословне интелигенције (БИ), који пружа методе и механизме за интеграцију и обраду података, генерисање напредне аналитике и визуелизацију резултата. Предложени оквир олакшава инжењеринг безбједности саобраћаја и побољшава резултате ХСИП-а. У саобраћајном инжењерству и безбједности саобраћаја развијено је неколико приступа.

Тренутно, за преглед мреже за појединачне локације (сегменте коловоза, рампе и раскрснице), ХСМ је препоручио коришћење очекиване/прекомијерне учесталости незгода, методом ЕБ.

Скрининг мреже на нивоу коридора, важан је за доносиоце одлука јер омогућава рангирање деоница, а не локација, како би се обезбједила хомогена инфраструктура за смањење промјена на релативно кратким удаљеностима. Унапређења се препоручују за дугачке деонице путева који би могли укључивати више локација са потенцијалом за унапређење безбједности саобраћаја [8].

Преглед мреже на нивоу коридора важан је за такве програме да идентификују коридоре који имају потребу за унапређењем безбједности. Неколико студија је користило уочену учесталост незгода, стопе незгода или индекс озбиљности незгода за преглед. Коришћење посматраних фреквенција незгода резултира пристрасношћу запремине, док употреба стопа незгода резултира пристрасношћу дужине сегмента; поред тога, коришћење посматраних незгода резултира пристрасношћу регресије на средњу вриједност.

За скрининг коридора, у [7] су користили незгоде које су се догодиле само на главним путевима на раскрсницама. Занемаривање интеракција главних и споредних карактеристика пута на раскрсницама утиче на предвиђену учесталост незгода, што доводи до погрешне процјене очекиване учесталости незгода. Друге студије нису претраживале деонице које су имали потенцијал за унапређење безбједности, већ је умјесто тога процијенио учесталост незгода на унапред агрегираним веб локацијама [3].

4. ПРИМЕНА АЛАТА У ПОСЛОВНОЈ ИНТЕЛИГЕНЦИЈИ

4.1 ГИС

Током протекле деценије, географски информациони системи (ГИС) постали су све важнији као подржавајућа и омогућавајућа технологија у пословном окружењу, а посебно у сфери пословне интелигенције (БИ).

Све више компанија, посебно у финансијским услугама и индустрији осигурања, почеле су да примењују ГИС у покушају да боље разумеју сложено и динамично пословно окружење. Менаџери у компанијама више користе ГИС технологију као резултат широке употребе ГПС-а за пословање и разоноду увођењем веб ресурса са омогућеном картом, попут „Google Earth“.

Локација је критична компонента у готово свакој пословној трансакцији. Иако многи подаци имају димензију локације, било да се ради о купцима, продавницама, складиштима или другој имовини, ове се информације ријетко користе у традиционалној анализи пословне интелигенције. Да би стекле максималну вриједност из све веће количине података, компаније морају да искористе елемент локације како би стекле дубљи увид у пословање како би побољшале конкурентност и пословање.

Комбинујући аналитичку моћ база података са географским могућностима карата, омогућава пословним корисницима да истражују и анализирају односе између географских података и пословних података. Док су алати за БИ идеални за анализу ко, шта и када, ова анализа даје одговоре на питања у вези са тим гдје.

ГИС је кључан у преошћивању јаза између интерних и екстерних података. Векторске и растерске топографске карте, ваздушне фотографије, пописни и други демографски подаци, па чак и сателитски снимци могу постати изузетно вриједне пословне информације, ако се анализирају заједно са интерним подацима у заједничком оквиру.

ГИС има способност да помогне БИ професионалцима да брже пронађу више података и информација повезивањем многих унутрашњих и спољних „цијеви“ података кроз локацију. У ствари, унапријеђени ГИС систем пословне интелигенције постаје права „све на једном месту“.

Због велике количине података које компаније похрањују о готово свим могућим аспектима свог пословања, можда је неизбежно да се понекад догоде грешке које деградирају корисност и вјеродостојност информација. Код многих типова података, попут великих клијентских база података, неке од ових грешака може бити тешко или чак немогуће открити у табличним форматима базе података. Често се неке од ових грешака могу открити тек када се ти подаци увуку у ГИС кроз процес геокодирања и мапирају или просторно анализирају.

Примарна разлика између ова два је у томе што се ГИС фокусира на локацију, а БИ се фокусира на пословне системе. Организацију која купује систем пословне интелигенције првенствено занима разумјевање начина на који могу побољшати своју

организацију анализом унутрашњих система, запослених и конкуренције. С друге стране, организација са ГИС-ом је заинтересована за анализу просторних односа и образаца. Они затим користе ове информације да одреде најбољу локацију за компаније, утврде ризике у сврхе, идентификују локацију и коришћење комуналних услуга, и разумеју понашање и кретање људи кроз јавни превоз и малопродајне локације.

На крају, ГИС и пословна интелигенција два су различита алата који раде заједно како би организацији доставили више информација него што би то могли сами. Додавањем компоненте локације у пословну анализу организација може боље разумети просторне односе, обрасце и трендове. Ова информација о локацији помаже у побољшању пословања у индустријама и подручјима као што су усмјеравање и логистика, комуналије, осигурање и управљање ризицима, малопродаја, планирање комерцијалних некретнина и корисничка услуга.

4.2 „SAFETY ANALYST

Safety Analyst“ је систем управљања безбједношћу саобраћаја који омогућава да се постигну циљеви без визије повећањем тачности мрежних скрининга и идентификовањем ефикасних безбједносних програма коришћењем напредне аналитике и визуелизације података.

„Safety Analyst“ има следеће могућности:

- Управља ризицима, али и ублажава их
- Идентификује изложеност ризику према категорији средстава, саставља регистар ризика и процјењује који ризик је приоритет за планирање, анализу и ублажавање.
- Постоји могућност повећања перформанси програма

Користећи моћне упите и интегрисане ГИС/ЛРС алате за презентацију могуће је извршити напредну анализу и самим тим повећати перформансе програма.

- „Harness Analytics“

Могуће је креирати оптималне планове рада користећи предиктивно моделирање да би се одредило најбоље вријеме за примјену превентивног третмана који одржава елементе, компоненте и њихову матичну структуру у добром стању.

- Генерисати потпуније извјештаје

Потребно је визуелизовати жаришне тачке и друге податке у табеларним, графичким и ГИС извјештајима заснованим на мапама да би пратили учинак у односу на циљеве.

- Дефинише контрамјере

Генерише интерактивне извјештаје који саопштавају ефекте инвестиционих одлука и перформансе мреже у односу на различите мјере.

- Означава проблематична подручја

Користи проактивне или реактивне моделе да би идентификовао велики број локација незгода и да би се увидјели фактори који доприносе томе.

- Испуњава стандарде

Подржава развојне безбједносне програме који су у складу са Приручником за безбједност на путу, као и другим правилима и законима.

Иако постоји више извора података који пружају огромну количину потребних података за „Safety

Analyst“, недостајали су различити атрибути података или су непотпуни. Већина података који недостају је прикупљена користећи “Google Earth”. Развијен је алат за прикупљање података, који служи за издвајање података као и за креирање “ArcGis“ датотека. Ова способност је олакшала развој и интеграцију база података.

„Safety Analyst“ захтјева да сви прикупљени подаци буду интегрисани коришћењем [9]:

- 1) руте;
- 2) пута, окружења и километраже;
- 3) трасе, деонице и растојања; или
- 4) пресека и удаљености.

Са „Safety Analyst“ могуће је унаприједити ефикасност планова за безбједност саобраћаја примјеном системске и поновљиве праксе управљања безбедношћу — генерисањем плана управљања и праћењем учинка у односу на план. Користећи моћне упите и интегрисане ГИС алате за презентацију могуће је извршити напредну просторну анализу да би била брзо идентификована проблематична подручја. Користећи проактивне или реактивне моделе за симулацију или идентификовање локације са великим бројем незгода, укључујући факторе који доприносе томе под различитим условима, да би боље разумјели како ублажити ризике. Постоје препоручени пројекти за постизање дефинисаних циљева смањења незгода са погинулима, одговорности, незгода са мањом материјалном штетом и још много тога, широм путне мреже или дуж коридора.

5. ЗАКЉУЧНА РАЗМАТРАЊА

Велике организације и мултинационалне компаније већ су имплементирале рјешење за пословну интелигенцију. Мада је имплементација ове врсте система изузетно напорна и скупа, користи су се показале много више. Као што је доказано у раду, интегрисање чувања и експлоатације података може донети важне предности. Главна предност у коришћењу пословне интелигенције је могућност претварања података у информације.

Предности могу бити важне за менаџмент компаније при доношењу стратешких одлука, али такође могу помоћи и руководиоцима одјељења, аналитичарима или било ком другом члану тима који се суочава са доношењем одлука. Анализа интелигентних података је одувек била важна јер се кроз ову анализу ствара интелигенција.

Данашњи путеви имају све предиспозиције да буду безбједнији за вожњу из дана у дан, а са доступном анализом великог броја података требало би и да буду безбједнији. Свакодневни возачи могу да спријече потенцијалне саобраћајне незгоде, повреде и смртне случајеве јер имају приступ најбољим подацима и технологији. Подаци се могу прикупљати и анализирати у реалном времену и кроз историјска истраживања. Што се више ових података приказује возачима, мања је могућност да се догоде незгоде и прекршаји.

6. ЛИТЕРАТУРА

- [1] Anda, I., Isah, R. O., & Enesi, F. A. (2017). A Safety Data Model for data analysis and decision making.
- [2] Dedić N., Stanier C., Measuring the Success of Changes to Existing Business Intelligence Solutions to Improve Business Intelligence Reporting, Беч, 2016
- [3] Hamidi, A., Fontaine, M. D., & Demetsky, M. J. (2010). A Planning-Level Methodology for Identifying High-Crash Sections of Virginia's Primary System (No. FHWA/VTRC 11-R4). Virginia Transportation Research Council.
- [4] Hancock, J., Toren, R., Practical Business Intelligence With SQL Server 2005, Addison Wesley Professional, 2006.
- [5] Negash., S., Business Intelligence, Kennesaw State University, 2004.
- [6] Reinschmidt, J., Francoise, A. Business Intelligence Certification Guide, IBM International Technical Support Organization, San Jose, CA, 2000.
- [7] Shimko, G. & Walbaum, C., City of Edmonton traffic safety strategy. Road Safety Session of the 2010 Annual Conference of the Transportation Association of Canada, Halifax, Nova Scotia., 2010.
- [8] Tjandra, S. A. (2014). Business Intelligence system for traffic data integration: linking roadway, collision and traffic flow data to improve traffic safety. NATMEC 2014 Improving Traffic Data Collection, Analysis, and Use.
- [9] Veeramisti, K., *A Business Intelligence Framework for Network-level Traffic Safety Analyses*, University of Nevada, Las Vegas, 2016.
- [10] Wang, H., Wang S., A Knowledge Management Approach to Data Mining Process for Business Intelligence, Industrial Management & Data Systems, 2008.
- [11] World Health Organization. WHO. (2020). The Top 10 Causes of Death.

Кратка биографија:

Љубинка Видовић је мастер рад на Факултету техничких наука из области Безбедност саобраћаја одбранила 2022. год.

**АНАЛИЗА ЖЕЛЕЗНИЧКИХ НЕСРЕЋА НА ПУТНИМ ПРЕЛАЗИМА НА
ТЕРИТОРИЈИ АП ВОЈВОДИНЕ****ANALYSIS OF RAILWAY ACCIDENTS AT ROAD CROSSINGS IN THE TERRITORY
OF AP VOJVODINA**

Букашин Бурсаћ, Гордан Стојић, Факултет техничких наука, Нови Сад

Област – САОБРАЋАЈНО ИНЖЕЊЕРСТВО

Кратак садржај – У оквиру овог истраживачког рада анализиране су железничке несреће на путним прелазима на територији АП Војводине у периоду од 2016. до 2021. године. Друштвени трошкови које проузрокују ове несреће су веома високи и доводе до смањења безбедности саобраћаја. Из тог разлога анализа ових нежељених догађаја нам је веома важна како бисмо уочили одговарајуће трендове у погледу њихове просторне и временске појаве и одредили одговарајуће превентивне мере како бисмо смањили њихову појаву у будућности.

Кључне речи: Несреће и незгоде на путним прелазима, железнички саобраћај, безбедност железничког саобраћаја

Abstract – As part of this research document, railway accidents at road crossings on the territory of AP Vojvodina were analyzed in the period from 2016 to 2021. The social costs caused by these accidents are very high and lead to a decrease in traffic safety. For this reason, the analysis of these unwanted events is very important to us in order to observe appropriate trends in terms of their spatial and temporal occurrence and to determine appropriate preventive measures in order to reduce their occurrence in the future.

Keywords: Accidents and incidents on the level crossing, Railway traffic, railway traffic safety

1. УВОД

Безбедност саобраћаја представља један од најважнијих показатеља сваког вида саобраћаја и један је од основних критеријума при избору начина превоза како људи, тако и робе. Под безбедношћу у железничком саобраћају подразумевају се услови које морају испуњавати железничке пруге, железничка возила и постројења, објекти, уређаји и опрема који се користе у железничком саобраћају, железнички радници и друга лица, као и други услови који су од значаја за остваривање безбедног, уредног и несметаног одвијања железничког саобраћаја [1].

Путни прелазни су посебна врста раскрсница где се укрштају железнички и друмски саобраћај. Овај део

саобраћајне површине служи и друмском и железничком саобраћају. На овим раскрсницама често се дешавају саобраћајне несреће и незгоде, које обично имају тешке последице.

За потребе израде рада коришћена је база података о несрећама и незгодама на територији Акционарског друштва за управљање јавном железничком инфраструктуром „Инфраструктура железнице Србије“ у периоду од 2016. до 2021. године.

Анализа несрећа и незгода на путним прелазима вршена је само у односу на путне прелазе који су под надлежношћу јавне железничке управе. Нису коришћени подаци за туристичко-музејску железницу, као део јавне железничке инфраструктуре, као ни подаци и преглед путних прелаза индустријске железнице и индустријских колосека, код којих је Актом о индустријским колосецима и посебним уговорима регулисано прикључење индустријског колосека на јавну железничку инфраструктуру [2].

2. БЕЗБЕДНОСТ ПУТНИХ ПРЕЛАЗА**2.1. Појам путних прелаза**

Место на којем се састају два вида саобраћаја са различитим техничким и технолошким карактеристикама је место које генерише безбедносне ризике у саобраћају. То место на линијским инфраструктурним објектима је место укрштаја пруге и пута. Место укрштаја пруге и пута може бити денивелисано при чему су безбедносни ризици сведени на незнатне. Прописи који уређују област укрштања пруге и пута, како законски тако и подзаконски, усвајају, на жалост, различите термине о једном истом појму. Данас налазимо термине у прописима који се односе на укрштај пруге и пута као што су: путни прелаз, укрштај, прелаз пута преко пруге, место укрштања железничке пруге и пута.

Терминолошка неусклађеност истог појма настала је због прописа који доносе стручњаци за област друмског и железничког саобраћаја без усаглашавања. Сходно датом појашњењу у овом раду коришћен је термин „путни прелаз“, који је дефинисан и материјалним законом који се односи на уређење железничког система у Републици Србији – Закон о железници.

2.2. Елементи безбедности на путним прелазима

У овом раду дефинисани су следећи елементи за безбедно регулисање и одвијање железничког и друмског саобраћаја на путним прелазима:

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији је ментор био др Гордан Стојић, ред. проф.

1. нормативно уређење изградње, одржавања, експлоатације и организације саобраћаја на путним прелазима,
2. инфраструктурни елементи на путним прелазима,
3. сигнализација железничког и друмског саобраћаја,
4. стање и опрема железничких возила и
5. учесници у саобраћају и њихова карактеристична понашања на путним прелазима.

2.2.1 Нормативно уређење изградње, одржавања, експлоатације и организације саобраћаја на путним прелазима

Начин безбедног одвијања, како железничког тако и друмског саобраћаја, на путним прелазима је уређено у законској и подзаконској регулативи. То је оквир који дефинише обавезе у поступању свих учесника у саобраћају на путном прелазу. Закони који уређују област путних прелаза су:

- 1) Закон о железници
- 2) Закон о безбедности у железничком саобраћају
- 3) Закон о путевима,
- 4) Закон о безбедности саобраћаја на путевима

2.2.2 Нормативно уређење изградње, одржавања, експлоатације и организације саобраћаја на путним прелазима

Путни прелаз, укључујући и средства за осигурање путног прелаза, је један од елемената који чине железничку инфраструктуру. Због тога се путни прелаз не посматра као посебан грађевински објекат већ као део линијског инфраструктурног објекта. Ипак и путни прелаз, као елемент железничке инфраструктуре, је уређен у складу са одређеним условима. Ти услови су: просторни, грађевински, саобраћајни, безбедносни.

2.2.3 Сигнализација железничког и друмског саобраћаја на путним прелазима

Железничка сигнализација на путним прелазима уређена је Правилником о врстама сигнала, сигналних ознака и ознака на прузи („Сл.гласник РС” број 51/20 од 14.04.2020. године) а сигнализација у друмском саобраћају Правилником о саобраћајној сигнализацији („Службени гласник РС”, бр. 85/2017 и 14/2021).

У железничком саобраћају постоје сигнали само за аутоматске уређаје на путним прелазима. Могу бити:

- 1) контролни светлосни сигнали;
- 2) помоћни контролни светлосни сигнали, по потреби;
- 3) сигнали укључне тачке.

Сигнални знаци који се дају наведеним сигнаlima намењени су учесницима железничког саобраћаја на путном прелазу, а то је састав возног особља, пре свега машиновођи и његовом помоћнику који непосредно управљају железничким вучним возилима.

Учесници у друмском саобраћају се саобраћајним знаковима на путу обавештавају и упозоравају о наиласку на путни прелаз, врсти осигурања саобраћаја на путном прелазу и удаљености од путног прелаза и др.

2.2.4 Учесници и њихова карактеристична понашања на путним прелазима

Највећи број несрећа и незгода на местима укрштања пута и железничке пруге своје узроке има у субјективним личним грешкама и пропустима учесника у друмском саобраћају. Изузетно је мало регистровано несрећа и незгода за које је одговорна железница. Приликом истраживања фактора настанка грешака код возача у друмском саобраћају уочене су следеће грешке: Грешке и фактори грешке током војње, грешке у перцепцији, грешке у дијагнози, грешке у прогнози, грешке при доношењу одлука, извршне грешке, опште грешке, ванредне ситуације [2].

3. ОСНОВНЕ СТАТИСТИЧКЕ МЕТОДЕ

Постоји више дефиниција статистике. Према једној статистика је саставни део активности научних, образовних, привредних и других институција. Статистика је научни метод који се користи за прикупљање, приказивање, анализу и тумачење различитих врста података.

Према другој статистика је скуп метода које доприносе да се дође до веродостојних закључака и одлука у условима неизвесности. Статистика се односи на скуп нумеричких података о стању посматране појаве [3].

Статистику делимо на: дескриптивну статистику и инференцијалну (аналитичку) статистику.

Друга подела статистике односи се на технике које се користе у статистици. Разликују се:

1. Параметарска статистика - прорачуни се базирају на нормалној (Гаусовој) расподели
2. Непараметарска статистика – спроводе се тестови који не морају подразумевати нормалност дистрибуције података којима располажемо [3].

Технике које се користе у параметарској статистици су следеће: аритметичка средина, стандардно одсупање (девијација), анализа варијансе (енг. Analysis of variance - ANOVA), Пирсонов продукт – коефицијент корелације итд.

Код непараметарске статистике користе се следеће технике: Спирманов коефицијент корелације, Хи-квадрат тест (χ^2 тест), Колмогоров Смирнов тест, Андерсон-Дарлингтонов тест, Крускал-Валисов тест, медијана, модоу и сл.

3.1. Дефиниција и класификација хипотеза

Хипотеза се може дефинисати као изјава или предвиђање о једној популацији или односу између две или више популација. Разликују се истраживачке и статистичке хипотезе.

Истраживачка хипотеза представља исказ о очекивањима/предвиђањима истраживача. Оне су проверљиве изјаве или предвиђање о повезаности или разлици између варијабли. Статистичке хипотезе сажимају истраживачке хипотезе повезујући их са параметрима популације. Могу се поделити на: Нулту хипотезу - H_0 и, Алтернативну хипотезу - H_1 [4].

4. ВРЕМЕНСКА ДИСТРИБУЦИЈА НАСТАНКА НЕСРЕЋА НА ПУТНИМ ПРЕЛАЗИМА

За потребе овог рада, анализирана је интерна база података предузећа „Инфраструктура железнице Србије“ а.д. у периоду од 2016. до 2021. године у оквиру које је обрађена временска дистрибуција појаве несрећа и незгода на путним прелазима на територији АП Војводине и то као месечна, седмична и дневна (часовна) дистрибуција.

4.1. Месечна дистрибуција

Укупан број железничких несрећа на путним прелазима дат је на слици 1.



Слика 1. Број несрећа на путним прелазима по месецима у периоду од 2016. до 2021. године на територији АПВ.

У августу месецу, као месецу са највећим бројем несрећа на путним прелазима на територији АПВ у анализираном периоду догодило се 14 несрећа од чега чак њих укупно седам (50 %) чине налети воза на трактор и то на пружним прелазима без полубраника или браника, обезбеђеним друмским саобраћајним сигнализима и осигураним троуглом прегледности.

У осталим случајевима дошло је до налета воза на путнички аутомобил и то такође на путним прелазима обезбеђеним зonom потребне прегледности и друмским саобраћајним знацима, док је забележен и један случај налета воза на прелазу обезбеђеним аутоматским уређајем са полубраницима на друмско путничко возило које се провлачило између спуштених полубраника. Такође, у два случаја забележен је и налет воза на пешаке. У свим случајевима одговорност за несрећу пада на терет возача односно трећих лица, а узрок код свих несрећа јесте непажња.

4.2. Седмична дистрибуција

У оквиру анализе несрећа и незгода на путним прелазима у периоду од 2016. до 2021. године, обрађена је и седмична дистрибуција појаве саобраћајних несрећа и незгода на територији АП Војводине (Слика 2.).



Слика 2. Седмична дистрибуција несрећа на путним прелазима на територији АПВ у периоду од 2016. до 2021. године.

4.3. Часовна дистрибуција

Анализирајући време настанка укупног броја несрећа на путним прелазима на територији АП Војводине у периоду од 2016. до 2021. године, долазимо до информације да се највећи број несрећа догађа у јутрањим часовима и то у периоду између 8 и 9 часова. Најмањи број несрећа на путним прелазима на територији АП Војводине дешава се у раним јутарњим и касним ноћним часовима (Слика 3.).



Слика 3. Часовна дистрибуција броја несрећа на путним прелазима на територији АПВ у периоду од 2016.-2021. године.

5. ТЕСТИРАЊЕ ПАРАМЕТАРСКИХ ХИПОТЕЗА ПОЈАВЕ ВРЕМЕНСКОГ НАСТАНКА ЖЕЛЕЗНИЧКИХ НЕСРЕЋА НА ПУТНИМ ПРЕЛАЗИМА НА ТЕРИТОРИЈИ АП ВОЈВОДИНЕ

За потребе овог рада а на основу саме природе и врсте података који су били на располагању, коришћен је тест двофакторске анализе варијансе различитих група – ANOVA и то преко програмског додатка Data Analasys у Microsoft Excel рачунарском програму. Тест је спроведен на основу временске расподеле појаве настанка несрећа на путним прелазима на територији АП Војводине у периоду од 2016-2021. године и то у односу на специфичне периоде појаве несрећа у току године, седмице и дана.

5.1. Двофакторска анализа варијансе различитих група – ANOVA појаве несрећа на путним прелазима у односу на специфичне периоде у години.

Анализирајући летњи период и период пролећа, узимајући при томе месец јул и август као референтне представнике за летећи период и месец март и април као референте представнике за период пролећа, постављена је нулта хипотеза (H_0) да не постоји

значајна разлика између броја несрећа на путним прелазима на територији АП Војводине у периоду од 2016. до 2021. године у летњим месецима и месецима пролећа.

Како је добијени резултати, применом двофакторске анализе варијансе различитих група преко додатка Data Analysis у Microsoft Excel рачунарском програму, показују да је вредност $F=13.23 > F(0.05)=4.74$, одбацујемо нулту хипотезу (H_0), тј. долазимо до закључка да постоји значајна разлика у броју несрећа на путним прелазима на територији АП Војводине у периоду од 2016. до 2021. године у летњим месецима и месецима пролећа. Међутим унутар група добини резултати показују да је $F=1.8 < F(0.05)=3.10$, тј. да немамо основа да одбацимо хипотезу (H_0).

5.2. Двофакторска анализа варијансе различитих група – ANOVA појаве несрећа на путним прелазима у односу на специфичне периоде у седмици

Анализирајући седмичну појаву несрећа на путним прелазима на територији АП Војводине у периоду од 2016. до 2021. године (Слика 2.) постављена је нулта хипотеза (H_0) да не постоји значајна разлика између броја несрећа у данима викенда (петак и субота) и радним данима (среда и четвртак).

Како добијени резултати, применом двофакторске анализе варијансе различитих група преко додатка Data Analysis у Microsoft Excel рачунарском програму, показују да је вредност $F=0.07 < F(0.05)=4.74$ и унутар група $F=0.23 < F(0.05)=3.10$, немамо основа да одбацимо нулту хипотезу H_0 тј. не постоји значајна разлика у појави броја несрећа на путним прелазима у данима викенда и радним данима.

5.3. Двофакторска анализа варијансе различитих група – ANOVA појаве несрећа на путним прелазима у односу на специфичне часовне периоде у оквиру дана

На основу анализе појаве укупног броја несрећа на путним прелазима на територији АП Војводине у периоду од 2016 до 2021. године у погледу времена њиховог настанка унутар дана, постаљена је нулта хипотеза (H_0) и то да не постоји значајна разлика између броја несрећа које се јављају пре подне (између 8, 9, 10 и 11 часова) и у вечерњим периодима дана (између 19, 20, 21, и 22 часа).

Добијени резултати, применом двофакторске анализе варијансе различитих група преко додатка Data Analysis у Microsoft Excel рачунарском програму показују да је вредност $F=6.31 > F(0.05)=4.11$, стога одбацујемо нулту хипотезу (H_0), тј. долазимо до закључка да постоји значајна разлика између броја несрећа на територији АП Војводине у периоду од 2016. до 2021. године које се јављају пре подне (између 8, 9, 10 и 11 часова) и у вечерњим часовима (између 19, 20, 21, и 22 часа). Међутим унутар група добијени резултати показују да је $F=0.311 < F(0.05)=2.47$, тј. да немамо основа да одбацимо хипотезу (H_0).

6. ЗАКЉУЧАК

У оквиру овог рада анализирана је појава железничких несрећа на путним прелазима на територији АП Војводине.

У оквиру рада постављено је неколико нултих хипотеза за које су применом двофакторске анализе варијансе различитих група у Microsoft Excel рачунарском програму добијени различити резултати. Тако је резултат теста показао да постоји значајна разлика у броју несрећа на путним прелазима на територији АП Војводине у периоду од 2016. до 2021. године у летњим месецима и месецима пролећа. Применом истог теста утврђено је да на територији АП Војводине у периоду од 2016. до 2021. не постоји значајна разлика у броју несрећа на путним прелазима у дане викенда и радним данима као и да не постоји значајна разлика у појави броја несрећа на путним прелазима пре подне (између 8, 9, 10 и 11 часова) и после подне (између 13, 14, 15 и 16 часова). Такође је утврђено да постоји значајна разлика између броја несрећа које се јављају пре подне (између 8, 9, 10 и 11 часова) и у вечерњем периодом дана (између 19, 20, 21, и 22 часа).

7. ЛИТЕРАТУРА

- [1] Ерцеговац П., докторска дисертација: Модел за процену и компарацију ризика од настанка несрећа и незгода на путно-пружним прелазима, Факултет техничких наука, Нови Сад, 2021.
- [2] Стојић Г., Танацков И., Ерцеговац П., Елаборат о дефинисању елемената за процену ризика од настанка железничких несрећа на путно - пружним прелазима на територији АП Војводине, 2022.
- [3] Шолак Њ., Статистика у економији и менаџменту, Младеновац, 1999.
- [4] Стојић Г, материјал са предавања и вежби из предмета Моделирање у саобраћају и транспорту, Факултет техничких наука у Новом Саду.

ЗАХВАЛНИЦА

Рад је финансиран са пројекта: Развој иновативних решења у функцији развоја саобраћаја и транспорта, Департамента за саобраћај, Факултета техничких наука.

Кратка биографија:



Вукашин Вурсаћ рођен је у Новом Саду 1994. год. Мастер рад на Факултету техничких наука из области саобраћајног инжењерства одбранио је 2022. год.
контакт: vukasinbursac94@gmail.com



Др Гордан Стојић, редовни професор, рођен је у Куманову 1971. год., дипломске и магистарске студије завршио на Саобраћајном факултету у Београду, а докторску дисертацију одбранио на Факултету техничких наука 2010. године. Бави се техничко-технолошким истраживањем и организацијом, пре свега, железничког саобраћаја и транспорта као и развојем модела у саобраћају и транспорту. Контакт: gordan@uns.ac.rs

**MAPIRANJE, LOKALIZACIJA I NAVIGACIJA PO NERAVNOM TERENU
AUTONOMNO VOĐENOG VOZILA PROMJENLJIVE GEOMETRIJE****MAPPING, LOCALIZATION AND NAVIGATION ON UNEVEN TERRAIN OF
AUTOMATED GUIDED VEHICLE WITH VARIABLE GEOMETRY**

Aleksandar Kičić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – Rad opisuje mapiranje, lokalizaciju i navigaciju po neravnom terenu autonomno vođenog vozila promjenljive geometrije. Zadatak rada bilo je istražiti i implementirati algoritme za automatsko mapiranje, lokalizaciju i navigaciju po neravnom terenu za autonomno vođeno vozilo. Pri tome bilo je potrebno na osnovu dostupne literature proučiti postojeće algoritme za automatsko mapiranje, lokalizaciju i navigaciju i odabrati odgovarajuće, prilagoditi odabrani pristup problemu navigacije na neravnom terenu i implementirati predloženo rješenje na autonomno vođenom vozilu. Predloženo je rješenje u obliku lidar filter čvora koji kao ulaz ima nefiltrirane vrijednosti distanci čitanja lidara i uglova rotacije sa inercijalnog senzora. Na osnovu ovih podataka filter daje kompenzacioni vrijednosti na određenu temu unutar ROS-a. Na kraju je eksperimentalno provjereno predloženo rješenje i upoređeno sa postojećim rješenjem uz pomoć Vicon sistema.

Ključne reči: AGV, mapiranje, lokalizacije, navigacija

Abstract – The paper describes mapping, localization and navigation on uneven terrain of an autonomously driven vehicle with variable geometry. The task of the work was to research and implement algorithms for automatic mapping, localization and navigation on uneven terrain for an autonomously driven vehicle. In doing so, it was necessary to study the existing algorithms for automatic mapping, localization and navigation based on the available literature and select the appropriate ones, adapt the selected approach to the problem of navigation on uneven terrain and implement the proposed solution on an autonomously driven vehicle. A solution was proposed in the form of a lidar filter node that has unfiltered values of lidar reading distances and rotation angles from the inertial sensor as an input. Based on this data, the filter publishes compensated values on a specific topic within ROS. Finally, the proposed solution was tested experimentally and compared with the existing solution with the help of the Vicon system.

Keywords: AGV, mapping, localization, navigation

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milutin Nikolić, vanr. prof.

1. UVOD

Proizvodnja stalno nastoji ka efikasnijim i jeftinijim načinima realizacije svojih konačnih produkta. Fleksibilni sistemi proizvodnje (FMS) su korišćeni u potrazi za isplativijim i vremenski efikasnijim procesima [1]. Postavljanje nove proizvodne linije predstavlja veliku investiciju za kompaniju, a da se manje promjene realizuju na već postojećim proizvodnim linijama može dovesti do zastoja, smanjenja produktivnosti i zahtjevaju veliko ulaganje kapitala.

Alternativa ovom tradicionalnom metodu bi bila ta da se u sistem ugradi fleksibilna linija. Ovaj pristup bi se razlikovao od tradicionalnog, linearnog u pogledu: rasporeda, njegove mogućnosti obrade i sposobnosti da proizvodi više nego jedan predmet. Još jedna prednost bi bila sposobnost promjene sistema bez uticaja na ukupnu proizvodnju.

Automatizovana vozila su bitna komponenta ove fleksibilnosti proizvodne linije. Termin eng. *Automated Guided Vehicles* se odnosi na vozila koja su u stanju da se kreću bez ljudske intervencije i sposobna su da autonomno navigiraju prostor u kome se nađu. Mogu se koristiti kao alternativa za fiksne transportne trake koje transportuju nedovršeni proizvod kroz faze proizvodnje sekvencijalno. AGV bi mogao uzimati i ostaviti (eng. *pick and place*) nedovršeni proizvod na bilo koju fazu proizvodnje što omogućava fleksibilniji proizvodni proces. Najnovija verzija AGV-a se često naziva Autonomno inteligentno vozilo (AIV), engl. *Autonomous Intelligent Vehicle*.

1.1. Postavka problema

Najčešće primjenjivani autonomi roboti u uslovima proizvodnje programirani su da prate unaprijed zadate linije putanja ili bar kodove koji se nalaze na ključnim mjestima u fabrici. Vozila su rigidne konstrukcije i nemaju mogućnost artikulacije što im dozvoljava da budu što kompaktinijeg dizajna i mogućnost što veće nosivosti tereta.

U uslovima proizvodnje gdje dolazi do kontaminacije radnog prostora oko mašina i pokretnih traka uviđena je potreba za ravojem autonomno vođenog vozila (AGV-a) koji bi imao sposobnosti da prelazi preko ovakvog terena. Sistem je razvijen primjenom koncepta artikulacije šasije i upotrebom četiri omnidirekciona točka tipa mekanum (eng. “*mecanum*”).

Navigacija u promjenljivim i kontaminiranim uslovima nije moguća upotrebom predefinisanih kretanja uz praćenje obilježja na tlu ili zidovima fabrike. Za ove

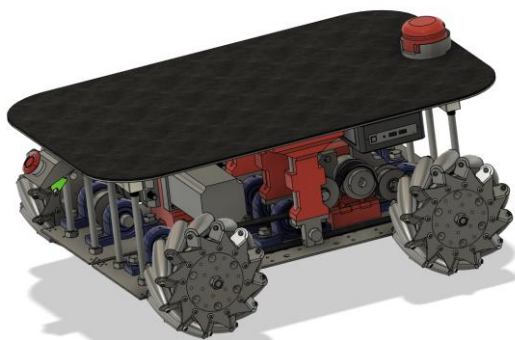
potrebe navigiranja robota unutar radnog prostora, koriste se metode navigacije upotrebom naprednijih senzora. Putanje se generišu na osnovu prvobitno kreirane mape koja se osvježava novim informacijama u toku rada vozila uz sisteme detekcije prepreka ili samih radnika. Senzor koji se prvobitno koristi na ovakvim vozilima jeste lidar. Pored lidara koriste se i inercijalni senzori (eng. “*Inertial Mesurment Unit*”) i kamere koje posjeduju mjerenja dubine pored normalne RGB kamere.

Trenutno ponuđeni paketi navigacije prostora imaju mogućnost korišćenja 2D ili 3D lidara ali pretpostavljaju da se vozilo kreće horizontalno, u ravni sa tlom. Ovakva pretpostavka ima negativan efekat u prelasku terena koji nije ravan jer se baza robota gdje, se obično montiraju senzori, naginje što pravi negativan uticaj na procjenu daljine objekata na mapi mjerenih sa sistemom lidara. U zavisnosti od ugla, tačke koje lidar dobija, koje predstavljaju razdaljinu od objekata u stvarnosti, imaju vrijednosti koje odstupaju od stvarnih rastojanja kada bi robot bio paralelan sa podlogom upravo zbog ugla gledanja.

2. MEHANIČKA KONSTRUKCIJA

Mehanika robota projektovana je tako da bi se omogućilo kretanje AGV-a po terenu koji nije idealo ravan već je ispunjen preprekama koje bi robot trebao da pređe uz što manje naginjanje površine za nošenje tereta i samog tereta. Zahtjevi nosivosti bili su 100 kg postavljenih na gornju noseću ploču koja je prekrivena gumiranim materijalom radi poboljšanja trenja između robota i predmeta koji nosi. Gabaritne dimenzije robota su 776 mm x 628 mm x 250 mm a osno rastojanje točkova je rastojanje 444 mm i po širini 526 mm.

Na slici 1 dat je prikaz mehaničke konstrukcije sklopljenog robota:



Slika 1: Mehanička konstrukcija vozila

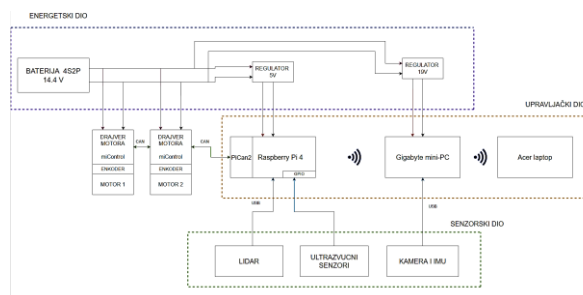
3. ELEKTRONIKA

U sledećem poglavlju dat je pregled elektronike korišćene za pokretanje i upravljanje sistema i dati su senzori koji su korišćeni za odometriju, mapiranje prostora i navigaciju kao i sigurnosni senzori. Blok šema elektronike data je na slici 2.

3.1. Energetski sistem

Napajanje robota vrši se iz četiri Turnigy 4s 14.4V Li-Po baterije, po dvije vezane u paralelu pa tako dva seta

vezana u serijsku vezu radi ostvarivanja većeg kapaciteta i potrebnog napona za rad drajvera. Baterije ovog tipa su odabrane radi njihove mogućnosti da isporuče relativno velike strujne zahtjeve koji pri punom opterećenju robota dosežu i 20A, velikog su kapaciteta od 5000 mAh u kompaktnom pakovanju i sa niskom težinom baterija.



Slika 2: Blok šema elektronike

Motori izabrani za pogon vozila su firme Groschopp i modela EGK65-30NA, to su motori bez četkica sa permanentnim magnetima. Nominalni obrtni moment je 0.35 Nm pri radu na 24V nominalnog napona. Proizvode snagu od 120 W i imaju 6000 rmp brzinu okretanja

Tabela 1: Energetski zahtjevi sistema

Motor + drajver x2	120 W x 2	240 W
RaspberryPi + lidar	15 W	15 W
Mini-PC	36 W	36 W
US senzori	1 W x 10	10 W

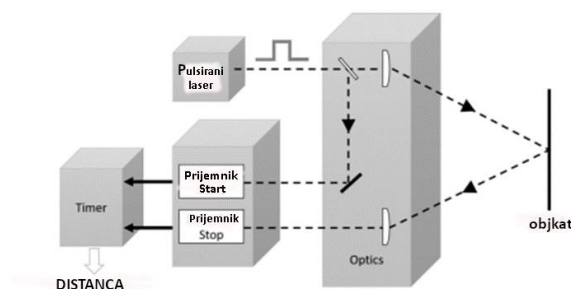
4. SENZORI

Senzora koji su korišćeni na vozilu su lidar, inercijalni senzor unutar Intel kamere i ultražvučni senzori. Senzori su korišćeni u svrhu mapiranja, lokalizacije i navigacije vozila u prostoru kao i u sigurnosne funkcije i detekciju prepreka ili ljudi koji nailaze na putanju kretanja.

4.1. LiDAR

Termin LiDAR (Light Detection and Ranging) se prvi put pojavio 1963. godine. Bila je to kombinacija reči „svjetlo” i “radar”. Princip rada je i dalje isti, refleksija svjetlosti i izračunavanje vremena leta (eng. “Time of Flight” TOF), što se može bolje vidjeti na slici 3.

Projektovanjem optičkog signala na objekat/površinu reflektovani povratno rasijani signal se prima i obrađuje da bi se odredila udaljenost. Ovo omogućava stvaranje oblaka tačaka okoline.



Slika 3: Princip rada lidara

Formula je vrlo jednostavna, udaljenost D do mete se mjeri na osnovu povratnog kašnjenja (ΔT) svjetlosnih talasa koji putuju do mete.

$$D = c \cdot \Delta T \quad (1)$$

gdje je: D - rastojanje objekta c - brzina svetlosti ΔT - vreme potrebno svetlosti da putuje

4.2. Inercijalna mjerna jedinica IMU

Inercijalna mjerna jedinica, IMU (eng. “*Inertial measurement unit*”), obično se sastoji od dva senzora sa 3 ose, akcelerometra i žiroskopa. Opciono, može biti prisutan i treći senzor - magnetometar. U prvom obliku oni su sposobni da mjere šest veličina, mjerenje ubrzanja (ms^2) i ugaone brzine (rad/s). U drugom slučaju dodavanjem magnetne orijentacije (μT esla) broj mjernih veličina se povećava na devet.

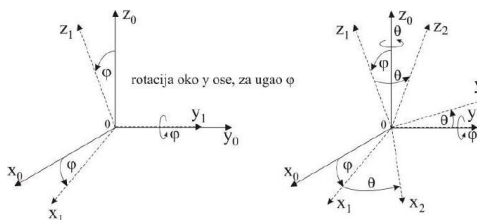
5. SOFTVERSKO REŠENJE

Za implementaciju ovog rada korišćen je *Robot Operating Sistem* (ROS), verzija *Foxy Fitzroy*. To je skup softverskih biblioteka i alata otvorenog koda posebno dizajniranih za robotske aplikacije i za jednostavnu integraciju nekoliko softverskih dijelova potrebnih za rad robota, od drajvere do najsavremenijih algoritama. Glavne softverske biblioteke i alati koji se koriste za ovu implementaciju su sledeće:

- Ubuntu 20.04.5 LTS (Focal Fossa): Operativni sistem
- ROS *Foxy Fitzroy*: Ros distribucija
- Lokalizacija robota: Ovaj paket je kolekcija čvorova za procjenu stanja i fuziju senzora.
- Imu alati: Sadrži različite alate za IMU uređaje, koristi se za pokretanje *madwick* filterskog čvora odgovornog za kreiranje orijentacije iz podataka Intel Imu.
- Rviz: Glavni alat za 2D/3D vizualizator u ROS-u
- Nav2 navigacioni paket

5.1. Lidar filter

Kada robot prelazi preko neravnog terena lidar se rotira oko x , y i z ose zaredom (φ - fi, θ - theta i ψ - psi) kao što je prikazano na kordinatnom sitemu na slici 5.



Slika 5. Transformacije koordinatnog sistema preko ZYX Ojlerovih uglova

Usljed rotacija kordinatnog sistema lidara podaci koji se koriste u navigacionom paketu za formiranje lokalne i globalne mape troškova nisu tačni jer se očitavanje distanci vrši u novom, pomjerenom koordinatnom sistemu. Ovakav ofset podataka uzrokuje netačnost pozicioniranja robota i mogućnost sudara sa fiksnim objektima na mapi.

5.2. Predloženo rešenje

Vektor distance (rastojanje tačke u prostoru do centra lidara) $\mathbf{d}_1$ u novom, pomjerenom sistemu može se izraziti preko jednačine 9:

$$\vec{d}_1 = R_1^0 \cdot \vec{d}_0 \quad (9)$$

gdje je $\mathbf{d}_1$ distanca u pomjerenom ks dok je $\mathbf{d}_0$ distanca u referentnom kordinatnom sistemu koji ortogonalan na teren bez nagiba, R_1^0 je matrica rotacija izmjedju ks.

Da bismo kompenzovali distancu lirada potrebno je da predstavimo $\mathbf{d}_0$ u odnosu na $\mathbf{d}_1$:

$$\vec{d}_0 = R_0^1 \cdot \vec{d}_1 \quad (10)$$

Gdje su R_0^1 i $\vec{d}_1$:

$$R_1^0 = \begin{bmatrix} \cos\theta & \sin\theta\sin\varphi & \sin\theta\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi \\ -\sin\theta & \cos\varphi & \cos\theta\cos\varphi \end{bmatrix} \quad (11)$$

$$\vec{d}_1 = \begin{bmatrix} \cos\alpha \\ \sin\alpha \\ 0 \end{bmatrix} d_1 \quad (12)$$

α – ugaono rastojanje trenutne tačke očitavanja lidara

Množenjem jednačina 11 i 12 dobijamo:

$$\vec{d}_0 = \begin{bmatrix} \cos\theta\cos\alpha + \sin\theta\sin\varphi\sin\alpha \\ \cos\varphi\sin\alpha \\ -\sin\theta\cos\alpha + \cos\varphi\sin\varphi\sin\alpha \end{bmatrix} d_1 \quad (18)$$

Intenzitet d_0 u preslikan u xy ravni je:

Mjerenja dobijena sa inercijalnog senzora koji je nagnut kada vozilo prelazi preko neravnog terena data su preko uglova φ i θ . Navigacioni paket koji je korišćen pretpostavlja da se robot nalazi u horizontalnoj ravni smatrajući da su prepreke vertikalne. Da bi smo ispravili distance mjerenja koje su mjerenje u nagem stanju naći ćemo intenzite vektora $\mathbf{d}_0$:

$$|d_0| = \frac{\sqrt{d_1^2(\cos\theta\cos\alpha + \sin\theta\sin\varphi\sin\alpha)^2 + d_1^2(\cos\varphi\sin\alpha)^2}}{\quad} \quad (19)$$

Predloženo rješenje je implementirano u ROS2 čvor čiji su ulazi dvije teme i to: `/scan/unfiltered` i `/imu/rpy`. Na prvu temu publiše `rplidar` čvor i tip poruke je `LaserScan` koji nam daje niz dužina tačaka u prostoru raspoređenih na određenim uglovima u krugu. Druga tema dolazi od filtera za intercijalni sistem i u obliku poruke `Vector3Stamped` daje nam informacije o `roll`, `pitch` i `yaw` uglovima. Kao izlaz iz čvora data je tema `/scan` koja je predstavlja filtriran oblik ulaznih podataka.

6. EKSPERIMENT

Testiranje će se vršiti uz pomoć *Vicon* sistema kamera. Pokazano je kako se robot navigira u prostoru sa preprekama sa i bez korišćenja lidar kompenzacionog filtera predloženog kao rešenje u prethodom poglavlju.

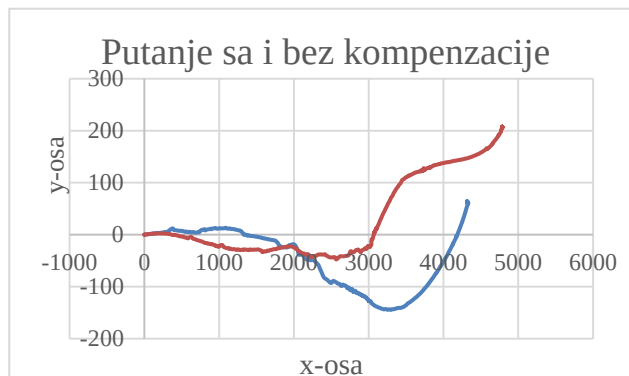


Slika 6. Kordinatni sistemi Vicon sistema, robota i mape

6.1. Rezultati eksperimenta

Nakon snimanja dobijamo rezultate dva načina navigacije, bez predloženog lidar filtra koji kompenzuje kretanje robota vozila po neravnom terenu i sa lidar filtrom. Koordinatni sistem kamera i robota dovedeni su u isti kordinatni početak naknadnom obradom podataka. Kao krajnja koordinata navigacije tada je tačka (4400, 50).

Pri putanji kretanja vozila u xy ravni bez kompenzacije u odnosu na referentni Vicon sistem uočavamo grešku po x osi od 336 mm i grešku po y osi od 207 mm i vizuelno možemo uvidjeti grešku u orijentaciji od 45 stepeni. Dok pri putanji kretanja vozila u xy ravni sa lidar kompenzacijom uočavamo grešku po x osi od 75 mm i grešku po y osi od 11 mm i vizuelno možemo uvidjeti grešku u orijentaciji od 15 stepeni.



Slika 7. Putanja AGV sa i bez lidar kompenzacije

7. ZAKLJUČAK

Prema zahtjevima nosivosti i brzine i načina kretanja vozila konstruisano je autonomno vođeno vozilo sa četiri omnidirekciona točka ali po dva su pokretana jednim motorom. Prvi dio zadatka riješen je upotrebom postojećih paketa za navigaciju, slam i lokalizaciju. Robotski sistem prilagođen je radu unutar ROS-a, prilagođena je postojeća robotska platforma za komunikaciju unutar ovog sistema i to uz korišćenje standarizovanih tipova poruka.

Kao rješenje problema navigacije po neravnom terenu implementiran je čvor filter unutar ROS-a koji prikuplja neobrađene podatke o distancama sa lidarskog čvora i podatke o nagibu robota po određenim osama sa intercijalnog senzora unutar kamere.

Eksperimentalnim putem potvrđena je efikasnost predloženog lidar kompenzacionog filtra na osnovu mjerenja referentnog Vicon sistema. Prvo mjerenje je urađeno bez upotrebe filtra i snimljeni su rezultati, a drugo mjerenje urađeno je sa upotrebom filtra. Nakon poređenja podataka došli smo do pozitivnih rezultata rada filtra. Potvrđeno je da se vozilo našlo na poziciji bližoj zadatoj u slučaju sa korišćenjem lidar filtra, a orijentacija je vizuelno izgledala bliže zadatoj.

Kao dodatak navigaciji robota u prostoru, planirano je implementirati lidarsku odometriju (eng LOAM – Lidar odometry and mapping) koja bi služila kao dodatak trenutnoj odometriji i poboljšala bi preciznost pozicioniranja vozila. Trenutno je na robotu montirana Intel Realsence kamera koja ima mogućnost da obezbijedi oblak tačaka distanci predmeta koje kamera može da vidi, tako da sledeći dodatak bila bi implementacija vizuelne odometrije. Nakon ovih poboljšanja odometrije planirano je da se vozilo pripremi za rad unutar industrijskog okruženja u fabrici,

8. LITERATURA

[1] L. Lynch, T. Newe, J. Clifford, J. Coleman, J. Walsh, D. Toal: “Automated Ground Vehicle (AGV) and Sensor Technologies- A Review”, 2018 Twelfth International Conference on Sensing Technology.

[2] Applied sciences: “An overview of lidar imaging systems for autonomous vehicles”, link: <https://www.mdpi.com/2076-3417/9/19/4093>, Accessed: 15.09.2022.

[3] B. Ivanović, T. Milić: “Kalmanov filter”, link: <https://blaza.github.io/kalmanfilter/>, Accessed: 20.09.2022.

[4] Wikipedia: “Extended Kalman filter, link: https://en.wikipedia.org/wiki/Extended_Kalman_filter, Accessed: 17.09.2022.

Kratka biografija:



Aleksandar Kičić rođen je u Tesliću, RS, BiH 1998. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti robotike odbranio je 2021.god.

kontakt: acokicic@gmail.com

PROJEKTOVANJE KOMPAKTNE TOPLLOTNE PODSTANICE**DEVELOPMENT OF COMPACT HEATING SUBSTATION**Nikola Bajandić, *Fakultet tehničkih nauka, Novi Sad***Oblast – MEHATRONIKA**

Kratak sadržaj – U radu je objašnjen način projektovanja kompaktne toplotne podstanice. Prema zahtevu, urađen je proračun i odabir potrebne opreme i prikazan tehnički crtež toplotne podstanice. Urađene su hidraulična i električna šema sistema. Prikazan je način parametrizacije ulaznih i izlaznih signala. Urađena je sastavnica i kalkulacija sistema.

Ključne reči: Automatizacija, termodinamika, toplotna podstanica, programabilno logički kontroler

Abstract – This paper explains how to design a compact heating substation. Based on the requirements, the calculation and selection of the necessary equipment was done and the technical drawing of the heating substation was presented. The hydraulic and electrical schematics of the system were made. The method of parameterization of I/O signals is presented. The bill of materials and calculation of the system was done.

Keywords: Automation, thermodynamics, heating substation, programmable logic controller

1. UVOD

Sistem daljinskog grejanja je sistem za distribuciju toplote generisane centralizovanoj lokaciji kroz sistem izolovanih cevi za potrebe grejanja stambenih i nestambenih objekata i zagrevanje vode. On se sastoji od toplotnog izvora, toplifikacione mreže i toplotne podstanice. Prateći tok razvoja daljinskog grejanja može se podeliti u četiri generacije. U prvoj generaciji nosilac toplote je para. Ova tehnologija je korišćena do 1930. godine u SAD i Evropi. Para visokog pritiska je transportovana cevima u betonskim kanalima sa visokim operativnim temperaturama i malom efikasnošću prenosa toplote energije. Osnovni motiv za projektovanje sistema daljinskog grejanja prve generacije je zamena individualnih kotlova u stambenim objektima (smanjenje rizika od eksplozija).

Mane prve generacije su značajni toplotni gubici, zbog visokih temperatura, teške nesreće usled eksplozija i niža energetska efikasnost. U drugoj generaciji nosilac toplote je para. Ova tehnologija je korišćena od 1930. do 1970-ih godina. Korišćene su cevi u betonskim kanalima, veliki cevni izmenjivači i veliki ventili. Osnovni motiv za projektovanje sistema daljinskog grejanja druge generacije je ušteda goriva, koja nastaje korišćenjem kombinovane proizvodnje

toplote i električne energije. Najveća mana ovih sistema je nemogućnost kontrole potražnje toplotne energije. U trećoj generaciji nosilac toplote je voda pod pritiskom. Ova tehnologija je korišćena od 1970-ih godina. Korišćene su predizolovane cevi zakopane u zemlju, a toplotne podstanice u ovim sistemima su kompaktne sa pločastim izmenjivačima. Osnovni motiv za projektovanje sistema daljinskog grejanja treće generacije je sigurnost snabdevanja. Umesto nafte koriste se ugalj, biomasa, solarni i geotermalni sistemi. U četvrtoj generaciji nosilac toplote je voda pod pritiskom. Komponente koje se koriste su orijentisane na montažu, a cevi su pravilnije od fleksibilnijih materijala. Temperature snabdevanja su niže (80°C), pa su i gubici manji. Postoji mogućnost upotrebe niskotemperaturnih i obnovljivih izvora toplote. Koristi se kao deo pametnih energetske sistema. Osnovni motiv je podloga za dalji razvoj grejne infrastrukture i tehnologije, odnosno transformacija u održivi energetski sistem koji će se boriti protiv klimatski promena.

Izbor projektnih temperatura u sistemima daljinskog grejanja je složen i veoma važan korak u dizajniranju mreže daljinskog grejanja. Temperature u mreži utiču na količinu isporučene toplote energije, gubitke toplote u mreži, rad pumpi za transport vode, proizvodnju električne energije u kogenerativnim postrojenjima, na toplotu proizvedenu u toplotnim pumpama.

Snaga (P), koju korisnik primi u svojoj podstanici, zavisi od razlike temperatura između polaznog i povratnog voda (ΔT), masenog protoka ($\dot{m}$) i specifičnog toplotnog kapaciteta medijuma (C_p):

$$P = \dot{m} * C_p * \Delta T \quad (1)$$

Povećanje temperaturne razlike, za određenu isporučenu snagu, ima za posledicu smanjenje masenog protoka, samim tim i smanjenje utrošene električne energije za rad pumpi i troškova vezanih za njih, čemu se teži. Za postojeću mrežu daljinskog grejanja, samo promene temperatura i protoka mogu uticati na smanjenje distributivnih gubitaka toplote. Energetski balans mreže se izražava sledećom formulom:

$$\dot{Q}_{plant} + \dot{W}_{pump} = \dot{Q}_{loss} + \dot{Q}_{cons} \quad (2)$$

$\dot{Q}_{plant}$ - toplota proizvedena u toplani,

$\dot{W}_{pump}$ - snaga potrebna za rad pumpnih stanica

$\dot{Q}_{loss}$ - toplotni gubici duž mreže,

$\dot{Q}_{cons}$ - toplota kojom se snabdevaju podstanice.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Stevan Stankovski, red. prof.

Efikasna mreža daljinskog grejanja podrazumeva niske temperature snabdevanja, što povećava efikasnost proizvodnje i smanjuje gubitke toplote pri transportu, i visoku

temperaturnu razliku između polaznog i povratnog voda, što dovodi do smanjenja masenog protoka, a samim tim i do uštede energije za rad pumpe [1].

2. TOPLOTNA PODSTANICA

Toplotna podstanica je treći osnovni element sistema daljinskog grejanja. Namenjena je za regulisanu predaju toplotne energije iz primarne mreže toplifikacionog sistema sekundarnoj mreži kućne instalacije grejanja. Dva osnovna tipa toplotnih podstanica su sa direktnim i sa indirektnim priključkom. Kod podstanica sa direktnim priključkom, sistem daljinskog grejanja i kućna instalacija predstavljaju jedinstven hidraulični krug. Temperatura povratne vode primara je niža ili je jednaka temperaturi povratne vode u sekundaru. Ovakav sistem nije uvek primenljiv, zbog pritiska koji je potrebno održavati u mreži daljinskog grejanja, a problem predstavljaju i nečistoće iz kućne instalacije koje se prenose u mrežu daljinskog grejanja. Ovo rešenje je jednostavnije i jeftinije. Kod podstanica sa indirektnim priključkom, sistem daljinskog grejanja i kućna instalacija predstavljaju dva zasebna hidraulična kruga, koje razdvaja izmenjivač toplote. Pritisak u kućnoj instalaciji je nezavisan od pritiska u mreži daljinskog grejanja, a nečistoće iz kućne instalacije se ne prenose u mrežu daljinskog grejanja [2]. Hidraulične krugove možemo podeliti na krugove pod pritiskom i bezpritiskne krugove. Hidraulični krugovi pod pritiskom se dele na prigušni, razdelni krug, krug sa prolaznim i krug sa trokrakim ventilom. Bezpritisni hidraulični krugove se dele na mešne i dvostruke mešne krugove.

Karakteristika prigušnog hidrauličnog kruga jeste promenljiva količina vode u primarnom i sekundarnom krugu. Temperatura vode u primarnom i sekundarnom krugu je konstantna, a regulacija toplotnog opterećenja vrši se promenom protoka pomoću prolaznog ventila i zadatih parametara. Prednosti ovakvog sistema su dobra raznolikost i stoga je pogodan za kondenzacione kotlove i distributivne sisteme toplotne energije. Nedostaci se javljaju prilikom regulisanja protoka vode u cevovodu, jer se radna tačka pumpe menja pomeranjem ventila. Razlika pritiska koja se javlja ima uticaj na grejna tela.

Karakteristika razdelnog hidrauličnog kruga je konstantan protok vode u primarnom krugu, dok je u sekundaru promenljiv. Temperatura u primarnom i sekundarnom krugu je konstantna, a podešavanje toplotnog opterećenja u sekundaru vrši se promenom protoka. Prednost ovog hidrauličnog kruga je konstantna količina grejnog medijuma u primarnom krugu, što znači da nisu potrebne pumpe sa regulisanim protokom.

Nedostatak predstavlja temperatura u sekundaru koja je uvek jednaka maksimalnoj temperaturi vode u primarnom krugu, i nije moguće postići različite temperaturne režime u sekundarnom krugu.

Karakteristika kruga sa prolaznim ventilom je promenljiv protok vode u primarnom krugu, dok je u sekundarnom konstantan. Promenljiva temperatura na sekundaru se podešava prema zahtevu potrošača. Prednost ove izvedbe je mogućnost primene za sisteme sa nižom temperaturom u povratnom vodu, pogotovo kada je razlika temperatura između primarnog i sekundarnog kruga velika. Nedostatak je potreba za diferencijalnim pritiskom da bi se

mogao dimenzionisati kontrolni ventil. Kod većih dužina cevovoda postoji opasnost od smržavanja u sekundaru.

Karakteristika kruga sa trokrakim ventilom jeste konstantan protok vode u primarnom i sekundarnom krugu, dok je temperatura u sekundarnom krugu promenljiva. Prednost ovog hidrauličnog kruga jeste to što je voda koja prolazi kroz kontrolni ventil raspoloživa u svakom trenutku i koristi se prilikom povezivanja grejnih elemenata kod kojih je potrebna velika količina toplote u kratkom vremenskom periodu. Nedostatak je permanentni porast temperature u povratnom vodu, pa nije primenljiv za sisteme daljinskog grejanja i sisteme sa kondenzacionim kotlom [4].

3. IZBOR KOMPONENTI TOPLOTNE PODSTANICE

U tabeli 1. prikazani su uslovi rada toplotne podstanice.

Tabela 1. Uslovi rada toplotne podstanice

Potrebna snaga podstanice	$Q = 28000 \text{ W}$
Temperatura u potisnom / povratnom vodu (primar)	$T_{prim1} = 80 \text{ }^{\circ}\text{C}$ $T_{prim2} = 55 \text{ }^{\circ}\text{C}$
Temperatura u potisnom / povratnom vodu (sekundar)	$T_{sek1} = 60 \text{ }^{\circ}\text{C}$ $T_{sek2} = 45 \text{ }^{\circ}\text{C}$
Nominalni pritisak primar / sekundar	25 / 6 bar
Radni fluid primar / sekundar	voda / voda
Pad pritiska na potrošaču	$\Delta H = 25 \text{ kPa}$

Za određivanje dimenzije cevovoda potrebno je izračunati protok fluida G na primarnoj i sekundarnoj strani podstanice:

$$G = \frac{Q}{c_p \cdot \rho \cdot \Delta T} \quad (3)$$

$$G_{prim} = 0,98 \text{ m}^3/\text{h}, \quad G_{sek} = 1,63 \text{ m}^3/\text{h}$$

gde su:

$c_p = 4,18 \text{ kJ/kgK}$ - specifična toplota vode

$\rho = 1000 \text{ kg/m}^3$ - gustina vode

$\Delta T_{prim} = 25 \text{ }^{\circ}\text{C}$ - razlika temperatura na primaru

$\Delta T_{sek} = 15 \text{ }^{\circ}\text{C}$ - razlika temperatura na sekundaru

Usvojena dimenzija cevovoda primarne i sekundarne strane toplotne podstanice je DN25.

Izmenjivač toplote je uređaj u kom se razmenjuje toplota između dva fluida pri čemu se jedan fluid hladi, a drugi greje. Kada se razmena toplote obavlja u svrhu grejanja, fluid na višoj temperaturi se naziva grejni fluid, a onaj na nižoj temperaturi se naziva grejani fluid. U obrnutom slučaju, kada je cilj razmene toplote hlađenje, onaj fluid kojim se hladi naziva se rashladni fluid, a onaj koji se hladi – hlađeni fluid. Mogu se podeliti na rekuperativne i regenerativne. Kod rekuperativnih fluidi su prilikom razmene razdvojeni grejnom (rashladnom) površinom, a kod regenerativnih preko grejne površine prvo struji topliji fluid, ona akumulira izvesnu količinu toplote koju odaje onda kada preko nje struji hladniji fluid. Rekuperativni izmenjivači se prema smeru kretanja fluida mogu podeliti na izmenjivače sa unakrsnim tokom, sa istosmernim tokom i sa suprotnosmernim tokom. Prema konstrukciji izmenjivače toplote možemo podeliti na izmenjivač „cev u cev“, bubnjasti, spiralni, pločasti i izmenjivač sa snopom cevi. Za toplotnu podstanicu biram

pločasti izmenjivač toplote Kelvion GVH 240H-20, lemljen nerđajućim čelikom sa 20 ploča.

Kontrolni ventil se upotrebljava u zatvorenim sistemima grejanja i hlađenja za regulaciju protoka (dvokraki) i za regulaciju temperature mešne vode (trokraki). Teorijska protočna karakteristika kontrolnog ventila $k_{v,teo}$ se izračunava:

$$k_{v,teo} = \frac{G}{100 \cdot \sqrt{\Delta H}} \quad (4)$$

$$k_{v,teo,prim} = 1,96 \text{ m}^3/\text{h}, \quad k_{v,teo,sek} = 3,26 \text{ m}^3/\text{h}$$

Pad pritiska na kontrolnom ventilu Δp_v se računa:

$$\Delta p_v = \left(\frac{G}{100 \cdot k_{vs}} \right)^2 \quad (5)$$

$$\Delta p_{v,prim} = 15,37 \text{ kPa}, \quad \Delta p_{v,sek} = 16,61 \text{ kPa}$$

Pri odabiru ventila potrebno je obratiti pažnju na autoritet ventila a , gde on treba da iznosi 0,35 - 0,75, a ne sme pasti ispod 0,25, jer bi u tom slučaju sistem postao nestabilan. Autoritet kontrolnih ventila a se izračunava:

$$a = \frac{\Delta p_v}{\Delta H} \quad (6)$$

$$a_{prim} = 0,61, \quad a_{sek} = 0,66$$

Na osnovu $k_{v,teo}$ i zahtevanih uslova usvajamo kontrolni ventil na primaru Samson 3222 $k_{vs} = 2,5 \text{ m}^3/\text{h}$ dimenzije DN25 i aktuator Samson 5825-10 24 VDC, a na sekundaru Siemens VXP45 $k_{vs} = 4 \text{ m}^3/\text{h}$ dimenzije DN20 i aktuator Siemens SSB61 24 VDC.

Balansni ventil služi za postizanje hidrauličkog balansa između različitih krugova sistema. Balans se postiže podešavanjem i memorisanim zaključavanjem. Za dimenzionisanje balansnog ventila potrebno je odrediti diferencijalni pritisak koji treba da se apsorbuje Δp_{bv} :

$$\Delta p_{bv} = \Delta H - \Delta p_v \quad (7)$$

$$\Delta p_{bv} = 8,39 \text{ kPa}$$

Potrebna protočna karakteristika balansnog ventila k_{bv} se izračunava:

$$k_{bv} = \frac{G_{sek}}{100 \cdot \sqrt{\Delta p_{bv}}} \quad (8)$$

$$k_{bv} = 5,63 \text{ m}^3/\text{h}$$

Na osnovu k_{bv} i zahtevanih uslova usvajamo balansni ventil Oventrop VTR $k_{vs} = 8,9 \text{ m}^3/\text{h}$ dimenzije DN25. Iz tehničkog lista proizvođača zaključujemo da odabrano podešavanje ventila iznosi $POZ_{pod} = 4,3$, dok maksimalno podešavanje iznosi $POZ_{max} = 7$. Otvorenost ventila n treba da bude u granicama od 25% do 75%, zbog stabilnosti regulacije.

$$n = \frac{POZ_{pod}}{POZ_{max}} \quad (9)$$

$$n = 61\%$$

Cirkulaciona pumpa u sistemima grejanja pokreće fluid i na taj način se vrši razmena toplote. Mogu se podeliti u dve grupe sa suvim i mokrim rotorom. Svaka od ove grupe se može podeliti na navojne i prirubničke. Cirkulacione pumpe sa suvim rotorom imaju visok nivo efikasnosti (80-85%). Sve radne jedinice su odvojene od fluida zaptivnim prstenovima i hlade se pomoću zasebnog ventilatora. Ovaj tip pumpi zahteva visok kvalitet vode i redovnu zamenu maziva na rotirajućim delovima. Cirkulacione pumpe sa suvim rotorom imaju nižu efikasnost, ali su i manji potrošači energije. Radne lopatice i rotor koje su u stalnom kontaktu sa fluidom ne zahtevaju zasebno hlađenje, a fluid ima funkciju maziva. Rad pumpe je skoro nečujan, jer fluid apsorbuje zvuk. Na osnovu potrebnog protoka na sekundaru $G_{sek} = 1,63$

m^3/h i zahtevanog napora $\Delta H = 25 \text{ kPa}$, usvajamo pumpu proizvođača Wilo Yonos Pico 25/1-6.

Hvatač nečistoća je komponenta koja štiti opremu od nečistoća u instalaciji. Montira se pre ulaska fluida u izmenjivač toplote, na primarnog krugu na potisnom, a na sekundarnom krugu na povratnom vodu. Usvajamo hvatač nečistoća Danfoss FVF DN25 PN25 na primaru i Tiemme DN25 PN20 na sekundarnoj strani podstanice.

Kuglasta slavina je zaporni ventil čija je funkcija da u zatvorenom/otvorenom položaju prekine/omogući protok fluida kroz cevi. Usvajamo slavinu Itap DN25 PN20 na sekundarnoj strani toplotne podstanice.

Sigurnosni ventil štiti instalaciju od prekomernog pritiska. Preko ovog ventila je moguće odzračivanje toplotne podstanice. Montiran je na potisnom vodu sekundarnog dela podstanice. Usvajamo ventil Caleffi 311 R1/2" 3bar.

Nepovratni ventil dozvoljava protok fluida samo u jednom smeru uz minimalno mogući pad pritiska. Montiran je na povratnom vodu sekundarnog dela podstanice. Usvajamo nepovratni ventil Itap Europa R1" PN18.

Ekspanzion posuda služi za kompenzaciju dilatacije fluida usled promene temperature sekundarnog dela toplotne podstanice. Zapremina ekspanzione posude treba da bude minimalno 4% ukupne količine vode u sistemu. Usvajamo posudu Elbi ER-CE 12L PN10.

Termostati na osnovu zadate temperature daju informaciju koliko je potrebno otvoriti kontrolni ventil kako bi regulacija bila što kvalitetnija. Usvajamo termostat Siemens RAK-TW.1000HB.

Senzori temperature se mogu podeliti na termoelemente, ekspanzione, otporničke, poluprovodničke itd. Biramo otpornočki tip senzora koji rade na principu promene otpornosti elementa usled promene temperature. Usvajamo dva modela seznora temperature Siemens QAD21/201 i Siemens QAE2121.015.

Programabilno logički kontroler (PLK) je digitalni elektronički uređaj koji poseduje programabilnu memoriju za smeštanje instrukcija, kojima se realizuju različite funkcije u cilju upravljanja mašinama i procesima putem digitalnih i/ili analognih ulazno/izlaznih jedinica.

Osnovne celine PLK su centralna procesorska jedinica (CPU), memorija, komunikacioni deo, modul napajanja, ulazni i izlazni modul i modul za proširenje. CPU je mozak PLK-a, koja odlučuje koje se naredbe izvršavaju. Realizuje se pomoću mikrokontrolera za manji broj ulaza i izlaza, a za veći broj pomoću mikroprocesora. Memorija se deli na sistemsku na kojoj se nalazi operativni sistem i korisničku čiji se jedan deo koristi za čuvanje stanja ulaza i izlaza, a drugi za čuvanje stanja promenljivih.

Komunikacioni deo obezbeđuje komunikaciju sa računalom na kom se piše upravljački program i kasnije šalje u PLK. Modul napajanja osigurava potreban napon za komponente sistema i upozorava CPU ako napon nije odgovarajući. Ulazni modul prihvata ulazne signale i konvertuje logičke novoe napona spoljnog sveta u nivoe koje zahteva logička jedinica.

Na izlazni modul CPU dovodi signal na led i uključuje ga. Svetlost pobuđuje foto tranzistor, koji aktivira izlazni uređaj, obično relej koji vrši prekidanje naponskih i strujnih signala. Modul za proširenje se koristi u slučaju potrebe za povećanjem broja ulaza i izlaza. Potreban nam je PLK sa minimum 7 ulaznih i 2 izlazna priključka, pa

usvajamo kontroler Siemens POL648.00/STD sa ekranom Siemens POL895.51/STD. Moguće je na daljinu upravljati toplotnom podstanicom pomoću softvera Siemens Climatix IC.

Zavarivanje je proces spajanja dva ili više metalna dela istog ili približno istog hemijskog sastava, čime se dobija nerazdvojiva veza. Koristimo TIG (Tungsten Inert Gas) postupak zavarivanja koji je vrlo precizan, skup i relativno spor, ali daje visokokvalitetan spoj. Elektroda od volframa u inertnom zaštitnom gasu se ne topi, već usmeruje intenzivan električni luk na metal. Moguće je spojiti osnovni metal bez ili prema potrebi uz upotrebu dodatnog materijala. Koristi jednosmernu struju, sa minus polom na elektrodi, za zavarivanje nerđajućih čelika, titanijuma, bakra, čeličnih limova, dok za aluminijum, magnezijum i njihove legure koristi naizmeničnu. Zaštitni gasovi koji se koriste su argon i helijum [3].

4. DETALJAN PROJEKAT

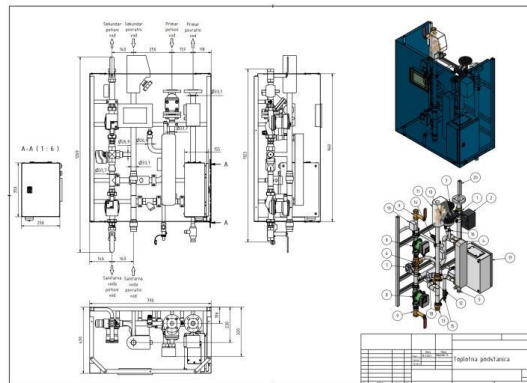
Na slici 1 prikazan je 3D model kompaktne toplotne podstanice realizovan u softveru Autodesk Inventor. Voda iz toplane prvo prolazi kroz hvatač nečistoća na primaru, koji čuva kapacitet izmenjivača toplote i produžava mu životni vek. Nakon hvatača prolazi kroz izmenjivač i stiže do primarnog kontrolnog ventila. Ventil reguliše protok na osnovu senzora temperature koji se nalazi na povratnom vodu primarnog dela podstanice. Svaka toplana ima definisanu minimalnu temperaturu koju potrošač mora da vrati u sistem. Sa druge strane izmenjivača toplote nalazi se sekundarni krug. Nakon što voda napusti izmenjivač dolazi do trokrakog ventila sekundara. Njegova funkcija je da reguliše protok u krugu grejanja uz pomoć senzora temperature, koji se nalazi na potisnom vodu kruga grejanja, kako bi potrošač u svakom trenutku imao adekvatnu temperaturu u prostoru koji se greje.



Slika 1. Izgled kompaktne toplotne podstanice

Zatim fluid stiže do cirkulacione pumpe koja pokreće fluid u sistemu. Balansni ventil, koji se nalazi na povratnom vodu, podešen je tako da sistem sistem radi sinhronizovano, da bi se izbegla mogućnost eventualnog hidrauličnog udara. Nepovratni ventil koji ne dozvoljava mešanje vode iz sistema kruga grejanja i sistema za sanitarnu vodu. Druga cirkulaciona pumpa služi za sistem grejanja sanitarne vode. Ona se uključuje i isključuje po potrebi kako bi dogrevala vodu u bojleru na 60 °C, zbog eliminisanja bakterija. Njen način rada definiše senzor temperature koje se nalazi u bojleru. Sigurnosni ventil,

koji se nalazi na potisnom vodu, ne dozvoljava da pritisak u sistemu bude veći od 3 bara. Kompletним sistemom upravlja PLK, koji se nalazi u upravljačkom ormanu. Na slici 2 prikazan je tehnički crtež toplotne postanice realizovan u softveru Autodesk Inventor.



Slika 2. Tehnički crtež kompaktne toplotne podstanice

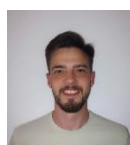
5. ZAKLJUČAK

U ovom radu izvršen je razvoj kompaktne toplotne podstanice po zahtevu kupca. Izvršen je proračun i izbor komponenti potrebnih za sistem, kao i upravljačke jedinice za realizaciju toplotne podstanice. Realizovane su hidraulična i električna šema i prikazan predmet i predračun sistema. U dodatku rada je prikazano na koji način je urađena parametrizacija ulaznih i izlaznih signala na odabranom PLK Siemens POL648.00/STD. Zaključuje se da se primenom toplotne podstanice zamenjuje rad radnika u kotlarnici i obezbeđuje sigurnost, smanjuje mogućnost pojave greške u sistemu, obezbeđuje lako upravljanje sistemom na daljinu, obezbeđuje se ekonomičnost i maksimalna iskorišćenost sistema, oslobađa prostor u kotlarnici pomoću zidne montaže uz mogućnost proširenja sistema ukoliko se pojavi potreba u budućnosti. Zadatak ovog rada rezultirao je uspešnom primenom stečenog znanja iz oblasti termodinamike, hidraulike, elektronike, automatizacije, 3D modeliranja i primene programabilno logičkih kontrolera.

6. LITERATURA

- [1] S. Ćirić, „Klimatizacija grejanje hladjenje“, Mašinski fakultet Beograd, Beograd, 1982
- [2] Đ. Kozic, B. Vasiljević, V. Bekavac, „Priručnik za termodinamiku i prostiranje toplote“, Gradjevinska knjiga, Beograd, 1983
- [3] O. Popović, R. Prokić, Cvetković, „Postupci zavarivanja“, Mašinski fakultet Beograd, Beograd, 2016
- [4] http://herzmediaserver.com/data/www/srb/literatura/hidraulika/Hidraulika_u_KGH_sistemima.pdf (pristupljeno 26.12.2021.)

Kratka biografija:



Nikola Bajandić rođen je u Novom Sadu 1994. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mehatronika – Robotika i automatizacija odbranio je 2022.god.
kontakt: nbajandic@gmail.com

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2022. godine učestvovali su sledeći recenzenti:

Aco Antić	Dragana	Marinko Maslarić	Nemanja Sremčev
Aleksandar	Konstantinović	Marko Lazić	Nemanja Tasić
Anđelković	Dragana Šarac	Marko Marković	Nenad Grahovac
Aleksandar Kovačević	Dragoljub Šević	Marko Todorov	Nenad Simeunović
Aleksandar	Drago Žarković	Marko Vekić	Nikola Vojnović
Kupusinac	Duško Bekut	Maša Bukurov	Petar Mirković
Aleksandar Ristić	Đorđe Vukelić	Matija Stipić	Platon Sovilj
Aleksandar Selakov	Goran Jeftenić	Mijodrag Milošević	Radivoje Dinulović
Aleksandra Radulović	Goran Savić	Milan Delić	Radomir Kojić
Aleksandra Pešterac	Goran Sladić	Milan Gavrić	Ratko Obradović
Andraš Anderla	Goran Švenda	Milan Marinković	Sandra Dedijer
Andrija Rašeta	Goran Tepić	Milan Mirković	Saša Medić
Atila Zelić	Gordan Stojić	Milan Rapajić	Slavica Mitrović
Bojan Batinić	Gordana Ostojić	Milan Rackov	Senka Bajić
Bojan Matić	Igor Dejanović	Milan Trivunić	Slobodan Morača
Bojan Tepavčević	Igor Peško	Milan Vidaković	Slobodan Šupić
Borislav Savković	Iva Šiđanin	Milena Krklješ	Srđan Popov
Branislav Atlagić	Ivana Mihajlović	Milica Kostreš	Srđan Vukmirović
Branislav Stevanov	Igor Maraš	Milica Miličić	Stevan Gostojić
Branka Nakomčić	Ivan Prokić	Miloš Simić	Stevan Grabić
Branko Milosavljević	Ivana Katić	Milovan Lazarević	Stevan Milisavljević
Branko Škorić	Ivana Maraš	Milja Simeunović	Stevan Stankovski
Damir Đaković	Ivana Miškeljin	Miodrag Milutinov	Strahil Gušavac
Danijela Ćirić	Jasmina Dražić	Miodrag Žigić	Svetlana Bačkalić
Danijela Gračanin	Jelena Atanacković	Mirjana Malešev	Svetlana Nikoličić
Danijela Lalić	Jeličić	Miroslav Zarić	Tamara Čeranić
Darko Čapko	Jelena Borocki	Mirko Borisov	Veran Vasić
Darko Reba	Jelena Demko Rihter	Mirko Raković	Vesna Stojaković
Darko Stefanović	Jelena Ivetić	Miro Govedarica	Višnja Žugić
Dejan Ecet	Jelena Radonić	Miroslav Kljajić	Vladimir Ilić
Dejan Lukić	Jelena Slivka	Miroslav Popović	Vladimir Katić
Dejan Reljić	Jelena Spajić	Miroslav Zarić	Vladimir Mučenski
Dejan Jerkan	Kalman Babković	Mitar Jcanović	Vlastimir Radonjanin
Dejan Movrin	Lazar Kovačević	Mladen Tomić	Vuk Bogdanović
Dejan Ubavin	Lidija Krstanović	Mladen Radišić	Vuk Vranjkovic
Dejana Nedučin	Ljiljana Popović	Nataša Samardžić	Zdravko Tešić
Dragan Ivanović	Ljubica Duđak	Nebojša Brkljač	Zoran Čepić
Dragan Ivetić	Magdolna Pal	Nebojša Radović	Zoran Jeličić
Dragan Jovanović	Maja Turk Sekulić	Nebojša Ralević	Zoran Papić
Dragan Pejić	Maja Petrović	Neda Milić Keresteš	Željen Trpovski
Dragan Ružić	Marija Silađi	Nemanja Kašiković	

