



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



ЗБОРНИК РАДОВА ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА

Едиција: Техничке науке - зборници

Година: XXXIV

Број: 3/2019

Нови Сад

Едиција: „Техничке науке – Зборници“

Година: XXXIV

Свеска: 3

Издавач: Факултет техничких наука Нови Сад

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

Уредништво:

Проф. др Раде Дорословачки

Проф. др Драгиша Вилотић

Проф. др Срђан Колаковић

Проф. др Владимир Катић

Проф. др Драган Шешилија

Проф. др Миодраг Хаџистевић

Проф. др Растислав Шостаков

Доц. др Мирослав Кљајић

Доц. др Бојан Лалић

Доц. др Дејан Убавин

Проф. др Никола Јорговановић

Доц. др Борис Думнић

Проф. др Дарко Реба

Проф. др Ђорђе Лађиновић

Проф. др Драган Јовановић

Проф. др Мила Стојаковић

Проф. др Драган Спасић

Проф. др Драгољуб Новаковић

Редакција:

Проф. др Владимир Катић, главни
уредник

Проф. др Жељен Трповски, технички
уредник

Проф. др Драган Шешилија

Проф. др Драгољуб Новаковић

Др Иван Пинђер

Бисерка Милетић

Језичка редакција:

Бисерка Милетић, лектор

Софија Рацков, коректор

Марина Катић, преводилац

Издавачки савет:

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

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

CIP-Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

378.9(497.113)(082)
62

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

Месечно

ISSN 0350-428X

COBISS.SR-ID 58627591

ПРЕДГОВОР

Поштовани читаоци,

Пред вама је трећа 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 28.09.2018. до 31.10.2018. год., а који се промовишу 22.03.2019. год. То су оригинални прилози студената са главним резултатима њихових мастер радова.

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

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

У овој свесци, са редним бројем 3., објављени су радови из области:

- машинства и
- електротехнике и рачунарства.

У свесци са редним бројем 4. објављени су радови из области:

- грађевинарства
- саобраћаја,
- графичког инжењерства и дизајна,
- архитектуре,
- мехатронике и
- инжењерства информационих система.

У свесци са редним бројем 5. објављени су радови из области:

- инжењерског менаџмента,
- инжењерства заштите на раду и заштите животне средине,
- геодезије и геоматике,
- сценске архитектуре и дизајна,
- биомедицинског инжењерства и
- анимације у инжењерству.

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

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

„Високо место у друштву најбољих“

Уредништво

SADRŽAJ

	STRANA
Radovi iz oblasti: Mašinstvo	
1. Željko Vlaović, Borivoj Stepanov, ANALIZA RADA VENTILA SIGURNOSTI U USLOVIMA PREGREVANJA U ZATVORENIM SISTEMIMA CENTRALNOG GREJANJA	407-410
2. Anđelko Aleksić, Milenko Sekulić, OPTIMIZACIJA PARAMETARA PROCESA REZANJA POMOĆU PRIRODOM INSPIRISANIH ALGORITAMA	411-414
3. Dragana Radaković, Sanja Bojić, MODELOVANJE SKLADIŠNIH SISTEMA PRIMENOM SIMULACIJA.....	415-418
4. Милорад Кузманчев, Драган Живанић, ПРОЈЕКТОВАЊЕ И РЕГУЛАЦИЈА РАДА ТРАКАСТОГ ДОЗАТОРА	419-422
5. Danijel Todorović, UPOREDNA ANALIZA SISTEMA ZA PRIPREMU SANITARNE TOPLE POTROŠNE VODE U HOTELIMA	423-426
6. Dragan Lazić, UPOREDNA ANALIZA SISTEMA GRIJANJA TOPLOTNIM PUMPAMA I HIBRIDNIM SISTEMIMA.....	427-430
7. Драгиша Гутовић, ПРИМЕНА ТОПЛОТНИХ ПУМПИ КОД ОБЈЕКТА РАЗЛИЧИТЕ ЕНЕРГЕТСКЕ ЕФИКАСНОСТИ	431-434
8. Petar Tošić, PRINUDNE OSCILACIJE VIBROUDARNOG SISTEMA.....	435-438
9. Nemanja Dačević, Marko Vilotić, ODREĐIVANJE KOEFICIJENTA TRENJA METODOM SLOBODNOG SABIJANJA KONUSNOG PRSTENA	439-442
10. Radenko Blažević, Đorđe Vukelić, TRIBODIJAGNOSTIKA KINEMATSKOG KOEFICIJENTA TRENJA METODOM BLOK NA DISKU.....	443-446
11. Nikola Đorđević, SIMULACIJE RAZLIČITIH MANEVARA AUTOMOBILA MODELIRANOG NEHOLONOMNIM SISTEMOM	447-450

	STRANA
12. Igor Milinski, ANALIZA NAPONSKOG STANJA PODZEMNOG CEVOVODA.....	451-453
13. Darko Beretić, UPOREDNA ANALIZA METODA PROGRAMIRANJA NUMA.....	454-457

Radovi iz oblasti: Elektrotehnika i računarstvo

1. Aleksandar Atanasovski, Vladimir Katić, EFEKTI POBOLJŠANJA PROGNOZE IZLAZNE SNAGE VETROELEKTRANA.....	458-461
2. Igor Gere, RAZVOJ WEB APLIKACIJA SA SPRING DATA REST NA PRIMERU APLIKACIJE ZA PRAĆENJE ZADATAKA.....	462-465
3. Nikola Ajduković, Željko Trpovski, DALJINSKI NADZOR I INTEGRACIJA OPREME U SCADA SISTEM	466-469
4. Ana Mitrović, MODULARNI SISTEM ZA OBRADU ANALOGNIH SIGNALA	470-473
5. Marko Vujić, Vladimir Katić, PRIMENA SCADA SISTEMA U SMART GRIDU	474-477
6. Aleksandra Mitrović, RAZVOJNO OKRUŽENJE ZA PROJEKTOVANJE I GENERISANJE KODA NEURONSKIH MREŽA.....	478-481
7. Borka Samardžić, VIZUELIZACIJA OBJEKTOG MODELA U MEMORIJSKIM BAZAMA PODATAKA.....	482-485
8. Milica Plavšić, MODELOVANJE DISTRIBUTIVNE ELEKTROENERGETSKE MREŽE UPOTREBOM NEO4J BAZE PODATAKA	486-489
9. Robert Šafer, Željko Trpovski, ARTS test okruženje za M2B Wi-Fi HaLow™	490-493
10. Vojislav Milivojević, RAZVOJ OPTIČKOG KOMUNIKACIONOG UREĐAJA ZA RAD U SLOBODNOM PROSTORU	494-497
11. Marko Striško, ASINHRONI PRINCIPI U JAVASCRIPT PROGRAMSKOM JEZIKU	498-501
12. Srđan Šobot, 4G LTE STANDARD MOBILNE TELEFONIJE I NJEGOVA OPENAIR INTERFACE (OAI) REALIZACIJA	502-505
13. Ognjen Kundačina, Marko Obrenić, Predrag Vidović, MATEMATIČKI MODEL TRANSFORMATORA SA NAMOTAJIMA SPREGNUTIM U SLOMLJENU ZVEZDU	506-509
14. Dušan Jeftić, RAZVOJ DISTRIBUIRANE, VISOKO SKALABILNE APLIKACIJE UZ <i>MICROSOFT ORLEANS</i>	510-513
15. Simiša Simić, Vladimir Katić, MODELOVANJE I SIMULACIJA MODULARNOG PRETVARAČA SA VIŠE NIVOA (MMC) U REALNOM VREMENU KORISTEĆI PREKIDAČKU FUNKCIJU	514-517
16. Zoran Sikirica, OSETLJIVOST REŽIMA DISTRIBUTIVNE MREŽE NA PROMENU OPTEREĆENJA I VREDNOSTI PARAMETARA.....	518-521
17. Боровой Јовановић, ПРЕГЛЕД И ИСПИТИВАЊЕ ГРОМОБРАНСКИХ ИНСТАЛАЦИЈА.....	522-525
18. Ivan Hrubik, REALIZACIJA PAMETNE KLOPKE ZA PRAĆENJE BROJA INSEKATA U POLJU	526-529

	STRANA
19. Marko Nedić, KOMPENZACIJA REAKTIVNE SNAGE I COST BENEFIT ANALIZA NJENOG ULAGANJA	530-533
20. Дејан Старчевић, Веран Васић, Ђура Орос, ОПТЕРЕЋЕЊЕ ДИЗЕЛ ЕЛЕКТРИЧНОГ АГРЕГАТА	534-537
21. Slavica Gajić, Boris Dumnić, VEKTORSKO UPRAVLJANJE SINHRONOM MAŠINOM SA STALNIM MAGNETIMA	538-541
22. Goran Cibula, Veran Vasić, Ђура Орос, ELEKTROMOTORNI POGONI U EKSPLOZIVNIM SREDINAMA	542-545
23. Светозар Стојковић, СОФТВЕРСКИ СИСТЕМ ЗА ПРЕПОРУКУ СТУДИЈСКОГ МОДУЛА НА СМЕРУ РАЧУНАРСТВО И АУТОМАТИКА	546-549
24. Srđan Lazarov, KRATKOROČNA PROGNOZA PROMENE POTROŠNJE USLED PRIMENE SAVREMENIH TEHNIKA REGULACIJE NAPONA U DISTRIBUTIVNIM MREŽAMA	550-553
25. Andrea Gavrilović, Vladimir Katić, PREGLED RASPOLOŽIVIH SOFTVERA ZA SOLARNE ELEKTRANE	554-557
26. Драгана Шкрбић, Хесус А. Оливер, Владимир Катић, МОДЕЛОВАЊЕ, АНАЛИЗА И ОПТИМИЗАЦИЈА КОНТРОЛЕ БЕЖИЧНОГ ПУЊАЧА БАТЕРИЈЕ ЗА ЕЛЕКТРИЧНИ АУТОМОБИЛ	558-561
27. Mirko Macanović, Octavian Cornea, Vladimir Katić, MODELOVANJE SAVREMENIH METODA SKLADIŠTENJA ELEKTRIČNE ENERGIJE	562-565
28. Марко Роган, Владимир Катић, ИСПИТИВАЊЕ ПАРАМЕТАРА КВАЛИТЕТА ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ НА НАФТНОМ ПОЉУ КОМПАНИЈЕ „НИС а.д. Нови сад“	566-569
29. Лазар Симић, Зоран Стојановић, РЕКОНСТРУКЦИЈА ТРАНСФОРМАТОРСКЕ СТАНИЦЕ „МЕТАЉКА“	570-573
30. Marko Jelača, PRIMENA AKTER PROGRAMSKOG MODELA NA MIKROSERVISNU ARHITEKTURU SERVISA ZA SKLADIŠTENJE PODATAKA U MEMORIJI	574-577
31. Dejan Živković, PRISTUP NADGLEDANJU INDIKATORA RADA IOT ČVORIŠTA U REALNOM VREMENU	578-581
32. Александра Максимовић, РЕГУЛАЦИЈА ТЕМПЕРАТУРЕ ТЕРМАЛ УЉА У ПРОЦЕСУ ПРЕДЖЕЛИРАЊА ПАСТЕ У ПОГОНУ ЗА ПРОИЗВОДЊУ PVC ПОДА.....	582-585
33. Aleksandar Kokotović, Marko Vekić, UPRAVLJANJE HVDC PRENOSOM ZASNOVANIM NA PRETVARAČIMA KAO NAPONSKIM IZVORIMA	586-589
34. Saša Lalić, ANALIZA PERFORMANSI ALGORITAMA UČENJA USLOVLJAVANJEM U OKVIRU STARCRAFT 2 OKRUŽENJA.....	590-593
35. Marko Adamović, KOMPRESIJA MULTISPEKTRALNIH SATELITSKIH SNIMAKA KORIŠĆENJEM HEVC STANDARDA	594-597
36. Nikola Škrbić, RAZVOJ APLIKACIJE ZA TRENUTNU RAZMENU PORUKA NA ANDROID SISTEMIMA U PROGRAMSKOM JEZIKU KOTLIN	598-601
37. Nina Babić, RAZVOJ INFORMACIONOG SISTEMA ZA AUTO-ŠKOLE.....	602-605
38. Марко Иветић, ФУНКЦИОНАЛНА ПАРАДИГМА У ПРОГРАМСКОМ ЈЕЗИКУ C#.....	606-609
39. Милица Говедарица, МЕТОДОЛОГИЈЕ УПРАВЉАЊА ПРОЈЕКТОМ СА АКЦЕНТОМ НА SCRUM	610-613

	STRANA
40. Десанка Тодић, MICROSOFT AZURE-OBA ARHITEKTURA BEZ СЕРВЕРА	614-617
41. Boško Šogura, ADAPTACIJA I ANALIZA SELENIUM TESTOVA U C# PROGRAMSKOM JEZIKU	618-621
42. Saša Momčilović, OSNOVNI PRINCIPI, ŠABLONI I PRAVILA ZA PISANJE ČISTOG KODA	622-625

**ANALIZA RADA VENTILA SIGURNOSTI U USLOVIMA PREGREVANJA U
ZATVORENIM SISTEMIMA CENTRALNOG GREJANJA****ANALYSIS OF THE SAFETY VALVE OPERATION UNDER OVERHEATING
CONDITIONS IN CLOSED HEATING SYSTEMS**

Željko Vlaović, Borivoj Stepanov, *Fakultet tehničkih nauka, Novi Sad*

Oblast- MAŠINSTVO

Kratak sadržaj – Cilj rada jeste analiza sigurnosnih ventila u uslovima pregrevanja, za koje nisu projektovani. Njihov radni temperaturski opseg je do 110 °C. Pri pregrevanju u sistemu temperatura može porasti preko 110°C i sistem može ući u oblast ključanja. U tim uslovima, prilikom ekspanzije do atmosferskog pritiska dolazi do isparavanja u ventilu, koje dovodi do smanjenja masenog fluksa, što je izračunato raspoloživim metodama, koje su detaljno opisane.

Ključne reči: pregrevanje u sistemu centralnog grejanja, isparavanje u ventilu, povećanje pritiska.

Abstract – The aim of this study is the analysis and evaluation of safety valves in conditions of overheating, for which they are not designed. Their working temperature range is up to 110°C and the system may enter the boiling region. Under these conditions, during expansion to atmospheric pressure, flashing may occur in valve, which lead to decrease of the mass flux. The mass flux is calculated with the available methods, which are described in detail.

Keywords: overheating in central heating system, flashing in safety valve, pressure increase.

1. UVOD

Standard SRPS EN 303-5:2016 (Kotlovi za grejanje na čvrsta goriva, ručno i automatski loženi, nazivne toplotne snage do 500 kW – Terminologija, zahtevi, ispitivanje i obeležavanje) propisuje kada se mogu ugraditi kotlovi na čvrsto gorivo u zatvorene sisteme grejanja. Jedino onda kada se proizvodnja toplote može brzo regulisati. To nije moguće u standardnoj konstrukciji kotla koja potiče iz 70tih godina prošlog veka. Kotlovi na pelet mogu da budu brzo regulisani.

Inženjerska praksa u Srbiji ne prati ovu odrednicu standarda. Sigurnosni ventili ne mogu biti projektovani za takve uslove zato što u slučaju pojave pregrevanja oni nastoje da održe pritisak u dozvoljenim granicama, ali nisu u stanju da odvedu toplotu, čime se pregrevanje ne zaustavlja. Pregrevanje se može definisati kao razlika između proizvedene toplote u kotlu i toplote predate korisniku putem radijatora.

Problem koji može da se desi jeste povećanje temperature u kotlu. Uzrok povećanja temperature u kotlu može biti otkaz rada regulatora promaje. Kod kotlova koji sagorevaju čvrsto gorivo, dovod vazduha se reguliše pomoću regulatora promaje, zatvaranjem klapne za vazduh. U slučaju povećanja temperature, regulator temperature prigušuje dovod vazduha, kako bi smanjio produkciju toplote. Čak i da regulator zatvori klapnu, određeni deo vazduha će se infiltrirati. Drugi uzrok može biti otkaz rada cirkulacione pumpe. Određen deo prirodne cirkulacije će još uvek postojati na osnovu razlike gustina, ali će se deo proizvedene toplote zadržavati unutar kotla, jer ne može biti prosleđen do potrošača. Takođe, postoji mogućnost da se vazduh nađe u sistemu formirajući vazdušni čep. Vazdušni čep predstavlja barijeru cirkulaciji. Pumpa nije u stanju da održava cirkulaciju i sistem proizvodnje toplote (kotao) je u potpunosti odvojen od sistema prijema toplote (radijatori).

Maksimalno pregrevanje je određeno maksimalnom snagom samog kotla. Usled povećanja temperature, u sistemu raste i pritisak. Pri temperaturama preko 100°C prilikom havarije dolazi do parne eksplozije. Svaki kotao je dimenzionisan da izdrži određen pritisak i ukoliko taj pritisak naraste preko maksimalnog dozvoljenog pritiska, u posudi dolazi do havarije. Sigurnosni ventil je poslednja linije zaštite. On reguliše pritisak u posudi, ali nema mogućnost regulacije temperature, odnosno mogućnost odvođenja toplote.

Sigurnosni ventili koji se najčešće ugrađuju u zatvorene sisteme centralnog grejanja, jesu italijanskog proizvođača “Calleffi”. U tehničkoj dokumentaciji sigurnosnog ventila koju je proizvođač propisao, radni opseg temperature je do 110°C.

Pod pretpostavkom da je promena izobarna (da ventil sigurnosti uspeva da održi pritisak na zadatom nivou) izvršena je analiza kapaciteta sigurnosnog ventila na osnovu proračunatog masenog fluksa. Predstavljene su neke od metoda za određivanje masenog fluksa i maksimalnog masenog protoka u odnosu na različite početne temperature i stepene suvoće. Takođe, predstavljeno je poređenje dobijenih rezultata [1].

2. DVOFAZNO STRUJANJE KROZ VENTIL

Postoje različite metode za proračunavanje masenog fluksa. Veoma bitan parametar je karakter strujanja kroz ventil. Da li je jednofazno, voda ili para ili dvofazno. Mnogi proračunski modeli za predviđanje dvofaznog strujanja kroz idealnu mlaznicu su dostupni u literaturi. Metode za proračun dvofaznog strujanja koje se trenutno smatraju najpogodnijim za dimenzionisanje ventila

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Borivoj Stepanov, docent.

sigurnosti, zasnovane su na homogenom ravnotežnom ili homogenom neravnotežnim modelu. Homogeni ravnotežan model implicira da je fluid koji struji kroz ventil dvofazna (gas-tečnost) mešavina, da je ona “dobro” izmešana da može da se predstavi kao jednofazni fluid sa svojstvima koje su kombinacija obe faze, pri tom da su te dve faze u mehaničkoj i termodinamičkoj ravnoteži. Ove pretpostavke su neophodne, ali ne i dovoljne za proračun, uz njih potrebno je i proračunati gustinu fluida u funkciji od pritiska i stepena suvoće [2].

2.1. Termodinamička ravnoteža

Uobičajena pretpostavka je da su gasna ili parna faza u termodinamičkoj ravnoteži sa tečnom fazom, što znači da su osobine smeše funkcija samo temperature, pritiska i njihovih udela. Drugim rečima, kada se pritisak u tečnosti smanjuje na pritisak zasićenja, isparavanje će se pojaviti odmah, ako je sistem u ravnoteži. Međutim, isparavanje je zapravo proces koji zahteva određeno vreme (npr. nekoliko milisekundi) da se u potpunosti razvije. Za to vreme tečnost može da struji i nekoliko centimetara u mlaznici ventila. Pod tim uslovima, količina proizvedene pare je mnogo manja nego što bi ona bila pod ravnotežnim uslovima, a gustina mešavine i maseni fluks su srazmerno veći. Eksperimentalni podaci su pokazali da je u velikom broju pojedinačnih jednokomponentnih sistema, dužina mlaznice od minimum 10 cm dovoljna da se uspostavi ravnoteža, što implicira na to da se strujanje u mlaznicama kraćim od 10 cm može smatrati neravnotežnim strujanjem [3].

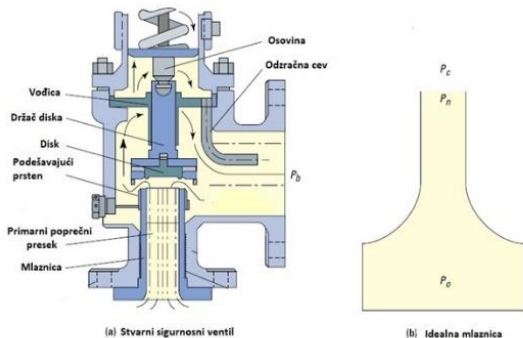
2.1.1 Pojava isparavanja u sigurnosnom ventilu

Radi lakšeg objašnjenja termina isparavanja u ventilu, najpre se mora definisati termin napon pare. Napon pare ili pritisak zasićene pare je pritisak iznad nivoa tečnosti. Na tom pritisku je uspostavljena ravnoteža između molekula koji isparavaju i onih koji se kondenzuju. Kada lokalni pritisak opadne ispod napona pare, tečnost će početi intenzivno da isparava [4].

3. DIMENZIONISANJE SIGURNOSNOG VENTILA

Potreban poprečni presek mlaznice se određuje na osnovu masenog protoka i masenog fluksa:

$$A = \frac{\dot{m}}{K_D \cdot G_0} \quad (1)$$



Slika 1. Stvarni sigurnosni ventil (levo) i idealna mlaznica (desno) [2]

Vrednost masenog fluksa je izračunata preko jednačine za izentropsku mlaznicu, pomoću usvojenog modela i pomnožena sa vrednošću koeficijenta isticanja, kako bi se

dobio stvaran maseni fluks. Kako je prikazano na slici 1, postoje sličnosti i razlike između idealne i stvarne mlaznice [5].

4. METODOLOGIJA PRORAČUNA POMOĆU DIREKTNE NUMERIČKE INTEGRACIJE

Prilikom proračuna ovom metodom, neophodno je poznavanje veličine termodinamičkih stanja. Ovde gustina predstavlja zavisnost promene pritiska, računata pomoću konstantne entropije, u opsegu od P_0 do P_a . Maseni fluks se izračunava korišćenjem formule [2]:

$$G_0 = \rho_n \cdot \left(-4 \sum_{P_0}^{P_n} \frac{P_{i+1} - P_i}{\rho_{i+1} + \rho_i} \right)^{\left(\frac{1}{2} \right)} \quad (2)$$

Što je manji usvojeni korak smanjenja pritiska, rezultati su precizniji. Da bismo odredili pritisak na kom dolazi do zagušenja, postupak proračuna ponavljamo smanjujući pritisak za usvojen korak, sve dok ne dobijemo maksimalni maseni fluks, počevši od $P_0 = P_n$. Ukoliko se maksimalni maseni fluks postigne tek na izlazu iz ventila ($P_n = P_a$), gde je P_a pritisak na izlazu iz ventila sigurnosti, atmosferski, neće doći do zagušenja protoka u samom ventilu. Ovaj metod ima široku primenu i primenljiv je na sve vrste fluida, pod bilo kojim uslovima. Metod se može proširiti i za neravnotežna strujanja, gde su dužine mlaznice kraće od 10 cm. Efekat neravnotežnog strujanja jeste da pojava isparavanja se neće desiti odmah kada pritisak opadne ispod pritiska zasićenja odnosno, kada pritisak dostigne pritisak zasićenja, isparavanje se neće u potpunosti razviti jer stepen suvoće ima manju vrednost od onog koji bi to imao za ravnotežno strujanje. Kako je gustina za ravnotežno dvofazno strujanje direktno zavisna od stepena suvoće, ona se računa po formuli:

$$\frac{1}{\rho} = \frac{x}{\rho_G} + \frac{(1-x)}{\rho_L} \quad (3)$$

Za neravnotežne uslove, gde je dužina mlaznice kraća od 10 cm, moramo pretpostaviti da stepen suvoće dostiže ravnotežno stanje x_e kako se dužina mlaznice približava dužini od 10 cm. Za mlaznice koje su manje od 10 cm, stepen suvoće u mlaznici se računa kao:

$$x = x_0 + (x_e - x_0) \cdot \frac{L}{10} \quad (4)$$

Kako se pritisak ispred mlaznice (P_n) bude smanjivao, maseni fluks (G_0) će rasti. Ukoliko dostigne svoj maksimum pre nego što pritisak u mlaznici opadne do atmosferskog, doći će do pojave zagušenja masenog protoka. Za početne uslove od 3,5 bara apsolutnog pritiska, temperature od 110°C, izračunavamo maseni fluks. Treba imati u vidu da se vrednost maksimalnog masenog fluksa izračunava pomoću formule:

$$G_n = \frac{G_0 \cdot K_D}{0,9} \quad (5)$$

gde je vrednost 0,9 preporučeni sigurnosni faktor [3].

5. METODOLOGIJA PRORAČUNA MASENOG FLUKSA POMOĆU OMEGA METODE

Ovaj metod je opisan u standardu API 520-1, i može se koristiti prilikom dimenzionisanja sigurnosnih ventila za

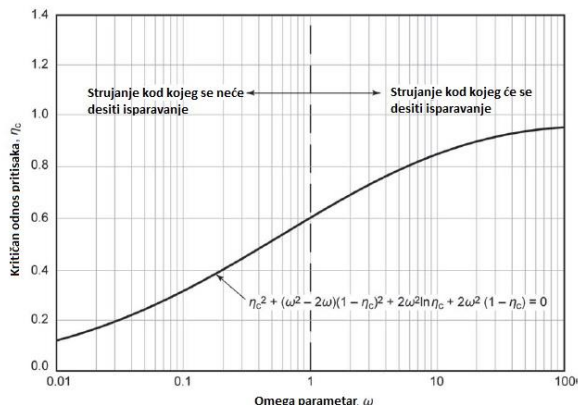
dvofazna strujanja. Metod koristi parametar omegu pomoću kojeg se određuje stanje:

$$\omega = \frac{x_0 \cdot v_{G0}}{v_0} + \frac{C_{f0} \cdot T_0 \cdot P_0}{v_0} \left(\frac{v_{fg0}}{v_0} \right)^2. \quad (6)$$

gde su:

- x_0 - početni stepen suvoće
- v_{G0} - specifična zapremina pare na početku $\left[\frac{m^3}{kg} \right]$
- C_{f0} - specifična toplota pri konstantnom pritisku $\left[\frac{J}{kg \cdot K} \right]$
- T_0 - temperatura fluida na ulazu u ventil $[K]$
- v_{fg0} - promena specifične zapremine $\left[\frac{m^3}{kg} \right]$
- P_0 - pritisak na ulazu u ventil $[Pa]$
- v_0 - specifična zapremina tečnosti na početku $\left[\frac{m^3}{kg} \right]$

Za slučaj kada je na početku tečnost suvozasiceana, udeo pare jednak je nuli. Omega parametar se izračunava iz formule (6). Na osnovu omega parametra i odnosa kritičnih pritisaka, konstatujemo da će doći do isparavanja tokom strujanja kroz ventil.



Slika 2. Odnos kritičnog pritiska i omega parametra [6]

Ukoliko je ispunjen uslov da je pritisak zasićenja veći od pritiska na izlazu iz ventila, tada se maseni fluks računa po formuli [5]:

$$\frac{G_c}{\sqrt{P_0}} = \frac{\eta_c}{\sqrt{\omega}}. \quad (7)$$

5.1. Metodologija proračuna za pothlađenu oblast sa visokim stepenom pothlađenja

Za slučaj kada je fluid na ulazu pothlađena tečnost, neophodna je provera uslova da li je tečnost sa slabim ili visokim pothlađivanjem. Formula za izračunavanje omega parametra je ista samo bez prvog člana, pošto je stepen suvoće jednak nuli. Za visoki stepen pothlađivanja maseni fluks se računa na osnovu formule [7]:

$$G_c = \sqrt{2 \cdot \rho_{f0} (P_0 - P_s)}. \quad (8)$$

5.2. Metodologija proračuna pomoću alternativne omega metode

Predstavljena metoda se može iskoristiti za projektovanje sigurnosnih ventila koji rukovode sa pothlađenom (uključujući i zasićenu) tečnošću na ulazu. Na ulazu ne

sme biti prisutna kondenzujuća para ili nekondenzujući gas. U zavisnosti od režima strujanja, tečnost će isparavati pre ili posle mlaznice. Korišćene formule se mogu primeniti na sve tečnosti. Formula za izračunavanje omega parametra je:

$$\omega_s = 9 \cdot \left(\frac{\rho_{l0}}{\rho_g} - 1 \right). \quad (9)$$

Na osnovu režima protoka i strujanja fluida, prema standardu API 520-1, za visoko pothlađeni režim strujanja [6]:

$$G = 1,414 \sqrt{\rho_{l0} (P_0 - P)}. \quad (10)$$

6. REZULTATI PRORAČUNA

Na osnovu dobijenih rezultata iz tabele 1 možemo primetiti da je najveća vrednost masenog fluksa dostignuta na pritisku od 1,45 bara (16893 $\frac{kg}{s \cdot m^2}$), što implicira da će doći do pojave zagušenja protoka u ventilu na tom pritisku.

Tabela 1. Dobijeni rezultati numeričkom integracijom za ravnotežno strujanje

P_N [bar]	ρ_n $\left[\frac{kg}{m^3} \right]$	$\frac{\Delta P}{\rho}$	$\Sigma \frac{\Delta P}{\rho}$	G_N $\left[\frac{kg}{s \cdot m^2} \right]$
3,5	951,05	0	0	0
3,45	951,05	-5,26	-5,26	2683,46
2,8	951,02	-5,26	-73,6	9871,9
1,45	950,9	-5,26	-215,5	16893,16
1,40	662,84	-13,34	-228,91	6859,40

U zavisnosti od toga u kakvom je stanju početna tečnost, maseni fluks će varirati. Uticaj temperature u pothlađenoj oblasti na maseni fluks kroz ventil je prikazan u tabeli 2, a uticaj stepena suvoće na maseni fluks kroz ventil je prikazan u tabelama 3, 4 i 5. Možemo zaključiti da sa povećanjem i temperature i stepena suvoće maksimalni fluks se smanjuje.

Tabela 2. Maksimalni maseni fluks u zavisnosti od početne temperature

t	99	110	114,5	119	124	137
Gn	17575	16893	15801	14635	13117	4516

Tabela 3. Maksimalni maseni fluks za korak promene stepena suvoće pare 0,001

x_e	0,001	0,002	0,003	0,004	0,005	0,006
Gn	2229	2220	2183	2146	2118	2098

Tabela 4. Maksimalni maseni fluks za korak promene stepena suvoće pare 0,01

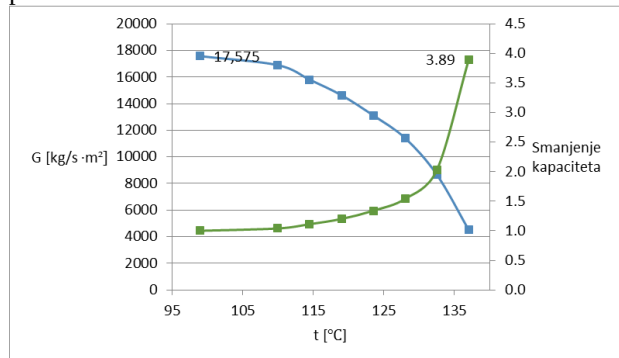
x_e	0,01	0,02	0,03	0,04	0,05	0,06	0,07
Gn	1985	1785	1643	1534	1448	1376	1314

Tabela 5. Maksimalni maseni fluks za korak promene stepena suvoće pare 0,1

x_e	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
Gn	1169	905	766	677	613	564	526	494

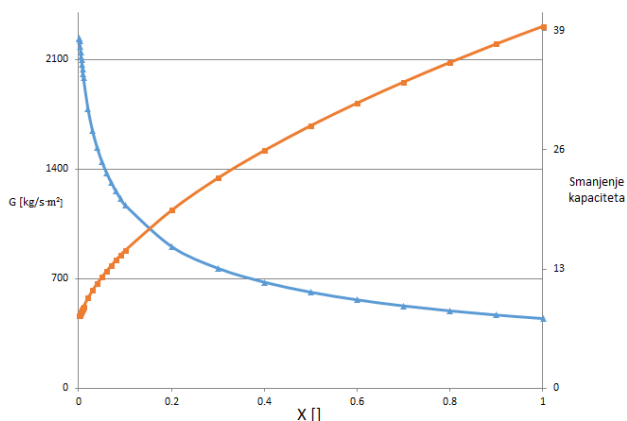
U tabeli 5 su prikazani rezultati maksimalnog masenog fluksa za različite početne temperature.

Sa povećanjem početne temperature u sistemu, maksimalni maseni fluks opada, što se može videti na slici 3 jer tečnost pri ekspanziji dostiže uslove za pojavu isparavanja. Istovremeno, kapacitet ventila se smanjuje i do 3,9 puta u odnosu na referentnu vrednost izmerenu na 99°C.



Slika 3: Dijagram zavisnosti maksimalnog masenog fluksa od povećanja početne temperature u sistemu i smanjenja kapaciteta ventila u odnosu na referentnu vrednost izmerenu na 99°C.

Analogno tome, slika 4 nam prikazuje rezultate maksimalnog masenog fluksa u odnosu na različit početni stepen suvoće. Zaključujemo da u prelaznoj oblast kapacitet ventila drastično opada.



Slika 4: Vrednosti maksimalnog masenog fluksa u zavisnosti od početnog stepena suvoće dobijenih i smanjenja kapaciteta ventila u odnosu na referentnu vrednost izmerenu na 99°C.

7. ZAKLJUČAK

Izvršena je analiza rada sigurnosnog ventila u zatvorenom sistemu centralnog grejanja pri pregrevanju. Zaključujemo da u slučaju otkaza rada regulacione opreme ili zastoja u cirkulaciji, sigurnosni ventil nije u stanju da reguliše povećanje temperature u zatvorenom sistemu. Sigurnosni ventil ne može da odvede toplotu iz sistema, on reguliše samo pritisak. U slučaju povećanja temperature preko 100°C , postoji opasnost od parne eksplozije. Sigurnosni ventil u takvim okolnostima ne poseduje dovoljan kapacitet da adekvatno odreguje. Na osnovu proračuna masenog fluksa u zavisnosti od početne temperature i stepena zasićenja, možemo zaključiti da će doći do isparavanja u sigurnosnom ventilu što dovodi do zagušenja protoka u ventilu. Vrednost masenog fluksa direktno zavisi od početnih uslova, pre svega temperature i pritiska na kom se sigurnosni ventil otvara.

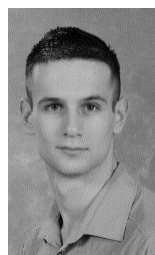
Sa povećanjem početne temperature u sistemu, maksimalna vrednost masenog fluksa opada, što negativno utiče na protok u samom ventilu. Isparavanje se dešava kada pritisak u mlaznici opadne ispod napona pare, prilikom strujanja kroz suženje mlaznice ("vena contracta").

U suštini, pojava isparavanja u sigurnosnom ventilu ograničava kapacitet ventila, i on nije u stanju da istisne zahtevanu količinu tečnosti. Kako stepen suvoće raste, maseni fluks opada, a srazmerno opada i kapacitet ventila. Na gornjoj graničnoj krivi kapacitet ventila opada do 38 puta. Na donjoj graničnoj krivi kapacitet ventila opada 7 puta. Dostizanje stanja na donjoj graničnoj krivi je u praksi realnije, što implicira da ventil koji je početno bio namenjen za kotlove do 72 kW, može podneti snagu od približno 10 kW. Ukoliko pretpostavimo da je prosečan kotao snage 30 kW, rezultat može biti ozbiljno ugrožavanje bezbednosti.

8. LITERATURA

- [1] SRPS EN 303-5:2016 - Kotlovi za grejanje na čvrsta goriva, ručno i automatski loženi, nazivne toplotne snage do 500 kW
- [2] Darby, R., Meiller, P. R., & Stockton, J. R. (2001). Select the Best Model, (May).
- [3] Darby, R., C. (2005). Size Safety-Relief Valves for Any Conditions, Published in Chemical Engineering, 112, (9), 42-50.
- [4] Bolle, L., Downar-Zapolski, P., Franco, J., & Seynhaeve, J. M. (1996). Experimental and theoretical analysis of flashing water flow through a safety valve. *Journal of Hazardous Materials*, 46(2-3), 105-116.
- [5] Darby R., & Raj P.C. (2004). Chemical engineering fluid mechanics, third edition, chapter 11, 291-301.
- [6] American Petroleum Institute. (2008). Sizing, Selection and Installation of Pressure Relieving Devices in Refineries, Part I - Sizing and Selection. API Standard 520, (December 2008).
- [7] Lees, F. P. (1996). Loss Prevention in the Process Industries: Hazard Identification, Assessment and Control, 1, 3776.

Kratka biografija:



Željko Vlaović rođen je u Subotici 1994. god. Završio gimnaziju „Isidora Sekulić“ u Novom Sadu 2013. godine, iste godine upisao osnovne akademske studije na Fakultetu tehničkih nauka iz oblasti Mašinstva – Energetika i procesna tehnika. Odbranio diplomski rad na temu Upravljanje energetskim performansama rashladnih postrojenja 2017. god. Trenutno student master studija na smeru Energetika i procesna tehnika.



Borivoj Stepanov rođen je u Novom Sadu 1976. Doktorirao na Fakultetu tehničkih nauka 2014 god., a od 2003 god. predaje predmete koji se bave termoelektričkim postrojenjima.



OPTIMIZACIJA PARAMETARA PROCESA REZANJA POMOĆU PRIRODOM INSPIRISANIH ALGORITAMA

OPTIMIZATION OF CUTTING PARAMETERS BY NATURE-INSPIRED ALGORITHMS

Anđelko Aleksić, Milenko Sekulić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj - U radu je prikazan kratak osvrt na značaj modelovanja procesa obrade kao i osnovne metode modelovanja. Dat je teoretski deo o optimizaciji procesa obrade sa kratkim pregledom metoda optimizacije, kao što su: lokalno pretraživanje, algoritam simultanog žarenja i algoritam slepog miša. Detaljno je opisan mravlji algoritam - ACO i algoritam roja čestica - PSO. Prikazana su eksperimentalna istraživanja i matematički model na osnovu kog je vršena optimizacija pomoću ACO, PSO i GA (genetski algoritam). Takođe je izvršeno poređenje rezultata dobijenih optimizacijom pomoću prirodom-inspirisanih algoritama (ACO, PSO i GA) sa jednom klasičnom optimizacionom metodom, Taguchi metod. Na kraju rada dat je prilog u kome su predstavljene print skrinovi korisničkog interfejsa programa MATLAB, pri optimizaciji pomoću ACO, PSO i GA.

Ključne reči: Optimizacija, prirodom inspirisani algoritmi, mravlji algoritam, optimizacija rojem čestica.

Abstract – This paper gives a brief overview of the importance of modeling the processing process as well as the basic modeling model. There is a theoretical part about optimization methods, such as: local search, simulated annealing and bat algorithm. Ant colony optimization and particle swarm optimization are described in detail. Experimental research and a mathematical model based on which ACO, PSO and GA (genetic algorithm) were performed were presented. Comparison of the results obtained by optimization with the nature-inspired algorithms (ACO, PSO and GA) was also compared with a classical optimization method Taguchi method. At the end of the work, there is an attachment in which the print screens of the MATLAB user interface are presented, with ACO, PSO and GA optimization.

Keywords: Optimization, nature-inspired algorithm, ant colony optimization, particle swarm optimization

1. UVOD

Predmet istraživanja rada predstavlja istraživanje na polju prirodom inspirisanih metoda za optimizaciju procesa obrade rezanjem. Metode optimizacije na kojima su vršena istraživanja su: optimizacija pomoću mravljeg algoritma (eng. Ant Colony Optimization – ACO), optimizacija pomoću rojeva čestica (eng. Particle Swarm Optimization – PSO) i optimizacija pomoću genetskog algoritma (eng. Genetic Algorithm).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milenko Sekulić.

Cilj istraživanja jeste ispitivanje mogućnosti efikasne primene ovih metoda (algoritama) u optimizaciji procesa obrade rezanjem. Mogućnosti ovih optimizacionih algoritama su ispitane na primeru optimizacije procesa obrade čeonim glodanjem. Kao funkcija cilja je odabrana minimalna vrednost srednje aritmetičke hrapavosti (Ra)_{min} [μ m].

2. MODELOVANJE PROCESA OBRADRE REZANJEM

Modelovanje procesa obrade predstavlja prvi neophodna korak za optimizaciju parametara procesa. Predstavlja razumevanje principa procesa obrade pomoću razvoja eksplicitnog matematičkog modela koji prezentuje realni sistem. Dakle, optimalni izbor parametara obrade je nezamisliv bez prethodnog definisanja pouzdanog matematičkog modela.

Svrha modelovanja procesa obrade je da se: realno predvide rezultati procesa, da se steknu nova znanja o pojedinim fazama procesa koja će pomoći projektovanju samog procesa, brzo i realno izvrši optimizacija procesa, izvedu zaključci o praćenju i upravljanju procesom.

U praksi se mogu sresti sledeće vrste modela: empirijski modeli, mehanistički modeli, analitički modeli, numerički modeli, modeli zasnovani na veštačkoj inteligenciji, modeli zasnovani na višefaktornom planu eksperimenta i hibridni modeli.

3. METODE ZA OPTIMIZACIJU PROCESA OBRADRE REZANJEM

Optimizacija predstavlja postupak definisanja najpovoljnijih rešenja za date početne uslove, iz skupa mogućih rešenja. Objekat optimizacije može biti neki proces u širem ili užem obliku ili sistem (tehnološki, obradni, proizvodno-pogonski ili neko drugo postrojenje, instalacija, proizvod, alat, itd.). Nema ograničenja kada je u pitanju izbor objekta optimizacije, jer postoji potreba da se svako tehničko ili tehnološko rešenje unapredi, što se itekako postiže primenom metoda optimizacije. Prema tome, nema tehničkih nauka niti inženjerskih problema, gde se manje ili više ne koriste metode optimizacije. Može se reći da gotovo ne postoji područje čovekove delatnosti, sistema ili procesa koji se na neki način ne bi mogao optimizovati.

Neka od opštih podela optimizacije jeste na: makrooptimizaciju (izbor optimalne varijante tehnološkog procesa) i mikrooptimizaciju (optimizacija rečzima obrade, geometrije alata, ...).

Za rešavanje problema optimizacije parametara procesa obrade koristi se širok spektar optimizacionih metoda i algoritama. U literaturi se mogu sresti različite podele

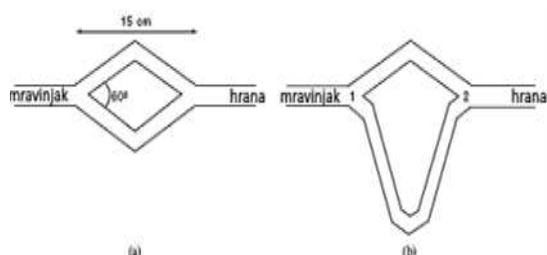
ovih metoda i algoritama. Optimizacioni algoritmi se, po mišljenju nekih autora, mogu svrstati u dva različita tipa: Tradicionalni i netradicionalni algoritmi [5].

4. MRAVLJI ALGORITAM

Termin „Ant Colony Optimization (ACO)“ uvodi Marco Dorigo u svom doktoratu iz 1992. godine [1], kao rezultat istraživanja kombinatornih optimizacionih pristupa. U početku, algoritam se primenjivao na problem trgovačkog putnika i problem udruživanja kvadranta. Od 1995. Dorigo i ostali rade na raznim proširenim verzijama početne ideje.

Mravi imaju sposobnost da pronađu najkraću putanju koja vodi od mravinjaka do izvora hrane bez ubotrebe vizuelnog znaka. Ovu sposobnost mravi realizuju tako što svojim kretanjem po tlu ostavljaju određenu količinu hemijske supstance poznate kao feromon. Pored toga, poseduju mogućnost prilagođavanja novonastalim promenama oko njih. Na primer, kada nađu najkraći put do izvora hrane, u slučaju pojave neke barijere na putu ili zakrčenja puta, pronalaze novi najkraći put.

Jedan od prvih eksperimenata, u laboratorijskim uslovima, provere sposobnosti mrava da nađu najkraću putanju jeste „eksperiment sa dva mosta“. Vršeno je dva različita načina ispitivanja, prvi je predstavljao most sa jednakim dužinama grana, dok drugi eksperiment je izveden sa različitim dužinama grana mosta, prikazano na slici 4.1.



Slika 4.1: Postavke eksperimenta dvostrukog mosta: a) sa jednakim granama i b) sa različitim dužinama grana [1]

U prvom slučaju (pod a) mravinjak kolonije mrava povezan je sa izvorom hrane preko dva mosta jednake dužine. Duž putanje između izvora hrane i gnezda, mravi deponuju feromon. Inicijalno, tj. na početku eksperimenta, svaki mrav nasumično bira jedan od dva mosta (slika 4.1a). Međutim, kao posledica slučajnog protoka mrava, posle nekog vremena na jednom od dva mosta pojaviće se veća koncentracija feromona koja će kao posledicu privući više mrava.

Ovo dodatno povećava koncentraciju feromona na tom mostu, što ga čini privlačnijim za mave, a kao posledica toga nakon nekog vremena čitava kolonija će težiti da koristi isti most. Deneubourgh i njegovi saradnici su ponovili eksperiment više puta i zapazili da će mravi odabrati jedan od dva mosta sa verovatnoćom od 50%. Ovakvo ponašanje na nivou kolonije, zasnovano na eksploataciji povratne informacije, mravi koriste kako bi pronašli najkraći put između izvora hrane i njihovog gnezda [1].

Goss i njegove kolege su razmatrali varijantu eksperimenta sa dva mosta u kojoj je jedan most značajno duži od drugog (slika 4.1b). U ovom slučaju stohastički protok kod inicijalnog odabira mosta je značajno

redukovan, a drugi mehanizam igra značajnu ulogu. Naime, mravi koji slučajnim izborom odaberu kraći most, prvi će stići nazad do mravinjaka. Kraći most će stoga primiti feromon ranije nego duži, a ova činjenica povećava verovatnoću da će mravi koji slede, odabrati kraći most pre nego duži most. Goss i njegovi saradnici su razvili matematički model posmatranog ponašanja.

Mravi, iako su slepi, mogu naći najkraću putanju do izvora hrane. Ova karakteristika mrava je na raspolaganju za rešavanje stvarnih problema korišćenjem određenih karakteristika i nekih dodataka. Karakteristike mrava koje stvaraju veštačke mrave su: komunikacija između mrava pomoću hemijske supstance feromon, preferiranje puteva sa većom količinom feromona, brže povećanje feromona na kraćim putevima, nego na dužim.

Karakteristike koje su dodate stvarnim mravima: žive u okruženju gde se vreme obračunava diskretno, potpuno su slepi i mogu pristupiti detaljima o problemu, mogu zadržati informacije za rešavanje određenog problema sa određenom količinom memorije [1].

Način funkcionisanja algoritma prikazan je pseudokodom (algoritam 4.1).

Algoritam 4.1: Pseudokod – ACO metaheuristika [1]

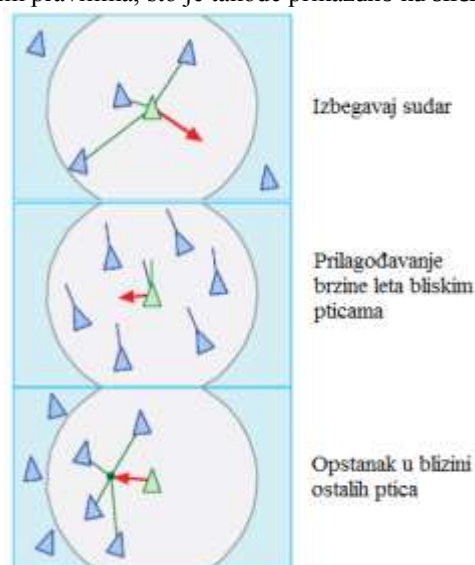
```

PROCEDURE ACO_metaheuristika()
  Podesi_Parametre();
  Inicijalizuj_Vrednosti_Traga_Feromona();
  WHILE (Nije_zadovoljen_kriterijum_za_zastavljanje) DO
    Raspored_aktivnosti
    Konstruisi_Mrave_I_Mravija_Rešenja();
    Primeni_Lokalnu_Pratragu(); {opciono}
    Podesi_Vrednosti_Feromona();
  END Raspored_aktivnosti
END WHILE
END PROCEDURE

```

5. OPTIMIZACIJA ROJEM ČESTICA - PSO

Optimizacija rojem čestica (eng. Particle Swarm Optimization – PSO) je stohastički algoritam koji se bazira na populaciji rešenja. Algoritam optimizacije rojem čestica otkriven je sasvim slučajno, pri pokušaju da se na računaru simulira kretanje jata ptica. Reynolds C.W. u svom radu 1986. godine razmatra jato ptica kao skup čestica gde svaka čestica (tj. ptica) svoj let prilagođava sledećim pravilima, što je takođe prikazano na slici 5.1:



Slika 5.1: Način funkcionisanja jedinke u jatu [2]

Sam algoritam je u određenoj meri inspirisan i sociološkim interakcijama između pojedinaca u populaciji. Svaka čestica je potencijalno rešenje razmatranog problema i koristi sopstveno iskustvo i iskustvo susednih čestica da bi odabrala kako da se kreće u prostoru pretrage. U fazi inicijalizacije svakoj čestici je dat slučajno odabrani položaj i početna brzina. Pozicija čestice predstavlja rešenje problema i zato ima vrednost datu funkcijom cilja. Dok se kreću u prostoru pretrage čestice pamte položaj, tj. poziciju najboljeg rešenja koje su našle. U svakoj iteraciji algoritma svaka čestica se kreće brzinom koja je suma tri komponente: stare brzine, brzine koja pokreće česticu ka poziciji u prostoru pretrage gde je prethodno pronašla najbolje rešenje do tada i brzine koja pokreće česticu prema lokaciji u prostoru pretrage gde su susedne čestice našle najbolje rešenje do tada. PSO algoritam je primenjen na velikom broju različitih problema i još je jedan primer veštačko-inženjerskog SI sistema.

Roj čestica je u prirodi stohastičan, on iskorišćava vektor brzine kako bi ažurirao trenutnu poziciju svake čestice u roju. Vektor brzine se ažurira na temelju pamćenja svake čestice, što konceptijski odgovara autobiografskoj memoriji, kao i na temelju znanja koje je stekao roj kao celina. Pozicija čestice u jatu je ažurirana na temelju socijalnog ponašanja roja koji se prilagođava svom okruženju stalnim traženjem boljih pozicija tokom vremena.

Numerički, pozicija x čestice i u iteraciji $k + 1$ je ažurirana na sledeći način:

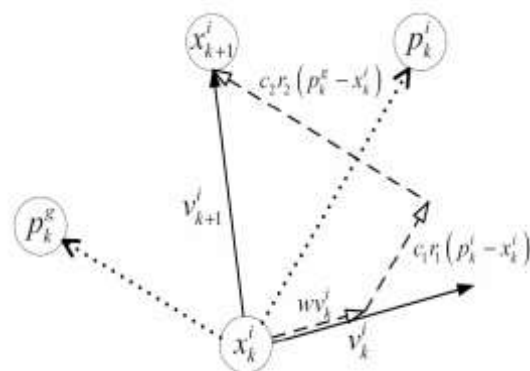
$$x_{k+1}^i = x_k^i + v_{k+1}^i \Delta t \quad (1)$$

pri čemu je v_{k+1}^i pripradajući ažurirani vektor brzine, a Δt je vrednost vremenske step funkcije. Vektor brzine svake čestice se računa kao:

$$v_{k+1}^i = \omega v_k^i + c_1 r_1 \frac{(p_k^i - x_k^i)}{\Delta t} + c_2 r_2 \frac{(p_k^g - x_k^i)}{\Delta t} \quad (2)$$

pri čemu je v_k^i vektor brzine u iteraciji k , p_k^i i p_k^g predstavljaju najbolju poziciju čestice ikad i globalnu najbolju poziciju čitavog roja sve do trenutne iteracije k , dok r predstavlja nasumični broj iz intervala $[0,1]$. Preostali članovi su konfiguracijski parametri koji igraju važnu ulogu u konvergencijskom ponašanju PSO-a. Član c_1 (kognitivni, samospoznajni parameter) predstavlja stepen poverenja u globalnom najboljem rešenju (najbolje pronađeno rešenje od jata kao celine). Uglavnom se uzima $1,8 < c_1 = c_2 < 2,2$. Poslednji član ω je iteracijska varijabla koja je iskorišćena za kontrolisanje istraživačkih sposobnosti roja tako da skalira vrednost trenutne brzine, te na taj način utiče na iznos ažuriranog vektora brzine. Većim vrednostima interacijske varijable (ω) vršimo globalno pretraživanje zbog toga što se ažurirani vektor brzine brže povećava dok zadavanjem manje vrednosti inercijske varijable vrednost ažuriranog vektora brzine postaje manja pa se tako novi položaj čestice ograničava na manje područje prostora istraživanja, tj. omogućavamo lokalno pretraživanje.

Na slici 5.1 prikazana je pozicija čestice i ažuriranje vektora brzine na prethodno opisan način u dvodimenzionalnom prostoru. Isto tako je vidljivo da će ažurirani položaj čestice zavisiti ne samo od najboljih pozicija roja i čestice, nego i od veličine konfiguracijskih parametara.



Slika 5.1: Pomeranje položaja čestice i ažuriranje vektora brzine [3]

6. EKSPERIMENTALNA ISTRAŽIVANJA

Eksperimentalna istraživanja, objašnjena u ovom radu, bazirana su na doktorskoj disertaciji B. Savkovića [4]. Istraživanja su sprovedena na Fakultetu tehničkih nauka, u Laboratoriji za konvencionalne tehnologije obrade.

Tabela 6.1: Izmerene i računске vrednosti sa prosečnom greškom eksperimenta [7]

R. br.	Rezultati za R_a [μm]:		Greška eksperimenta
	Eksperimentalni	Računski	
1	1,074	1,052	2%
2	1,081	1,117	4%
3	1,743	1,840	10%
4	1,645	1,838	19%
5	1,058	0,929	13%
6	1,023	0,951	7%
7	1,898	1,804	9%
8	1,734	1,738	0,4%
9	1,214	1,203	1%
10	1,236	1,203	3%
11	1,227	1,203	2%
12	1,175	1,203	3%
13	1,132	1,122	1%
14	1,253	1,146	11%
15	0,999	1,123	12%
16	2,556	3,617	30%
17	1,245	1,336	9%
18	1,237	1,118	12%
19	1,134	1,122	1%
20	1,235	1,146	9%
21	0,991	1,123	13%
22	2,489	3,617	31%
23	1,239	1,336	10%
24	1,221	1,118	10%
Prosečna greška:			9%

Modelovanje procesa obrade glodanjem izvedeno je na osnovu višefaktornog plana eksperimenta. Rezultati dobijeni eksperimentalnim ispitivanjem prikazani su u tabeli 6.1.

Obradom podataka dobijen je trofaktorni model drugog reda za srednju aritmetičku hrapavost obrađene površine:

$$R_a = C v^{p_1} s_1^{p_2} a^{p_3} *$$

$$\cdot \exp \left[p_{11}(\ln v)^2 + p_{22}(\ln s_1)^2 + p_{33}(\ln a)^2 + p_{12} \ln v \ln s_1 + p_{13} \ln v \ln a + p_{23} \ln s_1 \ln a \right] \quad (3)$$

gde v , s_1 i a predstavljaju režime obrade, v - brzina rezanja [m/s], s_1 – pomak alata po zubu [mm/z], a - dubina rezanja [mm]. Ostale vrednosti predstavljaju konstantu C i eksponente (p_1 , p_2 , p_3 , p_{11} , p_{22} , p_{33} , p_{12} , p_{13} , p_{23})

7. OPTIMIZACIJA PROCESA OBRADJE GLODANJEM

Optimizacija procesa čeonog glodanja, odnosno određivanje optimalnih vrednosti režima obrade, sprovedena je primenom različitih prirodom-inspirisanih algoritama. U tu svrhu su korišćene tri jednokriterijumske metode: Optimizacija pomoću mravljeg algoritma, Optimizacija rojevima čestica i Genetski algoritam.

Funkcija cilja u sva tri slučaja je bila minimalna vrednost srednje aritmetičke hrapavosti R_a , odnosno $(R_a)_{min}$.

Sam proces optimizacije je sproveden u programskom paketu MATLAB. ovaj softver ima brojne module, koji omogućavaju izvršavanje skrip koda za svaku od pobrojanih optimizacionih metoda.

Granične vrednosti ulaznih parametara (režima obrade) predstavljene su tabelom 6.2.

Tabela 6.2: Prikaz graničnih vrednosti ulaznih parametara

Naziv parametra	Minimalna vrednost	Maksimalna vrednost
v [m/s]	2,356	5,864
s_1 [mm/z]	0,089	0,223
a [mm]	0,5	2,6

Analiza dobijenih rešenja

Rezultati dobijeni pomoću različitih optimizacionih tehnika su predstavljeni u tabeli 6.3. Optimalne vrednosti dobijene pomoću različitih prirodom-inspirisanih algoritama (ACO, PSO i GA) se u potpunosti podudaraju. Ova činjenica pokazuje da je proces optimizacije obrade čeonim glodanjem sprovedena kvalitetno. Na kraju su optimalne vrednosti upoređene sa jednom klasičnom metodom optimizacije procesa obrade, Tagučići metod.

Tabela 6.3: Optimalne vrednosti ulaznih parametara pri procesu čeonog glodanja

Metoda optimizacije	OPTIMALNI PARAMETRI REZANJA				
	Zahtevana izlazna performansa procesa	v [m/s]	s_1 [mm/z]	a [mm]	
ACO	R_a [μm]	0,8236	2,36	0,1011	2,6
PSO	R_a [μm]	0,8236	2,36	0,1011	2,6
GA	R_a [μm]	0,8236	2,36	0,1011	2,6
Tagučići metod	R_a [μm]	0,875	2,35	0,089	2,6

8. ZAKLJUČAK

Na osnovu cilja istraživanja, koji predstavlja ispitivanje mogućnosti efikasne primene ovih metoda (algoritama) u optimizaciji procesa obrade rezanjem sa određenom funkcijom cilja $(R_a)_{min}$ i eksperimentata koji su izvršeni, može se zaključiti sledeće:

- potrebno je prethodno eksperimentalno istraživanje kako bi se formirao matematički model,
- bez kvalitetnog i pouzdanog matematičkog modela ne mogu se očekivati kvalitetna optimizaciona rešenja,
- da bi se izvršila optimizacija pomoću mravljeg algoritma, potrebno je uvesti kaznene faktore, koji omogućavaju ograničenje prostora pretrage, što nije slučaj za PSO,
- potreba za skript kodom određene metode (algoritma),
- brzina pronalaska optimalnih rešenja bolja kod PSO metode (u odnosu na ACO),
- vreme pronalaska rešenja je neizvesno, ali je zagarantovano.

Postignuti rezultati, dobijeni pomoću različitih metoda (algoritama ACO, PSO i GA), navode na zaključak da su sve tri navedene metode prikladne i veoma pouzdane metode za optimizaciju parametara čeonog glodanja.

9. LITERATURA

- [1] Dorigo M., Birattari M., Stützle T.: „*Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique*“, IRIDIA – Technical Report Series, September 2006.
- [2] Reynolds C.: „*Boids*“, 1986. Homepage: <https://www.red3d.com/cwr/boids/> 2.10.2018.
- [3] Li J. G., Yao Y. X., Gao D., Liu C. Q., Yuan Z. J.: „*Cutting parameters optimization by using particle swarm optimization (PSO)*“, Applied Mechanics and Materials, Vol. 10-12, 2008., pp. 879–883.
- [4] Savković B.: „*Doktorska disertacija: Modeliranje funkcije obradivosti pri procesu obrade glodanjem*“, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, 2015.
- [5] Sekulić M., Pejić V., Brezocnik M., Gostimirović M. Hadžistević M.: „*Prediction of surface roughness in the ball-end milling process using response surface methodology, genetic algorithms, and grey wolf optimizer algorithm*“, Advances in Production Engineering & Management, Vol. 13 (1), 2018., pp. 18-30.

Kratka biografija:



Andelko Aleksić rođen je u Sremskoj Mitrovici 1994. god. Master rad na Fakultetu tehničkih nauka iz oblasti procesa obrade skidanjem materijala odbranio je 2018. god.



Milenko Sekulić rođen je u Prijepolju 1966. Doktorirao je na Fakultetu tehničkih nauka 2007. god., a od 2017. je zvanju redovnog profesora. Uža naučna oblast su procesi obrade skidanjem materijala.

MODELOVANJE SKLADIŠNIH SISTEMA PRIMENOM SIMULACIJA**MODELING OF STORAGE SYSTEMS BY APPLICATION OF SIMULATION**Dragana Radaković, Sanja Bojić, *Fakultet tehničkih nauka, Novi Sad***Oblast –MAŠINSTVO**

Kratak sadržaj – U ovom radu predstavljen je odabir odgovarajućeg skladišnog sistema koji će biti instaliran u već postojećem skladišnom objektu. Postoje tri predloga rešenja, od kojih je izabran optimalan pomoću simulacija koje su rađene u softveru Enterprise Dynamics.

Ključne reči: skladišni sistemi, skladištenje, simulacije, Enterprise Dynamics.

Abstract - This paper presents selection of the appropriate warehouse system that will be installed in the already existing warehouse facility. There are three suggested solutions, of which the most optimal is chosen using simulations developed in Enterprise Dynamics.

Keywords: warehouse systems, storage, simulations, Enterprise Dynamics.

1. UVOD

Neophodan korak u optimizaciji tokova roba jeste optimizacija skladišnih sistema. S obzirom da skladištenje, koji je neophodan segment u tokovima roba, stvara velike troškove samim tim je velika potreba za optimizacijom tog procesa. Simulacija predstavlja jednu od priznatih metoda optimizacije. Efikasan je alat za planiranje, organizaciju i optimizaciju proizvodnih i logističkih procesa. Pomoću simulacionog modela moguće je sistemsko posmatranje i uočavanje eventualnih nepravilnosti, uskih grla, problema koji se javljaju u skladišnim sistemima i mogućnosti njihovog otklanjanja, i to ne samo u slučaju kada je skladište već napravljeno, nego i u samom procesu planiranja.

Pod skladištenjem se podrazumeva privremeno ili trajno čuvanje robe, u cilju amortizacije prostornih i vremenskih neravnomernosti koje se javljaju u tokovima roba, uz minimizaciju troškova, podizanja kvaliteta usluge ili obezbeđenja uslova da se proces realizuje [1].

2. SKLADIŠNI SISTEMI

U ovom radu se pristupilo rešavanju problema sa odabirom skladišnog sistema koji treba biti instaliran u jednom delu postojećeg skladišnog objekta i analizirano je koje od rešenja je optimalno i to korišćenjem pomenutog softvera, na osnovu specifikacija i karakteristika svakog od njih. Postoje različiti tipovi skladišnih sistema, od kojih su tri tipa bila predlog rešenja za instalaciju i odnose se na potencijalnu primenu konvencionalnih (klasičnih), pokretnih ili prolaznih (drive in) regala.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila doc. dr Sanja Bojić.

U zavisnosti od parametara skladišta kao kapaciteta, zahteva skladištenja, tipa robe koja se skladišti i dr, postoje i različiti sistemi, čija konstrukcija zavisi od njihove namene, i po kojoj razlikujemo: nepokretne, pokretne, prolazne "drive in" regale, regale protočnih skladišta, regale specijalne namene, od kojih će se u nastavku napomenuti samo ona koja su deo ovog rada.

Klasični regali su najčešće konstruisani od vertikalnih ramnih nosača i horizontalnih nosača na koje se odlažu palete, a koji su istovremeno i vezni elementi prostorne rešetke. Kod ovog tipa moguće je pristupiti svakoj paleti.

Pokretni regali objedinjavaju dobra svojstva blok sistema i skladištenja u redovima. Obezbeđuju kompaktnu strukturu skladišta, ali omogućavaju i direktan pristup svakom materijalu (što kod blok sistema nije moguće).

Prolazni - drive in regali predstavljaju najprostije rešenje za formiranje regalnog skladišta po blok sistemu. Viljuškar koji se kreće duž regalnih hodnika, manipuliše paletnim jedinicama, pri čemu se palete slažu po visini jedna iznad druge, tako što se oslanjaju na kratke oslonce koji su fiksirani za regalnu konstrukciju. Svaka paleta po visini ima svoje tačke oslonca tako da palete nisu izložene pritisku od strane paleta koje se nalaze u vertikalnim slogovima [2].

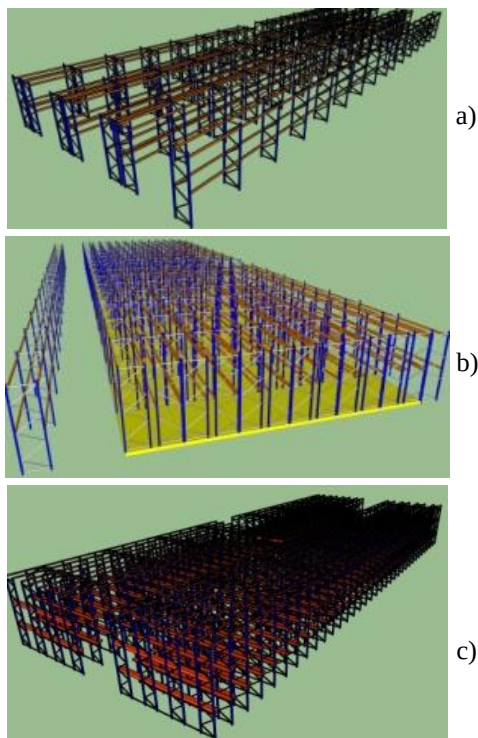
3. PRIMENA SIMULACIJA ZA PLANIRANJE I OPTIMIZACIJU SKLADIŠTA KOMPANIJE

U skladišnom objektu dimenzija 79.8 x 60.4 m, skladišti se roba široke potrošnje, gde bi se u jednom delu od 1000 m² trebalo smestiti piće. Roba bi bila primljena u skladište na standardnim Euro paletama, ukupne težine 1000 kg. Postavlja se pitanje da li će investitor, u zavisnosti od količine robe koju očekuje, investirati samo u opremu koja će biti postavljena u tom delu skladišta, ili će morati proširiti objekat.

Koristeći program SketchUp, modelirana su sva tri moguća rešenja, kako bi se na postojećoj kvadraturi rezervisanoj za skladištenje pića moglo napraviti poređenje svih karakteristika koje su od značaja za donošenje odluke. Pri tome je veoma važno da se maksimalno prilagodi zahtevima klijenata i da krajnji odabir bude optimalan po pitanju funkcionalnosti i ekonomičnosti.

Na slici 1 su prikazani modeli razmatranih mogućih rešenja skladišnog objekta.

U tabeli 1 su dati podaci o potencijalnom broju paletnih mesta za svaki tip regala, kao i budžetska analiza koja je dobijena na osnovu karakteristika i specifikacija sve tri razmatrane varijante.



Slika 1. Prikaz regala modeliranih u SketchUp programu, a) klasični, b) pokretni, c) prolazni regali

Tabela 1. Broj paletnih mesta za svaki tip regala i odgovarajuća budžetska analiza

Tip regala	Konvencionalni	Pokretni	Prolazni
Ukupan broj paletnih mesta	1155	2478	1980
Budžetska analiza	23.100 €	178.200 €	97.600 €

Da bi se došlo do modela koji će dati potrebne informacije za odabir skladišnog sistema, moralo se poći od podataka skladišta koji će nam biti od značaja pri samom pristupanju simulacijama.

U skladištu se nalazi oko 10 000 artikala, čiji broj varira u odnosu na to da li su neki proizvodi sezonski ili ne, da li je za neke od njih povećanja potražnja u određenom delu dana, nedelje, meseca, godine itd. Količina i vrsta robe takođe zavisi od pomenutih uticaja, kao i frekvencija ulaska i izlaska robe iz skladišta.

Što se kapaciteta skladišta tiče, ono sadrži 1778 paletnih mesta nosivosti paleta 1000 kg, 440 paletnih mesta sa nosivošću paleta od 500 kg, 1206 m² komisionih nivoa nosivosti 200 kg/paletnom mestu i 587 komisionih paleta na podnom delu. Objekat sadrži 6 ulaznih rampi, od koji se 3 koriste po potrebi i 2 izlazne (standardne). Od transportne opreme za manipulisanje paletama poseduju 7 paletnih kolica i 2 čeonu viljuškara.

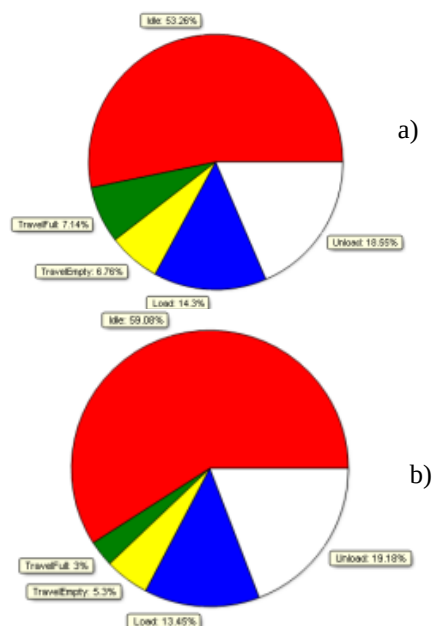
Konstrukcija regala sva tri slučaja je modelirana tako da se prilagodi prvenstveno zahtevu površine i zapremine koja je na raspolaganju u tom delu objekta. Dalje, formirane su tabele sa svim karakteristikama i specifikacijama regala, budžetskim analizama, radi sagledavanja prednosti i mana svakog koncepta u cilju dobijanja optimalnog rešenja ovog projekta.

4. SIMULACIONI MODELI I REZULTATI SIMULACIJE PREDLOŽENIH REŠENJA

Na slici 2 prikazan je model skladišta sa klasičnim regalima, a odmah ispod i rezultati koji su od značaja. Napomena: simulacije su sprovedene za 8h rada, odnosno jednu radnu smenu.



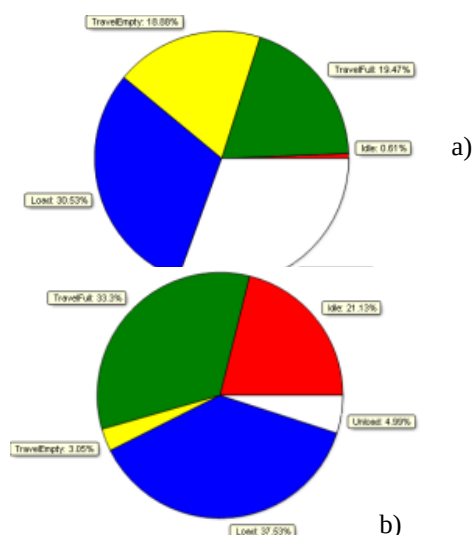
Slika 2. Prikaz simulacionog modela u softverskom paketu Enterprise Dynamics



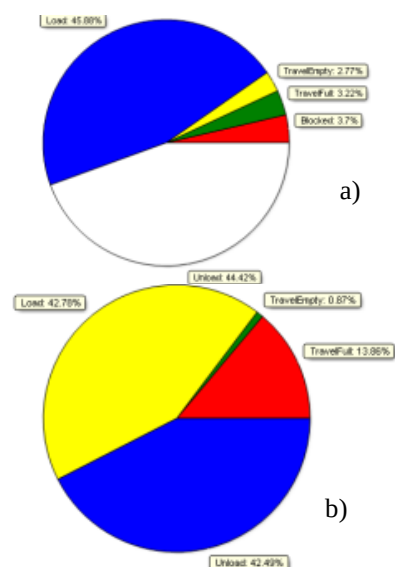
Slika 3. Grafički prikaz rada viljuškara a) koji opslužuje 6 ulaznih rampi, b) koji vrši uskladištenje paleta u sistem

Rezultati simulacija ukazuju na sledeće:

- viljuškar koji se nalazi na ulazu u skladište u toku radnog vremena je iskorišćen svega 47%, ali je potreban, da bi se smanjilo vreme čekanja na utovar/istovar,
- viljuškari za uskladištenje robe su nedovoljno iskorišćeni, svega 40%, iz razloga što rade tek kada je roba pristigla u ulaznu zonu,
- 7 paletnih kolica koja komisioniraju i otpremaju robu iz skladišta imaju zadovoljavajuću iskorišćenost, oko 80%, tokom radnog vremena,
- viljuškari za dopunu komisionog dela regala su maksimalno iskorišćeni, što ukazuje da frekvencija ulaza i izlaza robe u potpunosti opravdava korišćenje dva viljuškara za dopunu robe.

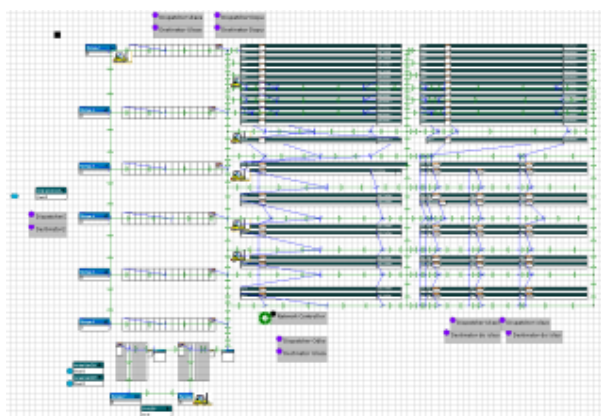


Slika 4. Grafički prikaz rada viljuškara a) koji vrši spuštanje paleta u komisioni deo regala i b) koji vrši komisioniranje robe

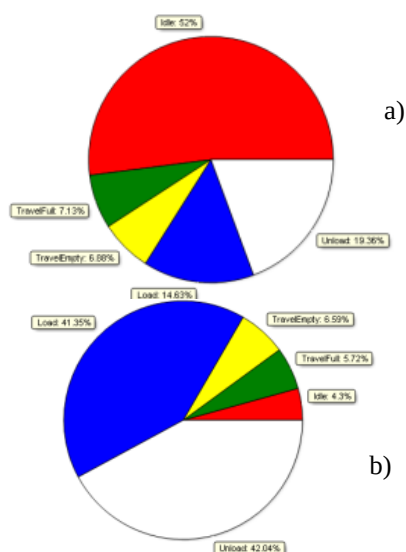


Slika 7. Grafički prikaz rada viljuškara a) koji vrši spuštanje paleta u komisioni deo regala i b) koji vrši komisioniranje robe

Na slici 5 prikazan je simulacioni model sa pokretnim regalima.



Slika 5. Prikaz simulacionog modela sa pokretnim regalima u softverskom paketu Enterprise Dynamics



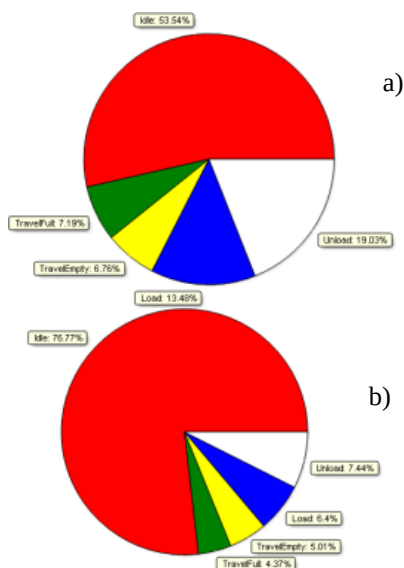
Slika 6. Grafički prikaz rada viljuškara a) koji opslužuje 6 ulaznih rampi, b) koji vrši uskladištenje paleta u sistem



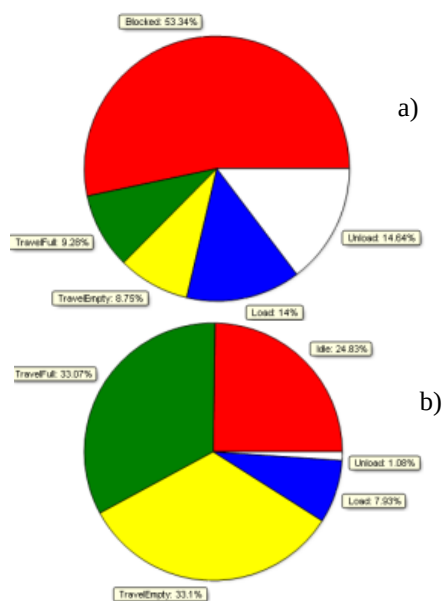
Slika 8. Prikaz simulacionog modela sa prolaznim regalima u softverskom paketu Enterprise Dynamics

S obzirom na tehnologiju skladištenja primenom prolaznih regala, u kojoj nema standardnih komisionih pozicija a samim tim ni potrebe za njihovom dopunom, viljuškari koji su u prethodnim modelima vršili dopunu, u ovom modelu su prenamenjeni za uskladištenje robe. Na taj način udvostručen je kapacitet opreme za uskladištenje robe u odnosu na prethodne modele. Rezultati simulacija

sa prolaznim regalima, nam pokazuju značajno veću iskorišćenost viljuškara na ulaznoj/izlaznoj zoni u odnosu na prvi model, jedan viljuškar 60% a drugi 25%, odnosno kumulativno 85%. To je prvenstveno usled dužih ciklusa rada nego u prvom slučaju, a kraće nego u drugom. Paletna kolica za komisioniranje imaju približno istu iskorišćenost.



Slika 9. Grafički prikaz rada viljuškara a) koji opslužuje 6 ulaznih rampi, b) koji vrši uskladištenje paleta u sistem



Slika 10. Grafički prikaz rada viljuškara a) koji vrši spuštanje paleta u komisioni deo regala i b) koji vrši komisioniranje robe

Napomena: Zbog identičnosti zahteva rada transportne opreme u skladištu i samim tim sličnosti dobijenih rezultata, prikazani su rezultati samo za pojedinu manipulativnu mehanizaciju koji su dovoljni da bi sestvorila celokupna slika o sistemu.

5. ZAKLJUČAK

Skladišni sistemi su integralni deo svakog toka materijala i stvaraju velik trošak koji značajno utiče na cenu proizvoda.

Iz tog razloga, treba uzeti u obzir uticajne faktore i korišćenjem programa za simulacije doći do mogućeg rešenja. Takođe, ne sme se zaboraviti to da su simulacije samo alat koji omogućuje brže dobijanje povratnih informacija, tako da se mora voditi računa o načinu unošenju podataka i njihovoj validnosti, kako bi na izlazu dobili što tačnije rezultate.

Na osnovu karakteristika svakog tipa regala, budžetske analize i rezultata simulacije, može se zaključiti da bi kapacitet u razmatranom skladištu sa konvencionalnim regalima bio najmanji, sa 1155 paletnih mesta i ukupnom cenom koštanja 23.100 €, dok je kod prolaznih povećan za 825 paletnih mesta sa porastom cene za 74.500 €, a kod pokretnih za 1323 paletna mesta i uvećanom cenom za čak 155.100 €. Istovremeno, iskorišćenost transportne opreme je znatno veća usled znatno dužih ciklusa rada ove opreme.

Pokretni i prolazni regali omogućavaju veći broj skladišnih mesta na istom prostoru, ali zahtevaju veće investicione troškove i mnogo veće angažovanje manipulativne opreme, što stvara dodatne troškove kako za samu opremu tako i za radnu snagu. Na osnovu toga, može se zaključiti da jedino u slučaju značajnog porasta zahteva za uskladištenjem dolazi do opravdanosti dodatnih investicija u pokretne ili prolazne regale. Investitor je taj koji bira i shodno njegovom očekivanju vezano za porast tokova roba u skladištu i mogućnosti ulaganja proizilazi odluka o izboru skladišnog sistema. U ovom konkretnom slučaju izabrano je da se instaliraju konvencionalni regali pošto se u skorije vreme ne očekuje porast robnih tokova u tolikoj meri koji bi opravdao značajno veće troškove koji nastaju u slučaju druge dve razmatrane opcije. Zaključeno je da druge dve opcije pružaju porast kapaciteta koji u ovom slučaju ne bi bili dovoljno iskorišćeni.

6. LITERATURA

- [1] M. Georgijević, „Tehnička logistika”, Zadužbina Andrejević, 2011.
- [2] B. Davidović, „Unutrašnji transport”, Visoka tehnička škola strukovnih studija, Kragujevac, 2012.

Kratka biografija:



Dragana Radaković, rođena u Kninu, Republika Hrvatska, 1994. godine. Zvanje diplomiranog inženjera stekla je 2017. godine. Master rad na Fakultetu tehničkih nauka iz oblasti mašinstva odbranila 2018. godine.



Sanja Bojić, rođena je 1981. u Karlovcu, Republika Hrvatska. Doktorirala je na Fakultetu tehničkih nauka 2013. god., a od 2014. je u zvanju docenta. Oblast njenog interesovanja su logistika i simulacije.

ПРОЈЕКТОВАЊЕ И РЕГУЛАЦИЈА РАДА ТРАКАСТОГ ДОЗАТОРА**DESIGN AND WORK REGULATION OF THE BELT FEEDER**Милорад Кузманчев, Драган Живанић, *Факултет техничких наука, Нови Сад***Област – МАШИНСТВО**

Кратак садржај – У овом раду је приказана неопходна контролно управљачка опрема потребна за регулисан рад и мерења на тракастом дозатору. Неопходне електро компоненте су имплементирани на постојећи специјални тракасти транспортер тако да су формирале једну целину функционалног тракастог дозатора. Циљ је био да се добије функционалан тракасти дозатор који би послужио за практична мерења, лабораторијске вежбе и као практичан уређај за даља испитивања.

Кључне речи: тракасти дозатор, тракасти транспортер, регулација рада

Abstract – In this paper it is shown the essential control equipment necessary for the regulated work and measurement on the belt feeder. Necessary electro components are implemented to the existing special conveyor belt, therefore they have formed one whole of the functional belt feeder. The aim was to obtain a functional belt feeder which would serve for practical measurement, laboratory exercises and as practical device for further testing.

Keywords: belt feeder, conveyor belt, work regulation

1. УВОД

Дозирање је саставни део сваког процеса код савремене индустријске производње. Назив уређаја или машине за дозирање, је углавном дефинисан техничким решењем којим се остварује транспорт у току рада, па тако постоје проточни, тракасти, вибрациони и други дозирни уређаји.

Тракасти дозатор омогућава транспортовање жељене количине материјала унутар задатог временског периода, односно постизање жељеног протока материјала. Састоји се обично од кратког тракастог транспортера са механизмом за мерење масе (вага) и регулационих органа, при чему исти континуално и аутоматски мери проток расутог материјала који пролази преко транспортне траке, без прекида тока материјала [3].

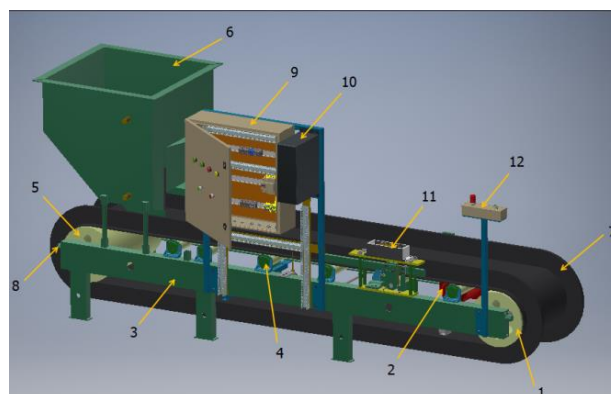
Осим мерења масе материјала неопходно је истовремено мерење и брзине траке транспортера.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Драган Живанић.

2. КОНСТРУКЦИЈА ТРАКАСТОГ ДОЗАТОРА

Тракасти дозатор, слика 1, чине два бубња, од којих је бубањ (1) погонски који преноси обимну силу на траку путем трења, а бубањ (5) затезни. Бескрајна трака (7) која прелази преко њих представља вучни и носећи елемент транспортера. Да би се створили услови за пренос обимне силе путем трења, трака мора да буде затегнута, што се остварује преко затезних навојних вретена (8) који су приказани на затезном бубњу. Погонски бубањ добија кретање од моторедуктора (2) посредством ланчастог преносника. Трака се на горњој (радној) страни ослања на ваљчане слојеве (4) чиме се постиже потребан положај, односно спречава превелики угиб траке. Утовар материјала је на задњем крају траке преко усипног коша (6) са шибером, са којим је могуће регулисати количину материјала која се утовара на траку [1]. Мерна вага служи за мерење транспортованог материјала по траци, а састоји се од ваљчаног слоја, који се са механизмом у виду клацкалице повезује са мерном ћелијом. У командном орману лица места (9) смештени су следећи елементи: PLC, релеји, моторна заштита, тастери, сигнализација. Поред ормара је постављен фреквентни регулатор (10). Сви ови елементи су постављени на носећу конструкцију тракастог дозатора (3) која је ослоњена на подлогу. Трака транспортера на својим крајевима, целом дужином има бочне стране у облику „завесица“, чиме се повећава површина попречног пресека материјала, односно капацитет транспортера.



Слика 1. Приказ основних елемената тракастог дозатора

2. ПРИНЦИП РАДА ТРАКАСТОГ ДОЗАТОРА

Притиском на NO тастер старт покреће се погон тракастог дозатора, при чему се укључује зелена сијалица, слика 2. Уколико је бункер празан,

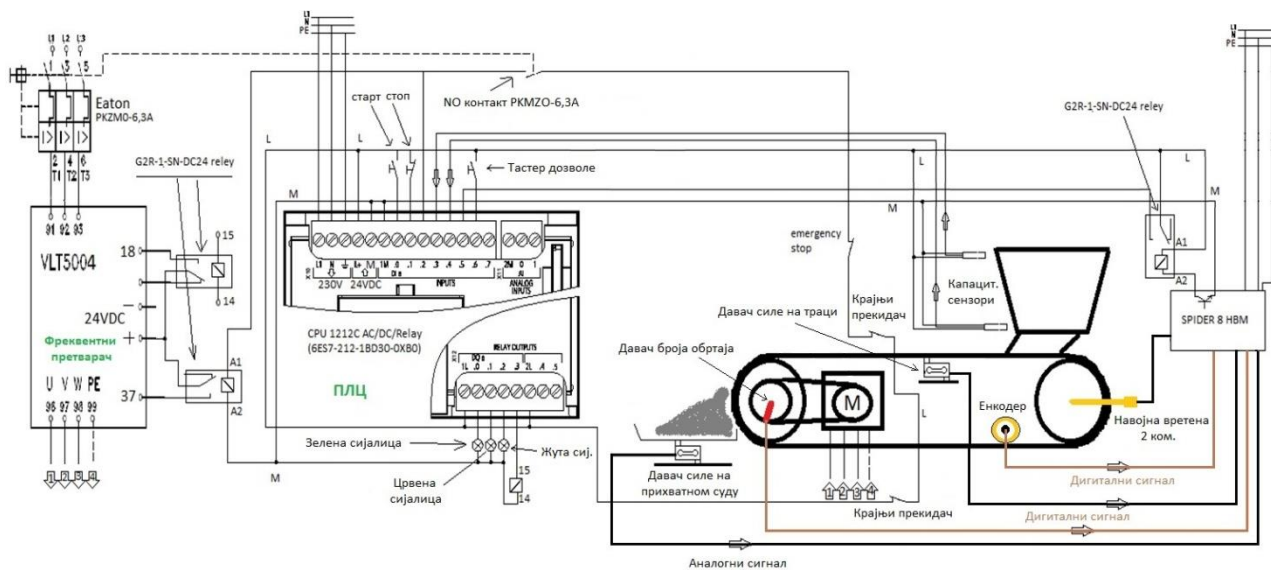
капацитивни сензор нивоа на бункеру, који је намештен при дну, детектује да нема материјала и на основу тога *PLC* укључује „додавач материјала” који је у корисничком програму и практичном делу задатка представљен жутом сијалицом (у правом погону то би била нека друга машина која би допремала материјал до дозатора или утоварни бункер, силос и сл.). Током утовара материјала долази до пуњења бункера и у једном тренутку материјал прекрива сензоре у бункеру, што даје сигнал *PLC* да искључи жуту сијалицу тј. замишљени „додавач материјала”.

Након што је бункер напуњен, потребно је да оператер који ради на тракастом дозатору притисне *NO* - тастер дозволе (тастер оператера). Након притиска тастера дозволе не значи да ће мотор траке почети да ради, јер то зависи од више услова: први услов је да допремач материјала у бункер не сме да ради, што је испуњено у овом случају. Други услов је да у прихватном суду нема материјала. Трећи услов тиче се безбедности оператера и опреме - стога је неопходно да није притиснут (*NC*) печуркасти тастер (*emergency stop*), да крајњи прекидачи који детектују евентуално бочно померање траке буду у нормалном положају (*NC*) и да је помоћни контакт моторног

заштитног прекидача затворен. Уколико су сви ови услови испуњени и оператер притисне тастер дозволе, мотор траке почиње да ради, а материјал се премешта из бункера у прихватни суд.

У том тренутку наступа аквизициони мерни уређај *SPIDER 8 HBM* на коме се задаје количина материјала која се жели претранспортовати - проток. Након доспећа материјала на траку, проточна вага, тј. мерна ћелија мери тежину материјала, енкодер мери брзину обртања траке, и из та два улазна сигнала *SPIDER* прерачунава проток. Када се претранспортује задата количина материјала, *SPIDER* активира релеј који даје сигнал да *PLC* искључи мотор траке. На прихватном суду је постављен давач силе, који шаље аналогни сигнал, на основу ког *SPIDER* мери да ли је претранспортована задата количина материјала, што ће имати велику улогу у калибрацији проточне ваге.

SPIDER 8 је, такође, задужен да обради сигнале са осталих сензора. На њега је повезан бесконтактни давач броја обртаја бубња и два навојна вретена, на којима су налепљене мерне траке и као таква целина чине мерну ћелију која мери силу затезања траке. Све те податке *SPIDER 8* помоћу *HBM* софтвера *CATMEN* приказује на рачунару.



Слика 2. Шематски приказ повезивања свих елемената на тракастом дозатору

3. ОДАБИР ОПРЕМЕ

3.1. Фреквентни претварач

Фреквентни регулатори, слика 3, су уређаји који управљају брзином обртања асинхроних и синхроних електро мотора, при чему је потребно мењати фреквенцију и напон, да би постигли жељени број обртаја. Данас у свим аутоматизованим погонима стандардно се користи трофазни мотор са фреквентним регулатором, па је сходно тој чињеници кључно одабрати адекватан фреквентни регулатор. Они се бирају на основу оптерећења, у складу са функцијом коју обављају и осталим техничко-економским параметрима. У овом случају, погон тракастог дозатора је нисконапонски трофазни кавезни асинхрони мотор (Север, каталoшки број

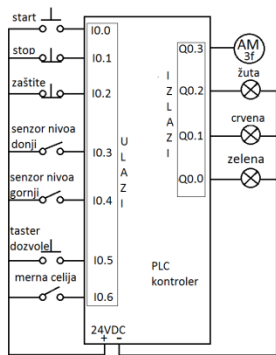
ZK80 B-4) номиналне снаге 0,75 kW, преко редуктора који је са погонским бубњем повезан ланчаницима и ланцем. За његову регулацију употребљен је претварач *Danfoss VLT 5000 Series-type 5004* номиналне снаге од 2,2 kW.



Слика 3. Конфигурација система управљања

3.2. Програмабилни логички контролер PLC

PLC је дигитални електронски уређај који користи програмабилну меморију за памћење наредби којима се захтева извођење специфичних функција, као што су логичке функције, секвенцирање, пребројавање, мерење времена, израчунавање у циљу управљања различитим машинама и процесима [2]. Након што је дефинисана и описана опрема погона тракастог дозатора и након што је установљен начин рада, могло се приступити избору контролера и начину аутоматизације која ће бити реализована помоћу контролера *SIMATIC S7 PLC* фамилије. Овде је приказан упрошћен модел пројектног задатка, слика 4, који ће програмеру дати потпуну слику о томе шта је повезано на улазе и излазе PLC. На основу тога је лакше схватити шта треба да се дефинише у лествичастом дијаграму, како би PLC на прави начин вршио своју функцију и одговорио на све захтеве приликом управљања поменутиим погоном.



Слика 4. Упрошћен модел пројектног задатка

3.3. Аквизациони мерни уређај SPIDER – 8 HBM

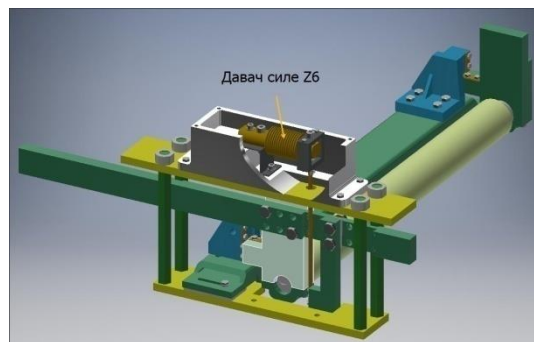
Аквизациони мерни уређај *SPIDER 8*, слика 5, је више-канална електронска јединица за паралелно мерење динамичких величина помоћу рачунара. Једноставно се повезује са рачунаром преко прикључка за штампач и брзо припрема за мерење. Сваки канал обезбеђује побуду за пасивне четворополне претвараче, филтере и властити претварач. Мали софтверски пакет, који је инсталиран у систем *Spider 8*, ради под оперативним системом *MS Windows* и омогућава аквизицију података (континуалну или периодичну) и њихово снимање у датотеке. HBM је развио и софтверски пакет *CATMAN* који омогућава произвољно дефинисање начина на који ће се приказати резултати мерења, потпуну (тренутну и накнадну) анализу резултата, произвољну интеракцију између корисника и софтвера, контролу и помоћ при мерењу.



Слика 5. SPIDER 8 HBM

3.4. Давачи силе

Давачи силе служе за мерење механичких вредности нпр. силе преводећи је у електронски облик (напон). Састоје се од мерних трака које су чврсто залепљене за опружни елемент, најчешће четири које су повезане у Витстонов мост. На тракастом дозатору су примењена четири давача силе. Један је постављен на мерној ваги и мери силу материјала на траци, слика 6. Други је постављен испод прихватног суда и мери тежину материјала, слика 7. Трећи и четврти давач су исти, направљени су дорадом навојних вретена за затезање бубња и мере силу у траци, слика 8.



Слика 6. Давач силе постављен на мерној ваги



Слика 7. Давач силе постављен испод прихватног суда



Слика 8. Давач силе на затезном бубњу

3.5. Остала опрема

Да би тракасти дозатор у потпуности функционисао и био безбедан за употребу, потребна је додатна помоћна опрема. Ту спадају тастери за укључивање-искључивање, светлосна сигнализација, тастер за хитно искључење (*emergency stop buttons*) и крајњи прекидачи који детектују бочно померање траке са бубња. Сви они, уколико дође до њихове активације, доводе до прекида рада тракастог дозатора.

За заштиту електро мотора и фреквентног претварача од струјног преоптерећења користи се моторни заштитни прекидач *PKZM0*. Присутни су капацитивни сензори за мерење нивоа материјала у бункеру и релеји за галванско раздвајање комуникације између *PLC-a* и *SPIDER-a*. На повратној страни траке постављен је ротациони енкодер, слика 9, који се

креће по унутрашњем делу траке [4]. Упоређивањем података о брзини погонског бубња и брзини траке може се контролисати и евентуално детектовати проклизавање траке на погонском бубњу.



Слика 9. Изглед ротационог енкодера

5. МЕТОДОЛОГИЈА ИСПИТИВАЊА

Циљ испитивања је утврђивање тачности мерног уређаја, калибрација или баждарење. Пре пуштања у рад тракастог дозатора, неопходно је проверити све електро спојеве визуелно и мерним инструментом. Потребно је извршити правилно затезање траке како би се њена оса покlopила са подужном осом транспортера. Прво стартовање транспортера се врши са минималним бројем обртаја електро мотора, уз контролу положаја траке на бубњевима како би се уочила евентуална могућност бочног померања траке током кретања. Неопходно је проверити функционалност сигурносне опреме, тастера сигурности и бочних искључивача. Сви мерни сензори се повезују на Spider 8, након чега се врши њихова калибрација - мерна вага је калибрисана помоћу ланца, а мерни сандук помоћу тегова који су еталонирани. По потреби се додатно коригује затегнутост траке, све док давачи силе не покажу исту вредност затегнутости и док трака током рада не постигне правилан положај. Тек тада је могуће приступити мерењу, а у конкретном случају за пробно испитивање је коришћена житарица јечам. Прво је сниман сигнал празне траке у раду са сензора на мерној ваги, а након тога је из утоварног бункера утоваран јечам на траку тако да се на мерној ваги мерила маса претранспортованог јечма и саме траке.



Слика 10. Пуштање тракастог дозатора у рад

Са друге стране, да би се проверила тачност мерења на мерној ваги, вршено је мерење количине материјала који је доспевао у прихватни суд. Одзимањем укупне измерене масе материјала и траке и измерене масе траке се добила укупна количина материјала који је претранспортован и тај податак је упоређен са измереном укупном масом на прихватном суду.

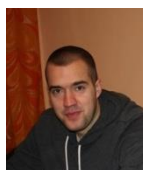
6. ЗАКЉУЧАК

У овом раду је посебна пажња била посвећена начину аутоматизације и регулације рада тракастог дозатора. Рад је имао више фаза. У првој фази је израђен 3D модел целог транспортера, предвиђена су места поставке и уградње свих неопходних компоненти. Затим је било потребно дефинисати алгоритам управљања, а потом се приступило практичном раду где је комплетан уређај са свим уграђеним компонентама требало довести у функционално стање. Током целокупног пројектовања наилазило се на низ проблема који су се решавали у ходу што овакви радови управо и захтевају. Искоришћена је сва постојећа опрема која је била доступна и употребљива, набављена је остала потребна опрема, при чему се због ограничених материјалних средстава тежило што једноставнијим и јефтинијим решењима, која су задовољила све захтеве које је требало испоштовати.

6. ЛИТЕРАТУРА

- [1] Ј. Владић, "Непрекидни и аутоматизовани транспорт", ФТН, Нови Сад 1991.
- [2] Д. Марчећ, "Програмабилни логички контролери и комуникациони протоколи у електроенергетици", ФТН, Нови Сад 2014.
- [3] В. Васић, "Систематизација машина за паковање, транспортни системи и испитивање тракастог дозатора", Мастер рад, ФТН, Нови Сад 1999
- [4] http://www.usas.no/index.php?page=shop.getfile&file_id=25&product_id=38&option=com_virtuemart&Item_id=2 (приступљено у јуну 2018.)

Кратка биографија:



Милорад Кузманчев је рођен у Новом Саду 1989. год. Дипломски рад на Факултету техничких наука из области Машинства – Мотори СУС одбранио је 2015.год.

контакт: miloradkuzz@gmail.com



Драган Живанић је рођен 1972. год. у Сремској Митровици. У звању доцента на ФТН-у у Новом Саду од 2014. год.

UPOREDNA ANALIZA SISTEMA ZA PRIPREMU SANITARNE TOPLE POTROŠNE VODE U HOTELIMA**COMPARATIVE ANALYSIS OF SYSTEMS FOR HEATING OF DOMESTIC HOT WATER IN HOTELS**

Danijel Todorović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – U radu su prikazane teorijske osnove sistema za pripremu sanitarne tople potrošne vode (STPV), sa fokusom na sisteme koji su primjenjivi u hotelima. Iznijete teorijske osnove su korištene za izradu rješenja sa konvencionalnim i obnovljivim izvorima energije i izvršena je tehno-ekonomska analiza gdje jevršeno upoređivanje sistema sa konvencionalnim izvorom energije i sistema sa obnovljivim izvorima energije.

Ključne reči: Sanitarne topla potrošna voda,, Izvor energije, Analiza

Abstract – This paper describes theoretical basis of domestic hot water (DHW) systems, focusing to the systems that are apliable for hotel solutions. The presented theoretical foundations have been applied for conceptual design of DHW system with conventional energy source and systems with renewable energy sources and comparative techno-economic analysis of the systems.

Keywords: Domestic hot water, Power source systems, Analysis

1. UVOD

Hoteli se ubrajaju u neke od najvećih potrošača energije u svijetu i za objekte koji imaju jako veliki uticaj na životnu sredinu iz razloga što uvijek moraju da imaju određenu količinu finalne energije spremene za korištenje, pored toga hoteli su jedan od osnovnih elemenata turizma kao privredne grane. S' obzirom da su hoteli nekretnine prije svega, za povećanje kapaciteta, samim tim poboljšanje poslovnih rezultata su potrebne jako visoke investicije, tako da je prihvatljiviji smjer smanjenje operativnih troškova. Troškovi za energiju mogu da dostignu 50% ukupnih troškova, pa se upravo u tom dijelu prvo prave uštede. Prema [1], prosječna popunjenost kapacitea hotela u južnoj i istočnoj Evropi je 65-70% na godišnjem nivou, pa je kao referentna vrijednost za analizu uzeta popunjenost od 68%.

2. ZADATAK RADA

Zadatak master rada je da se prikaže sistem sa konvencionalnim izvorom energije i dva sistema sa obnovljivim izvorima energije, izradi idejno rješenje za

sva tri sistema, dimenzionišu elementi sistema i da se izvrši uporedna tehno-ekonomska analiza sistema.

Prema iznijetim teorijskim osnovama izvršen je proračun elemenata svakog sistema, potrošnje energije u eksploatacionom periodu za hotel u Novom Sadu.

Izrada idejnog projekta (rješenja) za potrebe finansijske analize podrazumijeva pravilan odabir osnovnih elemenata poput izvora energije, akumulacionih rezervoara, izmjenjivača toplote i cirkulacionih pumpi. Pored mašinskih elemenata potrebno je predvidjeti i način automatskog upravljanja.

3. PODJELA SISTEMA ZA PRIPREMU STPV I HIGIJENSKI ZAHTEJEVI

Podjela sistema za pripremu tople vode može da se izvrši prema vrsti izvora toplote, načinu zagrijavanja vode, broju mjesta za uzimanje potrošne vode, prema pritisku vode u izmjenjivaču toplote, prema sistemu zagrijavanja i prema veličini postrojenja, s' tim da se u praksi za sisteme za pripremu STPV naječešće uzima u obzir prema izvoru toplote:

- kotlovi na gas, čvrsto gorivo, tečno gorivo...
- solarni kolektori za zagrijavanje vode
- toplotne pumpe

Druga podjela koja se često koristi za odabir je način (sistem) zagrijavanja:

- Akumulacioni sistemi kod kojih se potrebna količina vode zagrijava u rezervoarima
- Protočni sistem kod kojih se zagrijava samo količina vode koja je potrebna u tom trenutku
- Kombinovani sistem

Prilikom dimenzionisanja sistema za pripremu STPV, posebno izvora toplote, potrebno je uzeti u obzir probleme koji se mogu pojaviti. Jedan od najvećih je svakako pojava bakterije legionele koja izaziva legionarsku bolest.

Legionela se razvija u stajaćoj vodi pri temperaturama 32-42°C, a uništava se na temperaturi 60-65°C, tako da je potrebno predvidjeti kapacitet za zagrijavanje vode na temperaturu 60°C ili izvor koji može da dogrije vodu do 65°C u periodu od par sati nedeljno.

4. ELEMENTI SISTEMA

Osnovni elementi sistema za pripremu STPV primjenjenim u hotelima su:

NAPOMENA:

Ovaj rad proistekao je iz master rada, čiji mentor je bio prof. Aleksandar S. Andelković.

- Izvor energije (toplote)

Izvor toplote koji može direktno ili indirektno da zagrijava STPV. Izvori koji direktno zagrijavaju vodu moraju biti izrađeni od nerđajućeg čelika.

- Rezervoar za STPV

Rezervoar za skumulaciju STPV su izrađeni do nerđajućeg čelika ili običnog čelika, ali su iznutra obloženi staklastim materijalima kako voda ne bi bila u dodiru sa čelikom. Mogu da budu bez izmjenjivača ili da sadrže cjevaste izmjenjivače toplote, jedan ili dva, koji se koriste za zagrijavanje STPV tehničkom vodom. Dimenzionisanje rezervoara STPV se vrši na osnovu dnevnih potreba i vremena zagrijavanja vode.

- Rezervoari za tehničku vodu

Rezervoari za tehničku vodu mogu biti deo sistema koji su dimenzionisani i po protočnom i po akumulacionom sistemu, zavisno od snage izvora energije.

- Izmjenjivači toplote

Za indirektno zagrijavanje sanitarne vode se najčešće koriste pločasti izmjenjivači toplote koji se biraju za suprotno-smjerno strujanje fluida. Mogu da budu lemljeni ili rastavljeni, zavisno od tipa radnog fluida koji se koristi za zagrijavanje STPV.

-Cirkulacione pumpe

Cirkulacione pumpe se ugrađuju uvijek na strani STPV, odnosno sekundarnoj strani sistema, a radna kola i kućišta moraju biti izrađena od nerđajućeg čelika ili mesinga, dok se sa primarne strane ugrađuju u standardnoj izvedbi.

- Cijevna mreža i cijevna armatura

Cijevna mreža treba da bude izrađena od materijala pogodnih za STPV poput bakra, nerđajućeg čelika ili eventulano plastike koja može da zadrži karakteristike pri temperaturama od 60°C. Kompletne cijevne armature poput hvatača nečistoće, zaustavnih i (automatskih) balansnih ventila treba da bude izrađena od mesinga ili nerđajućeg čelika. Ekspanzione posude treba da budu nazivnog pritiska 10bara.

5. PRORAČUN POTREBNE KOLIČINE VODE I ELEMENATA SVAKOG SISTEMA

5.1 Količina STPV i zapremina rezervoara za STPV

Na osnovu preporuka iz [2] za dnevnu količinu STPV za ugostiteljske objekte prikazanih u tabeli 5.1 i na osnovu arhitektonsko-građevinskih osnova izvršen je proračun potrebne količine STPV na dnevnom nivou za 100% popunjenosti hotela. Količine STPV po hotelskoj jedinici i ukupna količina STPV na dnevnom nivou su prikazani u tabelama 5.2 i 5.3.

Tabela 5.1: Preporučene dnevne količine STPV [2]

Potrošno mjesto	Litara na dan po sobi	
	60°C	45°C
Restorani po gostu	4-20	12-30
Hoteli		
Sobe sa kupatilom	100-150	140-220
Sobe sa tušem	50-100	70-120

Tabela 5.2: Dnevne potrebe STPV za kafe i restoran

Prostorija	Broj ljudi	Dnevna količina STPV po osobi na 60°C	Ukupna količina STPV dnevno na 60°C
Restoran	76	20	1520
Cafe-restoran	60	8	600

Tabela 5.3: Dnevne potrebe STPV za sobe i apartmane

Prostorija	Broj prostorija	Dnevna količina STPV po prostoriji na 60°C	Ukupna Količina STPV dnevno na 60°C
Dvokrevetne sobe sa tuš kabinom u kupatilu	144	100	14400
Apartmani sa katom u kupatilima	9	130	1170

Ukupno potrebna količina STPV dnevno pri popunjenosti kapaciteta od 100% na temperaturi 60°C je 17690l.

Potrebna zapremina rezervoara pri prosječnoj zauzetosti hotela na godišnjem nivou od 68%:

$$V_s = \frac{C}{c \cdot (t_0 - t_u)} \cdot b = \frac{1000 \cdot C}{1,16 \cdot 35} \cdot b = 24,5 \cdot C \cdot b \quad (1)$$

C - potrebna akumulaciona energija;

b- faktor dodatke zbog mrtvog prostora ispod površine grijača, b=1,1-1,2;

$$C = \dot{m} \cdot c_p \cdot \Delta t \quad (2)$$

$\dot{m}$ - maseni protok vode za 68% potrebnog kapaciteta [kg/s];

$c_p = 4,14 \text{ kJ/kg}$ specifični toplotni kapacitet vode za piće na 25°C;

$\Delta t = 60 - 25 = 35 \text{ K}$ - razlika u temperaturama vode prije ulaza u grijač i posle grijača [K];

$$\dot{m} = V_{STPV/dan} / 3600 \cdot 0,68 \cdot \rho_{H3O} \quad (3)$$

$V_{STPV/dan} = 17690 \text{ l}$ - potrebna količina STPV na dnevnom nivou pri 100% popunjenosti kapaciteta;

$\rho_{H2O} = 0,997 \text{ kg/dm}^3$ - gustina vode za piće na 25°C;

$$\dot{m} = 3,331 \text{ kg/s} \quad (4)$$

$$C = 482,66 \text{ kWh} \quad (5)$$

$$V_s = 13007,69 \text{ l} \quad (6)$$

Usvojena je ukupna zapremina rezervoara od 13000l. Zbog velike promjenjivosti potrebne količine STPV usvaja se čelijski tip konfiguracije rezervoara i to dva horizontalna od 5000l tip Storatherm Aquaload 5000 H i jedan vertikalni od 3000l tip Storatherm Aquaload 3000. Na osnovu dnevno potrebne energije za zagrijavanje STPV izvršen je proračun potrebne energije na godišnjem nivou uz gubitak toplote na cjevovodu i izmjenjivačima toplote od 5%:

$$Q_{god} = C \cdot 365 \quad (7)$$

$$Q_{god} = 176\,170,9 \text{ kWh} \quad (8)$$

$$Q_{uk} = 176\,170,9 \cdot 1,05 \quad (9)$$

$$Q_{uk} = 184\,979,45 \text{ kWh} \quad (10)$$

5.2 Sistem sa gasnim kotlovima

S obzirom da je kotao jedini izvor, sistem će biti dimenzionisan na 130% potrebnog kapaciteta za vrijeme zagrijavanja 2h i to tako da imamo dva kotla svaki po 65% potrebnog kapaciteta, kao zaštitu zbog moguće havarije na jednom od kotlova. Potreban kapacitet kotla:

$$\dot{Q}_K = \frac{\dot{Q}_{uk} \cdot z_B}{z_B + z_A} \quad (11)$$

$\dot{Q}_K$ - potreban kapacitet izvora energije;

z_A - broj časova zagrijavanja vode;

z_B - broj časova rada izvora energije;

$\dot{Q}_{uk}$ - potrebna toplota u kW za 100% kapaciteta;

$$\dot{Q}_{uk} = \dot{m} \cdot c_p \cdot \Delta t \quad (12)$$

$$\dot{Q}_{uk} = 754,6 \text{ kW} \quad (13)$$

Kapacitet jednog kotla:

$$\dot{Q}_K = \frac{\dot{Q} \cdot z_B}{z_B + z_A} \cdot 0,65 \quad (14)$$

$$z_A = z_B = 2h \quad (15)$$

$$\dot{Q}_K = 245,3 \text{ kW} \quad (16)$$

Izabran je kotao Gas 310 ECO Pro 285/5 nominalne snage 261kW pri temperaturnom režimu fluida 80/60°C i efikasnosti 88% na godišnjem nivou. Potrošnja gasa na godišnjem nivou za izabrani kotao;

$$B_{god} = \frac{3,6 \cdot 10^3 \cdot Q_{god}}{\eta_K \cdot H_d} \quad (17)$$

B_{god} - potrošnja goriva (energenta) u toku godine za zagrijavanje STPV [m³];

Q_{god} - potrebna energija za zagrijavanje STPV tokom cele godine sa prosečnom popunjenošću hotela 65% [Wh];

$\eta_K = 88\%$ - efikasnost kotla, uzima se iz dokumentacije;

$H_d = 33500 \text{ kJ/m}^3$ - donja toplotna moć prirodnog zemnog gasa;

$$B_{god} = 22589,1 \text{ m}^3 \quad (18)$$

Na osnovu predate količine toplote izmjenjivaču i rezervoarima i vremena rada kotlova od 2h dnevno, proračunato je vrijeme rada cirkulacionih pumpi, samim tim i potrošnja električne energije:

$$E_{aux} = (P_{cpK} + P_{cpSTPV}) \cdot n \cdot 365 \cdot 20 \quad (19)$$

E_{aux} - električna energija za cirkulacione pumpe u periodu eksploatacije od 20 godina;

P_{cpK} - ulazna snaga cirkulacione pumpe sa strane kotla [W];

P_{cpSTPV} - ulazna snaga cirkulacione pumpe sa STPV [W];

n - broj sati rada cirkulacionih pumpi dnevno;

$$E_{aux} = 17374 \text{ kWh} \quad (19)$$

Na osnovu cijene električne energije od 0,067eur/kWh i cijene gasa od 0,31 eur/m³, ukupni eksploatacioni troškovi sistema sa gasnim kotlovima kao jedinim

izvorom su 141216,53 eur, dok su investicioni troškovi za ceo sistema 55230,00 eur.

5.3 Sistem sa solarnim kolektorima

U softverskom programu T-sol je izvršena analiza sistema za zagrijavanje STPV sa solarnim kolektorima koji bi radili u bivalenciji sa gasnim kotlom kapaciteta 350kW, koji je dovoljan da zagreje količinu vode potrebnu pri popunjenosti kapaciteta od 100% za dva sata. Za jugo-zapadnu orijentaciju solarnih kolektora i na osnovu dostupne površine za smještanje, izvršena je simulacija u T-sol-u za potrebnu količinu STPV pri prosječnoj popunjenosti kapaciteta i za 37 solarnih kolektora. Rezultati simulacije su prikazani u tabeli 5.4:

Tabela 5.4: Rezultati simulacije izrađene u T-sol programu za sistem sa solarnim kolektorima

Instalisana snaga solarnih kolektora	65,01 kW
Instalisana bruto površina solarnih kolektora	92,87 m ²
Zračenje na aktivnu površinu solarnih kolektora	128,74 MWh
Energija solarnih kolektora	50,78 MWh
Neto energija solarnih kolektora dostavljena na izmjenjivač	45,45 MWh
Potrebna energija za zagrijavanje STPV	178 MWh
Doprinos solarnih kolektora ukupni potrebama za zagrijavanje sanitarne vode	44,06 MWh
Energija za zagrijavanje STPV od pomoćnog sistema	141,7 MWh
Solarna frakcija sistema	23,7 %

Na slici 5.1 je prikazana šema odabranog sistema:



Slika 5.1: Šema odabranog sistema

Na osnovu rezultata simulacije, pomoću izraza (17) proračunato je da je u toku godine potrebno je 17303,94 m³ gasa. Takođe na osnovu rezultata simulacije i obrazca (20) je proračunato vrijeme rada cirkulacionih pumpi, a zatim, potrošnja električne energije u eksploatacionom periodu, a samim tim i eksploatacioni troškovi. Eksploatacioni troškovi sistema sa solarnim kolektorima u predviđenom eksploatacionom periodu su 108843,43 eur, dok su investicioni troškovi sistema 78500eur.

Primjer proračuna vremena rada cirkulacione pumpe na godišnjem nivou na osnovu isporučene energije:

$$\tau = \frac{Q_{GODsolar}}{\dot{m} \cdot c_p \cdot \Delta t} \quad (20)$$

$Q_{GODsolar}$ - energija solara za zagrijavanje STPV na godišnjem nivou;

$\dot{m}$ - maseni protok rasoline sa strane kruga solarnih kolektora;

Δt - prosječna temperaturna razlika potisnog i povratnog voda;

5.4 Sistem sa toplotnim pumpama i rekuperacijom toplote iz sivih otpadnih voda - SOV

Na osnovu potrebne količine STPV na 60°C i prosječne temperature tople vode pri korištenju od 36,4°, prosječnih padavina u Novom Sadu u periodu od maja do septembra na površinu krova od 700m², proračunato je da u sistem svaki dan dolazi 28943,6 kg SOV pri popunjenosti kapaciteta 100 % i 1240 kg atmosferskih voda. Pretpostavljeno je minimalno vrijeme dotoka otpadnih voda od 2h, pa su izabrani rezervoari za STPV od 15000l. Na osnovu mjerenja vršenih na objektu u periodu od 11-17.7.2018. kao referentna temperatura je uzeta ona izmjerena 14.7.2018., 25,1°C, jer je mjerena u vrijeme jutarnjih pikova i posmatrana je promjena temperature od 2h. U tabeli 5.5 su prikazani rezultati očitavanja temperatura:

Tabela 5.5: Temperature SOV u period 11.-17.7.2018

Dan	Temperatura SOV	Popunjenost kapaciteta hotela
11.7.2018.	11.5 °C	33%
12.7.2018.	17.8 °C	62%
13.7.2018.	24.3 °C	88%
14.7.2018.	25.1 °C	86%
15.7.2018.	20.8 °C	88%
16.7.2018.	23.0 °C	62%

S' obzirom da je pretpostavljena temperaturna razlika potisnog i povratnog voda 5K, najniža temperatura SOV koja može da se koristi za odavanje toplote je 9°C. Na osnovu referentne i minimalne temperature SOV i prosječne temperaturne razlike potis/povrat proračunat je broj recirkulacija SOV i usvojene su tri recirkulacije što daje ukupan dnevni protok pri prosječnoj popunjenosti kapaciteta od 64200 l/h prosječnog temperaturnog režima 17/12°C. Na osnovu protoka i pretpostavljenog maksimalnog vremena rada od 16h, izabrane su dvije toplotne pumpe, svaka kapaciteta 17 kW koje bi radile u bivalenciji sa gasnim kotlom kapaciteta 350 kW. Ukupni investicioni troškovi sistema sa toplotnim pumpama kao i rekuperacijom toplote iz SOV su 62930eur, a eksploatacioni za procjenjeni eksploatacioni period 95855,27eur.

6. FINANSIJSKA ANALIZA

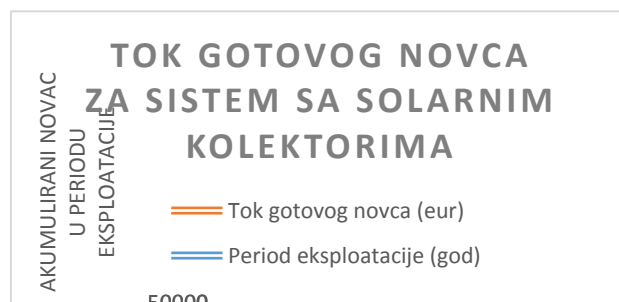
Na osnovu investicionih i eksploatacionih troškova sistema, izvršena je uporedna finansijska analiza gdje je vršena komparacija sistema sa obnovljivim izvorima energije u odnosu na sistem koji koristi konvencionalni izvor energije. U tabeli 6.1 su prikazani rezultati finansijske analize:

Tabela 6.1: Rezultati finansijske analize

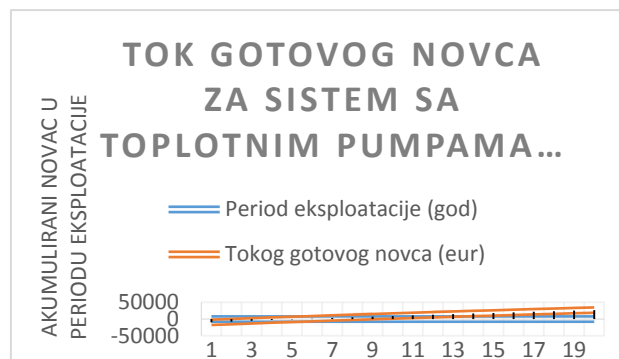
Sistem	B	POP	NPV
Solarni kolektori	32373,1	17	3907,75
Toplotne pumpe sa rekuperacijom iz SOV	46361,26	4,59	28221,49

B - ukupne uštede u troškovima eksploatacije za eksploatacioni period od 20 god [eur];

POP - dinamički period povrata novca [god];
NPV - neto sadašnja vrijednost novca [eur];



Dijagram 5.1: Period povrata investicije za sistema sa solarnim kolektorima



Dijagram 5. Period povrata investicije za sistema sa toplotnim pumpama i rekuperacijom toplote iz SOV

7. ZAKLJUČAK

U radu si iznijete teoretske osnove sistema za prpremu STPV uopšteno, sa posebnim osvrtom na sisteme pogodne za primjenu u hotelima, s' obzirom da se smatra da troškovi pripreme STPV učestvuju u ukupnim troškovima hotela za eneriju sa 25-50% zavisno od sistema. Na osnovu rezultata finansijske analize je primjetno da je sistem sa toplotnim pumpama najrentabilniji, dok se situacija kod sistem sa solarnim kolektorima obrnuta, uprkos visokim solarnim prinosima.

8. LITERATURA

- [1] www.statista.com
- [2] Grejanje i klimatizacija – Reknagel, Šprenger, Šramek, Čeperković
- [3] engineeringtoolbox.com
- [4] Valentine energije T sol

Kratka biografija:



Danijel Todorović rođen je u Tumarama, Lukavac, BIH 1992. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mašinstva - Termotehnika odbranio je 2018.god.



Aleksandar Andelković rođen je u Šapcu 1981. god. Doktorirao je 2015. god., a od 2015. je u zvanju docent. Oblast interesovanja su Nekonvencionalni sistemi grijanja i hlađenja.

UPOREDNA ANALIZA SISTEMA GRIJANJA TOPLOTNIM PUMPAMA I HIBRIDNIM SISTEMIMA**HEAT PUMP AND HYBRID HEATING SYSTEMS COMPARATIVE ANALYSIS**Dragan Lazić, *Fakultet tehničkih nauka, Novi Sad***Oblast - MAŠINSTVO**

Kratak sadržaj - U ovom radu izvršena je uporedna analiza mogućih rješenja različitih sistema grijanja toplotnim pumpama i hibridnim sistemima. Dato je objašnjenje procedure dimenzionisanja i izbora različitih dijelova sistema grijanja pomoću toplotnih pumpi sa različitim izvorima toplote. Izvršena je i tehnoe ekonomska analiza različitih sistema grijanja toplotnim pumpama. Na kraju rada date su i neke prednosti i mane poređenih sistema grijanja.

Ključne riječi: Grijanje, toplotne pumpe, izbor sistema

Abstract - In this paper, a comparative analysis of possible solutions for different heating systems with heat pumps and hybrid systems has been performed. Procedure of dimensioning and selection of different parts of the heating systems are also given. Techno-economic analysis of various heating systems by heat pumps with different heat sources was made. In conclusion, advantages and disadvantages of different heating systems were explained.

Keywords: Heating, heat pumps, System selection

1. UVOD

Kako bi se došlo do podatka o isplativosti nekog projekta, odnosno u ovom slučaju, sistema grijanja pomoću toplotne pumpe, potrebno je uporediti više različitih podataka, izvršiti odabir sistema, odrediti njegovu namjenu, izvršiti dimenzionisanje, izbor opreme i na kraju uporediti odabrani sistem sa nekim drugim konvencionalnim sistemom za grijanje. U ovom radu izvršena je analiza različitih sistema za grijanje i hibridnih sistema pomoću programskog paketa GeoTSOL [3]. U prvom dijelu rada, objašnjena je analiza različitih sistema grijanja u programskom paketu GeoTSOL.

Drugi dio rada se odnosi na tehnoe ekonomsku analizu različitih sistema grijanja sa toplotnim pumpama. Izvršeno je poređenje sistema sa vazdušnom toplotnom pumpom, toplotnom pumpom sa geotermalnim kolektorima, geotermalnim sondama, a na kraju je izvršena analiza hibridnog sistema za grijanje.

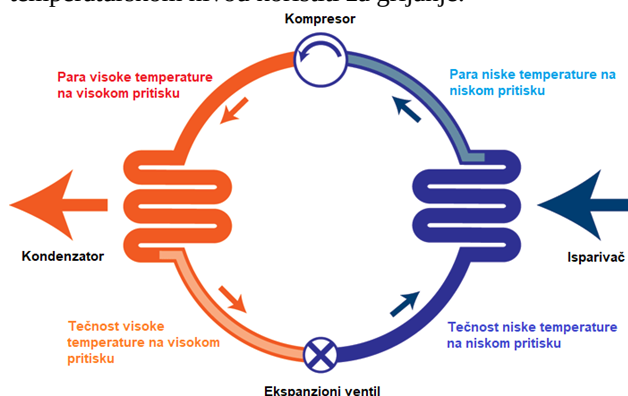
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Aleksandar Andelković.

Svi analizirani sistemi imaju dvojaku ulogu, odnosno, služe za grijanje i pripremu tople sanitarne vode. Nakon završene analize, izvršeno je poređenje bitnijih podataka dobijenih analizom. Na osnovu tih podataka, može se zaključiti da li postoji ekonomska opravdanost ugradnje sistema sa toplotnom pumpom.

2. PREGLED, PODJELA I ZNAČAJ PRIMJENE TOPLOTNIH PUMPI I HIBRIDNIH SISTEMA

Toplotna pumpa je uređaj koji pomoću uloženog rada, oduzima toplotu od okoline, koja se onda može na višem temperaturskom nivou koristiti za grijanje.



Slika 1. Osnovni ciklus toplotne pumpe

Na slici 1. prikazan je osnovni ciklus toplotne pumpe. Medijum koji se koristi za iskorišćenje toplote nekog niskotemperaturnog izvora toplote u toplotnoj pumpi naziva se rashladni medijum, a karakteriše ga niska temperatura ključanja. Ovaj medijum može da isparava na niskim temperaturama, pa u isparivaču on isparava preuzimajući toplotu iz okoline na niskom pritisku. Posle kompresora, ova para, sada visokog pritiska i visoke temperature, kondenzuje se u kondenzatoru predavajući toplotu okolini. Takva tečnost, nakon kondenzacije, prolazi kroz ekspanzioni ventil, čime se obara pritisak medijuma, nakon čega se dobija tečnost niskog pritiska i niske temperature. Takva tečnost ponovo prolazi kroz isparivač, čime se ciklus ponavlja. Osnovni izvori toplote [4] za toplotnu pumpu su: voda, vazduh, zemlja i okolina. Kako se toplotna pumpa može koristiti ne samo za grijanje, nego i za grijanje i pripremu tople sanitarne vode, sve su češći takvi sistemi. Kako bi se poboljšao rad sistema toplotne pumpe, moguće je kao podršku dijela sistema za pripremu tople sanitarne vode koristiti solarni kolektor. Ovakvi sistemi se nazivaju hibridnim sistemima.

3. PROJEKTOVANJE SISTEMA GRIJANJA TOPLOTNIM PUMPAMA POMOĆU PROGRAMSKOG PAKETA GEOTSOL

Prilikom projektovanja sistema grijanja pomoću GeoTSOL-a prvi korak podrazumijeva odabir lokacije. Pomoću potprograma MeteoSyn [1] bira se lokacija i povlače se klimatski podaci za tu lokaciju. Izvor klimatskih podataka je servis „Meteonorm“ [2]. Treba imati u vidu da su dostupni podaci samo do 2005.god.

Izvori toplote za toplotnu pumpu mogu biti [1] geotermalni kolektori, geotermalne sonde, spoljašnji vazduh, podzemne vode. Sistem može biti projektovan za grijanje, grijanje i pripremu tople sanitarne vode, a mogu se izabrati i hibridni sistemi. Prilikom izbora sistema, potrebno je voditi računa o tipu grijanja, odnosno, da li je to niskotemperatursko ili visokotemperatursko. Visokotemperatursko podrazumijeva radijatorsko grijanje, dok niskotemperatursko podrazumijeva zidno ili podno grijanje. Što se tiče sistema za pripremu tople sanitarne vode, on podrazumijeva definisanje podataka o potrošnji tople sanitarne vode po broju osoba. Može se izabrati sistem sa cirkulacijom tople sanitarne vode i bez. Treba imati u vidu da sistem sa cirkulacijom poskupljuje sistem, povećava gubitke, ali izbjegava se veće vrijeme odziva sistema, pa je topla sanitarna voda dostupna odmah.

Programski paket GeoTSOL omogućava izbor toplotne pumpe iz baze programa, ali je moguće unijeti i podatke o nekoj drugoj toplotnoj pumpi. Vrlo je bitno da se obrati posebna pažnja na izbor režima rada toplotne pumpe, odnosno da li će toplotna pumpa samostalno obezbjeđivati svu potrebnu količinu toplote ili će raditi u kombinaciji sa nekim pomoćnim izvorom toplote. Mogući režimi rada toplotne pumpe su: monovalentni, paralelni monoenergetski, alternativni monoenergetski i djelimično paralelni monoenergetski. Monovalentni režim rada predstavlja takav režim gdje toplotna pumpa obezbjeđuje cjelokupnu količinu potrebne toplotne energije. Kod paralelnog monoenergetskog režima rada, toplotna pumpa obezbjeđuje toplotnu energiju do određene ambijentalne temperature, a ispod te temperature, uključuje se pomoćni sistem koji služi kao podrška. Alternativni monoenergetski sistem funkcioniše slično kao prethodni sistem, samo što se ispod određene temperature toplotna pumpa isključuje. Djelimično paralelni monoenergetski režim rada podrazumijeva samostalni rad toplotne pumpe do neke temperature kada se uključuje pomoćni sistem, kada sistemi rade paralelno, a nakon temperature isključenja toplotne pumpe, pomoćni sistem nastavlja da radi samostalno. Pomoćni sistem podrazumijeva konvencionalni sistem grijanja, na primjer električni kotao, gasni kotao, itd.

Kod toplotnih pumpi sa geotermalnim sondama, potrebno je obratiti pažnju na dimenzionisanje primarnog kruga, odnosno na izvor toplote. Potrebno je definisati snagu cirkulacione pumpe, protok, temperaturu zemljišta, specifičnu količinu dobijene toplote po metru, nakon čega se dobija potrebna dužina sonde. Potrebna dužina sonde se dobija na osnovu jednačine [1]:

$$l_{geo} = \frac{P_{TP,hl}}{\dot{q}_L} [m]$$

gdje su:

$P_{TP,hl}$ – rashladni kapacitet toplotne pumpe,

$\dot{q}_L$ – specifična toplota koja se preuzima iz zemljišta.

Kod sistema sa geotermalnim kolektorima, potrebno je voditi računa o potrebnoj površini kolektora, dubini polaganja kolektora, snazi cirkulacione pumpe, protoku. Osim toga, kao i kod sistema sa geosondama, posebnu pažnju treba posvetiti definisanju specifičnog toplotnog kapaciteta zemljišta i količini dobijene toplotne energije.

Sistem sa vazdušnom toplotnom pumpom ima odmah dostupan izvor toplote koji ne zahtjeva investicije sa strane primarnog kruga, međutim, mora se izbjeći smrzavanje sistema, pa toplotna pumpa mora imati i sistem odmrzavanja. Tamo gdje se koristi podzemna voda kao izvor toplote, mora se definisati temperaturski opseg toplotnog izvora u relevantnom periodu.

Sistem sa toplotnom pumpom je najčešće opremljen akumulacionim („buffer“) rezervoarom, a ako se koristi i za pripremu tople sanitarne vode, opremljen je i rezervoarom tople sanitarne vode. Što se tiče solarnog sistema, njegovo definisanje podrazumijeva odabir solarnog kolektora određene snage, broj kolektora, ugao postavljanja, kao i potrebne dužine cjevovoda na osnovu čega se računaju i gubici.

Nakon definisanja i dimenzionisanja svih dijelova sistema, simulacijom se dobijaju podaci o efikasnosti sistema, potrošnji električne energije, dobijenoj količini toplotne energije, gubicima i tako dalje. Najvažniji indikator predstavlja sezonski faktor performansi (SPF- „seasonal performance factor“), odnosno pokazatelj korisnosti toplotne pumpe, odnosno cjelokupnog sistema, koji se računa prema [1] jednostavnoj formuli:

$$SPF = \frac{\text{benefiti}}{\text{troškovi}}$$

U dijelu finansijske analize vrši se analiza ekonomske opravdanosti sistema. Finansijskom analizom se dobija modifikovana stopa povrata, odnosno MIRR, koji predstavlja jasan indikator isplativosti, odnosno što je MIRR veći, sistem je isplativiji. Osim MIRR, dobija se i cijena grijanja po kWh toplotne energije, koja je značajna u pogledu poređenja sistema sa nekim drugim sistemima.

4. TEHNOEKONOMSKA ANALIZA RAZLIČITIH SISTEMA SA TOPLOTNOM PUMPOM

Izvršena je analiza četiri različita sistema sa toplotnim pumpama, odnosno sistem sa vazdušnom toplotnom pumpom, toplotnom pumpom sa geotermalnim sondama, geotermalnim kolektorima, kao i hibridnog sistema koji predstavlja kombinaciju toplotne pumpe sa geosondama i solarnih kolektora. Sistemi su opremljeni toplotnom pumpom, akumulacionim i rezervoarom tople sanitarne vode, kao i razvodima za grijanje i pripremu tople sanitarne vode. Kod hibridnog sistema, izabran je kombinovani rezervoar, tako da je izbjegnuta upotreba dvije posude.

U tabelama u nastavku su dati rezultati proračuna za različite sisteme sa toplotnim pumpama. Dati su sezonski faktori performansi, količine dobijene toplotne energije, kao i potrošnje električne energije.

Tabela 1. Rezultati proračuna sistema sa vazdušnom TP

Rezultati proračuna		Vrijednost
SPF toplotne pumpe		3,7
SPF sistema toplotne pumpe		3,7
Godišnja potrošnja električne energije		4430 kWh _{god}
Godišnje dobijena toplotna energija		16520 kWh _{god}
Iskorišćena energija	Za grijanje	13596 kWh _{god}
	Za TSV	2232 kWh _{god}
Gubici rezervoara		681 kWh _{god}

Tabela 2. Rezultati proračuna sistema TP sa geokolektorima

Rezultati proračuna		Vrijednost
SPF toplotne pumpe		3,4
SPF sistema toplotne pumpe		3,4
Godišnja potrošnja električne energije		4870 kWh _{god}
Godišnje dobijena toplotna energija		16541 kWh _{god}
Iskorišćena energija	Za grijanje	13596 kWh _{god}
	Za TSV	2233 kWh _{god}
Gubici rezervoara		686 kWh _{god}

Tabela 3. Rezultati proračuna sistema TP sa geosondama

Rezultati proračuna		Vrijednost
SPF toplotne pumpe		3,4
SPF sistema toplotne pumpe		3,4
Godišnja potrošnja električne energije		4819 kWh _{god}
Potrošnja el. energije za pomoćne sisteme		99 kWh _{god}
Godišnje dobijena toplotna energija		16545 kWh _{god}
Iskorišćena energija	Za grijanje	13596 kWh _{god}
	Za TSV	2233 kWh _{god}
Gubici rezervoara		690 kWh _{god}

Tabela 4. Rezultati proračuna hibridnog sistema

Rezultati proračuna		Vrijednost
SPF toplotne pumpe		3,4
SPF sistema toplotne pumpe		3,4
SPF cjelokupnog sistema (TP + solar)		3,9
Godišnja potrošnja električne energije		4220 kWh _{god}
Godišnje dobijena toplotna energija		16452 kWh _{god}
Iskorišćena energija		15813 kWh _{god}
Gubici		1872 kWh _{god}
Solarni sistem		
Udio solarnog sistema		15,9 %
Udio solarnog sistema u pripremi TSV		53,7 %
Efikasnost solarnog sistema		36 %
Dozračena toplota na aktivnu površinu kolekt.		7240 kWh _{god}
Dobijena topl. energija od solarnih kolektora		3411 kWh _{god}

U sledećoj tabeli dat je prikaz bitnijih podataka analize različitih sistema sa toplotnim pumpama. Može se primijetiti da je najjeftiniji sistem sa vazdušnom TP, dok je hibridni sistem najskuplji, zbog čega je i cijena grijanja najmanja kod sistema sa vazdušnom TP, a najveća kod hibridnog sistema.

Najveći SPF ima hibridni sistem od 3,9. Najveći MIRR se dobija kod sistema sa vazdušnom TP od čak 15,75%, ali

treba uzeti u obzir da je hibridni sistem više od tri puta skuplji, ali se i dalje ostvaruje pozitivan MIRR.

Tabela 5. Poređenje podataka različitih sistema sa TP

Sistem sa TP	Cijena sistema [€]	Cijena grijanja [€/kWh]	SPF	MIRR [%]
Vazdušna TP	4750	0,045	3,7	15,75
TP sa geotermalnim kolektorima	11050	0,065	3,4	4,05
TP sa geotermalnim sondama	14350	0,075	3,4	1,83
Hibridni sistem	18512	0,086	3,4/3,9*	0,23

* Napomena: zbog toga što se radi o hibridnom sistemu, date su vrijednosti SPF toplotne pumpe, kao i cjelokupnog sistema, respektivno.

5. ZAKLJUČAK

Kao što se može vidjeti iz svega prethodnog, u ovom radu dato je objašnjenje projektovanja sistema sa toplotnim pumpama u programskom paketu GeoTSOL.

Dato je objašnjenje načina izbora dijelova sistema kao i njihovog dimenzionisanja.

Drugi dio rada predstavlja tehnoe ekonomska analiza različitih sistema za grijanje i pripremu tople sanitarne vode pomoću toplotnih pumpi i hibridnih sistema. Poređenjem podataka dobijenih analizom različitih sistema, može se donijeti nekoliko zaključaka.

Sa strane efikasnosti sistema, najveću efikasnost, odnosno SPF ima hibridni sistem od 3,9 zbog toga što je količina električne energije utrošena za proizvodnju toplotne energije manja za količinu toplotne energije koja je dobijena putem solarnog kolektora, pa je sa strane odnosa utrošene električne i dobijene toplotne energije ovakav sistem najpovoljniji.

Nešto manji SPF ima sistem sa vazdušnom toplotnom pumpom, od 3,7. Najmanji SPF imaju sistemi sa geosondama i geokolektorima, od 3,4. Cijena grijanja predstavlja odnos godišnjih troškova sistema i dobijene toplotne energije iz sistema, pa pruža dobru informaciju o isplativosti ugradnje nekog od sistema.

Najmanju cijenu grijanja ima sistem sa vazdušnom toplotnom pumpom od 0,045 €/kWh, zatim slijede sistem sa geokolektorima i geosondama sa cijenama grijanja od 0,065 €/kWh i 0,075 €/kWh, respektivno. Najveću cijenu grijanja ima hibridni sistem od čak 0,086 €/kWh.

Visoke cijene grijanja kod hibridnog, sistema sa geosondama i sistema sa geokolektorima, rezultat su visoke cijene opreme, koja proističe iz specifičnosti načina iskorišćenja izvora toplote.

Dobar indikator isplativosti sistema predstavlja i MIRR, odnosno stopa povrata projekta, što je on veći, veća je i isplativost sistema.

Najveći MIRR ima sistem sa vazdušnom toplotnom pumpom, od 15,75%, a slijede sistemi sa geotermalnim kolektorima, geosondama sa MIRR od 4,05% i 1,83% respektivno, dok najmanji MIRR ima hibridni sistem od 0,23%, zbog visoke početne investicije.

Kako se vidi u prethodnom dijelu, najveću isplativost ima sistem sa vazdušnom toplotnom pumpom u pogledu korišćene tehnologije i cijene, zbog čega ovakav sistem u odnosu na ostale sisteme razmatrane u ovom radu ima očitu prednost.

7. LITERATURA

- [1] GeoT*SOL basic, Version 2.0, Planning and designing heat pump systems
- [2] <http://www.meteonorm.com/>, (pristupljeno 22.10.2017. 19:09h)
- [3] <http://www.valentin-software.com/en/products/heat-pumps/33/geotsol>, (pristupljeno 22.10.2017. 14:00)
- [4] H. Reknagel, E. Šprenger, Grejanje i klimatizacija, Interklima, Vrnjačka Banja, 2004.
- [5] http://www.mre.gov.rs/latinica/faq-energetska-efikasnost-obnovljivi-izvori.php#_Mere_podsticaja_z_1, (pristupljeno 3.10.2018. 07:00)

Kratka biografija:



Dragan Lazić rođen je u Bogenu, SR Njemačka 1992. god. Master rad odbranio je 2018. godine na Fakultetu tehničkih nauka iz oblasti Mašinstva – Termotehnika.



Aleksandar Andelković rođen je u Šapcu 1981. god. Doktorirao je 2015. god., a od 2015. je u zvanju docent. Oblast interesovanja su Nekonvencionalni sistemi grijanja i hlađenja.

ПРИМЈЕНА ТОПЛОТНИХ ПУМПИ КОД ОБЈЕКТА РАЗЛИЧИТЕ ЕНЕРГЕТСКЕ ЕФИКАСНОСТИ

APPLICATION OF HEAT PUMPS IN BUILDINGS WITH DIFFERENT ENERGY EFFICIENCY

Драгиша Гуговић, Факултет техничких наука, Нови Сад

Област – МАШИНСТВО

Кратак садржај - Задатак овога рада јесте да представи топлотне пумпе као енергетски и еколошки добро решење у системима грјања који се користе у стамбене сврхе. Описан је основни принцип на коме се заснива рад топлотне пумпе, топлотни извори које она користи и дата је основна подјела топлотних пумпи које се највише користе у овакве сврхе. Приказана је симулација топлотне пумпе ваздух/вода која се користи за грјање и припрему топлоте потрошне воде у домаћинству за три основна случаја: високо, просјечно и ниско енергетски ефикасан објекат. Сprovedена је анализа како енергетска ефикасност објекта утиче на енергетске, еколошке и економске параметре.

Кључне ријечи: Топлотна пумпа, GeoT*SOL, енергетска ефикасност.

Abstract – The aim of this paper is to present heat pumps as energy and environmentally sound solution in heating systems that are used for residential purposes. The basic principle the operation of the heat pump relies on is presented as well as thermal sources and the most commonly types of heat pumps used for such purposes. The simulation of an air-water heat pump used for water preparation and heating is given for three basic cases: high, average and low energy efficient facility. The analyses how energy efficiency takes a part in energy, ecological and economic parameters is showed as well.

Keywords: Heat pump, GeoT*SOL, energy efficiency.

1. УВОД

Највећи дио топлотне енергије потребан за грјање стамбених и других објеката гдје бораве људи обезбеђује се употребом фосилних горива.

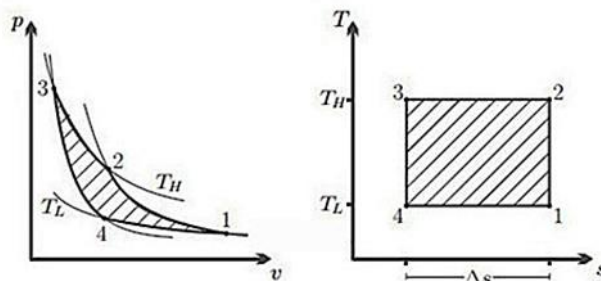
Коришћење енергије из обновљивих извора енергије постаје све популарније како расте свест о планирању будућности сваке земље када залихе фосилних горива буду при крају.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био доц. др Александар Анђелковић.

2. ПРИНЦИП РАДА И РАЗВОЈ ТОПЛОТНЕ ПУМПЕ

Топлотне пумпе у данашње вријеме се убрајају у нај-ефективније уређаје помоћу којих се добија топлотна енергија. за грјање у стамбеним зградама која би замијенила фосилна горива.

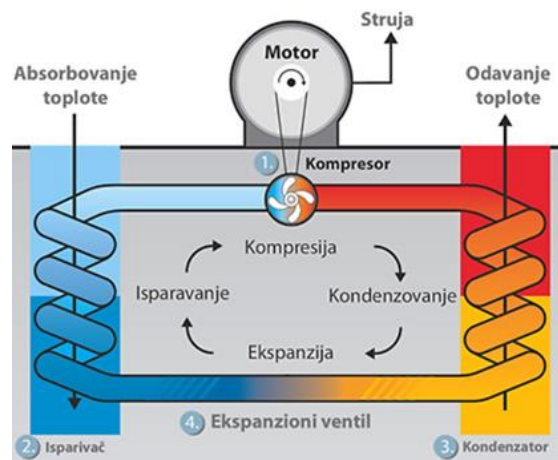


Слика 1: Карноов лијевокретни кружни циклус

Топлотна пумпа ради на принципу Карноов-ог кружног циклуса који чине двије изотермне и двије изентропне промјене стања. Овај циклус је реверзибилан што значи да сваки процес може да промијени смјер одигравања.

3. КОМПОНЕНТЕ ТОПЛОТНИХ ПУМПИ

Топлотне пумпе се састоје од четири главне компоненте: испаривача, компресора, кондензатора и експанзионог елемента.



Слика 2: Шема рада топлотне пумпе

Може се слободно рећи да су један од најважнијих елемената топлотних пумпи компресори. Задатак

компресора је да усава пару расхладног флуида из испаривача и да је дигне на виши енергетски ниво тј. виши притисак и температуру. Кондензатори су размјењивачи топлоте који представљају важан дио система топлотних пумпи. У њему се пара расхладног флуида кондензује предајућу топлоту кондензације флуиду којим се хлади тј. топлотном понору.

Испаривачи су размјењивачи топлоте који користе топлоту окружења тј. топлотног извора за испаравање течног расхладног флуида који циркулише кроз њега. Улога експанзионог вентила је да одржава константан притисак у испаривачу. Избор најпогоднијег расхладног флуида утиче прије свега на животну средину, сигурност, поузданост и трошкове које такав систем проузрокује.

4. ТОПЛОТНИ ИЗВОРИ И ПОНОРИ

Топлотни извор представља мјесто из кога се „црпи“ топлотна енергија а топлотни понор мјесто на кога са топлотна енергија преноси. Који ће се топлотни извор употребити зависи од расположивости, приступачности и енергетског нивоа извора. Топлотни извор треба да има високу и константну температуру током године и да има довољну издашност како би његово експлоатисање било на задовољавајућем нивоу. Ваздух је један од најприступачнијих топлотних извора и што је најважније има га у неограниченим количинама и бесплатан је (не захтијева никаква посебна одобрења).

5. ВРСТЕ ТОПЛОТНИХ ПУМПИ

На основу извора додатне енергије топлотне пумпе се дијеле на компресионе и абсорпционе. Основни извори за топлотну енергију пумпе су: ваздух, вода и земља. Када топлотна пумпа покрива 100% топлотног оптерећења каже се да ради у моновалентном режиму. За други случај када топлотна пумпа ради у спрези са неким другим извором енергије каже се да ради у бивалентном режиму. Такође је могуће остварити и друге начине повезивања топлотне пумпе и других обновљивих и необновљивих извора енергије које носе заједнички назив хибридни системи.

6. GEOT*SOL СОФТВЕР

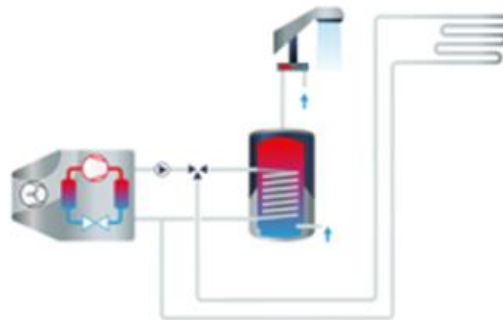
GeoT*SOL је професионални алат за планирање и пројектовање система топлотне пумпе. Помоћу њега можемо одабрати различите врсте и компоненте система, израчунати енергију и трошкове како бисмо остварили најбољи могући сезонски фактор учинка. GeoT*SOL такође симулира системе топлотне пумпе у складу са потребама у комбинацији са соларним термичким колекторима.

7. СИМУЛАЦИЈА ТОПЛОТНЕ ПУМПЕ ВАЗДУХ/ВОДА

У овом раду биће анализирана случај инсталације топлотне пумпе ваздух/вода помоћу којих се преноси топлота на систем нискотемпературног (подног) гријања. Такође је у симулацију узета у обзир и потрошња топле санитарне воде. Симулација је

одрађена на примјеру породичног стамбеног објекта нето површине 100 m². Анализа система ће бити одрађена на три базна случају тј. на примјерима:

- Високо енергетски ефикасног објекта (30 W/m²)
- Просјечно енергетски ефикасног објекта (45 W/m²)
- Ниско енергетски ефикасног објекта (90 W/m²)



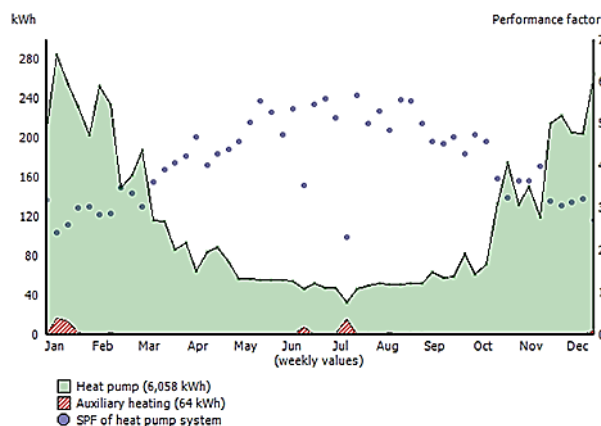
Слика 3: Систем ТП за гријање простора и ТПВ (ваздух/вода)

7.1. Топлорна пумпа ваздух/ваздух за високо ЕЕ објекат

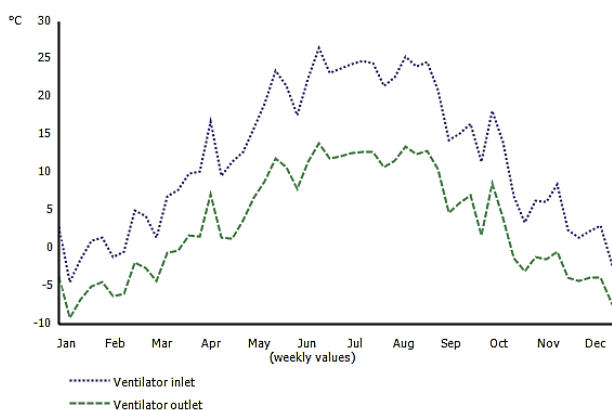
У овом случају ће бити анализиран случај када објекат има специфичну потрошњу топлоте 30 W/m² (класа K₁ разред R₄).

Табела 1: Резултати годишње симулације ТП ваздух/вода за високо ЕЕ објекат

Енергија потребна за гријање објекта:	2.918	kWh
Енергија потребна за ТПВ:	2.834	kWh
Енергија испоручена помоћу ТП:	6.058	kWh
Енергија испоручена електричним гријачем:	64	kWh
Сезонски фактор перформансе ТП:	3,5	-
Сезонски фактор перформансе система ТП:	3,4	-
Потрошња електричне енергије ТП:	1736	kWh
Потрошња електричне енергије гријача:	64	kWh
Потрошња ел. енергије пумпе топлотног извора:	0	kWh
Губици резервоара (бојлера):	369	kWh
Примарна уштеда енергије:	4.839	kWh
Смањење CO ₂ емисије:	1.198	kg



Слика 4: Енергетски дијаграм и промјена SPF фактора током године - ТП ваздух/вода високо ЕЕ објекат



Слика 5: Температурски дијаграм - ТП ваздух/вода високо ЕЕ објекат

За покривање вршних оптерећења користи се уграђен електрични гријач који помаже да се пројектовано тоplotно оптерећење оствари (Слика 4.). Ради стицања реалније слике о систему тоplotне пумпе ваздух/вода, овај систем је упоређен са системом гријања на гас (гасни котло-подно гријање).

Из табеле 14. може се видјети да је цијена kWh тоplotне енергије која је испоручена системом тоplotне пумпе ваздух/вода за 27,70 % мања од система гријања на гасни котло. Иако је цијена почетне инвестиције тоplotне пумпе ваздух/вода за око 3,75 пута већа од инвестиције за котло на гас ипак постоје годишње новчане уштеде јер се за сваку гријну сезону мора утрошити одређена количина гаса чија је цијена подложна мањим или већом осцилацијама.

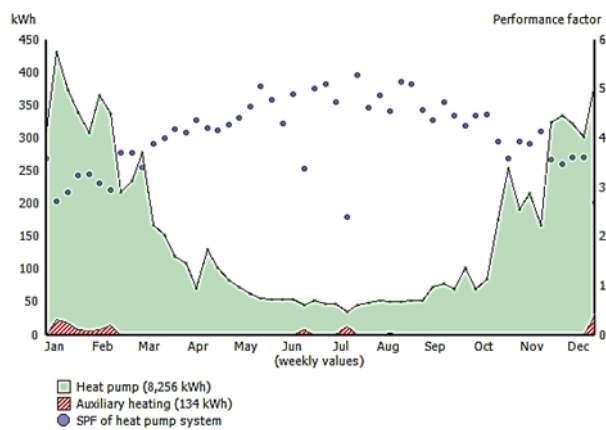
Ако би се у оваквом систему користила циркулација топле потрошне воде у циљу постизања већег комфора захтјеви за тоplotом енергијом би се повећали за 984 kWh.

7.2. Топлотна пумпа ваздух/ваздух за просјечно ЕЕ објекат

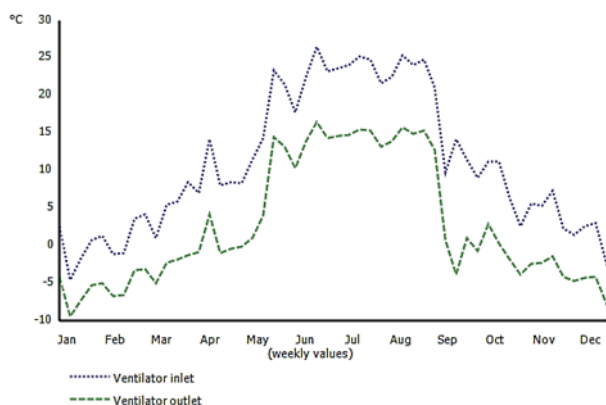
За случај просјечно енергетски ефикасног објекта 45 W/m^2 (класа K_2 разред R_6) резултати симулације за тоplotну пумпу ваздух/вода су приказани у следећој табели:

Табела 2: Резултати годишње симулације ТП земља ваздух/вода просјечно ЕЕ објекат

Енергетска потреба гријања простора:	5.189	kWh
Енергија потребна за ТПВ:	2.834	kWh
Енергија испоручена помоћу ТП:	8.256	kWh
Енергија испоручена електричним гријачем:	134	kWh
Сезонски фактор перформансе ТП:	3,7	-
Сезонски фактор перформансе система ТП:	3,5	-
Потрошња електричне енергије ТП:	2.240	kWh
Потрошња електричне енергије гријача:	134	kWh
Потрошња ел. енергије пумпе тоplotног извора:	0	kWh
Губици резервоара (бојлера):	368	kWh
Примарна уштеда енергије:	6.800	kWh
Смањење CO_2 емисије:	1.683	kg



Слика 6: Енергетски дијаграм и промјена SPF фактора током године - ТП ваздух/вода просјечно ЕЕ објекат



Слика 7: Температурски дијаграм - ТП ваздух/вода просјечно ЕЕ објекат

Због осцилација ваздуха температура ваздуха, температуре на улазу и на излазу из вентилатора осцилују што се може видјети из приложеног дијаграма (Слика 7.). Снижавањем температуре околног ваздуха коефицијент гријања опада што неповољно утиче на економску рачуницу. Оно што карактерише овај случај јесте већа специфична потрошња тоplotе по m^2 тј. 45 W/m^2 самим тим већи су и захтјеви за тоplotном енергијом (класа K_2 разред R_6).

Топлотна пумпа ваздух/вода је у бивалентној паралелној спрези са електричним гријачем од 1 kW који се активира када су пикови за тоplotном енергијом велики. Енергија коју испоручи овакав систем у овом случају је за 37,04 % већа од енергије коју испоручи систем у претходном случају. Ако би се користио систем са циркулацијом топле потрошне воде губици би се повећали за додатних 893 kW у односу на систем без циркулације. У поређењу са гасним котлом истих карактеристика цијена kWh тоplotне енергије за 69,36 % нижа од цијене гријања на гас.

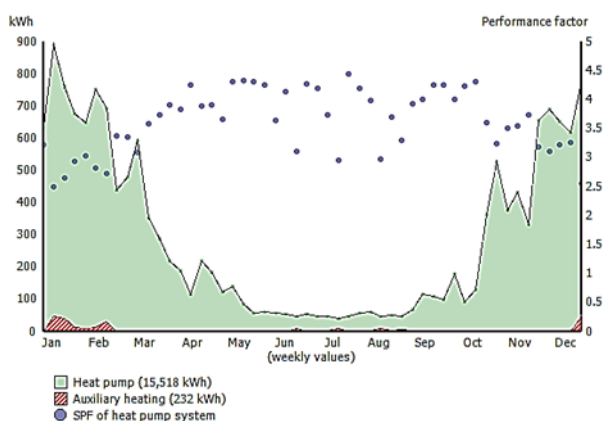
7.3. Топлотна пумпа ваздух/ваздух за ниско ЕЕ објекат

За случај ниско енергетски ефикасног објекта 90 W/m^2 (класа K_3 разред R_8) резултати симулације за

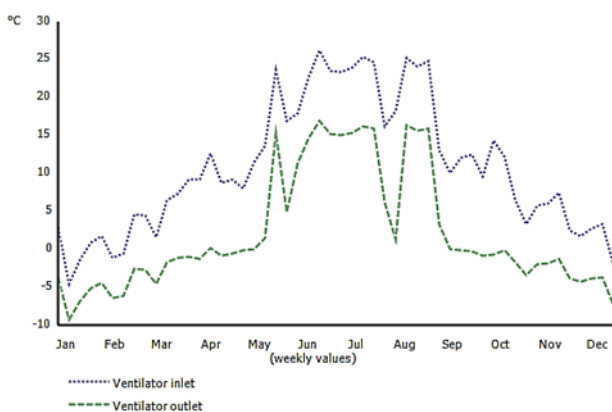
топлотну пумпу ваздух/вода су приказани у следећој табели:

Табела 3: Резултати годишње симулације ТП земља ваздух/вода ниско ЕЕ објект

Енергетска потреба гријања простора:	12.539	kWh
Енергија потребна за ТПВ:	2.834	kWh
Енергија испоручена помоћу ТП:	15.518	kWh
Енергија испоручена електричним гријачем:	232	kWh
Сезонски фактор перформансе ТП:	3,3	-
Сезонски фактор перформансе система ТП:	3,2	-
Потрошња електричне енергије ТП:	4.749	kWh
Потрошња електричне енергије гријача:	232	kWh
Потрошња ел. енергије пумпе топлотног извора:	0	kWh
Губици резервоара (бојлера):	376	kWh
Примарна уштеда енергије:	11.435	kWh
Смањење CO ₂ емисије:	2.830	kg



Слика 8: Енергетски дијаграм и промјена SPF фактора током године - ТП ваздух/вода за ниско ЕЕ објект



Слика 9: Температурски дијаграм - ТП ваздух/вода за ниско ЕЕ објект

Овдје је разматран случај када је специфична потрошња топлоте по m^2 тј. $90 W/m^2$ самим тим већи су и захтјеви за топлотном енергијом (класа K_3 разред R_8). Због слабије термоизолације тј. већих топлотних губитака у околину потребна је топлотна пумпа веће снаге у односу на претходна два случаја да би се остварило пројектовано грејно оптерећење. Топлотна

пумпа ваздух/вода је у бивалентној паралелној спрези са електричним гријачем од 2 kW који се активира када су пикови за топлотном енергијом велики. Енергија коју испоручи овакав систем у овом случају је за 87,72 % већа од енергије коју испоручи систем у случају просјечно енергетски ефикасног објекта а 2,57 пута већа од енергије која је потребна за ниско енергетски ефикасан објект. Ако би се користио систем са циркулацијом топле потрошне воде губици би се повећали за додатних 898 kW у односу на систем без циркулације. У поређењу са гасним котлом истих карактеристика цијена kWh топлотне енергије за 56,79 % нижа од цијене гријања на гас.

8. ЗАКЉУЧАК

Топлотне пумпе представљају револуционарно решење на пољу технике гријања због тога што користе природне ресурсе (њихову топлотну енергију) да би задовољиле енергетске (топлотне) потребе крајњих корисника.

Из симулације види се да најмању цијену гријања по kWh има случај ниско енергетски ефикасног објекта у износу од 3,731 дин./kWh с тим да захтијева три пута већу специфичну потрошњу по m^2 и мале инвестиције у термоизолацију објекта.

Најидеалнији случај јесте са високо енергетски ефикасним објектом специфичне потрошње од $30 W/m^2$ али захтијева већа инвестициона улагања у термоизолацију са цијеном гријања од 6,782 дин./kWh. Просјечно енергетски објект ($45 W/m^2$) је најбоље решење на енергетском и економском пољу са ценом гријања од 5,35 дин./kWh и највећим сезонским фактором перформансе од 3,9. Оно што је главни "камен спотицања" у коришћењу оваквих система јесте њихова цијена и заступљеност на тржишту.

9. ЛИТЕРАТУРА

- [1] А. Ченејац, Р. Бјелаковић, А. Анђелковић, Неконвенционални системи грејања: теоријске основе, примена и примери прорачуна, Факултет техничких наука – Универзитет у Новом Саду, Нови Сад, 2017.
- [2] Б. Јаћимовић, С. Генић, Топлотне операције и апарати, Машински факултет, Београд, 1992.
- [3] GeoT*Sol софтвер за прорачун топлотних пумпи, www.valentin-software.com

Кратка биографија:



Драгиша Гутковић рођен је у Требињу 1993. године. Основну и средњу школу завршио у Гацку. Основне студије Машинства – Енергетика и процесна техника на ФТН завршио 2017. године. Мастер рад на истом факултету одбранио 2018. године.

PRINUDNE OSCILACIJE VIBROUDARNOG SISTEMA

FORCED OSCILLATION OF VIBRO-IMPACT SYSTEM

Petar Tošić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO -Tehnička mehanika

Kratak sadržaj – U radu je analiziran vibroudarni sistem pobuđen kretanjem. Izvršena je analitička i numerička analiza slobodnih, prigušenih i prinudnih oscilacija. Prikazani su amplitudno frekventni dijagrami prinudnih oscilacija i izvršeno je poređenje analitičkog i numeričkog rešenja.

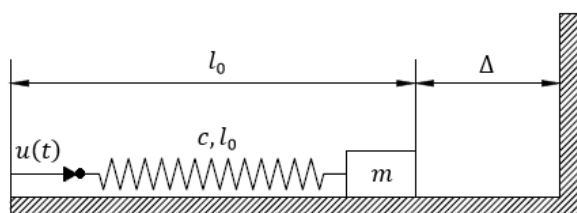
Ključne reči: oscilacije, udar, vibro udar

Abstract – In this paper are analysed forced oscillations of vibro-impact system. An analytical and numerical analysis of free, damped and forced oscillations was performed. Amplitude frequency diagrams of forced oscillations are presented and a comparison of the analytical and numerical solution was made.

Keywords: oscillations, impact, vibro-impact

1. UVOD

Tema ovog rada je analiza vibroudarnog sistema pobuđenog kretanjem. Sistem, prikazan na slici 1, sastoji se od lake linearne opruge, krutosti c i dužine l_0 u nenapregnutom stanju, na čijem se jednom kraju nalazi oscilator, mase m , koji može da klizi po horizontalnoj podlozi bez trenja, dok je drugom kraju opruge saopštena prinuda u vidu pomeranja $u(t)$. Prepreka kretanju, u vidu zida je postavljena sa desne strane sistema, na rastojanju Δ od ravnotežnog položaja oscilatora.



Slika 1

Ovakav model vibroudarnog sistema može da posluži u lakšem razumevanju rada široke klase vibroudarnih mašina [2],[3]. Razne mašine za bušenje, nabijanje, drobljenje i mlevenje, zabijanje rade na principu vibroudara. U ovom radu će biti analizirane slobodne oscilacije, biće ispitani slučajevi slobodnih oscilacija bez udara i slobodnih oscilacija sa udarom. Analiziraće se prinudne oscilacije, analitički i numerički, biće prikazan amplitudno frekventni dijagram i numerička rešenja diferencijalne jednačine kretanja za različite frekvencije prinude. Ispitaće se postojanje udar

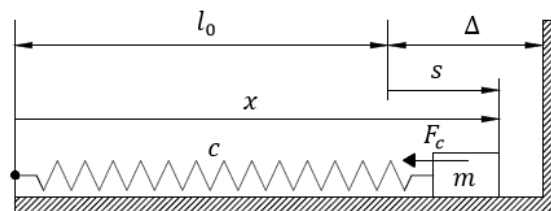
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miodrag Zuković.

nih oscilacija u zavisnosti od brzina u trenutku udara i frekvencije prinude. Ispitaće se postojanje vibroudarnih oscilacija u zavisnosti od frekvencije oscilacija, analiziraće se amplitudno frekventni dijagram dobijen numeričkim putem, zajedno sa amplitudno frekventnim dijagramom slobodnih oscilacija.

2. SLOBODNE OSCILACIJE

Počecemo naša razmatranja sa proučavanjem slobodnih oscilacija sistema.



Slika 2

Na slici 2 predstavljen je dati sistem u napregnutom stanju, koji se sastoji od opruge krutosti c dužine l_0 na čijem se jednom kraju nalazi masa m , koja može da klizi po horizontalnoj podlozi bez trenja dok je drugi kraj opruge fiksiran. Rastojanje između mase m u ravnotežnom položaju i zida je označeno grčkim slovom Δ . U radu se razmatra samo slučaj $\Delta > 0$. Položaj tačke je u potpunosti okarakterisan generalisanom koordinatom x .

U početnom trenutku položaj tačke je

$$x = l_0 + s \quad (1)$$

Uzeto je da je početna brzina oscilatora jednaka nuli

$$\dot{x}(0) = 0 \quad (2)$$

Slučaj bez udara se javlja kada je zadovoljen uslov

$$l_0 - \Delta < x_0 < l_0 + \Delta \quad (3)$$

Slučaj sa udarom javlja se kada je zadovoljen uslov

$$x_0 \leq l_0 - \Delta \quad (4)$$

Diferencijalna jednačina kretanja je data sa

$$m\ddot{x} = -F_c \quad (5)$$

Sa F_c je obeležena sila u linearnoj opruzi

$$F_c = c s \quad (6)$$

gde je s izduženje opruge.

Nakon uvođenja oznake $\omega^2 = \frac{c}{m}$, prethodna jednačina postaje

$$\ddot{x} + \omega^2 x = \omega^2 l_0 \quad (7)$$

Rešenje prethodne jednačine može se zapisati u obliku

$$x(t) = C_1 \cos(\omega t) + C_2 \sin(\omega t) + l_0 \quad (8)$$

Gde su C_1 i C_2 proizvoljne konstante koje se određuju uz pomoć početnih uslova

$$x(0) = x_0 = l_0 + s_0, \dot{x}(0) = 0 \quad (9)$$

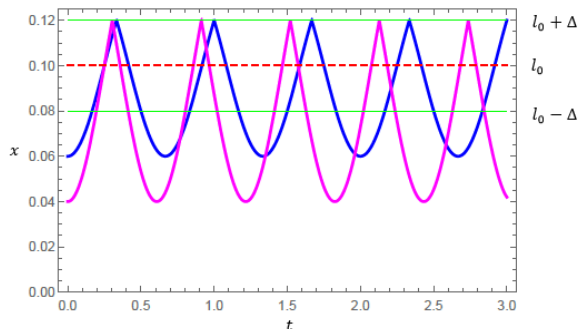
Ako se trenutak kada dolazi do udara obeleži sa t_u tada je položaj oscilatora u trenutku udara.

$$x(t_u) = l_0 + \Delta \quad (10)$$

Trenutak neposredno pre udara je označen sa t_u^- , a sa t_u^+ trenutak nakon udara. Veza između brzina u trenutku neposredno pre i nakon udara je data sa jednačinom

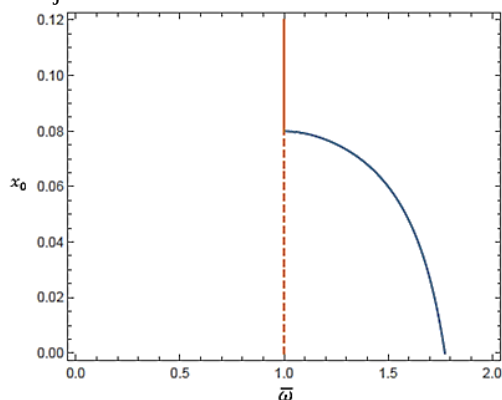
$$\dot{x}(t_u^+) = -k\dot{x}(t_u^-) \quad (11)$$

Rešena je diferencijalna jednačina (7) sa graničnim uslovima (9),(10),(11)



Slika 3. Oscilacije sa udarom rešenje za parametre $l_0 = 0.1\text{m}$, $\Delta = 0.2\text{m}$, $\omega^2 = 2\pi \text{ s}^{-1}$, $k = 1$, $x_0 = 0.06\text{m}$ i $x_0 = 0.04\text{m}$.

Na slici 3 prikazane su slobodne oscilacije sa udarom. Može se primetiti da se period oscilacija menja sa promenom amplitude, što nije slučaj kod slobodnih oscilacija bez udara.



Slika 4. Zavisnost frekvencije od pomeranja „back bone curve”

Na slici 4 prikazana je zavisnost kružne frekvencije od x_0 . U slučaju bez udara frekvencija je konstantna, a u slučaju sa udarom frekvencija se menja sa promenom amplitude odnosno početnog položaja x_0 .

3. PRINUDNE OSCILACIJE

Uvodi se prinuda $u(t)$ u naš sistem tako što se kraj opruge koji je bio fiksiran (slika 2) periodično pomera po harmonijskom zakonu

$$u(t) = u_0 \cos(\Omega t) \quad (12)$$

gde je u_0 amplituda prinudnog pomeranja, a Ω frekvencija prinude. Uvodi se i otpor kretanju vazduha kao

$$F_b = b\dot{x} \quad (13)$$

Pa se diferencijalna jednačina kretanja između dva udara definiše kao

$$m\ddot{x} = -F_c - F_b \quad (14)$$

Gde je F_c

$$F_c = c s = c(x - l_0 - u(t)) \quad (15)$$

U jednačinu (14) su zamenjeni izrazi (12),(13),(15) nakon sređivanja dobija se

$$\ddot{x} + 2\delta\dot{x} + \omega^2 x = \omega^2 l_0^2 + \omega^2 u_0 \cos(\Omega t) \quad (16)$$

Gde je $2\delta = \frac{b}{m}$, a $\omega^2 = \frac{c}{m}$. Ako se uvede smena

$$x = l_0 + y \rightarrow \dot{x} = \dot{y}, \quad \ddot{x} = \ddot{y} \quad (17)$$

Diferencijalna jednačina kretanja postaje

$$\ddot{y} + 2\delta\dot{y} + \omega^2 y = \omega^2 u_0 \cos(\Omega t) \quad (18)$$

Opšte rešenje diferencijalne jednačine se dobija kao zbir opšteg rešenja homogene jednačine i partikularnog rešenja same jednačine $y = y_h + y_p$. Homogeno rešenje zbog prisustva prigušenja vremenom iščezava $\lim_{t \rightarrow \infty} y_h = 0$. Usvaja se da opšte rešenje sadrži samo partikularni deo dok se homogeni deo može zanemariti $\lim_{t \rightarrow \infty} y = y_p$.

Partikularno rešenje je oblika

$$y_p = A_1 \cos(\Omega t) + A_2 \sin(\Omega t) \quad (19)$$

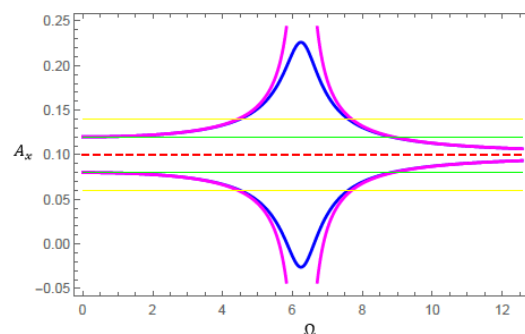
Konstante A_1 i A_2 se određuju tako što se pronađu odgovarajući izvodi partikularnog rešenja (19) i zamene se u diferencijalnu jednačinu (18). Preko konstanti A_1 i A_2 , dobija se jednačina za amplitudu prinudnih oscilacija.

$$A = \sqrt{A_1^2 + A_2^2} = \sqrt{\frac{\omega^4 u_0^2}{4\delta^2 \Omega^2 + (\Omega^2 - \omega^2)^2}} \quad (20)$$

Ukoliko se u (20) zameni da je $\delta = 0$ dobija se jednačina za amplitudu oscilacija bez prigušenja

$$A = \sqrt{\frac{\omega^4 u_0^2}{(\Omega^2 - \omega^2)^2}} \quad (21)$$

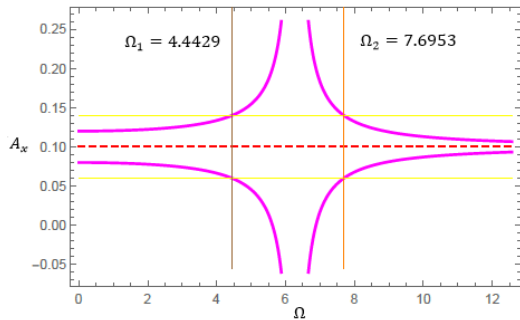
Zavisnost amplitude A_x ($A_x = l_0 \pm A$) od frekvencije Ω može se grafički predstaviti preko amplitudno frekventnog dijagrama slika 5.



Slika 5. Amplitudno frekventni dijagram

Na slici 5 su prikazane amplitudno frekventne karakteristike za prinudne oscilacije sa prigušenjem i bez prigušenja. Isprekidanom linijom na dijagramu je prikazan ravnotežni položaj oko kojeg se vrše oscilacije. Horizontalnim linijama je predstavljena prepreka, odnosno granice za postojanje udara. Kada je $\Omega \ll \omega$, tj. $\Omega \approx 0$ amplituda prinudnih oscilacija je približno jednaka u_0 , odnosno statičkom izduženju opruge usled pomeranja za vrednost u_0 . Kada je $\Omega = \omega = 2\pi \text{ s}^{-1}$ dolazi do rezonancije. U rezonantnoj oblasti amplituda prinudnih oscilacija bez prigušenja raste neograničeno, dok kod prinudnih oscilacija sa prigušenjem amplituda ima konačnu vrednost, koja zavisi od veličine prigušenja. Kada je $\Omega \gg \omega$, amplitude prinudnih oscilacija su vrlo male. Ovo osobina je važna u praksi jer ukazuje na

moćnosti da se prinudne oscilacije i pored dejstva prinude odstrane. Odstranjivanje, eliminisanje prinudnih oscilacija naziva se vibraciona izolacija [1]. Iz jednačine (21) može se odrediti vrednost frekvencije prinude, pri kojoj je amplituda prinudnih oscilacija jednaka granici udara Δ .

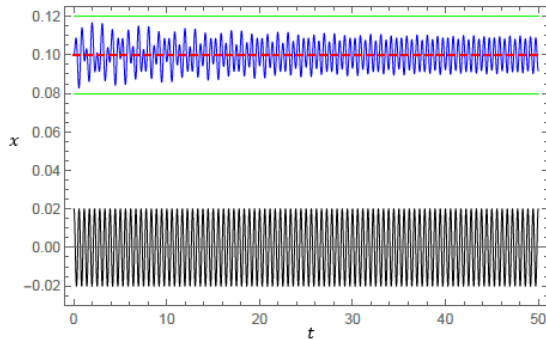


Slika 6. A/F dijagram i frekvencije prinude kada je $A = \Delta = 0.04\text{m}$ a $u_0 = 0.02\text{m}$ i $\omega = 2\pi\text{ s}^{-1}$.

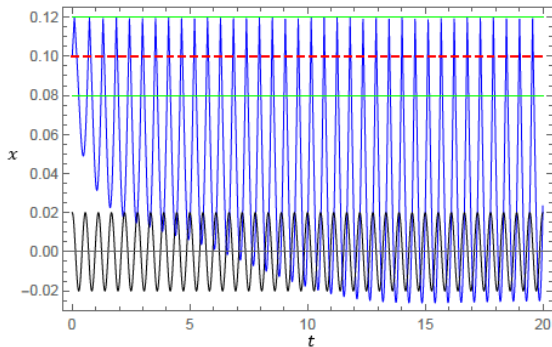
Diferencijalna jednačina kretanja (18), uz uslove udara

$$y(t_u) = \Delta, \dot{y}(t_u^+) = -k \dot{y}(t_u^-) \quad (22)$$

je rešena numerički za frekvenciju $\Omega = 1.8 \omega$.



Slika 7. Rešenje bez udara za parametre $\omega = 2\pi\text{ s}^{-1}$, $\delta = 0.5$, $k = 0.8$, $u_0 = 0.2\text{m}$, $l_0 = 0.1\text{m}$, $\Delta = 0.02\text{m}$, $y(0) = 0$, $\dot{y}(0) = 0$



Slika 8. Rešenje sa udarom za parametre $\omega = 2\pi\text{ s}^{-1}$, $\delta = 0.5$, $k = 0.8$, $u_0 = 0.2\text{m}$, $l_0 = 0.1\text{m}$, $\Delta = 0.02\text{m}$, $y(0) = 0$, $\dot{y}(0) = 0.2$

Za istu frekvenciju prinude postoje dva rešenja, rešenje sa udarom i bez udara, zavisno od početnog uslova za brzinu.

4. ANALIZA PRINUDNIH OSCILACIJA SA UDAROM

U ovom delu je zanemarena sila otpora vazduha, $\delta = 0$. Diferencijalna jednačina kretanja (18) tada postaje

$$\ddot{y} + \omega^2 y = \omega^2 u_0 \cos(\Omega t) \quad (23)$$

Ako se za početni trenutak izabere trenutak udara u zid t_u , tačnije trenutak neposredno nakon udara, a za krajnji, trenutak neposredno pre novog udara, $t_u + \frac{2\pi}{\Omega}$, granični uslovi se mogu zapisati kao

$$y(t_u) = \Delta, \quad y\left(t_u + \frac{2\pi}{\Omega}\right) = \Delta \quad (24)$$

$$\dot{y}\left(t_u + \frac{2\pi}{\Omega}\right) = v_u = v_u^-, \dot{y}(t_u) = -k v_u = v_u^+ \quad (25)$$

gde je sa $v_u^- = v_u$ obeležena brzina neposredno pre udara (brzina neposredno nakon udara je $v_u^+ = -k v_u$). Opšte rešenje jednačine kretanja se sastoji iz homogenog i partikularnog dela

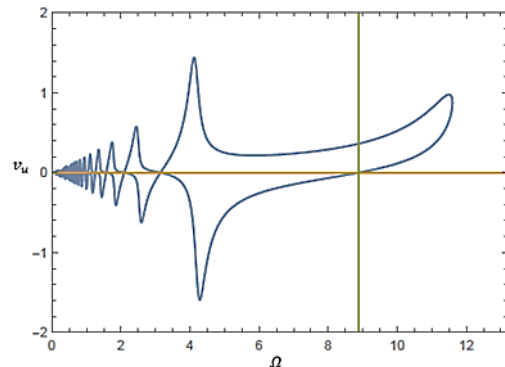
$$y(t) = C_1 \cos(\omega t) + C_2 \sin(\omega t) + A \cos(\Omega t) \quad (26)$$

Gde je

$$A = \frac{u_0 \omega^2}{\omega^2 - \Omega^2} \quad (27)$$

Iz izraza (24),(25),(26) rešavanjem i sređivanjem može se izvesti kvadratna jednačina za brzinu u trenutku udara.

$$v_u^2 \left(\frac{(-1+k)^2 (\Omega^2 - \omega^2)^2}{4u_0^2 \Omega^2 \omega^4} + \frac{(1+k)^2 (-\Omega^2 + \omega^2)^2 \cot\left(\frac{\pi\omega}{\Omega}\right)}{4u_0^2 \omega^6} \right) + v_u \left(\frac{(1+k)\Delta(-\Omega^2 + \omega^2)^2 \cot\left(\frac{\pi\omega}{\Omega}\right)}{u_0^2 \omega^5} \right) - 1 + \frac{\Delta^2 (-\Omega^2 + \omega^2)^2}{u_0^2 \omega^4} = 0 \quad (28)$$



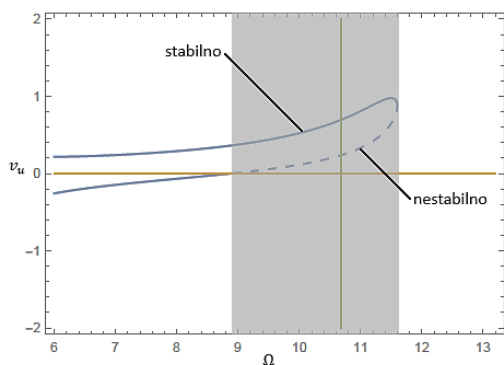
Slika 9. Brzine udara u zavisnosti od frekvencije ($\omega = 2\pi\text{ s}^{-1}$, $\Delta = 0.02\text{m}$, $k = 0.8$, $u_0 = 0.2\text{m}$)

Na slici 9 prikazana je brzina u trenutku udara v_u u zavisnosti od frekvencije Ω . Kvadratna jednačina za brzinu u trenutku udara ima u opštem slučaju dva rešenja. Negativna rešenja se neće razmatrati jer su fizički nemoguća. Vertikalnom linijom je predstavljena frekvencija za koju je brzina udara jednaka nuli i ona za podatke sa slike 9 iznosi $\Omega = \sqrt{2}\omega$.

Na delu dijagrama kada je $\Omega \ll \omega$ mogu se javiti oscilacije sa dva udara po periodu, oscilacije bez udara, kao i rešenja gde je teško definisati da li udar postoji te se ovaj deo neće razmatrati.

Pošto većina mašina radi u rezonantnom, a vrlo često i u postrezonantnom režimu za nas je zanimljivije da se ispituju frekvencije u ovim oblastima.

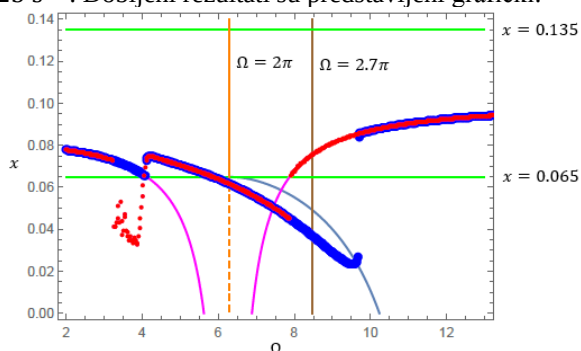
Detaljnijom analizom može se pokazati da u oblasti sa dva pozitivna rešenja sa udarom, osenčena oblast na slici 10, jedno rešenje – rešenje sa većom brzinom – je stabilno, dok je drugo – rešenje sa manjom brzinom – nestabilno.



Slika 10. Oblast sa dva pozitivna rešenja za brzinu udara

5. AMPLITUDNO-FREKVENTNI DIJAGRAM PRINUDNIH OSCILACIJA

U programskom paketu Wolfram Mathematica® numerički je rešena diferencijalna jednačina kretanja (18) sa uslovima (22), za vrednosti frekvencije u rasponu od $\Omega = 2 \text{ s}^{-1}$ do $\Omega = 14 \text{ s}^{-1}$. Počevši od $\Omega = 2 \text{ s}^{-1}$ povećavana je frekvencija za 0.025 s^{-1} . Za svaku vrednost Ω nađena je minimalna vrednost pomeranja tj. amplitude. Kao početni uslovi vrednosti za brzinu i pomeranje korišćeni su rezultati iz prethodne iteracije. Ponoviće se ista procedura numeričkog rešavanja jednačine kretanja, samo sada će se početi od najviše vrednosti za frekvenciju $\Omega = 14 \text{ s}^{-1}$. Frekvencija će se isto smanjivati za korak 0.025 s^{-1} . Dobijeni rezultati su predstavljeni grafički.



Slika 11. Amplitudno frekventni dijagram $k = 0.5, \delta = 0.05, \omega = 2\pi \text{ s}^{-1}, l_0 = 1\text{m}, \Delta = 0.035\text{m}, u_0 = 0.02\text{m}$

Na slici 11 su prikazani tačkasti dijagrami zavisnosti amplitude od frekvencije dobijeni numeričkim putem kao i „back-bone curve“ sa slike 4 i amplitudno frekventni dijagram za oscilacije sa prinudom bez prigušenja sa slike 5. Plavim većim tačkicama je prikazana zavisnost amplitude od frekvencije, dobijena povećavanjem frekvencije korakom od 0.025 s^{-1} , počevši od $\Omega = 2 \text{ s}^{-1}$. Crvenim manjim tačkicama je prikazana zavisnost amplitude od frekvencije, dobijena smanjivanjem frekvencije korakom od 0.025 s^{-1} , počevši od $\Omega = 14 \text{ s}^{-1}$. Numerički rezultati su približni kada postoji udarno rešenje i dobro prate krivu („back-bone curve“) amplitudno frekventnog dijagrama za oscilacije bez prinude. U prerezonantnoj oblasti dolazi do prekida numerički dobijene amplitudno frekventne krive.

Numerički dobijene vrednosti za amplitudu na mestu prekida sugerišu da u ovoj oblasti postoji udarno rešenje. Ukoliko se prikažu oscilacije za neku frekvenciju iz ove oblasti može se videti da postoje oscilacije sa udarom, međutim između dva udara postoji i neudarni „pik“. Ova oblast nije pogodna za izbor frekvencija pri projektovanju vibroudarnih mašina.

6. ZAKLJUČAK

U radu su analizirane prinudne oscilacije vibroudarnog sistema. U prvom delu rada ispitane su slobodne oscilacije, postavljen je uslov za postojanje udara, i analizirane su oscilacije bez udara, i oscilacije kada se udar javlja. Prikazana je zavisnost pomeranja (amplitude) od frekvencije „back-bone curve“ za oscilacije bez prinude. U nastavku su ispitane prinudne oscilacije. Prikazan je amplitudno frekventni dijagram prinudnih oscilacija sa prigušenjem i bez prigušenja.

Rešena je diferencijalna jednačina prinudnih oscilacija sa uslovima za postojanje udara za različite vrednosti frekvencije prinude. Za istu frekvenciju prinude dobijeno je rešenje bez udara i rešenje sa udarom u zavisnosti od početnih uslova. Izračunata je brzina u trenutku udara u zavisnosti od frekvencije. Kada postoje dva pozitivna rešenja za brzinu udara, jedno rešenje – rešenje sa većom brzinom – je stabilno, dok je drugo – rešenje sa manjom brzinom – nestabilno.

U petom delu je sprovedena numerička analiza, dobijen je amplitudno frekventni dijagram prinudnih oscilacija sa udarom. Kada postoji udar, numerički rezultati dobro prate „back-bone curve“ slobodnih oscilacije bez prinude. Pri izboru radne frekvencije, treba biti oprezan, jer se može desiti da se u želji da amplituda udara bude što veća, radna frekvencija odabere previše blizu oblasti u kojoj dolazi do prelaska udarnog u neudarno rešenje.

7. LITERATURA

- [1] Božidar D. Vujanović, Teorija oscilacija, Fakultet tehničkih nauka Novi Sadu, STYLOS, Novi Sad, 1995.
- [2] V. I. Babitsky, Theory of Vibro-Impact Systems and Applications, Foundations of Engineering Mechanics, Springer, 1998, ISBN 978-3-662-22534-9
- [3] Yang Liu, Marian Wiercigroch, Ekaterina Pavlovskaja, Hongnian Yu, Modelling of a vibro-impact capsule system, International Journal of Mechanical Sciences, Volume 66, 2013, Pages 2-11, ISSN 0020-7403

Kratka biografija:



Petar Tošić rođen je u Sremskoj Mitrovici 1991. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mašinstvo – Tehnička mehanika i dizajn u tehnici odbranio je 2018.god.

kontakt: tosic.pera91@gmail.com

ODREĐIVANJE KOEFICIJENTA TRENJA METODOM SLOBODNOG SABIJANJA KONUSNOG PRSTENA**FRICITION COEFFICIENT DETERMINATION BY CONICAL RING UPSETTING**Nemanja Dačević, Marko Vilotić, *Fakultet tehničkih nauka, Novi Sad***Oblast – MAŠINSTVO**

Kratak sadržaj – U radu je prikazan celokupan postupak određivanja vrednosti koeficijenta trenja metodom slobodnog sabijanja konusnog prstena u uslovima hladne obrade sa podmazivanjem uljem za materijal uzorka C45E.

Ključne reči: Koeficijent trenja, kalibracione krive, konusni prsten, sabijanje, numerička simulacija

Abstract – In this paper, a complete procedure of determination a friction coefficient using a conical ring upsetting test is presented. Friction coefficient is determined for cold upsetting process with oil lubrication for material of conical rings steel C45E.

Keywords: Friction coefficient, calibration curves, conical ring, upsetting, numerical simulation

1. UVOD

U postupcima plastičnim deformisanjem veličina kontaktnog trenja jedan je od odlučujućih faktora od kojih će zavisiti efikasnost projektovanog procesa i kvalitet dobijenog proizvoda. Kvantitativno određivanje kontaktnog trenja u obradama deformisanjem najčešće se vrši eksperimentalnim putem merenjem određenih pokazatelja veličine kontaktnog trenja [1,2]. U ovom radu prikazano je određivanje koeficijenta trenja metodom slobodnog sabijanja konusnog prstena u slučaju hladne zapreminske obrade sa podmazivanjem za materijal obratka C45E.

2. METODA SLOBODNOG SABIJANJA KONUSNOG PRSTENA

U zapreminskoj obradi plastičnim deformisanjem vrednost koeficijenta trenja najčešće se određuje eksperimentalnim putem. Eksperimentalne metode možemo da podelimo na direktne i indirektne metode. Direktne metode za određivanje veličine koeficijenta trenja podrazumevaju eksperimentalno merenje lokalne sile trenja ili nekih drugih pokazatelja u zavisnosti od primenjene metode i vrste obrade, dok indirektne metode podrazumevaju praćenje određenih geometrijskih veličina u uslovima podeljenog tečenja materijala.

Neke od najčešće primenjivanih indirektnih metoda su: metoda slobodnog sabijanja prstena, metoda konusnog pritiskivača, metoda slobodnog sabijanja konusnog prstena, metoda sabijanja dugačke lamele i tako dalje [3].

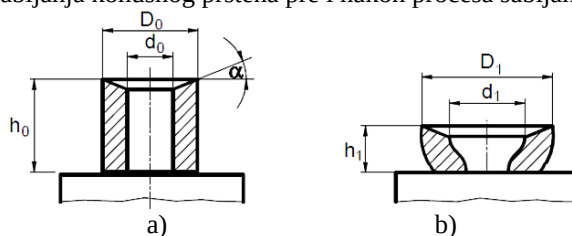
NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragiša Vilotić, red. prof.

Metoda slobodnog sabijanja konusnog prstena predstavlja varijaciju metode slobodnog sabijanja prstena razvijenu s ciljem prevazilaženja njenih nedostataka i povećanja tačnosti pri određivanju većih vrednosti koeficijenta trenja [4].

Osnovnu prednost ove metode predstavlja promenljiva veličina ugla konusa α . Ugao konusa najčešće se kreće u granicama između 10° i 20° . Pravilnim izborom veličine ugla konusa, ovom metodom moguće je eliminisati pojavu statičkog trenja i lepljenja materijala uzorka za alat, i na taj način obezbediti skoro ravnomernu brzinu tečenja materijala u radijalnom pravcu na kontaktnoj površini, pri bilo kojoj veličini koeficijenta trenja. Takođe, pri različitim vrednostima ugla konusa, menja se i osetljivost metode na različite veličine koeficijenta trenja. Primenom uzoraka sa većim uglom konusa postiže se veća osetljivost metode na veće vrednosti koeficijenta trenja, i obrnuto [4,5].

Na slici 1. prikazana je skica uzorka za metodu slobodnog sabijanja konusnog prstena pre i nakon procesa sabijanja.



Slika 1. Uzorak za metodu slobodnog sabijanja konusnog prstena [4] a) pre sabijanja b) nakon sabijanja

Uzorak sa konusnom kontaktnom površinom, prikazan na slici 1a, postavlja se na donji ravni alat i zatim sabija gornjim konusnim alatom. Koeficijent trenja određuje se na osnovu promene dimenzija visine h i prečnika D nakon sabijanja u odnosu na početne vrednosti. Pri većim koeficijentima trenja radijalno tečenje materijala biće manje, odnosno promena prečnika D biće manja, i obrnuto pri manjim koeficijentima trenja. Na osnovu rezultata merenja izračunava se relativna promena visine i prečnika nakon deformisanja u odnosu na početne vrednosti i dobijena vrednost se unosi na etalon dijagram za određivanje koeficijenta trenja. Etalon dijagram za ovu metodu, usled kompleksnosti geometrije uzorka, moguće je dobiti isključivo primenom numeričke simulacije sabijanja uzorka variranjem koeficijenta trenja.

Osnovni nedostaci primene ove metode jesu viša cena izrade uzoraka i odgovarajućih alata kao i pojava nižih vrednosti kontaktnih napona usled konusne kontaktne površine. Takođe, pojava nesaosnosti konusnog prstena i alata pri pozicioniranju negativno utiče na aksijalnu simetriju uzorka nakon sabijanja, što može dovesti do

nesigurnosti pri određivanju koeficijenta trenja s obzirom da je vrednost prečnika uzorka nakon deformisanja glavni pokazatelj kontaktnih uslova [4,5].

3. EKSPERIMENTALNA ISTRAŽIVANJA

Eksperiment sabijanja konusnih prstenova sastoji se u sabijanju pet uzoraka ugla konusa $8^\circ 30'$ i pet uzoraka ugla konusa 15° pomoću odgovarajućih alata. Umesto, uobičajenog inkrementalnog sabijanja jednog uzorka za dobijanje potrebnih promena dimenzija, primenom više uzoraka za sabijanje pri različitom hodu alata izbegava se neophodno podmazivanje nakon svakog inkrementa i remećenje sloja sredstva za podmazivanje.

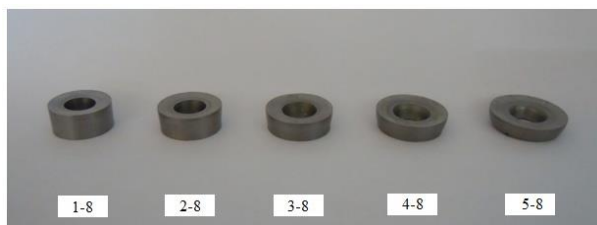
Kao sredstva za podmazivanje primenjivani su:

- konusna kontaktna površina - ulje za hladno oblikovanje metala „Modriča“
- ravna kontaktna površina - stearinska kiselina

Pretpostavlja se da je sloj sredstva za podmazivanje ujednačen po celoj kontaknoj površini i identičan na svakom uzorku. Važi kako za ulje, tako i za stearinsku kiselinu kao primenjivana sredstva za podmazivanje.

Za sabijanje uzoraka korišćena je hidraulična presa Sack&Kiessbach nominalne sile 6,3 MN. Sabijanje uzoraka izvršeno je pri brzini deformisanja od 1 mm/min

Na slici 2. prikazani su uzorci ugla konusa $8^\circ 30'$ nakon sabijanja, dok su na slici 3. prikazani uzorci ugla konusa 15° . Uzorci oznaka 1-8 i 1-15 sabijani su do dostizanja hoda alata od 1mm. Pri sabijanju svakog sledećeg uzorka hod alata je povećavan za po 1 mm, sve do dostizanja maksimalnog hoda alata od 5 mm pri sabijanju poslednjih uzoraka, oznaka 5-8 i 5-15.



Slika 2. Uzorci ugla konusa $8^\circ 30'$ nakon sabijanja

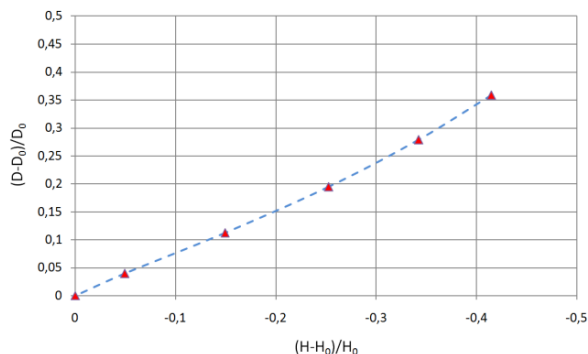


Slika 3. Uzorci ugla konusa 15° nakon sabijanja

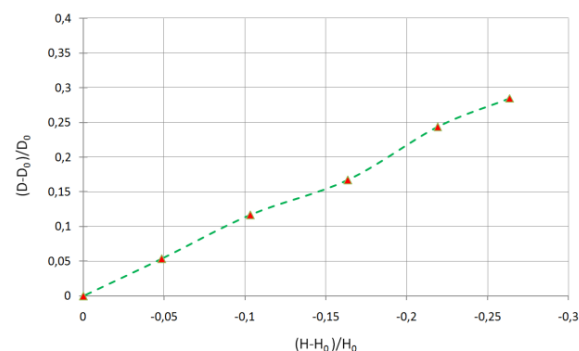
Za merenje dimenzija uzoraka pre i nakon procesa sabijanja korišćeno je digitalno pomično kljunasto merilo tačnosti 0,01 mm. S obzirom na veliki uticaj nesaosnosti uzoraka i alata na dimenzije uzoraka nakon sabijanja, a samim tim i na tačnost dobijene vrednosti koeficijenta trenja, merenje potrebnih dimenzija izvršeno je u četiri pozicije, a kao merodavna veličina korišćena je izračunata srednja vrednost izvršenih merenja.

Na osnovu izračunatih srednjih vrednosti prečnika D i visine h uzoraka nakon sabijanja, izračunate su relativne promene prečnika i visine, i formirane krive za procenu

koeficijenta trenja, prikazane na slici 4. za uzorke ugla konusa $8^\circ 30'$, i na slici 5. za uzorke ugla konusa 15° .



Slika 4. Kriva relativne promene visine i prečnika uzorka ugla konusa $8^\circ 30'$



Slika 5. Kriva relativne promene visine i prečnika uzorka ugla konusa 15°

4. DOBIJANJE ETALON DIJAGRAMA POMOĆU NUMERIČKE SIMULACIJE

Etalon dijagrami za određivanje koeficijenta trenja predstavljaju skup kalibracionih krivih, gde svakoj kalibracionoj krivoj odgovara određena vrednost koeficijenta trenja.

Kalibracione krive za određivanje koeficijenta trenja metodom slobodnog sabijanja konusnog prstena moguće je dobiti primenom numeričke simulacije, i u tu svrhu korišćen je programski paket Simufact.Forming 12.

Definisanje materijala za potrebe simulacije izvršeno je manuelno. Kriva tečenja materijala C45E određena je eksperimentalnim putem metodom Rastegajeva i za potrebe simulacije korišćen je njen analitički oblik:

$$K = 289,671 + 757,0366 \cdot \phi^{0,337698} \quad (1)$$

Ostali podaci o materijalu korišćeni za potrebe simulacije dati su u tabeli 1.

Tabela 1. Karakteristike materijala C45E

Modul elastičnosti [MPa]	210000
Gustina materijala [kg/m ³]	7800
Poisson-ov koeficijent	0,33

Primenjeni alati modelovani su kao apsolutno kruta tela usled čega nije potrebno izvršiti izbor materijala za iste. Kao i pri eksperimentalnom istraživanju, i u simulaciji je korišćena hidraulična presa sa konstantnom brzinom kretanja alata od 1 mm/min.

Definisanje kontaknog trenja izvršeno je pomoću Columb-ovog modela trenja. Izvršene su simulacije za vrednosti ko-

eficijena trenja za konusnu kontaktnu površinu od $\mu=0,00$ do $\mu=0,50$ u razmacima po 0,05. Za ravnu kontaktnu površinu, u svakoj simulaciji, korišćena je vrednost koeficijenta trenja $\mu = 0,012$.

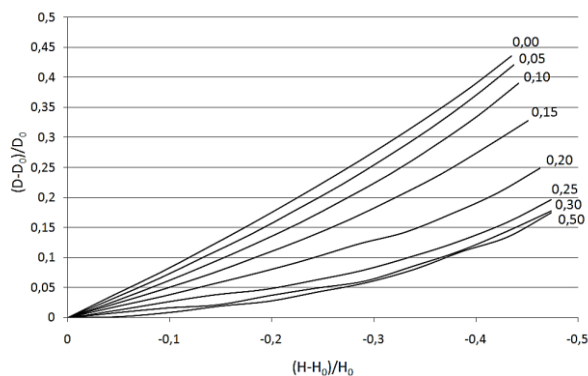
S obzirom da su i alat i uzorak rotaciono simetrični, kako bi se smanjilo vreme trajanja simulacije bez gubitka na tačnosti rezultata, umesto 3D simulacije, korišćena je 2D simulacija.

Za diskretizaciju modela na mrežu konačnih elemenata, kao konačni element izabran je kvadratni konačni element sa četiri čvora. Formiranje mreže konačnih elemenata izvršeno je na osnovu veličine konačnog elementa kako bi se obezbedila što približnija tačnost rezultata pri sabijanju uzoraka ugla konusa $8^\circ 30'$ i uzoraka ugla konusa 15° . Veličina konačnog elementa definisana je dužinom između dva susedna čvora, i u ovom slučaju iznosi 0,16 mm.

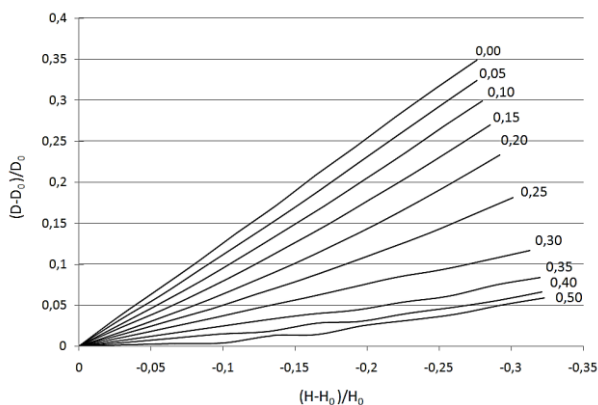
Sve simulacije sabijanja konusnih prstenova podešene su tako da kao izlaz pružaju deset rezultata do dostizanja hoda alata od 5 mm. S obzirom da je podešen broj rezultata koje softver beleži deset, kalibracione krive sastoje se od deset parova vrednosti relativne promene visine i prečnika uzoraka.

Na slici 6. prikazan je dobijeni etalon dijagram za uzorak ugla konusa $8^\circ 30'$, dok je na slici 7. prikazan dobijeni etalon dijagram za uzorak ugla konusa 15° .

Poređenje ova dva etalon dijagrama, za vrednosti ugla konusa $8^\circ 30'$ i 15° , zaključuje se da su uzorci ugla konusa $8^\circ 30'$ pogodniji pri određivanju koeficijenta trenja ugraničama od $\mu = 0,10$ do $\mu = 0,20$, dok uzorci ugla konusa 15° pružaju najveću osetljivost pri određivanju većih vrednosti koeficijenta trenja, u granicama od $\mu = 0,20$ do $\mu = 0,30$.



Slika 6. Etalon dijagram za određivanje koeficijenta trenja za ugao konusa $8^\circ 30'$

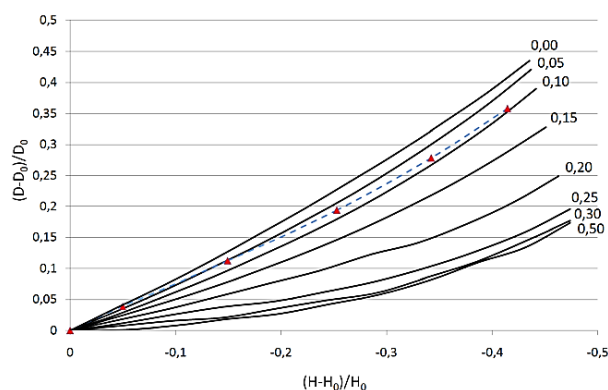


Slika 7. Etalon dijagram za određivanje koeficijenta trenja za ugao konusa 15°

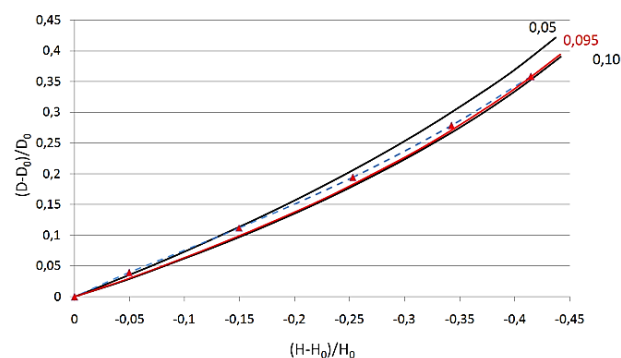
5. ODREĐIVANJE KOEFICIJENTA TRENJA

Određivanje vrednosti koeficijenta trenja izvršeno je komparacijom eksperimentalno dobijene krive za uzorke ugla konusa $8^\circ 30'$ i 15° sa odgovarajućim etalon dijagramom. Za vrednost koeficijenta trenja uzima se ona vrednost koja odgovara kalibracionoj krivoj sa najboljim preklapanjem sa eksperimentalno dobijenom tačkom pri ostvarenoj najvećoj deformaciji.

Poređenje eksperimentalno dobijene krive sa etalon dijagramom za ugao konusa $8^\circ 30'$ prikazano je na slici 8, sa koje se vidi da se vrednost koeficijenta trenja nalazi u granicama od $\mu = 0,05$ do $\mu = 0,10$. Kako bi se koeficijent trenja što bolje procenio, izvršene su dodatne simulacije za vrednosti koeficijenta trenja iz datog raspona. Najbolje podudaranje eksperimentalno dobijene tačke sa kalibracionom krivom dobijeno je pri vrednosti koeficijenta trenja $\mu = 0,095$, kao što je prikazano na slici 9.

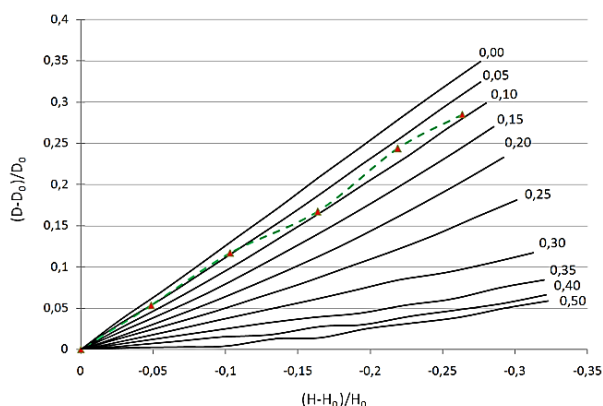


Slika 8. Poređenje eksperimentalno dobijene krive sa etalon dijagramom za ugao konusa $8^\circ 30'$

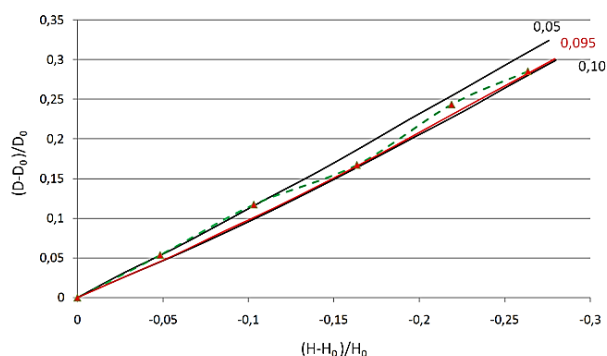


Slika 9. Određivanje koeficijenta trenja za ugao konusa $8^\circ 30'$

Na slici 10. prikazana je komparacije eksperimentalno dobijene krive sa etalon dijagramom za uzorak ugla konusa 15° . I u ovom slučaju, za približnije određivanje koeficijenta trenja neophodno je bilo izvršiti dodatne simulacije za vrednosti koeficijenta trenja od $\mu = 0,05$ do $\mu = 0,10$. Nakon izvršenih dodatnih simulacije zaključeno je da se najbolje podudaranje eksperimentalno dobijene tačke sa kalibracionom krivom dobija pri vrednosti koeficijenta trenja $\mu = 0,095$, kao što je prikazano na slici 11.



Slika 10. Poređenje eksperimentalno dobijene krive sa etalon dijagramom za ugao konusa 15°



Slika 11. Određivanje koeficijenta trenja za ugao konusa 15°

Kao vrednost koeficijenta trenja, određenog metodom slobodnog sabijanja konusnog prstena, usvaja se $\mu = 0,095$.

6. ZAKLJUČAK

Metodom slobodnog sabijanja konusnih prstenova primenom dva različita tipa uzoraka, ugla konusa 8° 30' i 15°, određena je vrednost koeficijenta trenja u slučaju hladne obrade sa podmazivanjem uljem za material uzorka C45E. Za oba tipa uzoraka dobijena je vrednost koeficijenta trenja: $\mu = 0,095$.

S obzirom da se u datim uslovima vrednost koeficijenta trenja kreće u granicama od $\mu = 0,08$ do $\mu = 0,15$ dobijeno rešenje se smatra odgovarajućim.

7. LITERATURA

- [1] H. Valberg, "Applied metal forming", Cambridge, Cambridge University Press, 2010.
- [2] V. Mandić, "Fizičko i numeričko modeliranje procesa obrade deformisanjem", Kragujevac, Fakultet inženjerskih nauka, 2012.
- [3] M. Plančak, Z. Car, M. Kršulja, D. Vilotić, I. Kačmarčik, D. Movrin, "Possibilities to measure contact friction in bulk metal forming", *Tehnicky vjesnik*, Vol. 19(4), pp. 727-723, 2012.
- [4] M. Teller, A. Klubakov, M. Franzke, J. Lohmar, G. Hirt, "Extended Conical Tube-upsetting Test to Investigate the Evolution of Friction Conditions", *Key Engineering Materials*, Vol. 716, pp. 157-164, 2016.
- [5] M. Teller, I. Ross, A. Temmler, R. Poprawe, S. Prünte, J. M. Schneider, G. Hirt, "Investigation of Friction Conditions in Dry Metal Forming of Aluminum by Extended Conical Tube-Upsetting Tests", *Key Engineering Materials*, Vol. 767, pp. 189-195, 2018.

Kratka biografija:



Nemanja Dačević rođen u Novom Sadu 1994. god. Osnovne studije završio je na Fakultetu tehničkih nauka u Novom Sadu 2017. godine. Master studije završava 2018. godine na Fakultetu tehničkih nauka u Novim Sadu na departmanu za Proizvodno mašinstvo, smer Savremene tehnologije oblikovanja materijala. kontakt: nemanjadacevic@gmail.com



Marko Vilotić rođen u Novom Sadu 1979. god. Diplomirao osnovnih i magistarskih studija, kao i doktorat stekao je na Fakultetu tehničkih nauka. Na Fakultetu tehničkih nauka (Katedra za tehnologije oblikovanjem i inženjerstvo površina) radi kao docent.

TRIBODIJAGNOSTIKA KINEMATSKOG KOEFICIJENTA TRENJA METODOM BLOK NA DISKU**TRIBODIAGNOSTICS OF THE KINETIC FRICTION COEFFICIENT BY BLOCK ON DISK METHOD**

Radenko Blažević, Đorđe Vukelić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PROIZVODNO MAŠINSTVO

Kratak sadržaj – U radu su prezentovani rezultati merenja kinematskog koeficijenta trenja na tribometru po metodi blok na disku. Blok i disk su izrađeni od čelika C45. Eksperimentalna istraživanja su sprovedena za različite nivoe broja obrta diska i pri različitim nivoima normalnog opterećenja bloka. Dobijeni rezultati su analizirani, diskutovani i komparirani. Na kraju su dati odgovarajući zaključci i pravci budućih istraživanja.

Ključne reči: Tribometar, kinematski koeficijent trenja

Abstract – The paper presents the results of measuring the kinetic friction coefficient on the tribometer by the disk on block method. Block and disc are made of steel C45. Experimental investigations were carried out for different levels of revolutions per minute, and at different levels of normal load. The obtained results were analysed, discussed and compared. At the end, appropriate conclusions and directions of future research are given.

Keywords: Tribometer, kinetic friction coefficient

1. UVOD

Tribologija je naučno-tehnička disciplina koja obuhvata istraživanja procesa trenja, habanja i podmazivanja, kao i postupaka za smanjenje i optimizaciju ovih procesa u različitim mašinskim sistemima [1]. U proizvodnim sistemima pod pojmom mehanički sistem podrazumevaju se sve vrste obradnih procesa, proizvodna i druga oprema. Obrada rezanjem odvija se, na primer, u mehaničkom sistemu koji čine alat i predmet obrade u relativnom kretanju. Pod pojmom tribo-mehanički sistem podrazumevaju se oni mehanički sistemi čije funkcionisanje zavisi u velikoj meri od veličine trenja i intenziteta habanja u zonama kontakta njihovih elemenata. Broj tribo-mehaničkih sistema u proizvodnim sistemima srednje veličine iznosi, po pravilu, nekoliko desetina hiljada. U njima se, za vreme trajanja proizvodnih procesa, pojavljuje trenje, u zonama kontakta koje treba savladati i habanje kritičnih elemenata, koje dovodi do zastoja proizvodnje zbog zamene pohabanih elemenata novim [2].

U većini slučajeva, u različitim pokretnim elementima mašina, uređaja i postrojenja, trenje je nepoželjna i štetna pojava. Na savladavanje sile trenja kao otpora kretanja bilo koje vrste, troši se znatna količina mehaničke energije.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Đorđe Vukelić, vanr. prof.

Kao posledica trenja na površinama u kontaktu dolazi do trošenja i zagrevanja površina. Međutim, trenje i trošenje su u nekim slučajevima neophodna i korisna fizička pojava.

Karakteristike tribo-mehaničkih sistema, u kojima se odvijaju procesi trenja i habanja, za vreme trajanja proizvodnih procesa, zavise od vrste materijala i uslova pod kojim se kontakt u njima ostvaruje (brzina relativnog kretanja, spoljašnje opterećenje, režim podmazivanja i sl.). Kontakt u osnovnim tribo-mehaničkim sistemima može biti između čvrstih elemenata izrađenih od različitih materijala, sa ili bez prisustva maziva [3].

Osnovni tribo-mehanički sistemi, bilo kojoj grupi da pripadaju, imaju iste ulazne, izlazne i izgubljene veličine. Ulazne veličine su masa, energija, kretanje, informacije i materijal. Izlazne i izgubljene veličine su iste vrste, ali sa promenjenim količinama. Izgubljene veličine su gubitak mase koji se javlja zbog postojanja procesa habanja u sistemu, gubitak energije kao posledica postojanja trenja u zonama kontakta i gubitak kretanja, koji se javlja kao posledica pojave klizanja i/ili kotrljanja [4].

Rešavanjem problema u zonama kontakta osnovnih tribo-mehaničkih sistema koji se odnose na smanjenje trenja i usporavanje procesa habanja, smanjuju se u značajnoj meri troškovi u proizvodnim sistemima. Primenom savremenih triboloških znanja na rešavanju problema trenja i habanja u osnovnim tribo-mehaničkim sistemima moguće je smanjiti [5]:

- Troškove održavanja i zamene pohabanih delova novim;
- Troškove proizvodnje koji nastaju zbog zastoja proizvodnih procesa izazvanih habanjem elemenata proizvodne i druge opreme;
- Troškove investicija u novu opremu zbog povećanja veka trajanja postojećeg;
- Troškove energije zbog smanjenja trenja;
- Troškove podmazivanja;
- Troškove radne snage.

Tribološke karakteristike čvrstih materijala (elemenata tribo-mehaničkih sistema) i maziva određene su [5]:

- Intenzitetom sile trenja odnosno koeficijenta trenja u zoni kontakta elemenata sistema;
- Intenzitetom habanja kritičnog elementa sistema.

Izbor materijala i tehnologije obrade kontaktnih površina, izbor tehnologija za poboljšanje triboloških svojstava kontaktnih slojeva i izbor maziva vrši se isključivo na bazi trenja i intenziteta habanja kritičnog elementa u osnovnom tribo-mehaničkom sistemu. Kako je merenje

sila trenja ili koeficijenta trenja u realnim uslovima veoma teško, a često i nemoguće, ostaje samo mogućnost da se kroz simulaciju realne kontaktne situacije na tribometrima izvrši određivanje triboloških karakteristika merenjem sa aspekta trenja ili sa aspekta habanja [5].

2. TEORIJSKE OSNOVE

Trenje predstavlja mehanički otpor koji nastaje na površini dodira dva tela, a pri kretanju jednog tela po drugom ili pri međusobnom relativnom kretanju. Izražava se silom trenja ili koeficijentom trenja.

Prema stanju dodirnih površina, trenje se deli na [6]:

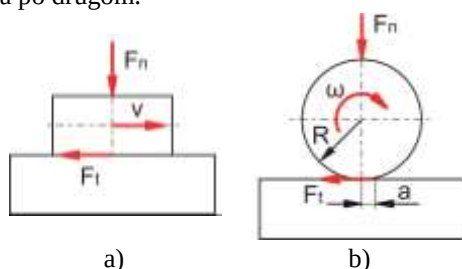
- suvo trenje - tela se dodiruju preko nepodmazanih površina,
- mokro trenje - tela se dodiruju preko podmazanih površina.

Prema agregatnom stanju tela, trenje se deli na [7]:

- unutrašnje trenje (viskoznost) - otpor međusobnom kretanju slojeva fluida (tečnosti i gasova) pod dejstvom kohezije i adhezije,
- spoljašnje trenje - otpor na površini dodira između čvrstih tela koji se suprotstavlja njihovom međusobnom kretanju.

Prema vrsti kretanja, trenje se deli na (slika 1) [8]:

- trenje klizanja - otpor koji se javlja na površinama dodira dva tela prilikom translacionog kretanja jednog tela po drugom,
- trenje kotrljanja - otpor koji se javlja na površinama dodira dva tela prilikom rotacionog kretanja jednog tela po drugom.



a) b)
Slika 1. Trenje
a) klizanja, b) kotrljanja

Sila trenja je proporcionalna normalnoj sili koja deluje na površine koje su u kontaktu, što se može prikazati sledećim izrazom [9]:

$$F_t = \mu \cdot F_n \quad (1)$$

gde je F_t –sila trenja, μ –koeficijent trenja i F_n –normalna sila. Po ovoj teoriji sila trenja ne zavisi od veličine površina koje su u dodiru i brzine klizanja, već zavisi jedino od opterećenja, stepena hrapavosti i prirode materijala elemenata.

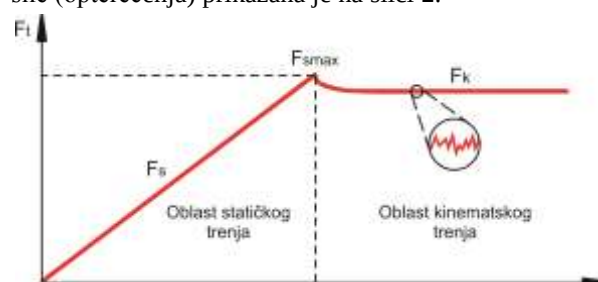
Koeficijent trenja ($\mu = F_t/F_n$), između ostalog, zavisi i od vrste materijala predmeta. Ako su tvrdoće kotrljajućeg tela i podloge dovoljno velike, a njihove površine idealne, može se pretpostaviti da će kod nekog opterećenja dodirno mesto biti tačka. Povećavanjem opterećenja, dolazi se u područje elastičnih deformacija, a dodirno mesto postaje površina. Usled savladavanja kotrljanja i deformacija, dolazi do porasta sile trenja i koeficijenta trenja. Iznad nekog graničnog opterećenja nastupa preopterećenje sa trajno plastičnim deformacijama.

Vrednosti koeficijenta trenja kotrljanja su znatno manje nego u slučaju klizanja u identičnim uslovima ostvarivanja kontakta [10].

Prema stanju kretanja tela, trenje se deli na [10]:

- trenje mirovanja (statičko trenje) - otpor koji se javlja na dodirnim površinama dva tela koja miruju,
- trenje kretanja (kinematsko trenje) - otpor koji se javlja na dodirnim površinama dva tela koja se kreću.

Zavisnost sile statičkog/kinematskog trenja klizanja od sile (opterećenja) prikazana je na slici 2.



Slika 2. Zavisnost sile statičkog/kinematskog trenja klizanja od sile tokom vremena

U stanju mirovanja sila trenja je na mestima kontakta površina veća ili jednaka sa rezultantom tangencijalnih sila koja izaziva kretanje. Trenje postoji na površini kontakta čvrstih tela i kada su tela u relativnom mirovanju, jer je potrebno uložiti znatnu silu da bi se izazvalo početno relativno kretanje. Nakon neke granične vrednosti sile telo se pokreće. Sila trenja potom raste do neke maksimalne vrednosti statičke sile trenja F_{smax} . Nakon toga sila trenja postaje kinematska sila trenja, a njena vrednost je nešto manja od statičke sile trenja $F_k < F_{smax}$. Potom se telo kreće savladavajući silu kinematskog trenja [10].

Vrste kinematskog trenja su [10]:

- trenje klizanja - pojava sile trenja ima za posledicu stvaranje toplote. To znači da važi zakon očuvanja energije: smanjivanje kinetičke energije tela prelazi u toplotnu energiju tela ili podloge (šibice funkcionišu upravo na tom principu gde ovaj efekat ima korisnu upotrebu);
- trenje kotrljanja - u slučaju kotrljanja tela po podlozi, trenje se javlja zato što se deo kinetičke energije troši na odvajanje od podloge, a deo na priljublivanje tela na podlogu. Sila trenja naročito dolazi do izražaja kada je podloga vlažna;
- trenje u tečnom sloju - u slučaju kretanja tela kroz neki fluid (tečnost ili gas) primećuje se usporavanje (sila trenja) čija veličina zavisi od vrste fluida kroz koji se telo kreće. Telo koje se kreće predaje deo svoje kinetičke energije atomima/molekulima fluida usled čega se delovi fluida počinju kretati (npr. nastajanje talasa).

3. ESPERIMENTALNA ISTRAŽIVANJA

Tribometar upotrebljen u eksperimentalnim istraživanjima (slika 3) koristi se za ispitivanje tribo-mehaničkih karakteristika različitih vrsta materijala u uslovima rada sa i bez primene maziva.



Slika 3. Tribometar

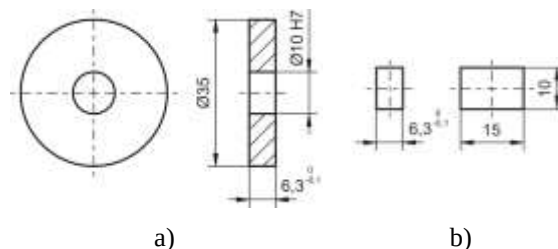
Osnovne karakteristike tribometra su:

- Fizički i merni opseg normalnog opterećenja pri merenju koeficijenta trenja: 0–140 N;
- Merni opseg sile trenja: 0–100 N;
- Opseg brzina klizanja: 0–3,6 m/s;
- Greška merenja < 1%;
- Uslovi merenja: sa i bez prisustva sredstva za hlađenje i podmazivanje sa mogućnošću regulacije protoka u intervalu 0–2 l/min;
- Fizički i konstruktivno onemogućen uticaj opterećenja na signal izmerene vrednosti sile trenja – onemogućeno mešanje signala;
- Mogućnost numeričkog i grafičkog prikaza rezultata merenja.

Procedura merenja svojstava na prikazanom tribometru je u skladu sa ASTM standardom [11]. Standardni uzorci diska i bloka sa kojima su izvršena eksperimentalna istraživanja su izrađeni od čelika C45. Geometrija diska i bloka je prikazana na slici 4.

Pre početka ispitivanja mere se parametri topografije diskova i blokova radi definisanja njihove hrapavosti. Merenje hrapavosti je izvršeno pomoću mernog uređaja "Surface Roughness Tester SR 9210".

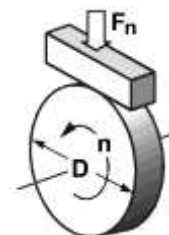
Hrapavost je merena po 10 izvodnica cilindra diska i u 10 pravaca kontaktne površine bloka. Srednja aritmetička vrednost hrapavosti diska je $R_a=1,604 \mu\text{m}$, a srednja aritmetička vrednost hrapavosti bloka je $R_a=1,602 \mu\text{m}$.



Slika 4. Geometrija a) diska, b) bloka

Merenje kinematskog koeficijenta trenja izvršeno je za kontakti par block on disk. Varirane su vrednosti normalnog opterećenja bloka F_n i broja obrta diska n (slika 5). Istraživanja su sprovedena za sedam različitih vrednosti normalnog opterećenja bloka $F_n = 5, 10, 15, 20, 25, 30, 35 \text{ N}$, i pet različitih brojeva obrta diska $N=120, 180, 240, 300$ i 360 obr/min . Izvršeno je $7 \times 5 = 35$ eksperimenata.

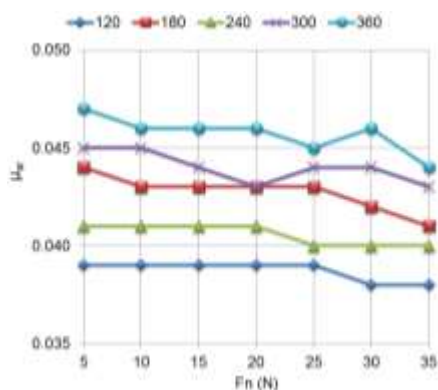
Vrednosti koeficijenta trenja su snimane tokom 30 sekundi. Eksperimentalna istraživanja su sprovedena u kontrolisanim uslovima, pri temperaturi 23°C , pritisku 1 bar i vlažnosti vazduha 55 %. Dobijeni rezultati, prikazani su u tabeli 1 i na slici 6.



Slika 5. Kontakti par blok na disku

Tabela 1. Rezultati eksperimentalnih istraživanja

n (o/min)	μ (-)	F_n (N)						
		5	10	15	20	25	30	35
120	μ_{sr}	0,039	0,039	0,039	0,039	0,039	0,038	0,038
	μ_{max}	0,041	0,042	0,042	0,041	0,041	0,041	0,042
	μ_{min}	0,035	0,034	0,037	0,036	0,036	0,035	0,034
180	μ_{sr}	0,044	0,043	0,043	0,043	0,043	0,042	0,041
	μ_{max}	0,046	0,047	0,047	0,046	0,046	0,044	0,044
	μ_{min}	0,036	0,036	0,035	0,037	0,036	0,036	0,036
240	μ_{sr}	0,041	0,041	0,041	0,041	0,040	0,040	0,040
	μ_{max}	0,046	0,046	0,045	0,045	0,044	0,043	0,042
	μ_{min}	0,036	0,037	0,037	0,036	0,036	0,037	0,036
300	μ_{sr}	0,045	0,045	0,044	0,043	0,044	0,044	0,043
	μ_{max}	0,047	0,046	0,046	0,046	0,046	0,046	0,046
	μ_{min}	0,042	0,042	0,041	0,041	0,042	0,039	0,039
360	μ_{sr}	0,047	0,046	0,046	0,046	0,045	0,046	0,044
	μ_{max}	0,049	0,049	0,048	0,047	0,047	0,047	0,046
	μ_{min}	0,042	0,042	0,044	0,043	0,042	0,042	0,042



Slika 6. Zavisnost srednje aritmetičke vrednosti kinematskog koeficijenta trenja kotrljanja od normalnog opterećenja i broja obrta

Na bazi tabele 1 i slike 6 mogu se izvesti sledeći zaključci:

- za kompletna eksperimentalna ispitivanja srednja aritmetička vrednost srednjih aritmetičkih vrednosti kinematskih koeficijenata trenja iznosi 0,042 sa standardnom devijacijom od 0,003,
- srednje vrednosti kinematskog koeficijenta trenja se kreću u rasponu od 0,038 (minimalna vrednost) do 0,047 (maksimalna vrednost),
- srednje vrednosti kinematskog koeficijenta trenja za pojedine brojeve obrta su približno konstantne vrednosti sa izuzetno blagom tendencijom smanjenja prilikom povećanja normalnog opterećenja,
- srednje vrednosti kinematskog koeficijenta trenja imaju malu tendenciju rasta sa povećanjem broja obrta za identične vrednosti normalnog opterećenja.

4. ZAKLJUČCI

S obzirom na savremene trendove koji se kreću u pravcu izrade i funkcionisanja tribo-mehaničkih sistema bez maziva ili uz njihovu minimalnu upotrebu eksperimentalna istraživanja su sprovedena bez prisustva maziva. Pored visoke cene jedan od osnovnih nedostataka maziva jeste to što predstavljaju opasne materije kako za radnu tako i za životnu sredinu. Kako bi se postigle značajne uštede i prevazišli troškovi kupovine, ali i troškovi prikupljanja i odlaganja maziva u toku i nakon upotrebe, tribo-mehanički sistemi koji funkcionišu bez upotrebe maziva, postaju sve aktuelniji. Tako, na primer, "suva" mašinska obrada dobija sve više na značaju sa razvojem i usavršavanjem prevlaka. Oslojeni elementi imaju znatno bolje karakteristike od neoslojenih, a obradu je moguće izvesti i bez upotrebe maziva - SHP-a.

Rezultati eksperimentalnih istraživanja za kontaktne parove napravljene od čelika C45, i predhodno definisane početne i eksperimentalne uslove, ukazuju da koeficijent trenja kotrljanja kreće oko 0,042. Tokom izvođenja eksperimentalnih ispitivanja koeficijent trenja nije prelazio vrednost od 0,049, a nije ni padoo ispod donje granice od 0,034.

Buduća istraživanja biće usmerena na merenja kinematskog koeficijenta trenja kontaktnih elemenata napravljenih od drugih materijala, sa različitim geometrijskim karakteristikama i pod različitim

eksperimentalnim uslovima (tipovi kontakta, mikroklimatski parametri, brojevi obrta, normalna opterećenja, bez ili sa primenom maziva, itd.). Takođe, u planu je da se izvrši merenje širine i traga habanja na kontaktnim površinama kontaktnih elemenata. Kontakt između bloka i diska ostvaruje se po liniji čija se širina (trag habanja) povećava sa vremenom ostvarivanja kontakta. Primenom adekvatnih merila može se snimiti profil pohabane površine, a promene visine (dubine) površine koja se haba u odnosu na tu referentnu (početnu) predstavlja linijski intenzitet habanja. Na taj način se može dobiti i vizuelna predstava o veličini pohabanog dela kontaktnih elemenata.

5. LITERATURA

- [1] A. Rac, „Osnovi tribologije“, Mašinski fakultet, Beograd, 1991.
- [2] B. Ivković, „Manufacturing Process Productivity Through Tribology“, *Robotics and Computer-Integrated Manufacturing*, Vol. 4, No. 1-2, pp.135-138, 1988.
- [3] S. Mitrović, M. Babić, D. Džunić, „Tribološka karakterizacija nanokompozita sa metalnom osnovom“, Fakultet inženjerskih nauka, Kragujevac, 2016.
- [4] B. Jeremić, „Tribološki procesi pri dinamičkom opterećenju“, Jugoslovensko Društvo za Tribologiju, Kragujevac, 1997.
- [5] B. Ivković, „Produktivnost i tribologija u industrijskim sistemima“, 8. *International Tribology Conference*, 8.-10. October, Belgrade, pp. 347-350, 2003.
- [6] A. Devaraju, „A review on important factors affecting dry sliding friction“, *Journal of Surface Science and Technology*, Vol. 32, No. 3-4, pp. 73-78, 2016.
- [7] J. Gandra, H. Krohn, R.M. Miranda, P. Vilaca, L. Quintino, J.F. Dos Santos, „Friction surfacing - A review“, *Journal of Materials Processing Technology*, Vol. 214, No.5, pp. 1062-1093, 2014.
- [8] K. Holmberg, A. Matthews, „Coatings Tribology, Properties, Techniques and Applications in Surface Engineering“, Elsevier, 1994.
- [9] J.P. Blau, „The significance and use of the friction coefficient“, *Tribology International*, Vol. 34, pp. 585-91, 2001.
- [10] S. Tanasijević, „Tribološki ispravno konstruisanje“, Mašinski fakultet, Kragujevac, 2004.
- [11] ASTM G77-17, „Standard Test Method for Ranking Resistance of Materials to Sliding Wear Using Block-on-Ring Wear Test“, ASTM International, 2017.

Kratka biografija:



Radenko Blažević rođen je u Banja Luci 1993. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mašinskog inženjerstva odbranio je 2018. god.

kontakt: blazevicradenko@gmail.com



Đorđe Vukelić rođen je u Novom Sadu 1974. godine. Doktorirao je na Fakultetu tehničkih nauka 2010. god., a od 2015 je zvanju vanrednog profesora.

SIMULACIJE RAZLIČITIH MANEVARA AUTOMOBILA MODELIRANOG NEHOLONOMNIM SISTEMOM

SIMULATIONS OF DIFFERENT MANEUVERS OF A CAR MODELED BY A NONHOLONOMIC SYSTEM

Nikola Đorđević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratka sadržaj – U ovom radu analizirana je dinamika automobila pri različitim manevrima pomoću Lineikinovog modela. Diferencijalne jednačine kretanja formirane su pomoću Lagranževih jednačina druge vrste za neholonomne sisteme. Različiti slučajevi kretanja automobila dobijeni su odgovarajućim izborom početnih uslova i upravljanja, a rešenja su prikazana grafički.

Ključne reči: dinamika vozila, Lineikin-ov model automobila, neholonomni sistem, simulacija kretanja.

Abstract – This paper analyzes the dynamics of a car with different maneuvers using the Lineikin model. Differential equations of motion are formed using Lagrange equations of the second kind for nonholonomic systems. Different cases of car motion were obtained with appropriate selection of initial conditions and control. The solutions are presented graphically.

Keywords: Dynamics of car, Lineikin car model, nonholonomic system, simulation of movement.

1. UVOD

Za analizu dinamike vozila koriste se različiti matematički modeli, čija kompleksnost zavisi od količine detalja, mehanizama i procesa obuhvaćenih modelom. Najsloženiji modeli bi bili kada bi se posmatralo kretanje vozila u 3D prostoru i kada se ne bi zanemario ni jedan stepen slobode, ali se postavlja pitanje opravdanosti upotrebe toliko složenog modela. Za slučaj kada je u interesu istraživanja samo jedan aspekt kretanja vozila, praktikuje se, radi pojednostavljenja posmatranje vozila u zasebnim ravnima kretanja. Tako postoje modeli za posmatranje longitudinalnog (uzdužnog), lateralnog (bočnog) i vertikalnog kretanja, koji su razmatrani od strane velikog broja autora, videti [1]. Za realno opisivanje kretanja vozila ne smeju se zanemariti mehaničke osobine pneumatika, videti [2], [3].

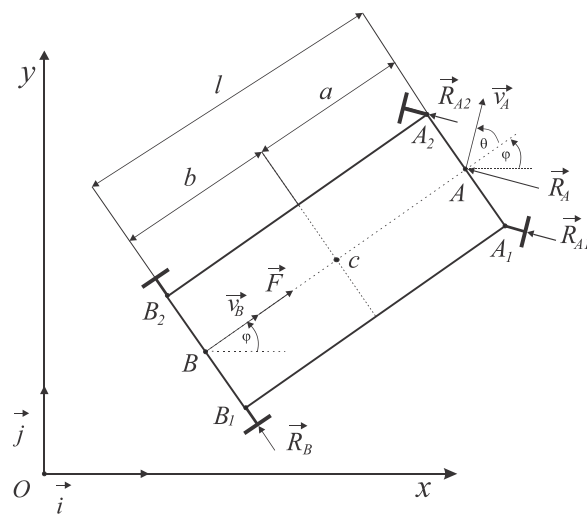
Kada se posmatra lateralno kretanje, veliki uticaj ima bočno povodenje točkova i otpori koji nastaju kao rezultat te pojave. Neki modeli koji uzimaju u obzir ovaj fenomen navedeni su u nastavku rada. Interesantno bi bilo uporediti rezultate dinamičke analize korišćenjem modela koji uzimaju i onih koji ne uzimaju u obzir bočno povodenje. U ovom radu proučeni su različiti slučajevi

kretanja vozila primenom Linei kinovog modela, koji uzima u obzir neholonomne veze usled kotrljanja točka po podlozi bez klizanja, ali ne obuhvata modeliranje deformabilnih svojstava pneumatika, videti [4]. Sa druge strane, razvijeni su razni softverski paketi za analizu dinamike automobila, pomoću kojih je moguće uzeti u obzir deformacije pneumatika i njihov uticaj na samo kretanje, te izvršiti pomenuto poređenje.

2. LINEIKINOV MODEL AUTOMOBILA

Lineikinov model opisuje kretanje automobila u horizontalnoj ravni Oxy referentnog koordinatnog sistema. Za proučavanje kretanja ovakvog modela moraju se usvojiti određene pretpostavke pre nego što se krene sa analizom.

- Obrtanje točkova nije relevantno.
- Nema proklizavanja između točkova i podloge.



Slika 1. Lineikinov model automobila

Na slici 1 dat je prikaz sledećih karakteristika sistema: l međuosovinsko rastojanje; a rastojanje prednje osovine od centra mase; b rastojanje zadnje (pogonske) osovine od centra mase; v_A brzina tačke A koja se nalazi na sredini prednje osovine; v_B brzina tačke B koja se nalazi na sredini zadnje osovine; Sila F predstavlja pogonsku silu vozila; φ ugao zakretanja vozila u odnosu na pozitivni smer x ose; θ ugao zakretanja prednjih točkova u odnosu na uzdužnu osu vozila. Brzine tačaka A i B se mogu napisati na sledeći način:

$$\vec{v}_A = \dot{x}_A \vec{i} + \dot{y}_A \vec{j}; \quad \vec{v}_B = \dot{x}_B \vec{i} + \dot{y}_B \vec{j} \quad (1)$$

Koordinate tačaka A i B su date sledećim izrazima:

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Miodrag Žigić, doc.

$$\begin{aligned} x_B &= x_C - b \cos \varphi; \quad y_B = y_C - b \sin \varphi \\ x_A &= x_C + a \cos \varphi; \quad y_A = y_C + a \sin \varphi \end{aligned} \quad (2)$$

Za opisivanje ovog sistema koristiće se Lagranževe jednačine druge vrste za neholonomne sisteme, čiji je osnovni oblik, videti npr. [5]:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = Q_j^* + \sum_{\beta=1}^l \lambda_{\beta} b_{\beta j} \quad (i=1, \dots, m) \quad (3)$$

Gde su q_j generalisane koordinate

$$q_j \quad (j=1, \dots, m), \quad (4)$$

a λ_{β} ($\beta=1, \dots, l$) Lagranževi množitelji. Generalisane koordinate predstavljaju parametre na osnovu kojih se može u svakom vremenskom trenutku jednoznačno odrediti položaj sistema. Za generalisane koordinate ovog sistema uzete su koordinate centra mase x_C, y_C , ugao zakretanja vozila φ i ugao zakretanja prednjih točkova θ . Jednačine neholonomnih veza, čiji je opšti oblik

$$\sum_{i=1}^n b_{\beta j} \dot{q}_j + b_{\beta} = 0 \quad (\beta=1, 2, \dots, l) \quad (5)$$

i kojih u analiziranom problemu ima dve, $l=2$, nameću ograničenja na brzine tačaka A i B koje predstavljaju sredinu prednje i zadnje osovine respektivno, pri čemu v_A i v_B imaju pravac prednjih odnosno zadnjih točkova, videti sliku 1.

$$\operatorname{tg} \varphi = \frac{\dot{y}_B}{\dot{x}_B}; \quad \operatorname{tg} (\varphi + \theta) = \frac{\dot{y}_A}{\dot{x}_A} \quad (6)$$

Iz prethodnih jednačina sledi da je broj stepeni slobode:

$$n = m - l = 4 - 2 \quad (7)$$

Lagranžijan L jednak je:

$$\begin{aligned} L = E_k - \Pi &= \frac{1}{2} M v_C^2 + \frac{1}{2} J_C \dot{\varphi}^2 + \frac{1}{2} J_A \dot{\theta}^2 \\ \Pi &= 0 \end{aligned} \quad (8)$$

Gde je: M masa vozila; J_C aksijalni moment inercije pri zakretanju vozila za ugao φ ; J_A aksijalni moment inercije pri zakretanju upravljačkih točkova. Generalisane sile nepotencijalnih dejstava označene su sa Q_j^* , $j \in \{x_C, y_C, \varphi, \theta\}$, i izvedene su iz izraza za virtualni rad nepotencijalnih sila

$$\begin{aligned} Q_{x_C} &= F \cos \varphi; \quad Q_{y_C} = F \sin \varphi \\ Q_{\theta} &= M_A; \quad Q_{\varphi} = 0 \end{aligned} \quad (9)$$

Uvrštavanjem jednačina (5), (8) i (9) u (3) dobija se sistem diferencijalnih jednačina kretanja neholonomnog sistema u obliku (10) pri čemu su jednačine neholonomnih veza dobijene korišćenjem izraza (2) i (6) i predstavljene izrazom (11).

Sistem diferencijalnih jednačina (10) i (11) predstavljaju jednačine kretanja automobila. U nastavku sledi rešavanje ovog sistema jednačina i simulacije kretanja automobila numeričkim rešavanjem sistema jednačina (10) i (11) za različite slučajeve početnih uslova i upravljanja.

$$\begin{aligned} M \ddot{x}_C &= F \cos \varphi + \lambda_1 \sin \varphi + \lambda_2 \sin(\varphi + \theta) \\ M \ddot{y}_C &= F \sin \varphi - \lambda_1 \cos \varphi - \lambda_2 \cos(\varphi + \theta) \\ J_C \ddot{\varphi} &= \lambda_1 b - \lambda_2 a \cos \theta \\ J_A \ddot{\theta} &= M_A \end{aligned} \quad (10)$$

$$\begin{aligned} \dot{x}_C \sin \varphi - \dot{y}_C \cos \varphi + b \dot{\varphi} &= 0 \\ \dot{x}_C \sin(\varphi + \theta) - \dot{y}_C \cos(\varphi + \theta) - a \cos \theta \dot{\varphi} &= 0 \end{aligned} \quad (11)$$

3. REŠENJE I REZULTATI SIMULACIJA

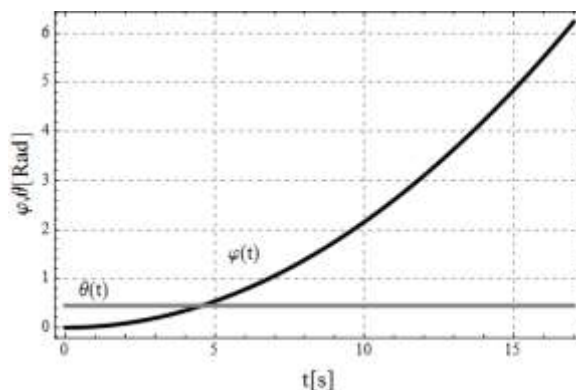
Vozilo za koje se vrši simulacija kretanja ima sledeće parametre.

$$\begin{aligned} M &= 1790 \text{ kg}; \quad l = 2.810 \text{ m}; \quad a = 1.300 \text{ m}; \\ b &= 1.510 \text{ m}; \quad J_C = 2600 \text{ kgm}^2; \quad J_A = 1 \text{ kgm}^2. \end{aligned} \quad (12)$$

Za potrebe analize, proučiće se različiti slučajevi, koji će se razlikovati u načinu upravljanja i početnim uslovima. Sistemom se upravlja pomoću pogonske sile F i obrtnog momenta M_A , koji utiče na ugao zakretanja upravljačkih točkova θ . Sistem diferencijalnih jednačina (10) i (11), uz odgovarajuće početne uslove, parametre sistema i upravljanje pri različitim maneuvrima, rešen je numerički.

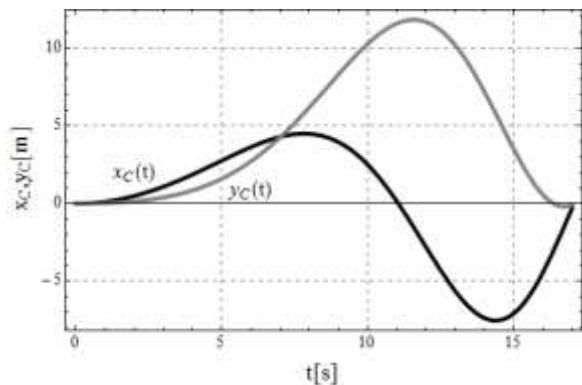
3.1. Prvi manevar, kretanje centra mase po kružnici

U ovom slučaju kretanja, pogonska sila F uzeta je kao konstanta i iznosi $F=500$ N. Kako bi se centar mase C kretao po kružnici, upravljački točkovi moraju biti zakrenuti pod određenim uglom koji se neće menjati tokom vremena, pa sledi da je $q(0)=0.45$ Rad $\approx 26^\circ$. Kako ugao θ treba da ostane konstantan, moment M_A mora biti jednak nuli, što se može videti i iz jednačine (10)₄. U početnom trenutku centar mase C vozila nalazi se u koordinatnom početku referentnog koordinatnog sistema i njegova početna brzina je jednaka nuli. Takođe, prednji deo vozila je usmeren prema pozitivnom delu ose x , ugao i ugaona brzina zakretanja vozila u početnom trenutku su jednaki nuli, kao i ugaona brzina zakretanja točkova. Na slikama 2 i 3 date su vrednosti generalisanih koordinata u funkciji vremena.

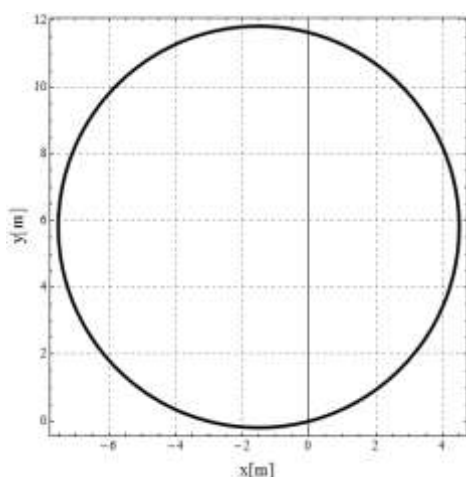


Slika 2. Uglovi φ i θ tokom prvog manevara

Sa slike 2 vidi se da će ugao θ ostati konstantan, dok će ugao φ tokom vremena da raste. Slika 3 prikazuje kretanje centra mase tokom vremena. U trenutku kada se koordinate x_C i y_C nakon početnog trenutka ponovo istovremeno nađu na osi x , centar mase C je obišao ceo krug.

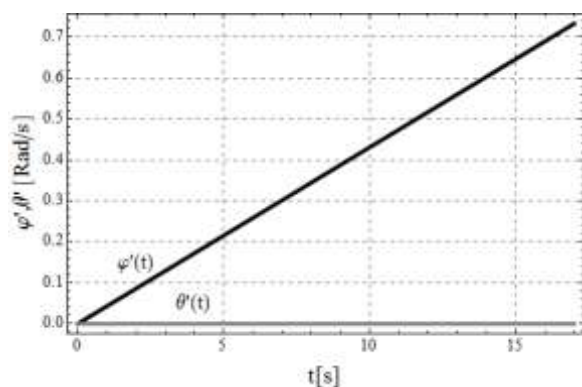


Slika 3. Koordinate centra mase x_C i y_C tokom prvog manevra.

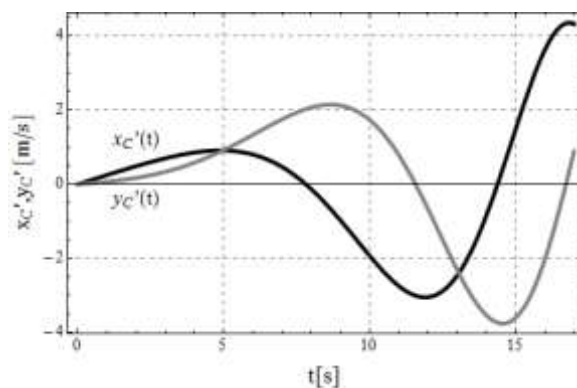


Slika 4. Trajektorija centra mase tokom prvog manevra

Na slici 4 prikazana je trajektorija centra mase C , koja je za ovaj slučaj kružnica. Trenutak kada centar mase obide ceo krug je $t=17,07$ s. Na slikama 5 i 6 prikazane su generalisane brzine u funkciji vremena tokom prvog manevra.



Slika 5. Ugaone brzine tokom prvog manevra



Slika 6. Projekcije brzine centra mase C tokom prvog manevra

Na slikama 2 – 6 dati su rezultati prvog slučaja kretanja na kojima se jasno može videti da će se centar mase vozila zaista kretati po kružnici. Ovakav način kretanja će se održati ukoliko sila trenja na mestu kontakta pneumatika i podloge ne pređe graničnu vrednost sile trenja. U tom slučaju došlo bi do proklizavanja točkova i do odstupanja centra mase sa kružne trajektorije prikazane na slici 4.

3.2. Drugi manevar, sinusna promena ugla zakretanja upravljačkih točkova

Za ovaj manevar pogonska sila će imati istu vrednost kao i u prethodnom slučaju, dok je moment M_A izražen na sledeći način

$$M_A = -Ak^2 \sin(kt) \quad (13)$$

gde je k frekvencija oscilovanja i za potrebe simulacije imaće vrednost 1 s^{-1} ; A je amplituda oscilovanja i njena vrednost je $0,45 \text{ Rad}$. Iz jednačine (10)₄ sledi da je ugao θ opisan funkcijom

$$\theta(t) = A \sin(kt). \quad (14)$$

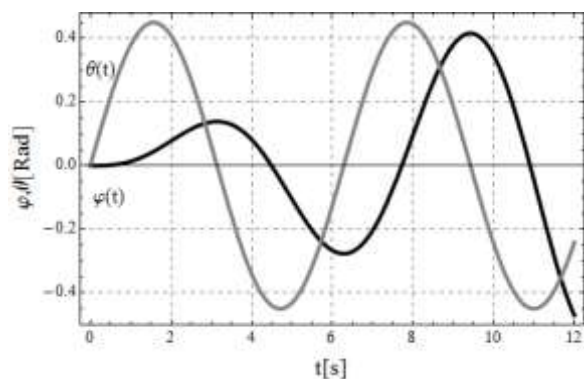
Centar mase, kao i u prethodnom scenariju, u početnom trenutku se nalazi u koordinatnom početku referentnog koordinatnog sistema i njegova početna brzina je jednaka nuli. Takođe, vozilo je usmereno prema pozitivnom delu ose x , ugao zakretanja vozila u početnom trenutku je jednak nuli, kao i ugaona brzina zakretanja. Ugao zakretanja točkova u početnom trenutku je jednak nuli, dok je ugaona brzina zakretanja točkova

$$\omega_\theta(0) = Ak. \quad (15)$$

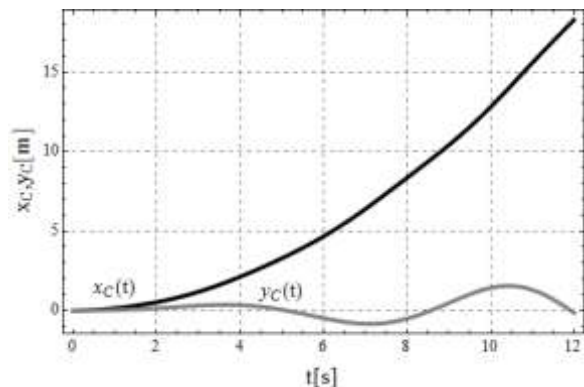
Na slikama 7 i 8 prikazane su generalisane koordinate sistema u funkciji vremena. Sa slike 7 može se videti da se upravljački točkovi zakreću po sinusnom zakonu.

Sa slike 9 očigledno je da će centar mase C vozila oscilovati oko x ose nepokretnog koordinatnog sistema. Ovo potvrđuje i slika 8, sa koje se vidi da generalisana koordinata y_C osciluje oko svog početnog položaja, dok x_C raste tokom vremena. Na slikama 10 i 11 prikazane su generalisane brzine u funkciji vremena tokom drugog manevra.

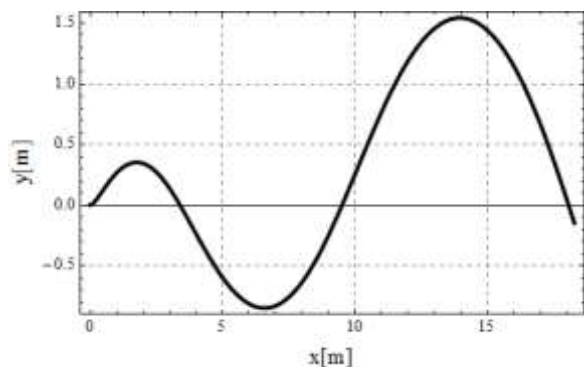
Na slikama 7-11 prikazani su rezultati drugog scenarija kretanja, na kojima se može videti da će se centar mase C vozila kretati po sinusnom zakonu sa rastućom amplitudom.



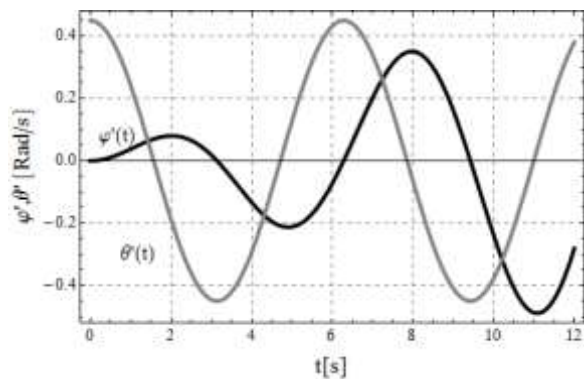
Slika 7. Promena uglova φ , θ tokom drugog manevra



Slika 8. Koordinate centra mase tokom drugog manevra



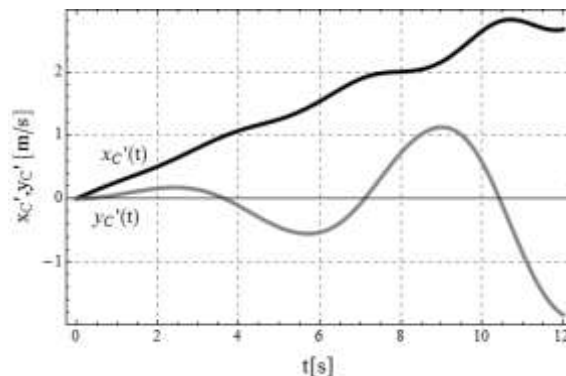
Slika 9. Trajektorija centra mase tokom drugog manevra



Slika 10. Ugaone brzine tokom drugog manevra

Porast amplitude je rezultat dejstva konstantne sile F na vozilo. Ovakav način kretanja, gde se centar mase C kreće sa ubrzanjem usled dejstva sile F , dovodi tokom vremena do sve većih bočnih sila koje dejstvuju na točak na mestu

kontakta sa podlogom. Analizom kontaktnih sila, za poznatu vrednost koeficijenta trenja između pneumatika i podloge, može se analizirati da li i u kom trenutku može da se javi proklizavanje točka.



Slika 11. Projekcije brzine centra mase C tokom drugog manevra

3. ZAKLJUČAK

U ovom radu prikazana su dva slučaja kretanja automobila. Ostali ispitivani slučajevi kao i njihova dublja analiza prikazani su u master radu. Lineikinov model automobila, koji je ovde korišćen za opisivanje kretanja, može da se zameni sa drugim kompleksnijim modelima koji uzimaju u obzir uticaj više faktora, kao što su obrtanje točkova, mehanička svojstva pneumatika itd., da bi se dobio realniji opis kretanja automobila. Neki od složenijih modela navedeni su u pomenutoj literaturi. Iako Lineikinov model ima određene nedostatke može se sa velikom tačnošću primeniti na jednostavnije slučajeve kretanja vozila. Jedna od mogućnosti koja može da se implementira na Lineikinov model jesu disipativne sile kao što su sile otpora pri kretanju vozila kroz materijalnu sredinu i sile trenja u slučaju proklizavanja točkova.

4. LITERATURA

- [1] W. Chen, H. Xiao, "Integrated vehicle dynamics and control", Hefei, Wiley, 2016.
- [2] G. Rill, "Road vehicle dynamics fundamentals and modeling", Boca Raton, CRC Press, 2012.
- [3] H.B. Pacejka, "Tyre and vehicle dynamics", Delft, Elsevier, 2006.
- [4] В.В.Добронравов, "Основы механики негोलомных систем", Издательство высшая школа, Москва, 1970.
- [5] D.T. Spasić, "Mehanika: osnove, opšte, proširenja.", Универзитет у Новом Саду, у припреми.

Kratka biografija:

Nikola Đorđević rođen je u Sremskoj Mitrovici 1991. god. Bečelov rad na Fakultetu tehničkih nauka iz oblasti Mašinstva-Tehnička mehanika i dizajn u tehnici odbranio je 2014. god.

ANALIZA NAPONSKOG STANJA PODZEMNOG CEVOVODA**STRESS ANALYSIS OF UNDERGROUND PIPELINE**Igor Milinski, *Fakultet tehničkih nauka, Novi Sad***Oblast – MAŠINSTVO**

Kratak sadržaj – U radu se iznose rešenja preopterećenja podzemnog naftovoda nominalnog prečnika 2 1/2". Za analizu naponskog stanja se koristi program CAESAR II. Analiza je izvršena u skladu sa standardom ASME B31.4

Ključne reči: Podzemni cevovod, Naftovod, CAESAR II.

Abstract – This paper provides solutions to overstressing of an underground pipeline 2 1/2" in diameter. CAESAR II software is used for stress analysis. The analysis is done in accordance to ASME B31.4 standard.

Keywords: Undergorund pipeline, Oil pipeline, CAESAR II.

1. UVOD

Podzemni cevovodi predstavljaju arterije i vene u sistemu eksploatacije nafte. Transport nafte podzemnim cevovima predstavlja najbezbedniji, najefikasniji i najekonomičniji metod transporta. Na projektovanje podzemnih cevovoda u praksi se obraća manje pažnje u odnosu na nadzemne cevovode. Ovo je pogrešno jer podzemni cevovodi nose sa sobom znatno veće rizike u pogledu bezbednosti postrojenja i zaštite životne sredine. Curenja, iako se mnogo teže otkrivaju, nisu ništa manje opasna nego kod nadzemnih cevovoda. Osim toga, sanacija na podzemnim cevovodima je dug i skup proces.

Do havarija na podzemnim cevovodima može doći i zbog neadekvatnog projektovanja cevovoda usled odsustva analize naponskog stanja. Analiza naponskog stanja (stress analiza) podrazumeva numeričko računanje i analizu sila i napona koji se javljaju na cevovodu usled dejstva težine cevovoda, pritiska fluida, temperature i spoljnih faktora. Analizom naponskog stanja identifikuju se problematične tačke na cevovodu i pruža se uvid u mogućnosti rešavanja istih.

2. ZADATAK RADA

Zadatak diplomskog-master rada, iz kojeg ovaj rad proističe, je analiza naponskog stanja podzemnog naftovoda prečnika 2 1/2" dužine 1218m, u skladu sa standardom ASME B31.4. Ukoliko se ustanovi da naftovod ne zadovoljava zahteve standarda, radom je potrebno predložiti rešenje problema. Naftovod je projektovan u okviru projekta „URP Povezivanje grma bušotina Ve-243. Ve-247“.

NAPOMENA:

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

3. ULAZNI PODACI

Naftovod je dimenzija 2 1/2" sch 40 (Ø73.0 x 5,2 mm). Ukupna dužina podzemnog dela iznosi 1218 m. Promene pravca na naftovodu izvedene su cevnim lukovima radijusa skretanja 320 mm. Materijal izrade naftovoda je ASTM A106 Gr.B. Karakteristike materijala date su u tabeli 1.

Tabela 1. Karakteristike materijala

Zatezna čvrstoća	415 Mpa
Napon tečenja	240 Mpa

Radni pritisak naftovoda iznosi 80 bar, a radna temperatura 40°C. Instalaciona temperatura je data u opsegu 5-21°C. Gustina radnog fluida (nafte) iznosi 900 kg/m³. Karakteristike zemljišta date su u tabeli 2.

Tabela 2. Karakteristike zemljišta

Koeficijent trenja	0,5
Gustina zemljišta	1810 kg/m ³
Prosečna dubina ukopavanja do vrha cevi	1000 mm
Ugao trenja zemljišta	27°

Razmatrani slučajevi opterećenja uzeti su prema preporukama iz standarda ASME B31.4. Usvojeni su sledeći slučajevi opterećenja:

L1 = W+T1+P1 (OPE) – Radno stanje

L2 = W+P1 (SUS) – Noseće stanje

L3 = L1-L2 (EXP) – Ekspanzija

4. REZULTATI PRORAČUNA

Proračun je izvršen za dve ekstremne vrednosti instalacione temperature, 5° C i 21° C. Rezultati proračuna pokazali su da je u oba slučaja naftovod preopterećen. Na pojedinim tačkama naftovoda javljaju se naponi skoro duplo veći od dozvoljenih. Očekivano, pri instalacionoj temperaturi od 21° C naponi su znatno manji nego pri 5° C ali je naftovod i dalje preopterećen. Rezultati proračuna prikazani su u tabelama 3 i 4.

Tabela 3. Rezultati proračuna (5°C)

Slučaj opterećenja	Maksimalni Napon [Mpa]	Dozvoljeni napon [Mpa]	Odnos [%]
L1	353,3	217,2	162,7
L2	260,7	217,2	120,0
L3	319,2	173,7	183,7

Tabela 4. Rezultati proračuna (21°C)

Slučaj opterećenja	Maksimalni Napon [Mpa]	Dozvoljeni napon [Mpa]	Odnos [%]
L1	208,4	217,2	96,0
L2	143,6	217,2	66,1
L3	174,4	173,7	100,4

Naftovod u izvedbi predloženoj projektom ne zadovoljava zahteve standarda ASME B31.4.

5. ANALIZA REZULTATA

Rezultati pokazuju da je cevovod Ve-247 u izvedbi predloženoj projektom izložen prevelikim naponima.

Razlog za ovakve rezultate je pre svega vrlo nepovoljna konfiguracija trase cevovoda. Na relativno maloj dužini cevovod često menja pravac i uslov aktivne dužine nije ispunjen. Posledica je ta da je veliki deo cevovoda u slobodnom stanju i da su prisutna pomeranja i veliki naponi u lukovima.

Drugi razlog koji bi mogli izdvojiti jeste relativno mali radijus skretanja lukova koji su korišteni na cevovodu. Za cevovod ovih dimenzija korišćenje lukova većeg radijusa skretanja ne bi puno povećalo ukupnu investiciju a doprinelo bi povoljnijem naponskom stanju.

Treba naglasiti i da geomehantički podaci o zemljištu nisu dobijeni ispitivanjem, već da su dostavljeni ulazni podaci pretpostavljeni od strane investitora. Ugao trenja zemljišta od 27° je veoma nepovoljan i znatno je uticao na velike napone dobijene proračunom.

6. METODE POBOLJŠANJA NAPONSKOG STANJA

Dugogodišnjom praksom projektovanja podzemnih cevovoda došlo se do nekih zaključaka o ponašanju podzemnih cevovoda. Po pravilu, problem preopterećenja se kod cevovoda koji menjaju pravac skoro uvek javlja na tačkama skretanja cevovoda, sa najvećom koncentracijom na mestima spoja kolena (luka) i prave cevi. Jasno je da pri ekspanziji slobodnog dela cevovoda zemlja vrši kontra opterećenje na cevovod i time izaziva velike napone koji ne bi bili prisutni da cev može slobodno da ekspandira.

Takođe treba naglasiti da će opterećenje bitno zavisiti od razlike instalisane i radne temperature. Tako na primer za cevovod sa razlikom temperature do 15° stepeni opterećenje na kolenima će biti sasvim zadovoljavajuće, dok za razliku od 50° stepeni u kolenima se mogu javiti naponi i nekoliko puta veći od dozvoljenih, za istu konfiguraciju cevovoda.

Imajući sve ovo u vidu razvijeno je više metoda za poboljšanje naponskog stanja cevovoda:

- Smanjivanje razlike temperatura
- Kompenzacioni jastuci
- Postavljanje anker blokova
- Korišćenje lukova većeg radijusa skretanja

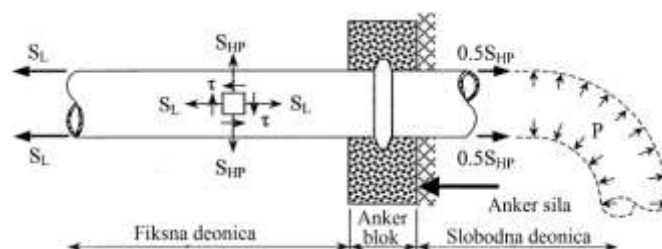
Smanjivanjem razlike radne i instalacione temperature smanjuje se efekat termičke ekspanzije, a time i naponi koji se javljaju u cevovodu. Jasno je da treba težiti što višoj instalacionoj temperaturi, odnosno težiti tome da se cevovod ugrađuje u letnjem periodu.

Kompenzacioni jastuci (slika 1.) su najviše našli primenu kod vrelovoda u grejnoj tehnici, dok se u naftnoj industriji manje koriste. Postavljanjem obloge od mekanog materijala oko cevovoda smanjuje se otpor zemlje ekspanziji cevovoda.



Slika 1. Kompenzacioni jastuci

Anker blokovima (slika 2.) stvaraju se fiksne tačke na cevovodu i smanjuje dužina usled koje dolazi do termičke ekspanzije. Praksa je da se blokovi ugrađuju ispred i iza skretanja na rastojanju od 6-10 prečnika cevovoda.



Povećanjem radijusa skretanja luka, smanjuje se vrednost faktora intenzifikacije napona (SIF) a time i koncentracija napona u tački skretanja. Dovoljnim povećanjem radijusa SIF dostiže vrednost približno $SIF = 1$, tako da u luku vlada napon skoro kao u pravoj cevi.

7. PREDLOŽENA REŠENJA

U cilju rešavanja problema preopterećenja razmatrana su dva rešenja.

Prvo rešenje predviđa postavljanje anker blokova u blizini kolena kako bi se umanjio efekat termičke ekspanzije i snizili naponi u cevovodu. Anker blokovi se postavljaju pre i posle svakog cevnog luka, na udaljenosti od oko 350 mm od luka. Za ovo rešenje naftovod bi zahtevao dodatnih 12 anker blokova. Rezultati proračuna dati su u tabeli 5.

Tabela 5. Rezultati proračuna (Anker blokovi)

Slučaj opterećenja	Maksimalni Napon [Mpa]	Dozvoljeni napon [Mpa]	Odnos [%]
L1	112,3	217,2	51,7
L2	22,2	217,2	17,1
L3	78,2	173,7	45,0

Iz rezultata se zaključuje da rešenje sa anker blokovima zadovoljava sve zahteve standarda ASME B31.4 i da naftovod nije preopterećen. Vrednosti sila koje deluju na anker blokove u ovom slučaju prikazane su u tabeli 6.

Tabela 6. Sile i momenti na anker blokovima

Anker blok	F _{AX} N	F _{LAT} N	F _{VERT} N	M _{B1} Nm	M _{B2} Nm	M _T Nm
1	85600	0	0	0	0	0
2	83200	1900	0	0	560	0
3	83200	1900	0	0	560	0
4	83300	1900	0	0	570	0
5	83300	1900	0	0	560	0
6	83300	1800	0	0	550	0
7	83300	1800	0	0	550	0
8	83200	1900	0	0	560	0
9	83200	1900	0	0	560	0
10	76300	3800	0	0	980	0
11	76300	3800	0	0	980	0
12	76200	3800	0	0	990	0
13	76200	3800	0	0	980	0
14	86600	0	0	0	0	0

Na prikazanu silu potrebno je dodati silu pritiska koja iznosi oko 24700 N.

Drugo rešenje predviđa korišćenje lukova većeg radijusa skretanja sa ciljem da se smanji lokalna koncentracija napona na mestima promene pravca naftovoda. Odabrani su lukovi radijusa skretanja 1500 mm umesto postojećih 320 mm. Rezultati proračuna dati su u tabeli 7.

Tabela 7. Rezultati proračuna (lukovi 1500 mm)

Slučaj opterećenja	Maksimalni Napon [Mpa]	Dozvoljeni napon [Mpa]	Odnos [%]
L1	119,3	181,0	56,7
L2	52,6	173,7	30,3
L3	82,5	217,2	37,9

Rezultati pokazuju da naftovod u ovakvoj izvedbi zadovoljava zahteve standarda ASME B31.4 i nije preopterećen. U ovom slučaju na naftovodu imamo samo dva anker bloka, na početku i kraju linije. Sile koje deluju na anker blokove date su u tabeli 8.

Tabela 8. Sile i momenti na anker blokovima

Anker blok	F _{AX} N	F _{LAT} N	F _{VERT} N	M _{B1} Nm	M _{B2} Nm	M _T Nm
1	85600	0	0	0	0	0
2	86600	0	0	0	0	0

Na prikazanu silu potrebno je dodati silu pritiska koja iznosi oko 24700 N.

8. ZAKLJUČAK

Rezultati dobijeni proračunom pokazali su da cevovod Ve-247 u izvedbi predloženoj projektom nezadovoljava zahteve standarda ASME B31.4 i da je izložen preopterećenju. Preopterećenje se manifestuje velikim koncentracijama napona na mestima promene pravca naftovoda. Predložena su dva moguća rešenja problema preopterećenja naftovoda.

Prvo rešenje podrazumevalo je ugradnju anker blokova na mestima skretanja naftovoda. Ovo rešenje zahtevalo bi ugradnju dodatnih 12 anker blokova, na koje bi delovala sila od približno 100 kN po anker bloku. Ovo predstavlja značajan investicioni trošak i zahteva obimne dodatne građevinske radove. Izvedba sa anker blokovima iako rešava problem preopterećenja nije ekonomski isplativa.

Drugo rešenje podrazumevalo je korišćenje lukova radijusa skretanja 1500 mm. Zahtevi standard ASME B31.4 zadovoljeni su i u ovom slučaju. Sa ekonomskog aspekta izrada i ugradnja navedenih lukova ne predstavlja veliki investicioni trošak.

Imajući sve ovo u vidu jasno je da je izvedba sa lukovima većih radijusa skretanja bolje rešenje sa tehno-ekonomskog aspekta.

9. LITERATURA

- [1] ASME B31.4 – Pipeline transportation systems for liquid hydrocarbons and other liquids, American Society of Mechanical Engineers, 2002
- [2] ASTM A106 – Standard specification for seamless carbon steel pipe for high-temperature service, American Society for Testing and Materials, 2000
- [3] NTC NIS – NAFTAGAS D.O.O : Uprošćeni rudarski projekat povezivanja grma bušotina Ve-243 i Ve-247, Novi Sad, 2016.
- [4] Liang-Chaun Peng: Pipe stress Engineering, ASME PRESS, 2009.
- [5] E.W. McAllister: Pipeline rules of thumb, handbook, Gulf publishing, 2002

Kratka biografija:



Igor Milinski rođen je u Novom Sadu 1989. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mašinstva – Energetika i šrpesna tehnika, odbranio je 2018.god.

UPOREDNA ANALIZA METODA PROGRAMIRANJA NUMA COMPARATIVE ANALYSIS OF CNC PROGRAMMING METHODS

Darko Beretić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MAŠINSTVO

Kratak sadržaj – Programiranje CNC mašina alatki predstavlja jednu od najvažnijih faza pripreme proizvodnje u savremenoj industriji. U radu se porede tri metode programiranja- ProgramGuide, ShopMill i CAM. Analizirani su struktura programa, definisanje zahvata obrade, vreme obrade, broj rečenica kao i vreme programiranja. U cilju dobijanja sveobuhvatnog zaključka gde je najpoželjnije primeniti neku od metoda.

Ključne reči: metode programiranja, ProgramGuide, ShopMill, CAM, poređenje metoda, zahvati obrade.

Abstract – Programming of CNC machine tools is one of the most important stages of production preparation in the modern industry. The paper compares three methods of programming - ProgramGuide, ShopMill, and CAM. The structure of the program, the definition of processing operations, the processing time, the number of sentences and the programming time are analyzed. In order to obtain a comprehensive conclusion where it is most appropriate to apply one of the methods it is most appropriate to apply one of the methods.

Keywords: Programming Methods, ProgramGuide, ShopMill, CAM, Comparison of Methods, Processing Procedures.

1. UVOD

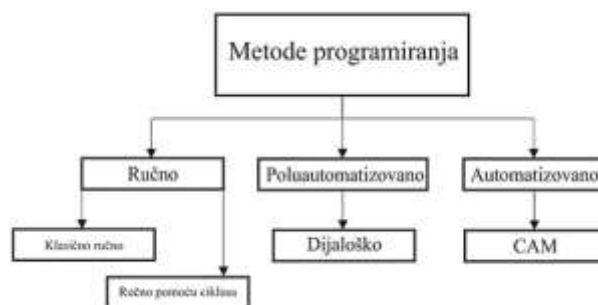
Svetsko tržište mašina alatki za obradu rezanjem danas se suočava s neprestanim promenama zbog mnogih faktora, koje se ogledaju u dinamici velike ponude novih i raznovrsnih proizvoda, kratkim vekom eksploatacije, visokim zahtevima kupaca za kvalitetom i pristupačnim cenama. U takvim uslovima globalne ekonomije koju karakteriše značajna tržišna konkurencija imperativ je brzo, jeftino i kvalitetno izraditi proizvod. Prema tome, sa aspekta ekonomičnosti postupaka, glavni ciljevi svih obrada su povećanje dimenzionalne preciznosti i kvaliteta proizvoda uz istovremeno povećanje brzine uklanjanja materijala (strugotine), kao i smanjenje vremena obrade po jedinici proizvoda, smanjenje brzine habanja alata i utroška električne energije. Da bi se zadovoljili zahtevi tržišta i kupaca neprestano se razvijaju nove metode programiranja numerički upravljanih mašina alatki. U ovom radu izvršena je analiza tri metode programiranja (ProgramGuide, ShopMill, CAM) sa stanovišta strukture programa, vremena obrade, definisanje zahvata obrade, broja rečenica i vremena programiranja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Slobodan Tabaković, red. prof.

2. METODE PROGRAMIRANJA NUMERIČKI UPRAVLJANIH MAŠINA ALATKI

Prema stepenu automatizacije metode programiranja numerički upravljanih mašina alatki se dele na (Slika 2.1): ručno, poluautomatizovano i automatizovano.



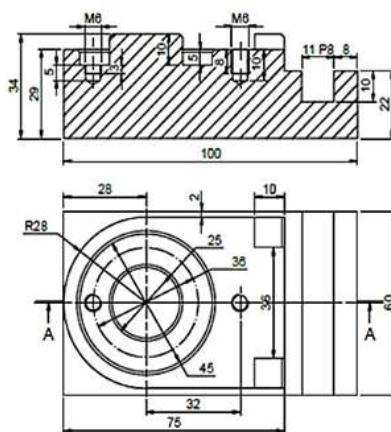
Slika 2.1 Metode programiranja NUMA

Bez obzira o kojoj se metodi programiranja radi, numerički upravljana mašina alatka dobija, kodirane, sve potrebne informacije za automatsku obradu materijala priprema [2].

ProgramGuide predstavlja metodu ručnog programiranja primenom ciklusa. Programski ciklusi predstavljaju tipizirane podprograme koji definišu geometrijske parametre pojedinih zahvata obrade dok se tehnološki kao što su brzina obrtanja glavnog vretena, vrsta alata, uključivanje/isključivanje podmazivanja i hlađenja i sl. definiše van ciklusa. Ciklusi obrade imaju za cilj eliminisanje ponavljanja naredbi pri ručnom programiranju i omogućavanje jednostavnih izmena upravljačkog programa. Neki od ciklusa pri obradi glodanjem su: Ciklus konturne obrade (eng. *Profile Machining Cycle*), Ciklus glodanja (eng. *Milling Cycle*), Ciklus bušenja (eng. *Drilling*) [4].

ShopMill je softver za poluautomatizovano programiranje numerički upravljanih mašina alatki namenjen upravljačkim sistemima Siemens Sinumerik. Kod ovog načina programiranja upravljačka jedinica ima integrisane dodatne funkcije pomoću kojih se izbegava klasično programiranje korišćenjem G-koda. Umesto toga, programiranje se izvodi kroz interaktivni dijalog između operatera i upravljačke jedinice, preko meke tastature i podprograma sa snimljenim ciklusima, uz snažnu grafičku podršku [1].

CAM (eng. *Computer Aided Manufacturing*) je automatizovana metoda programiranja koja se definiše kao računom podržano upravljanje procesom proizvodnje, odnosno računom podržana razrada tehnologije i automatizovano definisanje upravljačkog programa u skladu sa ISO specifikacijom. Da bi se pristupilo programiranju u CAM-u potrebno je definisati model obrade u nekom od CAD programskih sistema [3]. Tok programiranja primenom CAM-a je predstavljen na slici 2.2.



Slika 3.2 Crteži tela bežičnog punjača telefona i pokretne platforme

Slika 2.2 Tok aktivnosti u tehnologiji CAM programiranja

Poređenje metoda je realizovano za svaki zahvat pojedinačno, prilikom čega se posmatra način definisanja zahvata, struktura programa, vreme obrade i kao krajnje informacije broj rečenica u programu i potrebno vreme izrade upravljačkog programa.

Režimi obrade i vrsta alata u zahvatima je ista za sve metode programiranja s tim da su režimi obrade iskustvene vrednosti. Poređenje će biti realizovano sledećim redosledom programGuide → ShopMill → CAM (Slika 3.1).



Slika 3.1 Metode programiranja

U nastavku su predstavljeni crteži tela bežičnog punjača i pokretne platforme (Slika 3.2).

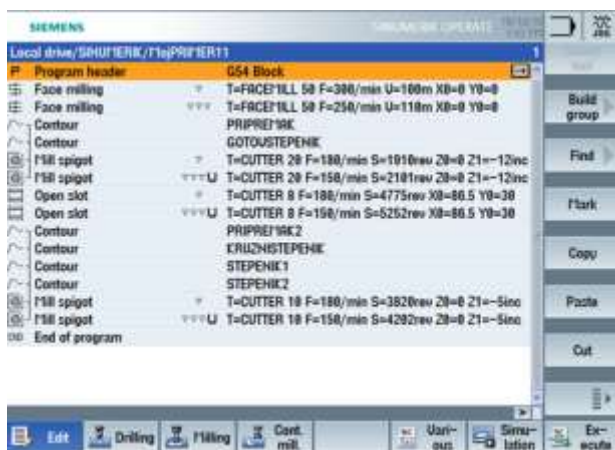
Telo bežičnog punjača je stepenasti prizmatični radni predmet, dok je pokretna platforma sačinjena od kontura i složenih oblika.

U tabeli 1. su predstavljeni režimi obrade po zahvatima za telo bežičnog punjača.

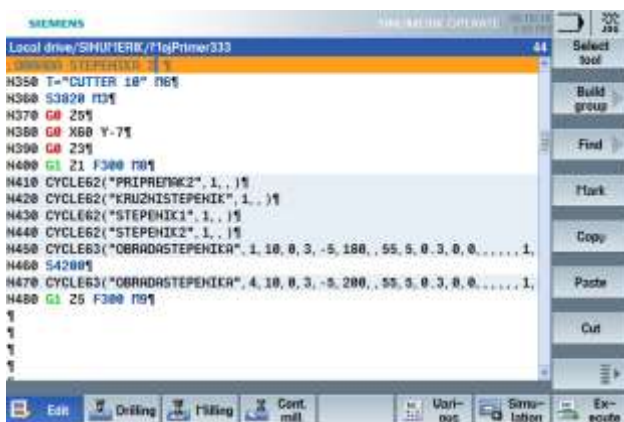
	Vrsta obrade	Alat	Dubina obrade [mm]	Pomak [mm/min]	Brzina rezanja [m/min]	Broj obrtaja [o/min]
Čeona obrada ravne površine	Gruba	Čeono glodalo Ø50	0,7	300	100	640
	Završna	Čeono glodalo Ø50	0,3	250	110	700
Obrada stepenika 1	Gruba	Vretenasto glodalo Ø20	4	180	120	1910
	Završna	Vretenasto glodalo Ø20	6	150	132	2100
Obrada žljeba	Gruba	Vretenasto glodalo Ø8	5	180	120	4775
	Završna	Vretenasto glodalo Ø8	5	150	132	5250
Obrada stepenika 2	Gruba	Vretenasto glodalo Ø10	5	180	120	3820
	Završna	Vretenasto glodalo Ø10	5	150	132	4200
Obrada kružnog džepa	Gruba	Vretenasto glodalo Ø6	5	180	120	6366
	Završna	Vretenasto glodalo Ø6	5	150	132	7000
Zabušivanje rupa	/	Zabušivač Ø5	1	150	/	800
Bušenje rupa	/	Burgija Ø5	5/10	150	/	2000
Urezivanje navoja M6 u rupe	/	Ureznik M6	3/8	50	/	50
Obaranje ivica	/	Alat za obaranje sa uglom 45°	1	150	/	1000

Kod metode CAM programiranja model obrade se definiše van upravljačkog programa, te se naknadno na njemu vrši definisanje zahvata. Međutim, kod programGuide-a i ShopMill-a se ne definiše model obrade, već pripremak u samom upravljačkom programu.

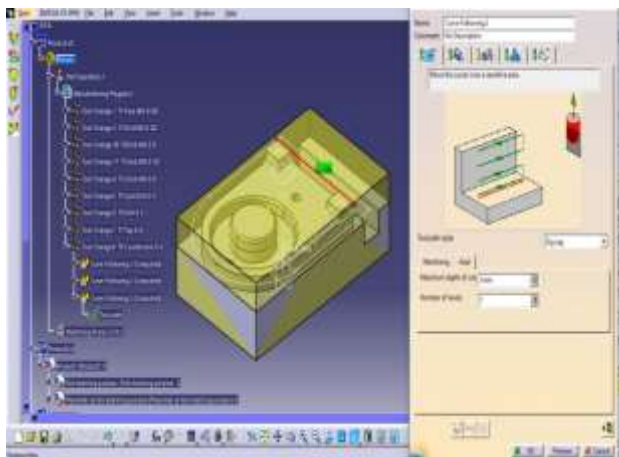
Struktura programa pojedinih zahvata za svaku metodu programiranja tela bežičnog punjača je predstavljena na slikama 3.3, 3.4 i 3.5.



Slika 3.3 Struktura programa u ShopMill-u



Slika 3.4 Struktura programa za zahvat obrade stepenika 2 u programGuide-u



Slika 3.5 Struktura programa u CAM-u

Nakon definisanja svih zahvata obrade pomoću ugrađenih mehanizama u samim programskim sistemima dobija se ukupno vreme obrade i broj rečenica u upravljačkom programu (tabele 2. i 3.). Treba napomenuti, u CAM metodi da bi se dobio upravljački program potrebno je izvršiti postprocesiranje. Pored toga, CATIA V5 poseduje mogućnost generisanja izveštaja o procesu obrade (slika 3.8).

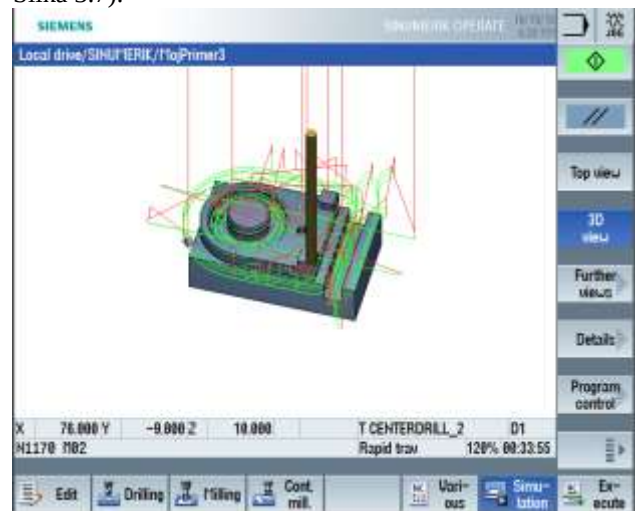
Tabela 2. Ukupno vreme obrade, broj rečenica, vreme programiranja

	ProgramGuide	ShopMill	CAM
Vreme obrade	33' 55''	37' 50''	30' 12''
Broj rečenica	262	37	485
Vreme programiranja	2 h i 05'	1 h i 15'	4 h i 35'

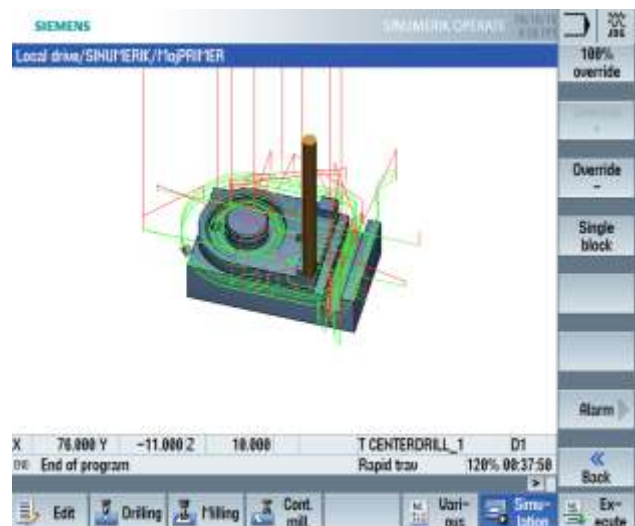
Tabela 3. Ukupno vreme obrade, broj rečenica, vreme programiranja za pokretnu platformu

	ProgramGuide	ShopMill	CAM
Vreme obrade	1 h 17' 29''	1h 21' 44''	1h 5' 49''
Broj rečenica	249	31	1514
Vreme programiranja	2 h i 25'	1 h i 55'	3 h i 10'

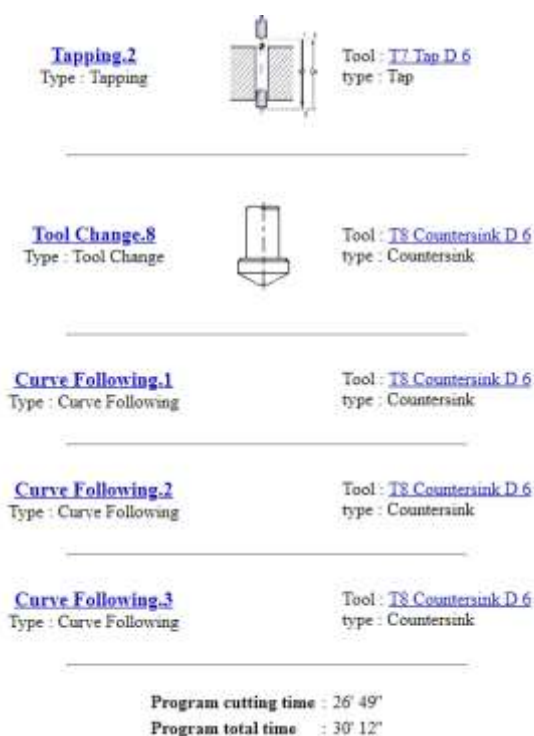
Kod programGuide-a i ShopMill-a u simulaciji je prikazano ukupno vreme obrade i putanja alata (Slika 3.6 i Slika 3.7).



Slika 3.6 Simulacija obrade programGuide



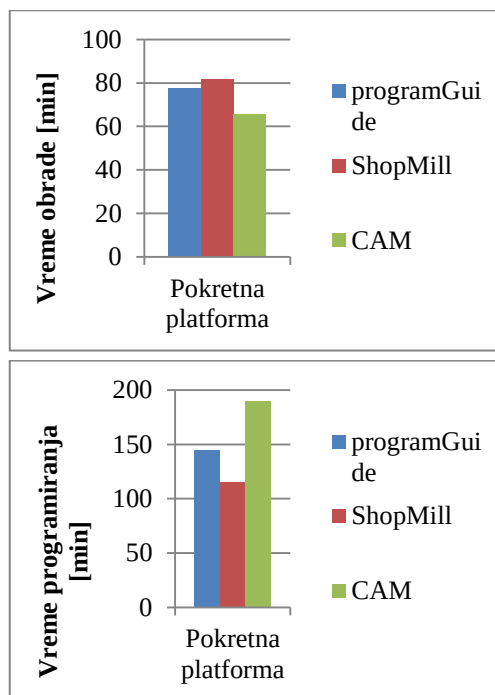
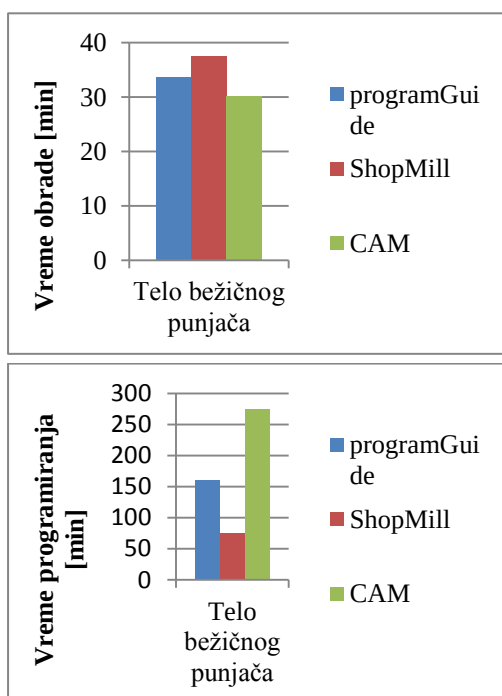
Slika 3.7 Simulacija obrade ShopMill



Slika 3.8 Izveštaj za CAM metodu programiranja sa ukupnim vremenom obrade (HTML)

4. ZAKLJUČAK

Za isti radni predmet primenom tri različite metode dobija se različito vreme obrade. Na primeru izrade upravljačkog programa za obradu tela bežičnog punjača može se uočiti da se automatizovana metoda pokazala najbolje sa stanovišta vremena obrade, ali je potrebno uzeti u obzir i druge parametre. Na primer da se ovaj radni predmet izrađuje u 10 primeraka, za vreme koje se utroši na programiranje u CAM-u primenom ShopMill-a bi imali gotove izradke. Takođe, da se radi i o većim serijama utroškom nekoliko desetina minuta više, postiže se ušteda u vremenu obrade od 6% primenom programGuide-a.



Prema tome, na osnovu rezultata dobijenih poređenjem ove tri metode može se zaključiti da je ShopMill pogodan za pojedinačnu i maloserijsku proizvodnju. Pored toga je jednostavan za upotrebu što olakšava obučavanje rukovoca ovog programskog sistema. Jednostavnim optimizovanjem putanje alata ručnim unosom pojedinih parametara u programGuide metodi postiže se smanjenje vremena obrade. Na taj način može se svrstati u metode pogodne za male i srednje serije. CAM je najkompleksnija metoda, zahvati obrade se definišu do detalja što umnogome povećava vreme programiranja, ali i smanjuje vreme obrade. Ova metoda je predviđena za srednje i velike serije, kada je potrebno postići maksimalna ušteda u vremenu obrade kao i definisati proces obrade na najvišem nivou.

5. LITERATURA

- [1] Čiča, Đ., Jokanović S.: Programiranje numerički upravljanih mašina alatki, Univerzitet Banjoj Luci, Mašinski fakultet, Banja Luka, 2014
- [2] Mećanin, V.: Programiranje obradnih process na CNC mašinama, Univerzitet u Kragujevcu, Mašinski fakultet Kraljevo, Kraljevo, 1997
- [3] Slavoković, R., Dučić, N.: Primena CAD/CAM tehnologija u proizvodnim procesima, Tehnički fakultet Čačak, Čačak, 2011.
- [4] Suh, S., Kang, S., Chung, D., Stroud, I.: Theory and Design of CNC Systems, Republic of Korea, 2008.

Kratka biografija:



Darko Beretić rođen je u Subotici 1995. god. Diplomirao na Fakultetu tehničkih nauka smer Proizvodno mašinstvo 2017. god. Master studije upisao iste godine na usmerenju CIM

DALJINSKI NADZOR I INTEGRACIJA OPREME U SCADA SISTEM**REMOTE MONITORING (TELEMETRY) AND EQUIPMENT INTEGRATION IN THE SCADA SYSTEM**

Nikola Ajduković, Željko Trpovski, Dejan Nemec, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Ovim radom opisana je realizacija jedne od faza konkretnog projekta daljinskog nadzora i telemetrijskih metoda kako bi se stvorili uslovi za integraciju kompleksnih sistema u jedan jedinstven geografski distribuiran sistem uz integraciju u aplikaciju za sistem za merenje, praćenje i kontrolu industrijskih procesa (SCADA – Supervisory Control And Data Acquisition).

Ključne reči: telemetrija, SCADA, daljinski nadzor, integracija

Abstract – One phase of the development of geographically distributed SCADA system, data acquisition over different communications media, different types of communication with review of latest telemetry technologies, data aggregation and sending over OPC server, visualisation and historization are described in this paper.

Keywords: Telemetry, SCADA, Geographically distributed system, integration.

1. UVOD

Konkretni projekat opisan radom realizovan je u cilju integracije različite opreme sa pogona Reg. 1, Reg. 2, Reg. 3 i Reg. 4 u jedinstveni SCADA (Supervisory Control And Data Acquisition) sistem.

Pod integracijom se podrazumeva povezivanje opreme locirane u različitim proizvodnim objektima u jedinstvenu komunikacionu infrastrukturu, kao i kreiranje odgovarajućih softverskih obrazaca (*template*) i instanci kojima su u SCADA aplikaciji prezentovani statusi pomenute opreme.

Opisani su tipični scenariji sa kojima se susretalo prilikom realizacije. Zatečena oprema bila je raznovrsna i u nekim slučajevima nije bilo moguće izvršiti čitanje zbog nestandardnih protokola za komunikaciju. U tim slučajevima pribegavalo svođenju nestandardnih signala na standardne korišćenjem koncentratora podataka, a u nekim slučajevima i zamenom zastarele opreme (redi slučaj).

Takođe zbog značajnih ušteda ređe se pribegavalo korišćenju danas rasprostranjenih telemetrijskih metoda prenosa podataka posredstvom GPRS-a (*General Packet Radio Service*) (ili 4G mreže) i sličnih, jer je cena

prenosa podataka posredstvom navedenog načina komunikacije u momentu realizacije bila nekoliko puta veća od današnjih. Danas proizvođači ove opreme imaju razrađene mehanizme za slanje podataka po događaju što u većini slučajeva omogućava višestruke uštede prilikom realizacije ovakvih projekata.

Razlog svih navedenih radova jeste dovođenje podataka do dispečerskih centara i do lica koja su podatke smeštala u bazu podataka. Ti podaci su morali biti stalno dostupni radi (u početku ručne) izrade izveštaja za nadređene, i plana proizvodnje. Naredne faze rezultirale su integracijom sa drugim sistemima kojima su podaci iz baze bili značajni za analizu/optimizaciju proizvodnje. Izveštavanje je u potpunosti automatizovano do nivoa slanja izveštaja na sve relevantne adrese: dvočasovnih, poludnevni, dnevnih, nedeljnih, mesečnih itd.. Realizacija jednog ovakvog projekta rezultirala je višestrukim uštedama u poslovanju.

Jedna od značajnih novih tehnologija za telemetrijske sisteme jeste korišćenje baterijski napajanih telemetrijskih uređaja kojima je vek trajanja baterije 5-7 godina čiji su dometi do 1.600 m uz korišćenje eksterne antene. Ovaj novi metod telemetrije pravio je uštede i na jediničnim lokacijama koje su udaljene samo nekoliko stotina metara vazdušnom linijom.

2. INTEGRACIJA OPREME

U ovom poglavlju ukratko će biti opisane telemetrijske metode koje su korištene prilikom integracije određene opreme i sistema u jedinstveni sistem za nadzor i upravljanje.

Sistemi koji su integrisani obuhvataju:

- 1) sisteme za merenje protoka gasa uz osvrt na uređaje korišćene za ovu namenu čiji izbor, odnosno scenario integracije je vršen u zavisnosti od zatečenog mernog uređaja, postojeće komunikacione infrastrukture i ostalih faktora koji su mogli učestvovati u izboru metode telemetrije (npr. da li lokacija poseduje napajanje, da li ima optičku vidljivost ka centru ka kom gravitira itd.),
- 2) Integracija ESP pumpi, gotovo u svim slučajevima vršena je posredstvom radio modema koji su u serijskoj komunikaciji i koji su u suštini stvarali tunel ka centru ka kom gravitiraju,
- 3) Integracija kompleksnih mernih sistema koja je uvek vršena posredstvom gejtveja (konvertera sa

NAPOMENA:

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

RS 485 na Ethernet odnosno MODBUS RTU na MODBUS TCP),

- 4) Integracija kompresora vršena je identično kao pod tačkom 3.

2.1. Telemetrijske metode

Na osnovu zahteva koje je postavio krajnji korisnik svojim tehničkim zadacima, vršen je u zavisnosti od različitih faktora, odabir metoda integracije određenog sistema. Svaka od izabranih metoda odabrana je u zavisnosti od sledećih faktora:

- 1) vrste uređaja koji treba integrisati,
- 2) zatečenih dostupnih komunikacionih portova na uređaju koji treba integrisati,
- 3) postojeće infrastrukture tj. da li postoji Wi-Fi, *serial wireless data network*, serijska komunikacija na kablu RS 232/422/485, GPRS, *Ethernet*, satelitska ili *fieldbus* (HART),
- 4) da li ka telemetrisanoj lokaciji postoji optička vidljivost,
- 5) postoji li napajanje na telemetrisanim lokacijama ili je potrebno obezbediti alternativni izvor napajanja za svu telemetrisanu opremu i sam sistem,
- 6) da li je na lokaciji ka kojoj gravitira telemetrisana oprema potrebno prikazati telemetrisani podatak ili je njegovo upisivanje relevantno samo za centre u kojima se vrši analiza.

Kako bi se izvršili odabiri odgovarajućih metoda telemetrije u nekim slučajevima se vršila analiza terena, pre svega kod sistema koji zahtevaju optičku vidljivost, kao što su različiti bežični sistemi u opsegu 2,4 GHz. Grafički prikaz koji je rezultat analize topologije terena koji se oslanja na precizne satelitske snimke uz unošenje parametara visine stuba na jednoj ili obe strane slabljenja na kablovima i konektorima i dobitak na anteni pomagao nam je uz podatke o rastinju i optičkoj vidljivost da precizno procenimo da li će se sa najpovoljnijim metodama telemetrije ostvariti željeni rezultati.

Kod sistema kod kojih nije postojalo napajanje telemetrisane lokacije vršila se analiza potrošnje celog sistema i dimenzionisanje u našem slučaju sistema solarnog napajanja za najgore uslove rada a prema tehničkom zadatku koji je propisao investitor koji su ujedno opisali i klimatske uslove koje sistem mora da zadovolji (zaključno sa sastavom tla za temelj i brzinom vetra i veličinom grada koje sistem mora da izdrži), a koji su bili daleko predimenzionisani u odnosu na najgore klimatske uslove koji su zabeleženi na toj teritoriji.

Takođe na izbor telemetrijske metode učestvovali su i faktori cene metode, gde se uvek insistiralo na smanjenju troškova tokom eksploatacije, tako da je investitor uvek davao prednost bežičnim metodama komunikacije u odnosu na ostale za rastojanja veća od 300 m.

Praksa je potvrdila da su metode eksploatacije koje su odabrane zaista donele ogromne uštede tokom eksploatacije (nije bilo značajnih havarijskih zastoja, osim kvarova nastalih recimo ljudskim nemarom ili od direktnog udara groma).

Što se tiče kasnije integracije u SCADA sistem i upisa podataka u bazu, korišćene su otvorene platforme za komunikaciju sa uređajima različitih proizvođača tzv.

OPC serveri (*Open Platform Communications*). Na ovaj standardni način i korišćenjem softverskog alata za razvoj SCADA aplikacije baziranog na sistemu proizvođača Wonderware (konkretno proizvod je Wonderware Sistem Platform) omogućen je razvoj sistema baziranog na razvijenim templejtima iz kojih su prilikom integracije svakog uređaja i omogućenog čitanje zahtevanih parametara posredstvom OPC servera „izvlačene“ instance.

Wonderware je u vreme realizacije ovog projekta bio jedini softver koji je omogućio ovakav način rada i koji je bio otvoren za sve komunikacione platforme što je rezultiralo integracijom sistema koji se sastoji od preko 60.000 tagova (promenljivih koje se čitaju sa različitih uređaja, lokacija i iz različitih geografskih regiona uz generisanje dodatnih internih promenljivih kojima su definisane npr. granice alarma). Deo podataka je automatski beležen u bazu podataka (*real time* bazu) baziranu na SQL-u iz koje su se radi kasnije analize povlačili podaci uz generisanje tabela, trendova (grafika različitih zavisnosti), radi rekonstrukcije određenih incidentnih situacija, izveštaja kao i radi predaje drugim sistemima koji su imali potrebe za određenim podacima.

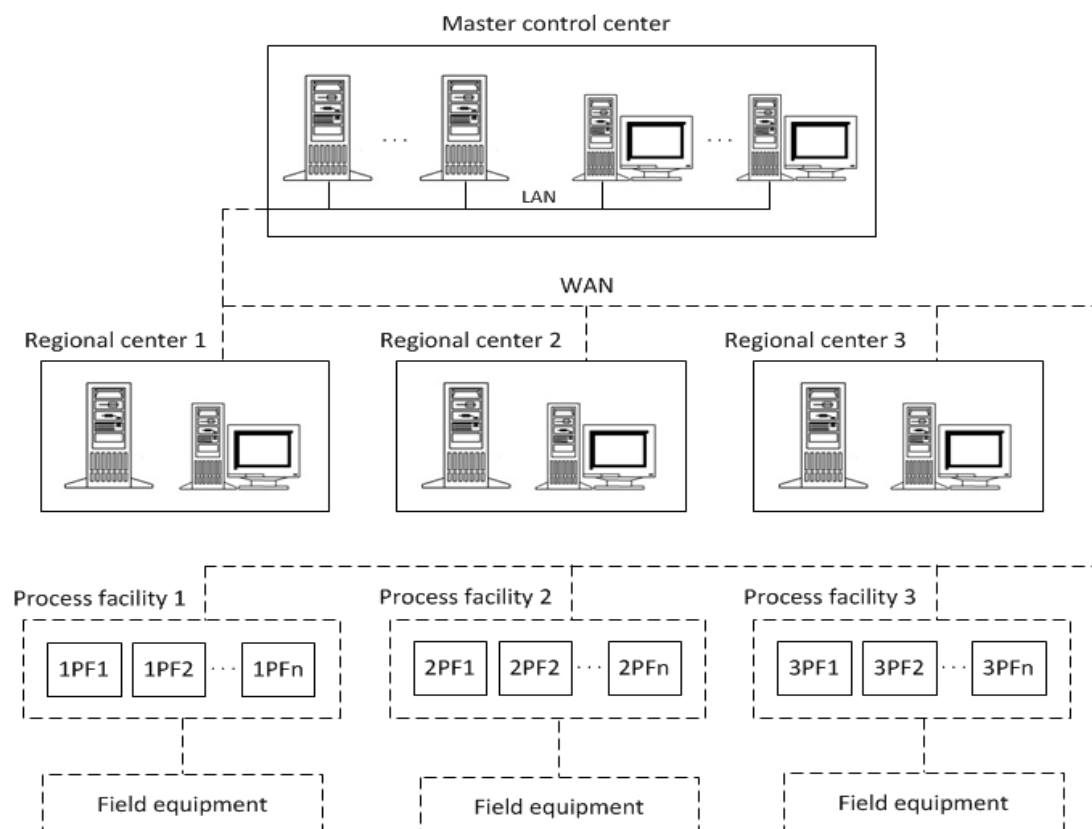
Struktura nadzorno upravljačkog sistema bazirana je na klijent server arhitekturi prikazanoj na slici 1. uz poštovanje zahteva krajnjeg korisnika da svi podaci budu vidljivi samo u glavnom dispečerskom centru, a da ostali imaju mogućnost pregleda samo dela sistema za koji su nadležni prema korisničkim nalogima. Svi podaci definisani kao alarmi i događaji se takođe beleže u bazu podataka kao i aktivnosti operatera u smislu vremena kada je određen alarm detektovan i kada je potvrđen. Kroz sistem se vrši monitoring nad komunikacionom infrastrukturom.

Otvorenost sistema rezultirala je njegovim korišćenjem ne samo kao sistem za nadzor i upravljanje nego i kao sistem posredstvom kog će drugi sistemi koji nemaju direktnu mogućnost za komunikaciju vršiti razmenu određenih setova podataka što je rezultiralo boljim performansama i ostalih sistema koji su ranije podatke primali npr. ručnim unosom.

U delu vezanom za konfigurisanje sistema za prenos podataka posredstvom različitih komunikacionih medija bilo žično ili bežično korišćeni su različiti softverski alati proizvođača opreme koja je odabrana za određeni metod telemetrije, a nekada i više proizvođača čija se oprema morala naći u nizu lanca za telemetrisanje određenog uređaja i rad sa nekim od softvera korišćenih za ovu namenu.

Svaki prenos podataka zahtevao je i modifikacije aplikacija na prijemnim stranama pre svega sa strane PLC-ova (programabilnih logičkih kontrolera) i tzv. mapiranje podataka u njihove registre radi daljeg čitanja. Najčešće korišćeni protokoli za razmenu podataka između PLC-a i SCADA-e su u ovom slučaju bili:

- 1) MODBUS TCP/IP,
- 2) ENRON MODBUS,
- 3) DNP 3,
- 4) Komunikacioni OPC protokoli za čitanje određene vrste PLC/RTU uređaja.



Slika 1. Arhitektura sistema

3. TEHNIČKI ZADATAK

Tehničkim zadatkom opisan je zadatak projekta uopšteno uz osnovne smernice za realizaciju u pogledu zahtevanih karakteristika opreme, podataka koje treba prikupiti i poželjnog načina za prikupljanje, gde ih vizualizovati. Opisan je celokupan obim radova i lokacije na kojima je radove potrebno izvršiti u skladu sa svim zakonima kojima jedan ovakav projekat podleže.

Dat je opis zatečenog stanja sistema i željenog sistema nakon završetka projekta. Definisani su nivo proširenja sistema sa aspekta licenciranog softvera izrade novih aplikacija modifikacija i unifikacija postojećih aplikacija kao i ostali radovi koje je potrebno uraditi da bi cilj radova bio ispunjen.

3.1. Razrada tehničkog rešenja

Pored opisanih metoda telemetrije u uvodnom delu rada u ovom delu izvršen je konkretan opis telemetrijskih paketa za konkretne slučajeve sistema za daljinski nadzor proizvodnih postrojenja. Navedeni su detaljni opisi centralnih i udaljenih telemetrijskih paketa koji su sačinjeni i konfigurisani potpuno u saglasnosti sa zahtevima navedenim u tehničkom zadatku.

Svi komunikacioni paketi sa aspekta napajanja poseduju UPS jedinicu na 24 VDC radi obezbeđenja kontinualnog napajanja na lokaciji i u slučaju nestanka napajanja električnom energijom.

Ovakav zahtev proističe iz potrebe da se signal o prestanku napajanja električnom energijom prenese do centra i nakon nestanka električne energije (tj. da se obezbedi neprekidno napajanje komunikacione infrastrukture kako bi se ovakvi podaci preneli, a koji su od suštinske važnos-

ti za sistem održavanja, sistem alarmiranja, analizu stanja sistema, ulaze u sistem za izveštavanje.

Da se nije ispunio ovakav zahtev došlo bi do bespotrebnih troškova izlaza kvalifikovanih zaposlenih na lokacije. Sistem obezbeđuje pristup kvalifikovanim stručnjacima sa udaljenih lokacija gotovo do poslednjeg elementa u sistemu za dijagnostiku. Softverski alati za konfigurisanje radio veza su poslednje tehnologije sa ugrađenom enkripcijom i frekvencijskim skakanjem, uređaji nisu vidljivi bez posedovanja ID-a, i na taj način je izbegnuta mogućnost ometanja.

Prilikom montaže antena, prenaponskih zaštita i kablova vršen je proračun slabljenja na svakom konektoru, na svakom metru kabla i kao i proračun dobitka na anteni kako bi se ostalo u zakonskim limitima. Posebno se vodilo računa o kvalitetu konekcija u RF delu jer je praksa pokazala da se najčešći problemi javljaju zbog korišćenja nekvalitetnih konektora, kablova i nepravilne instalacije.

Svaki spoj je vulkaniziran kako bi se sprečio prodor vlage u kabel i konektor, izrađena su mesta na prevojima kabla kako bi se ostvarilo otkidanje kapi sa kabla sa kog se sliva voda pre njegovog spoja sa konektorom (kritične su pozicije uvođenja RF kabla u razvodne ormare).

Kod svih metoda telemetrije gde se oprema nalazila u eksplozivnoj sredini (Ex-zona) moralo se posebno voditi računa da se oprema čita posredstvom uređaja koji imaju funkciju izolatora (izolacione barijere). Izolacione barijere imaju funkciju ograničavanja nivoa napona i struje koje je moguće emitovati u Ex zonu, na taj način se izbegava mogućnost pojave varnice i havarije u zoni opasnosti.

Kako bi se osvrnuli na trend razvoja opisan je i metod telemetrije korišćenjem baterijski napajanih uređaja čiji je projektovani vek trajanja baterije 5-7 godina.

Za ovu namenu korišćeni su uređaji proizvođača Schneider Electric pod nazivom Accutech. Suštinski ova vrsta opreme sa svojim baznim stanicama ne komunicira po standardnim protokolima ali se to od njih i ne očekuje jer standardni protokoli nisu optimizovani za prenos podataka u režimu smanjene potrošnje energije dok je kod ovih uređaja to suština.

Njima je moguće ostvariti komunikacije na rastojanjima do 1.600 m uz obavezno ispunjen uslov o optičkoj vidljivosti i ovo će uz razne varijacije na temu kod različitih proizvođača postati trend budućeg razvoja.

4. ZAKLJUČAK

Pored zaključka koji su navedeni u osnovnom radu a koji su definisani tehničkim zadatkom faze III, realizovana je dogradnja jedinstvenog sistema dispečerskih centara.

Ovo obuhvata sistem monitoringa na svim mestima i upravljanje na mestima na kojima to tehnologija proizvodnog procesa dozvoljava, zatim vizualizacije, arhiviranja i izveštavanja.

Ovo je realizovano u okviru projekta izvedenog u WonderWare ArchestrA IDE 3.1 okruženju, a sve u cilju realizacije jedinstvenog sistema nadzora nad objektima za proizvodnju nafte i gasa.

Takođe, ostvarene su komunikacione veze ARM računara, sa dispečerskim centrima mrežnom vezom TELEKOM (MPLS L3VPN).

Izrađene su potrebne softverske aplikacije i pušteno je u rad 9 novih radnih stanica sa novim licencama.

Nove aplikacije u ArchestrA okruženju izrađene su na svim zahtevanim objektima.

Pored novih aplikacija, izvršena je modifikacija i unifikacija postojećih aplikacija na još 13 objekata. Poboľjšani su Wi-Fi kanali veza ka ARM računarima u Reg. 3 kao i na 10 lokacija za bušotine na postojećim Wi-Fi sistemima.

Tehničkim zadatkom faza III bilo je potrebno projektovati, izvršiti nabavku, montažu i pustiti u rad šest novih podsistema i proširiti sedam postojećih sistema (od ukupno 15 postojećih) za daljinski nadzor ukupno još 26 bušotine u "ESP" metodi eksploatacije.

Svi zahtevani zadaci su uspešno izvršeni zajedno sa posebnim tehničkim zahtevima koji su zahtevali implementaciju Wi-Fi linkova uz poštovanje određenih pravila (razmena podataka Wi-Fi linkova mora biti kriptovana, unutar RS232/485 mreže obezbediti mogućnost ograničenja učesnika u mreži, autentifikacije

korisnika prilikom pristupa komunikacionim portovima kontrolera na glavnom (centralnom) objektu).

Sve telemetrijske metode našle su svoju primenu nekada u kombinaciji jedna sa drugom i došlo se do konačnog zaključka da je prilikom realizacije ovakvih projekata najbolje koristiti opremu renomiranih proizvođača uz praćenje standardnih baznih tehnologija kao i novih trendova razvoja telemetrijskih sistema.

5. LITERATURA

- [1] http://plcsystems.ru/catalog/asc/doc/RealFLO_spec_eng.pdf, Decembar 2018
- [2] https://www.moxa.com/product/MGate_MB3170_3270.htm, Mart 2019.
- [3] https://www.schneider-electric.com/en/product-range/61419-trio-licensed-radios?subNodeId=827471733en_WW, Mart 2019.
- [4] Manual, SCADAPack 32 Controller, Instalation, Operation and Maintenance, Schneider Electric, 2011 <https://www.plcsystems.ru/catalog/SCADAPack/doc/5232.pdf>, Mart 2019.

Kratka biografija:



Nikola Ajduković rođen je u Zrenjaninu 1980. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacije odbranio je 2018. god. kontakt: nikola023@gmail.com



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



Dejan Nemec rođen je 1972. god. Diplomirao, specijalizirao i magistrirao je na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva. Oblast interesovanja su telekomunikacije i obrada signala.

EFEKTI POBOLJŠANJA PROGNOZE IZLAZNE SNAGE VETROELEKTRANA
EFFECTS OF WIND FARM POWER FORECAST IMPROVEMENT

Aleksandar Atanasovski, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad*
Oblast - ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je analizirana prognoza izlazne energije vetroelektrana. Nakon opšteg opisa karakteristika energije vetra i njene konverzije u električnu, dat je pregled dostupnih metoda prognoze kao i njihova uloga u procesu upravljanja elektroenergetskim sistemom. Procenjeni su i pozitivni tehno-ekonomski efekti poboljšanja prognoze na rad sistema i vrednost energije vetra na tržištu.

Ključne reči: Vetroelektrane, prognoza izlazne snage, metode prognoze

Abstract – This paper deals with wind farm power forecasting. After general description of wind power characteristics and conversion into electricity, attention was focused on various power forecasting methods and models. Their importance and role in power system's operation and planning is explained. Benefits of forecast improvements are assessed in respect to reducing the integration problems, improving the power system and increasing the wind power value on the market.

Keywords: Wind farm, power forecasting, forecasting methods

1. UVOD

Globalna želja i potreba čovečanstva za korišćenjem čistih i obnovljivih izvora energije (OIE) uz postepenu smenu elektrana zasnovanih na fosilnim gorivima, dovodi do velikih promena u načinu funkcionisanja elektroenergetskih sistema. Planiranje i upravljanje sistemom, kao i mehanizmi trgovine električnom energijom se prilagođavaju potrebama OIE, koji zbog svoje intermitentne prirode predstavljaju veliki izazov.

Danas je od svih vrsta OIE najviše kapaciteta instalisano za korišćenje energije vetra [1]. Pored velikih benefita, zbog promenljive prirode vetra, odnosno zbog promenljive izlazne snage vetroelektrana, planiranje i upravljanje elektro-energetskim sistemom (EES) znatno je složenije. Da bi se to poboljšalo, razvijaju se različite metode predviđanja ili prognoze energije vetra, koje se više ili manje slažu sa stvarnim parametrima vetra.

Iako je i teoretski nemoguće postići 100% tačnu prognozu, svako njeno poboljšanje donosi niz prednosti. Ovo je već uočeno primenom najjednostavnije metode, kada je, u odnosu na period bez prognoziranja prosečna srednja kvadratna greška prognoze izlazne snage po satima za dan smanjena sa 30-35% na oko 15-20% [2,3].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Katić, red. prof.

To je omogućilo primenu vetroelektrana kao (donekle) predvidivog izvora, kojeg je moguće koordinirati sa termo, nuklearnim ili hidro-agregatima na način da se očuva stabilan i pouzdan rad EES-a.

Cilj ovog rada je da razmotri postojeće metode prognoze i uporedi efekte njihove primene na elektro-energetske i ekonomske parametre vetroelektrana.

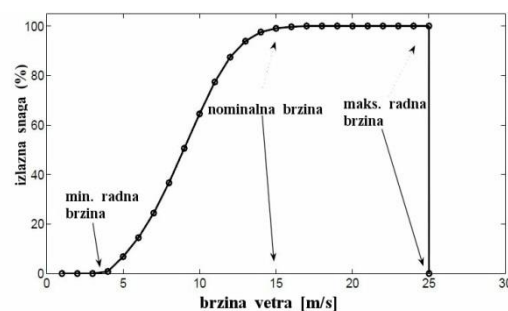
2. PROMENLJIVOST SNAGE VETRA

Mehanička snaga koju vetroturbina razvija na svom vratilu se definiše kao:

$$P_{meh} = C_p P_{vetra} = C_p \frac{1}{2} \rho A v^3 \quad (1)$$

gde je ρ gustina vazduha (standardna gustina pri temperaturi 15°C iznosi $\rho = 1,225 \text{ kg/m}^3$), C_p – koeficijent iskorišćenja snage turbine (obično 0,4 – 0,45; max 0,59 Betzova konstanta), A površina kruga koji pri obrtanju prebrišu lopatice, v brzina vetra u m/s.

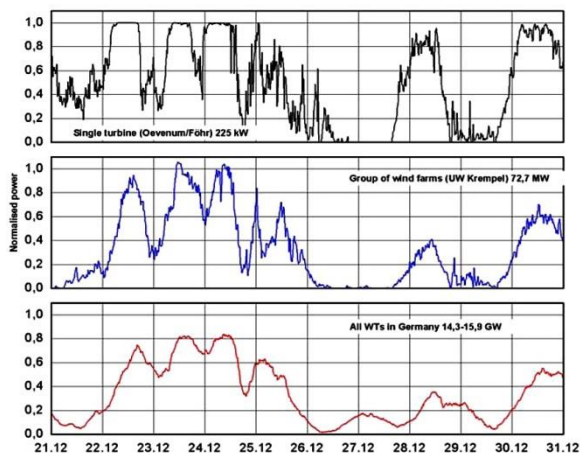
Koliku aktivnu snagu vetrogenerator može razviti pri različitim brzinama vetra proizvođači opisuju karakterističnom „S“ krivom snage (wind turbine power curve). Tipična kriva snage prikazana je na slici 1.



Slika 1 - Karakteristika snage vetrogeneratora [4]

Iz izraza (1) se uočava da je brzina vetra dominantna promenljiva koja utiče na izlaznu snagu vetroturbine. Na krivoj snage vetroturbine ta se kubna zavisnost odražava u delu između minimalne i nominalne radne brzine, gde mala promena brzine vetra izaziva velike promene izlazne snage. Samim tim i greška u prognozi brzine vetra (u ovom opsegu) dovodi do još veće greške prognoze izlazne snage vetroturbine, odnosno vetrogeneratora. Kako bi se umanjila ovakva nepredvidivost i prognoziranje olakšalo, proizvodnja električne energije se ne analizira na nivou jedne vetroturbine (VT), nego cele vetroelektrane (VE), a za potrebe sistema se često zajedno posmatraju i grupe vetroelektrana u jednom regionu ili čak celom EES. Time se ukupne varijacije izlazne snage smanjuju (Slika 2), jer se pojedinačne varijacije brzine vetra sa više VT u okviru VE međusobno potiru. Kod objedinjavanja više VE

imamo i pozitivan efekat geografske agregacije pri kojoj je način promene brzine vetra u nekom vremenskom intervalu različit na međusobno udaljenim lokacijama viševetroelektrana.



Slika 2 - Poređenje varijacija izlazne snage pojedinačne vetroturbine, jedne (ili grupe) vetroelektrana i svih vetroelektrana u jednom EES [2]

Problemi vezani sa integracijom proizvodnje električne energije iz VE uneki EES zavise od toga sa kojim procentom ona učestvuje u ukupnoj proizvedenoj energiji iz svih postojećih izvora u EES-u. Taj udeo se naziva nivo prodora energije vetra (u literaturi se kao sinonimi koriste termini „učesće“, „penetracija“ i sl.). Obično se smatra da su varijacije proizvodnje energije iz VE praktično zanemarljive prema uobičajenim varijacijama potrošnje u sistemu i da ne zahtevaju specijalne mere, dokle god proizvodnja svih VE u jednom sistemu ne prelazi 10% ukupne instalisane snage svih izvora tj. pri niskom nivou prodora snage vetra [2]. Ta granica (do koje se ne očekuje veliki uticaj i problem prilikom integracije energije vetra u sistemu) zavisi pre svega od veličine i fleksibilnosti sistema, tj. postojanja elektrana koje mogu brzo i u velikom rasponu da menjaju izlaznu snagu.

Za upravljanje sistemom su najvažnije tzv. kratkoročne varijacije izlazne snage VE, koje utiču na balans aktivnih snaga u sistemu. U njih spadaju sve one u rasponu od nekoliko sekundi do 24 ili 48 sati, jer se u tom rasponu vrše primarna, sekundarna i tercijarna regulacija aktivnih snaga u sistemu i određuje hidro-termo koordinacija agregata. Takođe, to je vremenski okvir za trgovinu električnom energijom na brzom (spot) tržištu, gde moraju ispoštovati ugovorene količine i plan isporuke energija iz VE, te se njihovoj prognozi posvećuje najviše pažnje.

3. METODE PROGNOZE SNAGE VE

Glavni problem je uspešno prognozirati buduću proizvodnju vetroelektrane na osnovu informacionih skupova sadašnjih i istorijskih poznatih promenljivih.

Neka se ima skup poznatih vrednosti izlazne snage vetroelektrane $\{P_t\}_{t \in T}$, gde je T skup vremenskih indeksa, a P_t snaga VE u trenutku t . Želi se da se predvidi buduća vrednost P u trenutku $t+k$ (k je horizont prognoze), na bazi poznavanja ponašanja procesa $\{P_t\}_{t \in T}$ u ranijem periodu (do trenutka t), tj. implicitno pretpostavljajući da će on nastaviti da se isto ponaša. Matematički to može da se napiše u sledećoj formi [4]:

(2)

$$P_{t+k|t} = g(P_t, \dots, P_{t-l_P}, \xi_t, \dots, \xi_{t-l_\xi}, \hat{\chi}_{t+1|\tau}, \dots, \hat{\chi}_{t+l_\chi|\tau}, \hat{q}_{t+k|t}) + \epsilon_t \\ = g(\Phi'_t, \hat{q}_{t+k|t}) + \epsilon_t \quad (2)$$

gde je ϵ_t beli šum, Φ'_t - informacioni skup promenljivih, a q_t broj dostupnih vetroturbina. U informacionom skupu su sadržane sve ukupne izlazne snage VE u prošlom periodu (P_t) kao i prošle (ξ) i prognozirane vrednosti eksternih promenljivih ($\hat{\chi}$). Cilj je naći najbolju aproksimaciju funkcije g koja objedinjujući te parametre iz informacionog skupa i prognozu broja dostupnih VT daje izlaznu snagu VE u budućem trenutku $t+k$.

Postoje dva osnovna problema u prognozi izlazne snage dobijene iz vetra. Prvi je transformacija postojećih numeričkih podataka o vremenskim uslovima u izlaznu snagu elektrane, a drugi je pažljiva selekcija faktora iz istorijskih podataka, koji će uticati na proizvodnu snagu u budućnosti. Vremenom su se izdvojila 2 glavna tipa metoda prognoze za rešavanje ovih problema: fizički i statistički [5]. Pored ovih postoje i drugi poput: „naive reference“ - referentni metod, ANN/AI, hibridni i na kraju probabilistički [4,5].

Najjednostavniji metod se naziva „persistence“ prognoza i u suštini je zasnovan na pretpostavci da se vetar u budućem trenutku $t+k$ ponaša tačno onako kako se ponašao do trenutka t kada su poslednji put zabeležene njegove karakteristike, odnosno da će buduća izlazna snaga vetroelektrane biti identična onoj poslednji put izmerenoj [4,5]. Ponekad se koristi i uopštavanje da će buduća vrednost biti jednaka prosečnoj vrednosti prethodnih n merenja. Performanse ovakvog jednostavnog pristupa, gde se uopšte ni ne uzimaju u obzir eksterni parametri terena i atmosfere, drastično opadaju što je taj budući trenutak dalji. Međutim, pokazalo se da u kratkim vremenskim intervalima k (otprilike 4-6 sati) tj. veoma kratkom horizontu prognoze, ovaj metod daje prilično tačne rezultate te se zbog toga on uzima kao referenca za poređenje svih drugih, složenijih i naprednijih metoda [4,5].

Fizičke metode se zasnivaju na obradi meteoroloških podataka pomoću takozvane „parametrizacije“, kojom se stvara statistički model zasnovan na empirijskim podacima o parametrima atmosfere i reljefa (lokalni termalni efekti, turbulencije, aerodinamička dužina hrapavosti okolnog terena itd.) merenim ili procenjenim na tačnoj lokaciji vetroelektrane [5]. Dalje postoje dva pristupa. Prema jednom se koristi logaritamski zakon i procena hrapavosti terena i stepena viskoznog trenja vazduha za proračun brzine vetra na tačnoj visini rotora. Drugi pristup je stvaranje vrlo detaljnog modela terena oko vetroelektrane i kretanja fluida oko svake turbine, za šta se koriste *Computational Fluid Dynamics* (CFD) tehnike modelovanja.

Statističke metode prognoze su zasnovane na jednom ili više modela koji uspostavljaju vezu između istorijskih vrednosti izlazne snage VE, istorijskih i prognoziranih vrednosti meteoroloških parametara i izmerene snage vetra [5]. Ovde se uopšte ne ulazi u objašnjavanje i analiziranje samih fizičkih pojava niti između posmatranih promenljivih mora da postoji neka fizička korelacija. Parametri modela se procenjuju isključivo na osnovu skupa dostupnih istorijskih podataka (bila to brzina vetra ili izlazna snaga iz VE), i redovno tokom

rada u realnom vremenu podešavaju na osnovu trenutnih merenja. Tako dobijen model onda predviđa koje će se vrednosti dobiti u budućnosti.

4. EKONOMSKI EFEKTI POVEĆANJA TAČNOSTI PROGNOZE PROIZVODNJE VE

Realna tržišna vrednost proizvedene energije i njene prognoze se izračunava prema sledećem izrazu [3]:

$$Vrednost = \frac{P_{prog} * C_{ug} + G_{poz} * N_{bpoz} - G_{neg} * N_{bneg}}{P_{up}} \quad (3)$$

gde je: *Vrednost* – fiksna cena koju treba platiti prilikom trgovanja direktnom OTC pogodbom (*Over-the-counter market*), tako da proizvođač električne energije iz VE ima istu zaradu kao da je trgovao energijom na berzi; P_{prog} – prognozirana količina energije vetroelektrane, ugovorena dan pre isporuke; C_{ug} – tržišna cena za tu energiju, ugovorena dan pre isporuke; G_{poz} – greška prognoze kada je realna proizvodnja VE veća od prognozirane (pozitivno odstupanje); N_{bpoz} – naknada za višak energije usled pozitivnog odstupanja u odnosu na prognozu. Ovaj višak energije je moguće prodati na tržištu, ali samo po nižoj ceni, u rasponu 0 - 92% od tržišne cene, tako da je i u ovom slučaju vlasnik VE ima umanjeње prihoda; G_{neg} – greška prognoze kada je realna proizvodnja VE manja od prognozirane (negativno odstupanje); N_{bneg} – naknada za angažovanu balansnu energiju usled negativnog odstupanja u odnosu na prognozu (Ovde TSO po višoj ceni, do 108% od tržišne, naplaćuje vlasniku energiju koja je morala biti „dokupljena“ da bi se ispunio ugovoreni plan); P_{up} – ukupna proizvedena energija u posmatranom periodu.

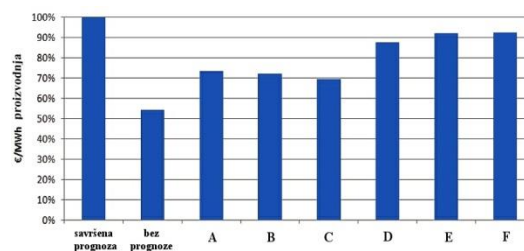
Naknade zbog disbalansa izazvanih greškama u prognozi, koje proizvođač plaća, dovode do gubitka prihoda i smanjenja tržišne vrednosti kompanije. Relativni gubici prihoda (naknade plaćene zbog disbalansa u odnosu na tržišnu cenu) se nazivaju „gubici disbalansa“. U radu su korišćeni podaci dostupni iz dve studije, belgijske [3] i američke [6].

Iz prve studije je na primeru jedne belgijske vetroelektrane snage 8MW, a na osnovu podataka iz 2009. god. analiziran uticaj primene naprednih metoda prognoze. Rezultati su prikazani na slici 3.

Prvo je kao osnova za poređenje izračunata vrednost proizvedene električne energije, koju je moguće postići na tržištu direktnim ugovaranjem u uslovima idealne, 100% tačne prognoze. Ovo je nazvano „savršena prognoza“. Procena tog OTC (*Over-the-Counter*) trgovanja ekvivalenta je 78,3 €/MWh.

Suprotno od toga je teoretska situacija gde se prognoza namerno uopšte ne uvažava (bez prognoze) ili jako retki slučajevi kada se desi da ugovorne strane predlože neku razmenu, ali se taj ugovor ne potvrdi od strane proizvođača/prodavca energije pre nego što se tržište za „dan unapred“ zatvori. U tom, najlošijem slučaju, procenjena OTC vrednost energije je 42,5 €/MWh, srednja kvadratna greška prognoze je 30,9%, i rezultujući gubici disbalansa su čak 46%.

Ostali rezultati prikazani na slici 3, dobijeni kada se primenjuju realne prognoze su sledeći:



Slika 3 - Uticaj poboljšanja prognoze na tržišnu vrednost energije dobijene iz vetra [3]

A. Prognoza je proporcionalna prosečnoj godišnjoj proizvedenoj enegiji na nivou jednog dana – procenjena OTC vrednost energije je 57,5 €/MWh, srednja kvadratna greška prognoze 24%, i rezultujući gubici disbalansa su oko 26,5%.

B. Prognozaprma „*persistence*“ metodi, zasnovanoj na prosečnoj proizvodnji tokom prethodnog dana - procenjena OTC vrednost je 56,5 €/MWh, srednja kvadratna greška prognoze je 26,9%, i rezultujući gubici disbalansa su oko 28%.

C. Prognoza prema „*persistence*“ metodi, zasnovanoj na proizvodnji tokom 2 sata neposredno pre zaključivanja ponude - procenjena OTC vrednost je 54,5 €/MWh, srednja kvadratna greška prognoze je 31,1%, a gubici disbalansa su oko 28%.

D. Prognoza prema savremenim (*state-of-the-art day-ahead*) metodama - procenjena OTC vrednost je 68,6 €/MWh, srednja kvadratna greška prognoze pada na 15,5% i rezultujući gubici disbalansa na 14%.

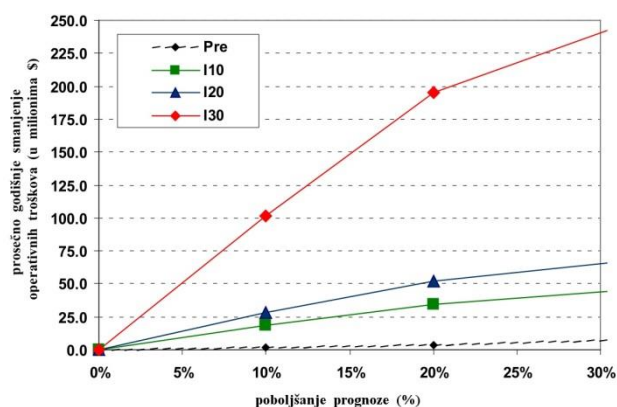
E. Efekti koji se dobijaju savremenim metodama prognoze pri unutar-dnevnom trgovanju, gde se ponude podnose na svaka 4 sata (*4-hour rolling intra-day market*). Procenjena OTC vrednost je oko 72,1 €/MWh, srednja kvadratna greška prognoze pada na 9,2% i rezultujući gubici disbalansa na 8,6%.

F. Efekti koji se dobijaju savremenim metodama prognoze pri unutar-dnevnom trgovanju, gde se ponude se podnose na svaka 2 sata (*2-hour rolling intra-day market*). Procenjena OTC vrednost je oko 72,4 €/MWh, srednja kvadratna greška prognoze pada na 8,5% i rezultujući gubici disbalansa na 7,3%.

Vidi se da je metoda F najbliža idealnoj, savršenoj prognozi i da se njom postižu najmanji gubici disbalansa.

Druga studija [6] je rađena 2011. godine za potrebe američke regulatorne organizacije WECC (*Western Electricity Coordinating Council*). Cilj je bio da se pokaže da li i kolike uštede pri upravljanju sistemom donosi poboljšanje tačnosti postojećih metoda prognoze (onih koje su u vreme analiziranih podataka studije smatrane za „*state of the art*“ - SOA) za 10, 20 i 30%. Pri tome su na osnovu planova izgradnje novih vetrelektrana pretpostavljena 3 moguća scenarija što se tiče budućih nivoa prodora snage vetra u odnosu na postojeće stanje: 10%, 14% i 24%. Efekti poboljšanja tačnosti prognoze su sagledani kroz analizu eksploatacionih troškova, troškova nedostatka operativne rezerve, količine potisnute i neiskorišćene energije.

Na slici 4 prikazano je prosečno godišnje smanjenje eksploatacionih troškova sistema sa poboljšanjem prognoze. Pod eksploatacionim troškovima sistema se podrazumevaju godišnji promenljivi troškovi, pre svega pogonski troškovi stavljanja agregata u pogon i njegovog zaustavljanja, troškovi goriva i promenljivog rada agregata, kao i troškovi održavanja. Tu ne spadaju fiksni troškovi poput investicionih troškova ili vraćanja dugova i sl.



Slika 4 - Prosečno godišnje smanjenje eksploatacionih troškova sistema sa poboljšanjem prognoze [6]

Za scenario I20 (14% učešća energije vetra u WECC) rezultati pokazuju da poboljšanje prognoze od 10% donosi prosečnu uštedu godišnjih eksploatacionih troškova od 28 miliona US dolara (\$, USD), a poboljšanje prognoze za 20% štodi čak 52 miliona \$. Za scenario I30 (24% učešća energije vetra u WECC) rezultati pokazuju da poboljšanje prognoze od 10% donosi prosečnu uštedu godišnjih eksploatacionih troškova od 100 miliona \$, a poboljšanje prognoze za 20% štodi 198 miliona \$.

Nedostatak operativne rezerve je situacija kada u sistemu nema dovoljno dostupnih proizvodnih kapaciteta da se ispune svi zahtevi nametnuti potrebama operativne rezerve. Pri ovoj situaciji proizvodnja energije u sistemu je i dalje dovoljna da podmiri zahteve potrošača, ali je nema dovoljno da popuni propisani nivo npr. rotirajuće regulacione ili havarijske rezerve. Godišnji nedostatak operativne rezerve je ukupna suma svih količina nedostajuće energije tokom radnih sati jedne kalendarske godine. U EES koji i po „klasičnim“ pravilima sigurnosti i pouzdanosti imaju pravilno određenu sigurnosnu marginu kapaciteta operativne rezerve, ovakve situacije se jako retko dešavaju, čak i kada se uvodi energija dobijena iz VE. Stoga su rezultati pokazali zanemarljivo poboljšanje u scenarijima I10 i I20 (prodor snage vetra u sistemu od 10% i 14%) i tek pri scenariju I30 i učešću energije vetra od 24% i više imamo primetne uštede sa poboljšanjem prognoze proizvodnje VE. Tada prosečni godišnji nedostatak operativne rezerve opada sa 43 GWh na 24 GWh sa poboljšanjem prognoze od 10%, i na 15 GWh ako se prognoza poboljša 20%.

Slična je situacija i što se tiče potisnute i neiskorišćene energije, kada se višak energije iz VE odbacuje i ne može da bude prihvaćen u mreži. Rezultati studije pokazuju da poboljšanje „dan-unapred“ prognoze proizvodnje VE nema značajnih efekata za scenarije I10 i I20. Za scenario I30 se dobija daprosečna godišnja potisnuta energija opada sa 655 GWh na 628 GWh (ušteda 4%) sa

poboljšanjem prognoze od 10%, i na 613 GWh (ušteda 6,3%) ako se prognoza poboljša 20%.

5. ZAKLJUČAK

Poslednjih godina načinjeni su veliki pomaci u prognozi izlazne snage VE, usavršavanjem metoda prognoze i primenom savremenih softverskih alata. Iako su fizičke i statističke metode, koje su danas u standardnoj upotrebi širom sveta, već ostvarile značajan napredak u odnosu na najranije i najjednostavnije *persistence* metode, još postoji prostora za poboljšanje tačnosti primenom „stateofheart“ probabilističkih, hibridnih metoda, kao i kombinovanjem postojećih.

Srednju kvadratnu grešku prognoze moguće je danas, uz određene uslove, spustiti i ispod 10%. Analizom podataka i primera iz literature pokazano je da i relativno malo smanjenje greške prognoze donosi velike uštede, kako u novcu, tako i u energiji i da je isplativo i vlasnicima/operatorima VE i operatorima sistema i tržišta.

6. LITERATURA

- [1] ***, „WIND IN POWER 2017 - Annual combined onshore and offshore wind energy statistics“, Wind Europe, Brisel, 2018.
- [2] F. Van Hulle, P. Gardner, „Wind Energy - The Facts: Part II – Grid Integration“, EWEA, 2009.
- [3] R. Belmans, W. D’haeseleer, G. Palmers, J.C. Maun, „Balancing wind energy in the grid: an overall, techno-economic and coordinated approach SD/EN/02A Final Report“, Brussels, 2011.
- [4] L. Järvenpää, „Production Planning of a Wind Farm Based on Wind Speed Forecasting“, M.Sc. Thesis, Aalto University, 2010. god.
- [5] P. Pinson, „Estimation of the uncertainty in wind power forecasting“, Ph.D. Thesis, MINES ParisTech, Paris, 2006.
- [6] R. Piwko, G. Jordan, „Impacts of Improved Day-Ahead Wind Forecasts on Power Grid Operations“, GE Energy Consulting, New York, 2011.

Kratka biografija:



Aleksandar Atanasovski rođen je 1986. godine u Smederevu. Fakultet tehničkih nauka, studijski program Energetika, elektronika i telekomunikacije upisao je školske 2004/2005. Na studijama se opredelio za modul Elektroenergetski sistemi i diplomski-master rad odbranio 2018. god.



Vladimir A. Katić, red.prof. rođen je 1954. godine u Novom Sadu. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu 1978. god., a magistrirao i doktorirao na Univerzitetu u Beogradu 1981. i 1991. godine, respektivno. Oblasti interesovanja su energetska elektronika, obnovljivi izvori električne energije, kvalitet električne energije i električna vozila.

RAZVOJ WEB APLIKACIJA SA SPRING DATA REST NA PRIMERU APLIKACIJE ZA PRAĆENJE ZADATAKA**DEVELOPMENT OF WEB APPLICATIONS WITH SPRING DATA REST WITH AN EXAMPLE OF ISSUE TRACKER APPLICATION***Igor Gere, Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu prikazan je razvoj demonstrativne web aplikacije korišćenjem Spring Data REST modula popularnog framework-a Spring. Ukazane su neke prednosti i mane korišćenja modula koji je u fokusu ovog rada.

Ključne reči: Java, Spring, JPA, REST

Abstract – This paper presents development of web application using Spring Data REST module of the popular framework Spring. Some advantages and drawbacks of module which is in the focus of this paper are shown.

Keywords: Java, Spring, JPA, REST

1. UVOD

Tema ovog rada jeste modul Spring Data REST, jedan od modula Spring framework-a (radni okvir). Pomenuti modul će biti iskorišćen prilikom implementacije demonstrativnog projekta. Iskustva stečena prilikom implementacije će biti navedena u okviru ovog rada.

Funkcionalnosti koje su implementirane u demonstrativnoj aplikaciji korišćenjem Spring framework-a su mogle biti implementirane i na druge načine bez korišćenja Spring Data REST modula, ali bi to zahtevalo daleko složeniji kod. Ovaj rad služi i kao primer kako posmatrani modul ubrzava razvoj aplikacija.

Pored Spring Data REST koji predstavlja glavnu temu rada korišćen je i modul Spring Data JPA. Cilj je da se istraži na koji način se koristi Spring Data REST kroz implementaciju aplikacije koja predstavlja issue tracker (sistem za evidenciju zadataka).

Svrha primer aplikacije jeste implementacija funkcionalnosti korišćenjem Spring Data REST modula radi njegove analize i demonstracije, dok sam Spring framework kao takav nije u fokusu.

1.1. REST

REST (Representational State Transfer) predstavlja arhitektonski stil mrežno baziranih aplikacija, definisao ga je Roy Thomas Fielding u svojoj doktorskoj disertaciji [1]. Predstavlja set pravila koja se primenjuju za izgradnju web baziranih aplikacija. Nastao je na osnovu izbora pogodnih arhitektonskih stilova uzimajući u obzir njihove

pojedinačne prednosti i mane, a sa ciljem kreiranja stila koji je optimalan u opštem slučaju.

Od postavljenih pravila REST arhitekturom naglasio bih da je to klijent-server arhitektura kod koje je postavljen zahtev za unificiranim interfejsom. Unificirani interfejs olakšava izgradnju složenog sistema olakšavajući interakcije izađu njegovih komponenti. Kao negativnu osobinu unificirani interfejs smanjuje efikasnost sistema jer podaci nisu prilagođeni konkretnom slučaju korišćenja.

Svaku informaciju REST smatra resursom. Resursi ne moraju direktno predstavljati entitete u sistemu već njihovo značenje. Moguće je da isti entitet bude predstavljen kao dva različita resursa ili da više entiteta bude spregnuto u jedan resurs. Interakcija između komponenti sistema odvija se razmenom resursa koji u toj interakciji mogu biti predstavljeni svojim identifikatorom. U ovom slučaju identifikatorom se smatra URL (uniform resource locator), a ne primarni ključ entiteta koji je predstavljen resursom.

REST je stateless što znači da svaki zahtev ka REST sistemu treba da sadrži sve potrebne informacije za njegovu obradu i da se ne može računati na neko sačuvano stanje na serveru tzv. sesiju.

HATEOAS (Hypermedia as the Engine of Application State) je princip REST-a po kome je potrebno obezbediti način za otkrivanje i navigaciju interfejsom. Jednostavno objašnjenje se može naći na [2].

1.2. Spring i Spring Boot

Spring framework je open source (otvorenog koda) framework nastao 2002. godine i veoma je popularan u Java svetu. Namena mu je kreiranje kompleksnih aplikacija gde je jaka konkurencija EJB-u (Enterprise JavaBeans) [3]. Tokom godina evoluirao je uporedo sa programskim jezikom Java i trenutno se nalazi u verziji 5 koja kao minimalnu verziju Java zahteva verziju 8. Sastavljen je od mnoštva modula koje pokrivaju celokupni spektar potreba kompleksnih aplikacija.

Osnovni koncepti zastupljeni u Spring-u (Spring framework) su dependency injection / inversion of control (inverzija kontrole) i convention over configuration (konvencija pre konfiguracije). Inverzija kontrole je postignuta time što Spring aplikacija predstavlja kontejner za komponente koje mogu biti zavisne jedne od drugih. Svaka komponenta ume da kreira sebe, a komponente od kojih zavisi će joj biti pružene automatski od strane Spring kontejnera. Komponente se u terminologiji Springa nazivaju bean-ovi i one mogu biti različitog tipa.

NAPOMENA:

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

One dolaze konfigurisane po konvenciji, a njihovom korisniku je ostavljena mogućnost da konfiguraciju izmeni ako je potrebno. Na navedenim principima zasnovani su moduli od kojih se framework sastoji.

Ranije navedene glavne koncepte Spring framework-a pogotovu zapažamo pri radu sa Spring Boot modulom koji je korišćen kao osnova za razvoj primer aplikacije u okviru ovog rada. Spring Boot je osmišljen tako da ubrza razvoj aplikacija nudeći default (uobičajnu/podrazumevanu) konfiguraciju modula koji se uključuju u sastav razvijane aplikacije.

Takođe Spring Boot vodi računa o verzijama modula tako da je početak razvoja dodatno olakšan. Korišćenjem Spring Boot modula vrlo brzo se može preći na implementaciju biznis logike jer je dodatna konfiguracija od one sa kojom Spring Boot već dolazi često minimalna. Ovo je veoma olakšalo razvoj i dodatno podiglo popularnost Spring framework-a.

Spring framework je veoma fleksibilan tako da kada je potrebno uobičajna konfiguracija se lako može izmeniti i prilagoditi potrebama aplikacije.

1.3. Spring Data JPA

Spring Data predstavlja skup modula Spring framework-a zaduženih za rad sa skladištima podataka. Nudi veći broj podmodula orijentisanih ka konkretnom tipu skladišta podataka. Korišćeni podmodul Spring Data JPA [4] služi za rad sa relacionim bazama podataka. Pruža generički sloj za manipulaciju podacima i time znatno olakšava razvoj aplikacija.

Osnovna abstrakcija koju Spring Data JPA koristi je repository (repozitorijum). U okviru ovog modula postoje više vrsta repository interfejsa koji nude različite funkcionalnosti za pristup podacima.

Kreiranje konkretnog repozitorijuma vrši se nasleđivanjem interfejsa koji sadrži željene funkcionalnosti.

Interfejsi koji se nasleđuju su template (šablonski) sa prvim parametrom koji predstavlja tip entiteta koji repozitorijum obrađuje i drugim parametrom koji predstavlja tip primarnog ključa obrađivanog entiteta.

Repozitorijumi ponuđeni za nasleđivanje nude mnoštvo već postojećih metoda za rad sa entitetima kao što su standardne CRUD metode (create, read, update, delete - kreiranje, čiranje, ažuriranje i brisanje) kao i razne metode za osnovne pretrage kao što su metode za pronalaženje svih entiteta i metode za pronalaženje entiteta na osnovu primarnog ključa. Nije potrebno pisati implementaciju ovih metoda već će ona biti automatski pružena.

Pored nasleđenih metoda moguće je deklarirati i nove metode čije ime treba da prati specifikaciju koja je propisana u dokumentaciji i modul će i za njih automatski pružiti implementaciju.

Ovo predstavlja veoma moćan koncept koji znatno ubrzava razvoj sloja za manipulaciju podacima. Korisnicima modula je dodatno na raspolaganje stavljen i mehanizam gde oni sami mogu da odrede šta će definisane metode raditi tako da je pružena mogućnost i za izvršavanje veoma specifičnih funkcionalnosti.

Funkcionalnosti koje modul dodatno i automatski podržava su sortiranje i paginacija, a da bi ih modul za nas implementirao sve što treba da se uradi je da se navedu kao dodatni parametri deklariranih funkcija. Treba naglasiti da parametar za paginaciju u sebi već

sadrži podatke za sortiranje pa kada se koristi nije potrebno dodatno navoditi parametar za sortiranje.

1.4. Spring Data REST

Spring Data REST [5] je modul za ekstenziju sloja za manipulaciju podacima pružajući mogućnost da se definisani repozitorijumi otvore i izlože kao REST API. On zapravo predstavlja sam po sebi Spring aplikaciju i kao takav može da stoji uz već postojeću aplikaciju ili može da predstavlja osnovu pored koje bi se izgradila nova aplikacija.

2. SPECIFIKACIJA

Kao demonstrativnu aplikaciju mogućnosti Spring Data REST modula treba kreirati aplikaciju za issue tracking (praćenje zadataka i problema).

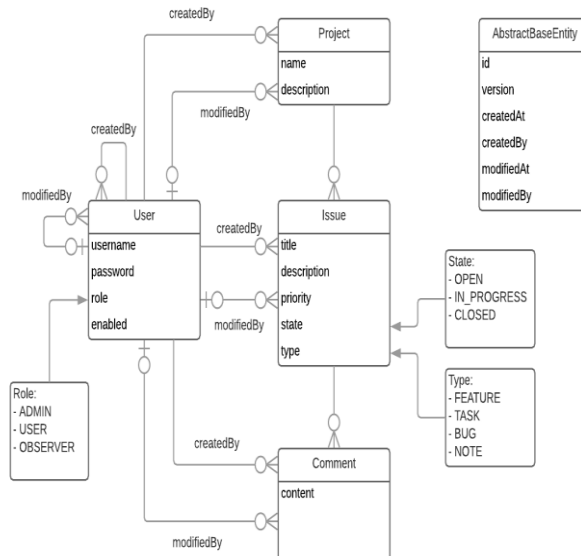
Funkcionalnosti aplikacije treba da su prilagođene demonstraciji mogućnosti posmatranog modula Spring-a i ne treba da pokrivaju potpuno domen problema issue trackinga.

2.1. Model podataka

Domenski model sadrži 3 tipa entiteta i to su po hijerarhiji poređani: project (projekat), issue (zadatak) i comment (komentar). Project sadrži kolekciju svojih issue-a, a oni imaju kolekciju svojih comment-a.

Svaki od tipova entiteta sadrži spopstvene informacije vezane za svoju namenu, a zajedničko za sve je da sadrže audit (revizione) podatke. Nije potrebno čuvati istoriju vrednosti već samo podatke o kreaciji i poslednjoj izmeni. Podaci od značaja za izmene su vreme i korisnik koji ih je učinio kao i redni broj verzije entiteta.

Pored entiteta direktno vezanih za domen problema aplikacija sadrži i korisnike. Postoje 3 tipa korisnika, to su admin (administrator), user (korisnik) i observer (posmatrač). Podrazumeva se da administrator ima veća prava od korisnika i da može da vrši iste stvari kao korisnik, dok korisnik ima veća prava od posmatrača i da može da vrši iste stvari kao i posmatrač. Dijagram modela dat je na slici 1.



Slika 1. Dijagram modela

2.2. Slučajevi korišćenja

Samo ulogovani korisnici smeju da pristupaju resursima sistema. Posmatrači imaju prava da vide sve podatke, ali ih ne mogu menjati. Korisnici mogu da kreiraju issue i comment-e, a mogu da vrše izmene nad entitetom tipa issue bez ograničenja dok comment entitet mogu da menjaju samo ako su ga oni i kreirali. Administrator kreira korisnike, ali ih ne može brisati. Brisanje korisnika nije moguće. Administrator kreira, ažurira i briše projekte, a može i da briše entitet tipa issue.

Tabelarni prikaz mogućnosti korisnika po entitetu dat je u tabeli 1.

Legenda:

- C - Pravo kreiranja (create).
- R - Pravo pregleda (read).
- U - Pravo izmene (update).
- U! - Uslovno pravo izmene (update).
- D - Pravo brisanja (delete).
- D! - Uslovno pravo brisanja (delete).

Tip korisnika entiteta	User	Project	Issue	Comment
Admin	C,R,U	C,R,U,D	C,R,U,D	C,R,U!,D!
User	R	R	C,R,U	C,R,U!,D!
Observer	R	R	R	R

Tabela 1. *Mogu'nosti korisnika*

2.3. Dodatne napomene

S obzirom da entitet korisnika sadrži password (lozinku) ona se ne sme čuvati u otvorenom obliku. Lozinka ne sme da bude dostupna preko REST API za čitanje, ali je dostupna za unos i izmenu.

3. IMPLEMENTACIJA

U ovom poglavlju biće opisana implementacija demonstrativne aplikacije sa fokusom na implementaciju njenog backend dela. Pregled implementacije će biti izložen prolazom kroz izabrane funkcionalnosti.

3.1 Osnovna podešavanja

U okviru osnovnih podešavanja moguće je podesiti baznu putanju na kojoj su izloženi REST repozitorijumi, strategiju izlaganja repozitorijuma preko REST interfejsa. Inicijalno identifikatori entiteta nisu uključeni u JSON odgovora, ali ih je moguće uključiti pojedinačno navodeći klase koje modeluju entitete.

U radu je korišćena zajednička bazna klasa za sve JPA entitete, ali identifikatore nije moguće uključiti navođenjem bazne klase.

3.2 Sigurnosna podešavanja

S obzirom da se repozitorijumi izlažu automatski kao REST API na osnovu izabrane strategije potrebno je posebnu pažnju posvetiti zaštiti.

Zaštita se ostvaruje na standardne načine koji su dostupni sa Spring Security, ali je kao novina moguće zaštitne anotacije koristiti i na metodama koje reaguju na događaje nad repozitorijumom. Ovim mehanizmom je moguće presresti i sprečiti neželjene akcije.

U radu sa REST repozitorijumima je uočeno da se u slučaju kada je kao autentifikacioni mehanizam uključena autentifikaciona forma za neautentifikovane pristupe dobija HTTP odgovor 302 umesto očekivanja 401 i 403.

Autentifikaciona forma je iz navedenih razloga isključena i za autentifikaciju se koristi drugi mehanizam (Basic Auth). Pošto je REST API namenjen konzumiranju od strane drugih aplikacija za slučaj kada su one izvršavane u web pretraživačima potrebno je posetiti CORS (Cross-origin resource sharing) u okviru sigurnosnih podešavanja.

3.3 Model

Model je sačinjen od standardnih JPA entiteta izvođenjem bazne klase koja u sebi sadrži identifikator i revizijske podatke. Pored klasa koje modeluju entitete korišćene su i klase za modelovanje projekcija koje služe za kontrolu izlaznog JSON-a time što se pojedini atributi ili asocijacije mogu uključiti ili isključiti. Uz pomoć projekcija moguće je dodati vrednosti koje se isračunavaju evaluacijom SpEL izraza.

Na žalost, u radu sa REST repozitorijumima nije moguće navesti više od jedne projekcije koja bi se primenila na JSON odgovor. Nedostatak oko broja upotrebljenih projekcija donekle može da se reši nasleđivanjem s obzirom da se projekcije definišu kao interfejsi.

Za olakšavanje razvoja korišćena je biblioteka Lombok koja generiše set/get metode za pristup atributima entiteta. Pored JPA/Hibernate anotacija u modelu su korišćene i anotacije specifične za Jackson, komponentu koja vrši serijalizaciju između Java objekata i JSON-a. Korisnička lozinka je ovim mehanizmom izbačena iz JSON odgovora, ali je korišćena anotacija koja dozvoljava izmenu kroz REST repozitorijume.

3.4 Repozitorijumi

Ključnu tačku u okviru modula Spring Data REST predstavljaju JPA repozitorijumi kojima se kreiraju REST resursi. Moguća je kontrola omogućenih HTTP metoda koje odgovaraju metodi ili metodama repozitorijuma. U slučaju entiteta korisnika u odgovarajućem repozitorijumu je onemogućeno brisanje tako što su sve metode brisanja bile označene anotacijom koja ograničava izlaganje.

Metode pretrage koje su definisane u okviru repozitorijuma je moguće pozvati preko REST-a jer su izložene na `/search/` podputanjama resursa pod istim nazivom. Na sve CRUD (create, update, delete) metode repozitorijuma moguće reagovati implementacijom *event handlera*. Ovaj princip je iskorišćen za transformisanje lozinke korisnika.

4. ZAKLJUČAK

Spring Data REST predstavlja dobro rešenje za generisanje REST API-ja aplikacija. Posebna prednost mu je to što lako može da koegzistira sa već postojećim ručno napisanim REST API-jem.

Nudi mogućnost prilagođavanja potrebama aplikacije, a nudi i klase za ručno generisanje REST interfejsa.

Pri radu je bitno dobro upoznati ponašanje modula jer nije baš sve onako kako bi se očekivalo, o ovome je najviše bilo reči u radu u sekciji implementacije demonstrativnog projekta.

S obzirom da Spring Data REST otvara generički REST API potrebno je značajnije posvetiti pažnju bezbednosnim aspektima jer vrlo lako i neželjeno mogu da se otvore funkcionalnosti i podaci koji ne bi smeli biti dostupni.

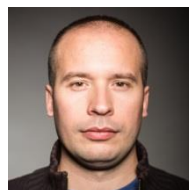
Kao što ni REST nije dobro rešenje za sve vrste aplikacije tako ni Spring Data REST nije rešenje za sve. Domen primene bi mu mogle biti aplikacije sa formama gde postoji obrada podataka jednostavnih struktura.

Aplikacije koje imaju složene izlazne podatke možda nisu pogodne za implementaciju korišćenjem Spring Data REST modula, ali i kod njih bismo savetovali da se deo aplikacije koji je pogodan za REST implementira korišćenjem pomenutog modula.

5. LITERATURA

- [1] <https://www.ics.uci.edu/~fielding/pubs-/dissertation/top.htm> (pristupljeno u avgustu 2018. godine)
- [2] <https://martinfowler.com/articles/-richardsonMaturityModel.html> (pristupljeno u avgustu 2018. godine)
- [3] https://en.wikipedia.org/wiki/Spring_Framework (pristupljeno u avgustu 2018. godine)
- [4] <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/> (pristupljeno u avgustu 2018. godine)
- [5] <https://docs.spring.io/spring-data/rest/docs/current/reference/html/> (pristupljeno u avgustu 2018. godine)

Kratka biografija:



Igor Gere rođen je u Novom Sadu 1982. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva - Računarske nauke i informatika odbranio je 2018. godine.

MODULARNI SISTEM ZA OBRADU ANALOGNIH SIGNALA**MODULAR SYSTEM FOR ANALOG SIGNAL PROCESSING**

Ana Mitrović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je dat opis modularnog sistema konstruisanog od različitih štampanih ploča, od kojih svaka ima posebnu funkciju za obradu analognog signala. Sistem je predviđen za korišćenje na predavanjima i vežbama u okviru osnovnih studija elektrotehnike kao sredstvo učenja. Dat je opis razvojnog softvera i prototipskog hardverskog rešenja, a prikazano je i testiranje hardvera sa rezultatima obrade signala.

Ključne reči: *analogna obrada signala, operacioni pojačavač, modularni sistem*

Abstract – *This paper contains the description of the modular system constructed from several different printed circuit boards, every one of them having its own function for analog signal processing. The system is designed to be used as a learning tool in bachelor studies of electronic engineering. A description of the development software and the prototype hardware solution is provided, as well as the results of hardware testing*

Keywords: *analog signal processing, operational amplifier, modular system*

1. UVOD

Operacioni pojačavač je veoma efikasan uređaj sa širokom primenom u elektronskoj industriji – koristi se za kondicioniranje signala, posebne prenosne funkcije, zatim u oblasti analogne instrumentacije i analognog računanja. Kola koja koriste operacione pojačavače karakterišu jednostavnost i preciznost.

Termin „operacioni pojačavač” nastao je zbog prvenstvene primene u analognim računarima, gde su se kola sa operacionim pojačavačima koristila za vršenje različitih matematičkih operacija. Tada je otkriveno da primena negativne povratne sprege oko pojačavača velikog pojačanja formira kolo sa preciznim karakteristikama pojačanja, koje u takvoj konfiguraciji zavisi pretežno od tipa povratne sprege.

Pravilnim izborom komponenata u povratnoj sprezi, kola sa operacionim pojačavačima mogla su da se koriste za sabiranje, oduzimanje, integraljenje, diferenciranje i slične matematičke operacije.

Dodatkom dioda ili tranzistora mogu se obavljati i neke nelinearne operacije nad signalima – množenje, deljenje, kvadriranje, logaritmovanje i sl.

NAPOMENA:

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

Kako su tehnike primene operacionih pojačavača postajale šire poznate, postalo je jasno da povratna sprema može da bude korisna u velikom broju primena u oblasti instrumentacije i automatskog upravljanja.

U današnje vreme operacioni pojačavači se pored primene u osnovnim kolima koriste i kao DC pojačavači, AC pojačavači, komparatori, dražveri servo ventila, oscilatori sa malom distorzijom, pretvarači efektivne vrednosti signala, multivibrator i mnogi drugi.

Može se reći da su mogućnosti primene operacionih pojačavača ograničene samo imaginacijom i znanjem korisnika.

Dobro poznavanje operacionih pojačavača i njihovih karakteristika otvara vrata ka boljem razumevanju sveta analogne elektronike i od velike je važnosti za studente elektrotehnike. Iz tog razloga je razvijen modularni sistem za obradu analognih signala koji objedinjuje neka od osnovnih kola u kojima se koriste operacioni pojačavači.

Zahvaljujući jednostavnom povezivanju i kombinovanju različitih modula, kao i mogućnosti posmatranja rezultata obrade na osciloskopu, ovaj sistem može da se koristi na vežbama i predavanjima u okviru studija elektrotehnike, kao sredstvo uz pomoć kog se studentima na zanimljiv način predstavlja osnovna primena operacionih pojačavača i obrada analognog signala.

2. OPIS SISTEMA

Predviđen je razvoj sistema koji se sastoji od osam štampanih pločica: jedne ulazne, na koju se dovode napajanje i ulazni signal, i sedam pločica sa različitim funkcijama – invertujući pojačavač, neinvertujući pojačavač, diferencijator, integrator, diferencijalni pojačavač, sabirač i oscilator. Svaka od navedenih pločica sadrži konektore postavljene tako da mogu jednostavno međusobno da se povezuju i da se na taj način gradi lanac različitih funkcija za obradu signala.

Kao početno rešenje napravljen je sistem od četiri prototipske pločice – pločica za dovodenje ulaznog signala i napajanja, neinvertujući pojačavač, sabirač i oscilator – koje će biti dalje prikazane u radu.

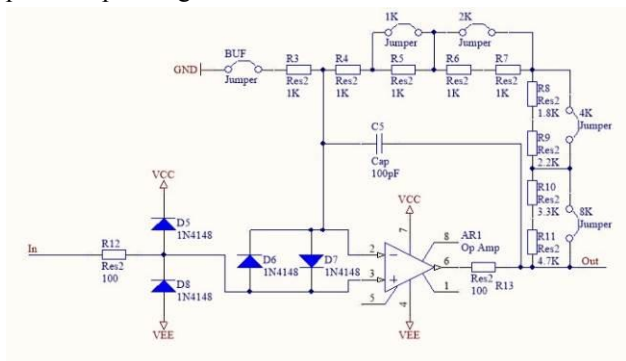
3. RAZVOJNI SOFTVER

Pločice su razvijane u programskom paketu Altium Designer, razvijenom od strane australijske softverske kompanije Altium Limited (do 2001. godine poznate kao Protel). Altium Designer je programski paket širokih mogućnosti koji se koristi za razvoj uređaja. Sadrži mogućnost crtanja elektronske šeme (pri čemu mogu da se definišu komponente, uključujući i njihov izgled na štampanoj ploči), kao i mogućnost razvoja same štampane ploče. U Altium Designer-u napravljena je posebna biblioteka u

kojoj je definisan izgled komponenata na štampanoj ploči. Nakon toga je za svaku od osam pločica izrađena elektronska šema, na osnovu koje je dalje projektovana štampana ploča. Razvoj štampane pločice biće prikazan na osnovu jednog od elemenata modularnog sistema - neinvertujućeg pojačavača.

3.1. Elektronska šema

U konfiguraciji neinvertujućeg pojačavača, ulazni naponski signal se dovodi direktno na neinvertujući ulaz pojačavača, što rezultuje time da je izlazni signal u fazi sa ulaznim – ne menja se znak. Kontrola povratnom spregom postiže se vraćanjem malog dela izlaznog napona na invertujući ulaz putem naponskog razdelnika.

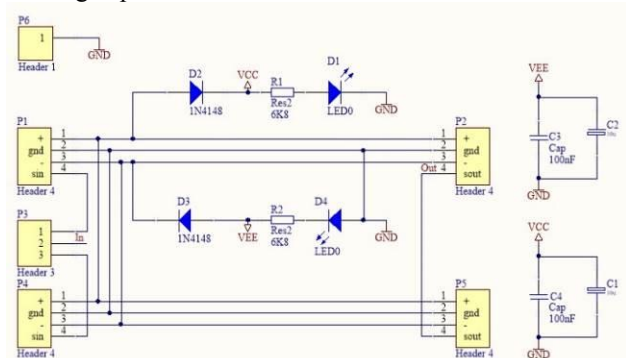


Slika 1. *Elektronska šema neinvertujućeg pojačavača - 1 deo*

Ovakvom konfiguracijom dobijamo pojačanje koje se definiše sledećom formulom:

$$A_{(v)} = 1 + \frac{Rf}{R3}$$

gde R_f predstavlja jedan otpornik ili zbir više otpornika u povratnoj sprezi (od R4 do R11 na slici 1). Na taj način, izborom otpornika pomoću različitih prekidača, postiže se pojačanje u opsegu od 2 do 17 u inkrementima po 1. Takođe, korisnik ima mogućnost da, isključivanem prekidača „BUF“, izabere da kolo funkcioniše kao jedinični pojačavač - bafer. Dioda i otpornik R12 na ulazu operacionog pojačavača služe za zaštitu kola od potencijalnog visokog napona na ulazu.



Slika 2. *Elektronska šema neinvertujućeg pojačavača - 2. deo*

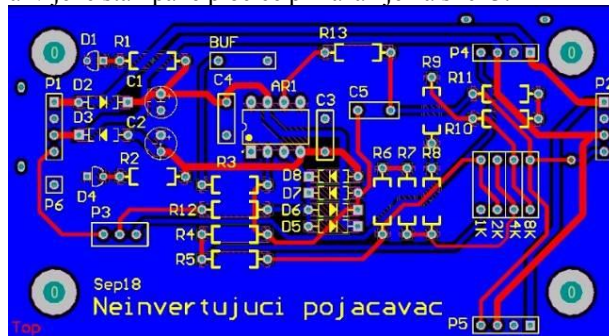
Sve pločice (osim ulazne) su konstruisane tako da imaju po dva ulazna i dva izlazna konektora, jedne naspram drugih, pri čemu je napajanje povezano na sva četiri konektora. Napajanje se preko dioda dovodi na ostatak kola – na taj način je kolo osigurano od pogrešnog spajanja napajanja. Na svakoj pločici su pored operacionih pojačavača

postavljena po dva kondenzatora za filtriranje i stabilizaciju napajanja. Na pločicama su montirane i po dve LED diode, koje služe kao indikatori prisustva napajanja.

Na šemi na slici 2 primećujemo i konektor P6, koji služi npr. za povezivanje sonde osciloskopa na masu, kao i P3, koji služi za izbor konektora sa kog dovodimo ulazni signal na pojačavač.

3.2. Razvoj štampane ploče

Na osnovu prikazane elektronske šeme projektovana je štampana ploča za neinvertujući pojačavač. Izgled razvijene štampane pločice prikazan je na slici 3.



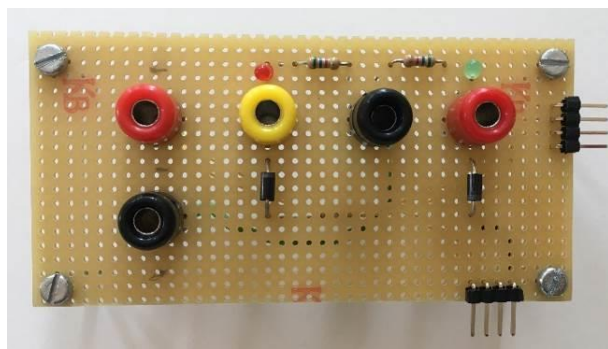
Slika 3. Razvijena štampana pločica neinvertujućeg pojačavača

4. HARDVER

Inicijalno hadversko rešenje predstavlja sistem od četiri prototipske pločice.

4.1. Pločica za dovodenje ulaznog signala i napajanja

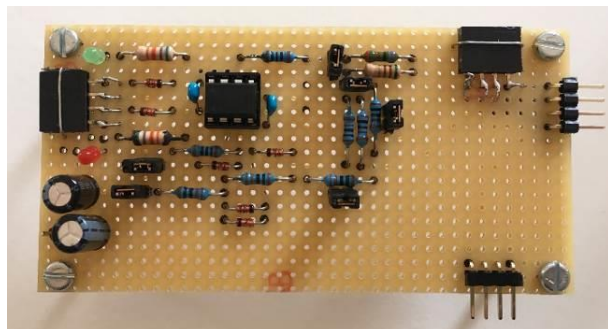
Na ovu pločicu postavljeno je pet priključaka – dva za ulazni signal (signal i masa) i tri za dovodjenje napajanja (+, - i masa). Ulazna pločica prikazana je na slici 4.



Slika 4. Pločica za dovođenje ulaznog signala i napajanja

4.2. Neinvertujući pojačavač

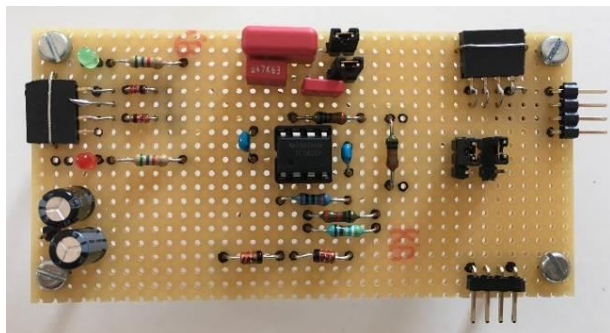
Pločica koja ima funkciju neinvertujućeg pojačavača prikazana je na slici 5.



Slika 5. *Neinvertujući pojačavač*

4.3. Oscilator

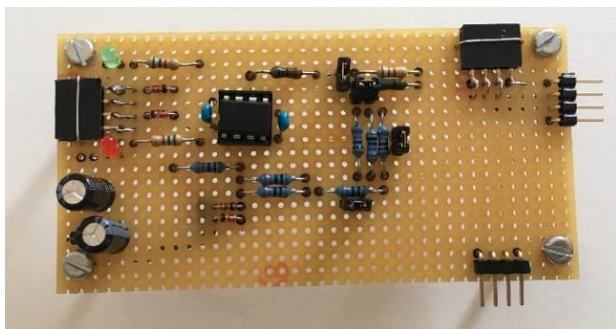
Oscilator je konstruisan tako da na izlazu može da daje povorku pravougaonih ili trougaonih impulsa, što korisnik može da bira postavljanjem kratkospojnika na određene konektore. Izborom kombinacije različitih kondenzatora u kolu omogućeno je podešavanje frekvencije izlaznog signala. Prototipska verzija oscilatora data je na slici 6.



Slika 6. Oscilator

4.4. Sabirač

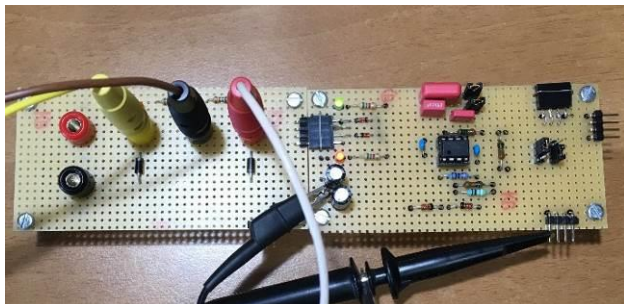
Slično kao kod neinvertujućeg pojačavača, kod sabirača kombinacijom različitih kratkospojnika postoji mogućnost izbora pojačanja zbira dva ulazna signala. Na slici 7 prikazana je pločica koja ima funkciju sabirača.



Slika 7. Sabirač

5. TESTIRANJE SISTEMA I REZULTATI

Modularni sistem za analizu analognog signala testiran je uz pomoć napajanja, ton generatora i osciloskopa. Na slikama 8 i 9 prikazano je povezivanje oscilatora na napajanje i izlaz oscilatora – povorka četvrtke i povorka trougaonih impulsa.

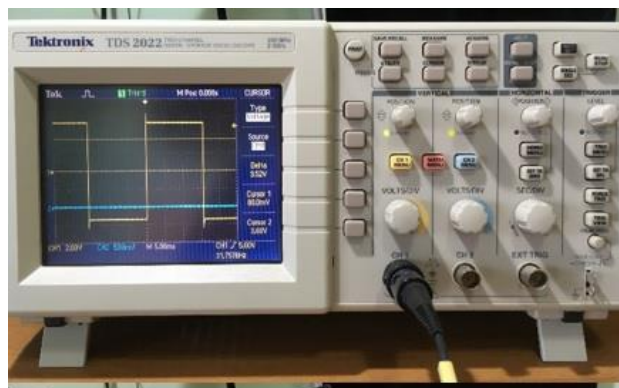


Slika 8. Povezivanje oscilatora na napajanje

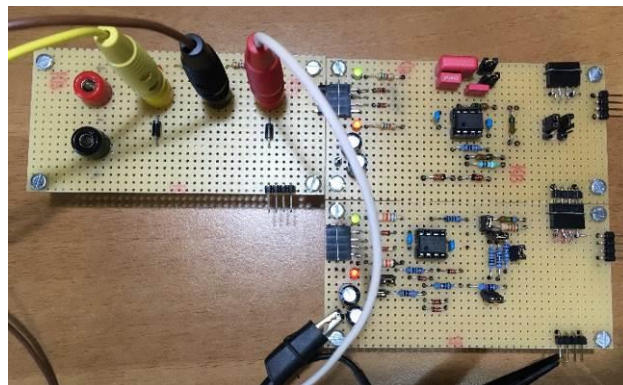
Ukoliko se na izlaz oscilatora koji generiše trougaoni napon amplitude 2,7 V poveže neinvertujući pojačavač i podesi pojačanje na 2 uključivanjem svih kratkospojnika na pločici, napon na izlazu je povećan na 5,4 V, kao što je i predviđeno. Slika 10 prikazuje povezivanje ove dve

prototipske pločice, dok je na slici 11 dat prikaz rezultata njihovog povezivanja na osciloskopu.

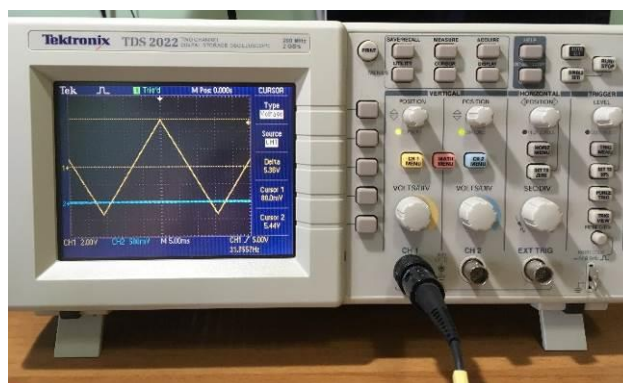
Na slikama 12, 13 i 14 prikazan je rezultat povezivanja pločica sa oscilatorom i sabiračem signala, pri čemu je na jedan ulaz sabirača dovedena sinusoida sa ton generatora, a na drugi ulaz signal sa oscilatora – četvrtke i trougaoni impulsi.



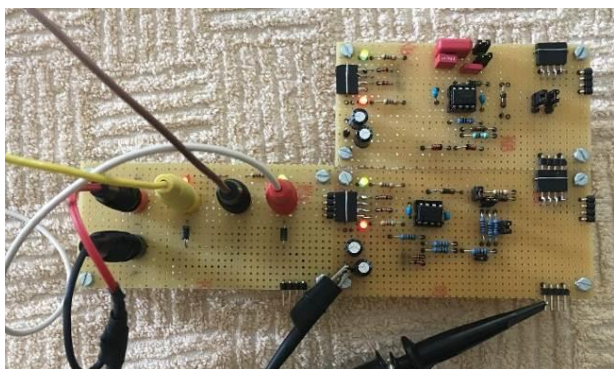
Slika 9. Izlaz oscilatora prikazan na osciloskopu



Slika 10. Kombinacija oscilatora i neinvertujućeg pojačavača



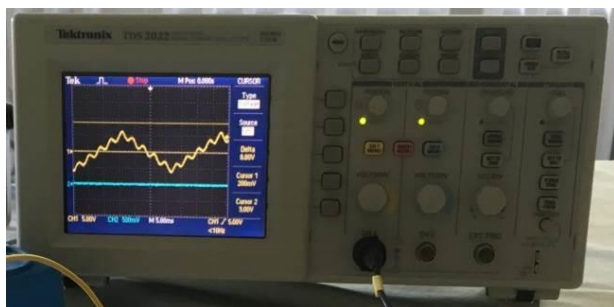
Slika 11. Izlaz neinvertujućeg pojačavača



Slika 12. Kombinacija oscilatora i sabirača



Slika 13. Sabiranje sinusoide sa ton generatora i povorke pravougaonih impulsa sa oscilatora



Slika 14. Sabiranje sinusoide sa ton generatora i trougaonog napona sa oscilatora

6. ZAKLJUČNA RAZMATRANJA

Razvijeni modularni sistem za obradu analognog signala pruža mogućnost učenja na interesantan način – omogućava studentima da jednostavnim povezivanjem pločica sa različitim funkcijama formiraju lanac funkcija i posmatraju promene signala. Na taj način oni na licu mesta vide konstrukciju i delovanje neinvertujućeg pojačavača, integratora, diferencijalnog pojačavača, oscilatora i sličnih kola sa operacionim pojačavačem i tako se upoznaju sa njegovim osnovnim primenama.

Postoji veliki broj modula koji mogu da se konstruišu na sličan način, te se očekuje dalji razvoj ovog modularnog sistema uvođenjem pločica sa novim funkcijama (npr. antilogaritamski pojačavač, instrumentacioni pojačavač i sl), kao i primena sistema na laboratorijskim vežbama iz više predmeta na Fakultetu tehničkih nauka u Novom Sadu i u drugim visokoškolskim ustanovama.

7. LITERATURA

- [1] Radojle Radetić, „Operacioni pojačavači – odabrana poglavlja“, Bor, 2013.
- [2] Bruce Carter, Thomas R. Brown, „Handbook of operational amplifier applications“, Texas Instruments Incorporated, 2001.
- [3] James K. Roberge, „OPERATIONAL AMPLIFIERS: Theory and Practice“, Massachusetts Institute of Technology, 1975.
- [4] <https://www.altium.com/> (19.09.2018.)

Kratka biografija



Ana Mitrović rođena je u Novom Sadu 1994. godine. Diplomirala je na Fakultetu tehničkih nauka u Novom Sadu, na katedri za električna merenja, 2016. godine.

PRIMENA SCADA SISTEMA U SMART GRIDU**APPLICATION OF SCADA SYSTEM IN SMART GRID**Marko Vujić, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO****2. SCADA SISTEMI**

Kratak sadržaj – U okviru ovog rada predstavljena je uloga SCADA sistema u Smart Grid-u. Opisana je arhitektura SCADA softvera koji zadovoljava potrebe Smart Grida. Predstavljani su parametri i kriterijumi koje moraju da ispune proizvođači SCADA softvera kako bi zadovoljili zahteve klijenata. Predstavljani su najbolji SCADA softveri i pojedinačno anlizirani u pogledu na ispunjavanje Smart Grid standard.

Ključne reči: SCADA sistemi, Smart Grid, ADMS

Abstract – This paper presents the role of the SCADA system in the Smart Grid. The architecture of SCADA software that meets Smart Grid needs is described. Featured are the parameters and criteria to be met by manufacturers of SCADA software in order to meet customer requirements. The best SCADA software is presented and individually described in terms of completing the Smart Grid standard.

Keywords: SCADA sistemi, Smart Grid, ADMS

SCADA (Supervisory Control And Data Acquisition) je sistem koji služi za automatizaciju opštih procesa, odnosno koji se koristi za prikupljanje podataka sa senzora i instrumenata lociranih na udaljenim stanicama i za prenos i prikazivanje tih podataka u centralnoj stanici u svrhu nadzora ili upravljanja. Prikupljeni podaci se obično posmatraju na jednom ili više SCADA računara u centralnoj (glavnoj) stanici. SCADA sistem u realnosti može da prati i upravlja i stotinama hiljada ulazno-izlaznih vrednosti. Uobičajeni analogni signali koje SCADA sistem nadzire (ili upravlja) su nivoi, temperature, pritisci, brzine protoka i brzine motora. Tipični digitalni signali za nadzor (upravljanje) su prekidači nivoa, prekidači pritiska, status generatora, releji i motori [1].

3. SMART GRID

Pametna mreža (Smart Grid) kao tehnologija naprednih elektroenergetskih mreža je ključna komponenta u polju primene obnovljivih izvora energije, odnosno jedan od temeljnih stubova na kojima počiva ideja obnovljivih izvora energije. Električna snaga uređaja i kontrola proizvodnje i distribucije električne energije su vrlo važni aspekti pametne mreže. Smart Grid napredne elektroenergetske mreže predstavljaju skup tehnologija koje omogućavaju bolju integraciju obnovljivih izvora u elektroenergetsku mrežu te uvode nove tehnologije i tehnološke inovacije koje omogućuju da klasična mreža funkcioniše na nešto drugačiji, stabilniji i precizniji način nego što je sada.

4. ULOGA SCADA SISTEMA U SMART GRID-U

SCADA sistemi imaju značajnu ulogu u Smart Grid-u i manifestuje se u pogledu što bržeg, efikasnijeg i pouzdanijeg prikupljanja informacija sa RTU (PLC), na osnovu kojih će se izvršiti određene komande, generisati alarmi kako bi detektovali istupanja iz normalnog stanja, tačno prikazalo stanje sistema kako bi operater imao jasnu sliku kompletnog sistema koji nadzire.

4.1. SCADA aplikacije za kontrolni centar i njeni korisnici

Funkcionalnosti u prenosnim i distributivnim mrežama koje imaju „Smart“ karakteristike ostvaruju se u kontrolnom centru pomoću nekoliko korisnih aplikacija vezanih za električnu uslugu, koje uključuju sledeće celine: SCADA, DMS, EMS, NIS (Network Information System), GIS (Geographic Information System). Tipična uloga, odnosno poslovi koje obavljaju ljudi koji su

1. UVOD

Tehničkim razvojem i stalnim unapređenjem proizvodnih procesa u svim sferama društva (industrije, privrede) postavljaju se zahtevi za što efikasnijim, sigurnijim i pouzdanijim praćenjem i upravljanjem tim procesima kako bi oni bili što produktivniji. Sistemi za prenos i distribuciju energenata (prirodni gas, nafta, električna energija, voda za grejanje), su prostorno raspodeljeni, složeni i neophodan je sistem koji će služiti za nadzor tih procesa. Za njihovu bezbednu i efikasnu isporuku zahteva se poznavanje trenutnog tehnološkog stanja ukupnog transportnog sistema, odnosno stanje svih parametara koji utiču na njihovu distribuciju i prenos. Razvojem računarskih sistema, kao i merne opreme omogućeno je daljinsko praćenje ovih parametara, i prikazivanje njihovih rezultata operaterima u fizički udaljenim stanicama koje su zadužene za njihovo praćenje. Kompletna automatizacija ovih procesa je omogućila razvijanje posebnih programa koji su namenjeni za praćenje proizvodnih procesa u industriji kao i praćenju procesa prenosa energenata.

Ovakva programska struktura ispraćena adekvatnim hardverskim sklopom predstavlja zaseban SCADA sistem. Ovi sistemi danas predstavljaju jednu od najznačajnijih uloga u navedenim procesima i njihova primena je sve učestalija.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Katić, red. prof.

performanse sistema, u zavisnosti od veličine mreže i topologije; Generisanje alarma, *event*-a (događaja), prikaz trenda promene vrednosti uz adekvatne dijagrame; Obezbeđivanje adekvatnog i jednostavnog korisničkog interfejsa; Ispunjavanje svih bezbednosnih uslova vezanih za zaštitu podataka i njihove razmene.

6. TRŽIŠTE SCADA SISTEMA U SVETU

Ulaganje u razvoj SCADA softvera se iz godine u godinu povećava. U ovom poglavlju su predstavljeni neki od proizvođača.

6.1 OSI SCADA

OSI SCADA softver na tržištu se ističe svojom *monarch* arhitekturom, koja predstavlja *real-time* sistemsku arhitekturu, podržanu visokim performansama koje zahteva današnje tržište i aplikacijama za praćenje i nadgledanje mreže, optimizaciju, složene mrežne operacije koje se koriste u prenosu električne energije, gasa, nafte. *Monarch* arhitektura uključuje sledeće sisteme: SCADA, DMS, EMS, GMS, NMS i MMS sistem. Predstavljena je kao fleksibilna arhitektura koja bez problema može da podrži u radu sistema sa milion tačaka i koja je proširiva za nove protokole i standarde. OSI ovakvom arhitekturom garantuje procesiranje milion transakcija po sekundi, kao i stotina hiljada alarma i *event*-a u minuti, kao i korisnički interfejs koji zadovoljava sve uobičajene tenderske kriterijume [3].

6.2 General Electric (GE)

GE SCADA proizvod se zasniva na *iPower* softverskoj aplikaciji čija je osnovna uloga obezbeđivanje svih rešenja koji zahteva moderni SCADA sistem. *iPower* obezbeđuje moćno softversko rešenje dizajnirano za kontrolne centre, koji imaju ulogu u nadgledanju, upravljanju električnim mrežama i kao takvo obezbeđuje upravljanje dinamikom mreže, čuvanje podataka u bazi, prikaz dinamike i topologije i mnoge druge značajne stvari koje mora moderni SCADA da podrži. Svojim HMI GE daje kvalitetno rešenje kojim zadovoljava sve kriterijume tržišta, kao i zahteve operatora [4].

6.3 Siemens SCADA

Kao odgovor na potrebe zahteva *Smart Grid*-a, Siemens nudi *Spectrum Power* softversko rešenje. Ova arhitektura omogućava brz i precizan pregled kompletne mreže, što operatorima daje brzu procenu statusa mreže, kao i donošenje ispravnih odluka na osnovu trenutnog stanja mreže. Važan deo *Spectrum Power* arhitekture je DSCADA (Distribution SCADA), koji utiče na poboljšanje pouzdanosti, efikasnosti i optimizacije celokupnog sistema. DSCADA prati sve telemetrijske i netelemetrijske operativne podatke, uključujući alarmiranje svih neželjenih uslova rada, izvršavanje naredbi operatera koje zadaju iz kontrolnih centara preko korisničkih interfejsa, kao i dodatne prelazne funkcije. Tri ključne stvari koje Siemens naglašava, koje karakterišu SCADA deo ADMS *Spectrum Power* aplikacije su pouzdanost, efikasnost i optimizacija [5].

6.4 Schneider Electric

Schneider Electric predstavlja jednog od globalnih lidera u proizvodnji SCADA softvera i *Smart Grid* aplikacija. Schneider Electric SCADA je integrisana u ADMS

softver i predstavlja FEP servis, koji obezbeđuje sve SCADA funkcionalnosti. Odlikuju ga izuzetne performanse, efikasnost i proširivost u pogledu implementiranja novih industrijskih i komunikacionih protokola. Posедуje funkcionalnosti koje omogućavaju operaterima jednostavnije praćenje. Integrisan je sa DMS delom softvera, koji obezbeđuje funkcionalnosti vezane za optimizaciju snaga u mreži, detektovanje kvara u mreži i dr. Implementiran je rad sa kalkuliranim tačkama, prikaz statistike, kao i dodatni aplikativni alati koji pomažu operaterima u praćenju mrežne dinamike.

6.5 ABB SCADA

ABB kao SCADA proizvod ističe softversku SCADA platformu u realnom vremenu za sigurno i efikasno upravljanje svim daljinskim i kontrolisanim operacijama u proizvodnji, prenosu i distribuciji električne energije. Ističe se ABB proizvod *MicroSCADA Pro* rešenje koje predstavlja aplikaciju na nivou jedne stanice (mini SCADA sistem samo za praćenje npr. jedne transformatorske stanice), kao i posebno distributivno generisano rešenje za integraciju i upravljanje obnovljivim izvorima energije (na pr. solarne FN elektrane, vetroelektrane i sl.) i na taj način omogućavaju daljinski nadzor, pojednostavljene operacije rada, analitiku i prognožiranje [6].

7. POJEDINAČNA ANALIZA DOSTUPNIH SCADA SISTEMA NA TRŽIŠTU

Na osnovu iznetih parametara koje nameće trenutno tržište SCADA sistema, i opisanih kompanija koji su lideri u proizvodnji SCADA softvera, možemo izvršiti pojedinačnu analizu svako softverskog rešenja.

OSI SCADA sa performansama koje ističe OSI (procesuiranjem milion tačaka u sekundi) možemo reći da takav sistem može da se primeni u nekom velikom geografskom prostoru sa velikim brojem potrošača, transformatorskih stanica, odnosno krajnjih korisnika. Razlika u odnosu na ostale SCADA sisteme navedene u ovom radu je mogućnost integracije na *Cloud* servis ovog SCADA rešenja, što donosi poboljšanje performantnosti sistema i smanjenje dodatnih troškova u vidu dodavanja hardvera za čuvanje podataka u bazi.

General Electric SCADA sistem poseduje sve potrebne karakteristike modernog SCADA sistema. General Electric na tržištu nam nudi ADMS softver koji ima posebnu DPF aplikaciju, čija je namena praćenje, upravljanje topologijom mreže u realnom vremenu. Ovakvo rešenje je predviđeno za velike geografske oblasti, jer DPF aplikacija prikuplja podatke sa više SCADA sistema, obrađuje iste, i prikazuje ih operaterima.

Siemens SCADA sistem predstavlja deo celokupne moćne *Spectrum Power* arhitekture, koja omogućava praćenje, kontrolu i komandovanje u distributivnoj i prenosnoj mreži. Ovaj SCADA sistem ima značajnu ulogu u uspostavljanju mnogih funkcionalnosti koje donose EMS i DMS funkcije, čiji je cilj poboljšanje rada generatora, smanjenje gubitka snage i kompletan rad sa mrežnom topologijom.

Schneider Electric SCADA kao deo ADMS softvera predstavlja podršku za funkcionalnosti koje donose DMS i EMS funkcije. Performanse ovog softvera su veoma

visoke i na taj način predstavljaju jednog od tržišnih lidera. Ovaj sistem je predviđen za praćenje velikog broja potrošača, poseduje distribuiranu arhitekturu, visok nivo sigurnosti. Omogućeno je praćenje preko *Web* aplikacija, kao i posebna aplikacija koja obezbeđuje obaveštenje terenskim radnicima o mestu i uzroku kvara.

ABB SCADA sistem je takođe integrisan u ADMS rešenje. Naglasimo rešenje u vidu ABB MicroSCADA Pro, koje obezbeđuje praćenje i nadziranje dela mreže posebnih karakteristika i zahteva praćenja. Ovaj SCADA sistem ima primenu kod vetroelektrana, solarnih elektrana, ili nekih specifičnih potrošačkih jedinica kao što su železnice, velika fabrička postrojenja itd.

8. ANALIZA NAJBOLJEG SCADA REŠENJA U SMART GRID-U.

Analizirajući zahteve koje nameće tržište, vezano za SCADA softvere i upoređujući već pojedinačno analizirane SCADA sisteme, mogu se doneti određeni zaključci oko izbora najpogodnijeg rešenja u zavisnosti od potrebe klijenata.

SCADA distribuirani sistem, integracija u ADMS softver: po ovom parametru svi SCADA sistemi su integrisani u ADMS. Ovo donosi ogromna poboljšanja, i kompletnu integraciju u Smart Grid rešenje. U ovom segmentu se ističu Schneider Electric i Siemens, ne samo po dobrim SCADA karakteristikama, već i po dodatnim EMS, DMS funkcijama.

Komunikacija: većina SCADA softvera koji su analizirani u ovom radu implementiraju zahtevane protokole (DNP3, IEC 101 i 104, IEC).

Proširivost i postavka softvera: prednost imaju softveri kod kojih je SCADA deo implementiran u vidu FEP servisa, a ne kao dodatna instalacija ili posebna aplikativna sekcija. Schneider Electric poseduje softver sa najboljim karakteristikama u ovom pogledu.

Smart Grid funkcionalnosti: SCADA sistemi koji su integrisani u ADMS imaju ulogu da svojom sigurnošću i pouzdanosti, doprinesu ostvarivanju funkcija koje implementiraju DMS i EMS delovi. Schneider Electric i Siemens imaju funkcije koje omogućavaju operaterima proračune mogućih dešavanja u sistemu, pre svake manipulacije operatera.

Performanse: izdvajaju se Schneider Electric (procesiranje milion vrednosti u sekundi, podrška za rad sa 10 miliona tačaka) i OSI SCADA koja ističe mogućnost procesiranja milion transakcija u sekundi i mogućnost generisanja 100 hiljada alarma u minutu.

Integracija na Cloud servise: Jedino je OSI SCADA trenutno integrisana na Cloud servis, dok Schneider Electric tu mogućnost najavljuje u novoj verziji softvera.

9. ZAKLJUČAK

U ovom radu je opisano funkcionisanje SCADA sistema u budućim pametnim mrežama (*Smart Grids*). Predstavljani su dostupni SCADA sistemi razvijeni od vodećih i najjačih kompanija u domenu elektrotehnike, energetike, automatike i softverskog upravljanja i izvršeno njihovo poređenje. Analizirano je najpovoljnije rešenje za primenu u pametnim mrežama.

10. LITERATURA

- [1] http://www.keep.ftn.uns.ac.rs/predmeti/ee2_3g_insys_protokoli/ISIP%20skripta%20-%20SCADA.pdf
- [2] A.Leonardi, K.Mathioudakis, A.Wiesmaier, F.Zeiger, „Towards the Smart Grid: Substation Automation Architecture and Technologies Hindawi Publishing Corporation“, *Advances in Electrical Engineering* Vol. 2014, Article ID 896296,.
- [3] <http://www.osii.com/solutions/products/scada.asp>
- [4] <https://www.ge.com/digital/sites/default/files/ge-digitals-hmi-scada-with-ipower-brochure.pdf>
- [5] <https://w3.usa.siemens.com/smartgrid/us/en/distribution-grid/products/distribution-scada-system-platforms/pages/dscada.aspx>
- [6] <https://new.abb.com/network-management/network-management/microscada-pro/sys600>

Kratka biografija:



Marko Vujić rođen je 1989. god. u Šapcu. Diplomirani inženjer elektrotehnike i računarstva postao je 2013. god. na Fakultetu tehničkih nauka. Master studije je upisao školske 2013/2014. god. na studijskom programu Energetika, elektronika i telekomunikacije, modul Elektro-energetika i završio ih 2018. god. Trenutno je medior software inženjer u Panon IT Novi Sad, outsourcing u Schneider Electric DMS NS.



Vladimir Katić rođen je 1954. god. u Novom Sadu. Doktorirao je na Univerzitetu u Beogradu 1991. god. Od 2002. god. je redovni profesor Univerziteta u Novom Sadu. Trenutno je prodekan Fakulteta tehničkih nauka. Oblasti interesovanja su energetska elektronika, obnovljivi izvori električne energije, električna vozila i kvalitete električne energije.

**RAZVOJNO OKRUŽENJE ZA PROJEKTOVANJE I GENERISANJE KODA
NEURONSKIH MREŽA****NEURAL NETWORK SOFTWARE DEVELOPMENT ENVIRONMENT WITH CODE
GENERATION CAPABILITIES**

Aleksandra Mitrović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je predstavljeno razvojno okruženje za kreiranje modela neuronskih mreža i generisanje koda na osnovu istih. Takođe, objašnjen je i grafički DSL koji se koristi za izradu modela i prikazan je primer upotrebe okruženja.

Ključne reči: *Neuronske mreže, generator koda, DSL, IDE.*

Abstract – *This paper presents a neural network model development environment with code generation capabilities. This paper also describes a DSL used for specifying neural network model and shows an example of its use.*

Keywords: *Neural networks, code generator, DSL, IDE.*

1. UVOD

Upotreba mašinskog učenja za rešavanje kompleksnih problema je najčešći izbor u poslednjih nekoliko godina. Popularnost ovog pristupa je značajno porasla u poslednjoj deceniji kada su računarske komponente postale dostupnije i efikasnije. Jedan od često korišćenih tipova mašinskog učenja predstavljaju neuronske mreže, čiji je domen upotrebljen za kreiranje jezika specifičnog za domen (eng. DSL). Cilj ovog jezika jeste da omogući brže kreiranje modela ekspertima, ali isto tako da i onim korisnicima koji tek zalaze u domen neuronskih mreža omogući kreiranje modela bez potrebe za poznavanjem nekog programskog jezika.

Kako arhitektura neuronskih mreža predstavlja usmeren graf, gde čvorovi predstavljaju različite operacije koje se sprovode nad ulaznim podacima, a veze predstavljaju prosledjivanje izlaza jedne operacije na ulaz drugih, kreiranje grafičkog DSL-a bi doprinelo boljem razumevanju toka podataka i redosledu primene operacija. Vizuelizacija jezika je omogućena u već postojećem rešenju FLCT [1] koji predstavlja razvojno okruženje za projektovanje i generisanje koda fazi logičkih kontrolera. Ovo okruženje je za DSL iz domena fazi logičkih kontrolera omogućilo vizuelizovano kreiranje šeme kontrolera, validaciju šeme, kao i generisanje koda spram zadate šeme. Rezultirajući kod se može upotrebiti za neki drugi program ili direktno postaviti na kontroler nekog fizičkog uređaja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Gordana Milosavljević, van.prof.

Uključivanjem još jednog atraktivnog domena u ovo okruženje otvara mogućnost kombinovanja dva jezika, jer su bazirani na osnovnom modelu jezika, gde se dodaju samo domenske specifičnosti.

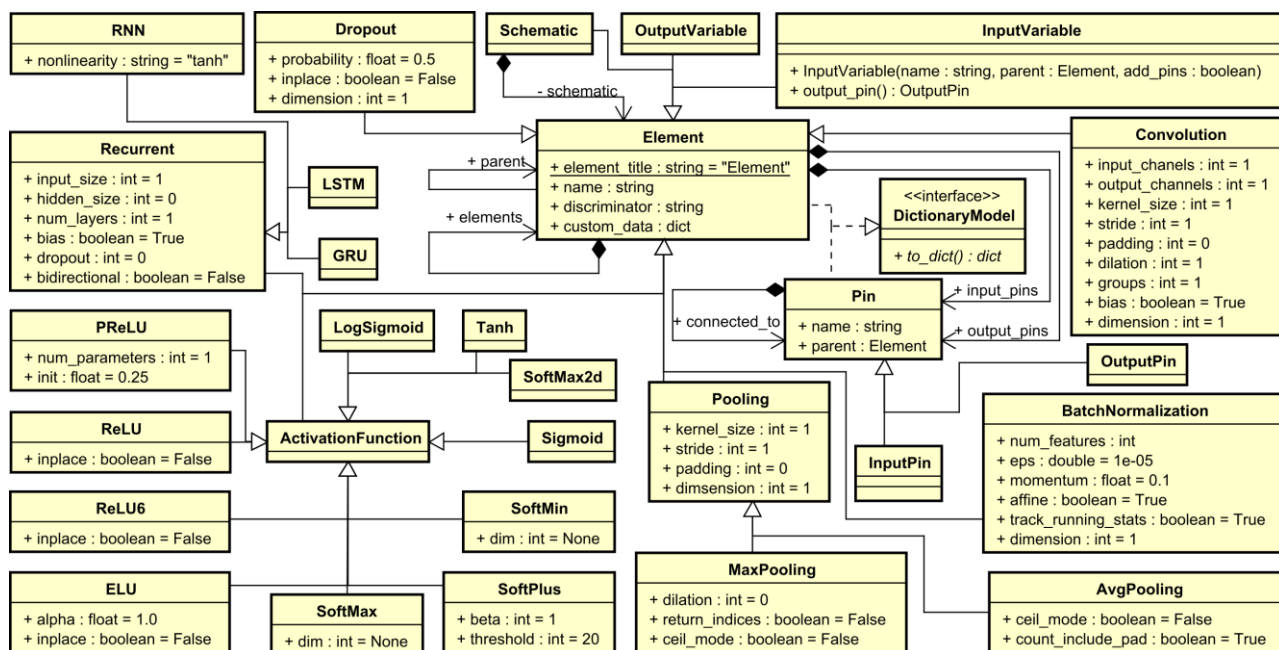
2. DSL ZA NEURONSKE MREŽE

Za formiranje ovog jezika u obzir je uzeta struktura i model prethodnog jezika za fazi logičke kontrolere, ali i biblioteka za rad sa neuronskim mrežama PyTorch [2].

2.1. Gramatika i činioci jezika

Za formiranje jezika koji se oslanja na strukturu grafa, potrebno je definisati osnovne pojmove koji odgovaraju čvoru i vezi. U ovom jeziku to predstavljaju *Element* i *Pin* redom. Uvođenjem termina iz neuronskih mreža koji predstavljaju operacije se kreiraju specijalizacije *Element-a*. Dok za *Pin* postoje samo dve specijalizacije koje bliže određuju da li se podaci prosledjuju elementu ili se prosledjuju iz elementa. Pomoću ovog jezika moguće je formirati hijerarhijsku strukturu, i imati ugnježdene elemente. Na vrhu hijerarhije se nalazi *Schematic* (srp. šema) koja predstavlja forward-pass modula neuronske mreže. Za dobavljanje podataka i isporuku koriste se *InputVariable* i *OutputVariable* (srp. ulazna i izlazna varijabla), koje predstavljaju specijalizacije *Element-a*. Činioce jezika, pored osnovnih koji su prethodno navedeni, predstavljaju i:

- aktivacione funkcije – služe za određivanje pobuđenosti neurona na osnovu ulaza, u ovoj verziji jezika podržane su samo neke, često korišćene, aktivacione funkcije;
- konvolucioni slojevi – primenjuju konvolutivnu neuronsku mrežu na ulazne podatke i rezultat isporučuju na izlaz elementa;
- dropout slojevi – služe za regulazaciju ulaznih podataka, koji u procesu obučavanja sprečavaju preobučanost;
- linearni slojevi – primenjuju linearnu transformaciju ulaznih podataka;
- slojevi za normalizaciju – vrše mapiranje ulaznih vrednosti na prethodno definisan skup vrednosti, pomaže u obučavanju kod previše velikih ili previše malih vrednosti i njihovih oscilacija;
- pooling slojevi – služe za redukciju dimenzionalnosti ulaznih podataka, i
- rekurentni slojevi – primenjuju rekurentnu neuronsku mrežu na ulazne podatke.



Slika 1. Dijagram klasa modela

Svi prethodno navedeni činioci predstavljaju operacije koje se izvršavaju na ulaznim podacima, dobijenih na ulazni pin elementa, a rezultat primene date operacije, prosleđuju na izlazni pin elementa. Na slici 1. dat je model činilaca jezika.

2.2. Model

Uz oslonac na prethodni DSL je kreiran i DSL za neuronske mreže. Generičke činioce poput *Element*, *Pin*, *InputVariable*, *OutputVariable* i *Schematic* su izdvojene kao zajedničke i predstavljaju osnovu za svaku drugu verziju DSL-a.

Kako je ideja ovog jezika bila kreiranje modula neuronske mreže za PyTorch biblioteku, i sami elementi su kreirani u skladu sa tim da podrže operatore iz ove biblioteke i njoj sličnih. Svaki *Element* sadrži sledeće podatke:

- name – predstavlja naziv datog elementa, služi za imenovanje varijable u programskom kodu, ukoliko se generiše,
- parent – definiše hijerarhiju uvođenjem roditeljskog elementa,
- elements – predstavlja kolekciju dece datog elementa,
- discriminator – omogućava tipizaciju elemenata,
- custom_data – definiše druge metapodatke koji će dopuniti svojstva elementa,
- input_pins – predstavlja kolekciju svih ulaznih pinova,
- output_pins – predstavlja kolekciju svih izlaznih pinova,
- element_title – predstavlja naslov čvora, služi za vizuelizaciju i najčešće opisuje tip čvora.

Pin predstavlja konektor elementa, i omogućava mu da se kroz njega dostave ili isporuče podaci.

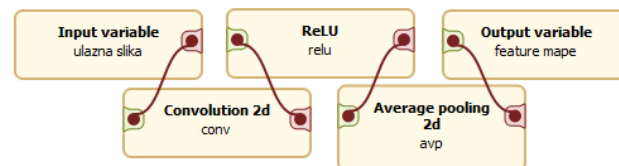
Atributi ove klase su *name*, *parent*, čije je značenje isto kao i kod *Elementa*, i *connected_to* koji predstavlja kolekciju pinova sa kojima je dati pin povezan.

Ove dve klase implementiraju *DictionaryModel* i redefinišu metodu *to_dict* koja treba da omogući serijalizaciju objekata ovih klasa radi perzistencije.

Još jednu specifičnu klasu predstavlja *Schematic* koja predstavlja arhitekturu neuronske mreže, odnosno forward-pass modula. Takođe, ima i ulaznu i izlaznu varijablu u kojoj dostavlja ulazne podatke ostalim elementima šeme, a rezultat primene svih operacija čuva u izlaznoj varijabli. Ostale klase, predstavljaju specijalizacije elementa i specifične operacije u domenu neuronskih mreža. Detaljnije objašnjenje se može pronaći u master radu.

2.3. Implementacija

Konkretna sintaksa jezika kreirana je po uzoru na prethodni jezik za fazi logiku, odnosno fazi logičke kontrolere. Tehnologije upotrebljene za kreiranje te sintakse su *Python* programski jezik, *PySide* biblioteka. Za ovu verziju aplikacije i samog jezika upotrebljena je nova verzija *PySide2* [3] i izvršena je neophodna migracija. Za vizuelizaciju ovog modela napravljena je wrapper klasa *SchematicModel* koja je kreirana kao *editable model* po smernicama Qt-a [4]. Na slici 2. dat je primer vizuelizacije odabranih elemenata jezika.



Slika 2. Primer šeme

Ova slika prikazuje čvorove, odnosno operacije, u obliku zaobljenih pravougaonika ispunjenih krem bojom. Pinovi se nalaze sa strana i na obodu čvora, i u zavisnosti da li je ulazni ili izlazni ima zelenu ili crvenu boju, redom. Veze između tih čvorova formiraju se u ulaznom i izlaznom pinu i predstavljene su kubnom linijom bordo boje.

Podebljan tekst u čvoru predstavlja njegov naslov, a tekst ispod naslova predstavlja naziv čvora. U zavisnosti od pravila jezika, čvorovi mogu imati više od jednog pina. Ovaj jezik je formiran tako da podaci uvek idu od izlaznog ka ulaznom pinu nekog drugog elementa.

Za svaki čvor na šemi moguće je vršiti izmenu boje ispunjenosti elementa, kao i izmenu boje bordure. Naslov elementa, odnosno čvora, je fiksiran, ali je naziv, kao i ostale specifične parametre, moguće promeniti.

4. APLIKACIJA

FLCT je aplikacija koja predstavlja okruženje za generisanje šema i koda na osnovu šeme fazi logičkog kontrolera [1]. Aplikacija je proširiva, uz pomoć plug-in framework-a koji je kreiran sa ciljem da bude minimalistički.

Proširenje aplikacije se trenutno ogleda u dodavanju novih generatora koda, ali proširenje može biti izvršeno i dodavanjem novih editora i drugih komponenti, što je u planu za dalji razvoj aplikacije. Grafički korisnički interfejs (GUI) je kreiran u *Python* programskom jeziku i uz korišćenje *PySide2* biblioteke.

Prikaz aplikacije dat je na slici 3. Generisanje koda na osnovu šeme omogućeno je uz pomoć biblioteke *Jinja2* [5].

Kreiranje šeme vrši se odabirom lokacije i naziva datoteke u kojoj će se šema čuvati. Populisanje novokreirane šeme vrši se primenom *drag&drop* operacija sa palete alata, koja se nalazi u *toolbox-u*, sa desne strane editora šeme. Izmena podataka o elementu vrši se dvoklikom na

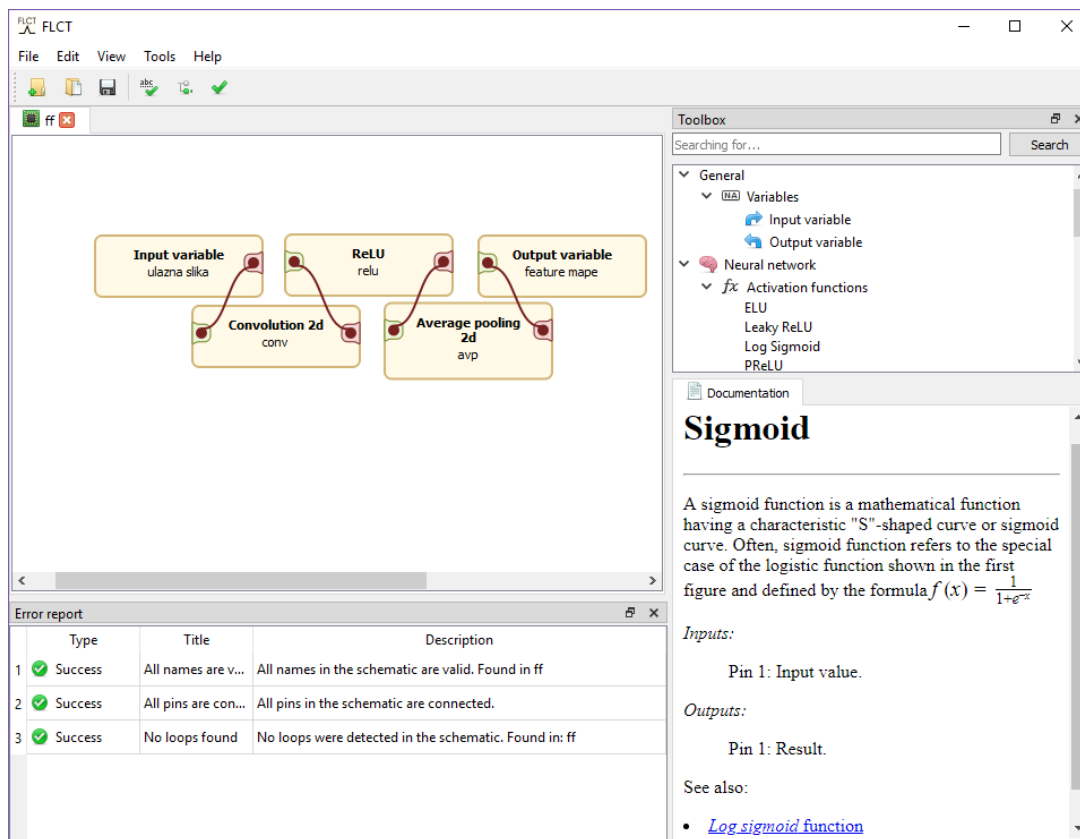
željeni element, nakon čega se prikazuje dijalog sa svim parametrima koje ima dati element, a varira u zavisnosti od konkretnog tipa elementa.

U procesu izmene čvora, moguće mu je prilagoditi i izgled, izmenom boja za ispunu čvora i oboda. Kreiranje se može izvršiti i uz pomoć prethodno definisanih primera, koji predstavljaju poznate arhitekture neuronskih mreža, i korisne su prilikom manjih izmena istih ili kod sagledavanja strukture, za kreiranje nove.

U aplikaciji je omogućena validacija šeme na osnovu zadatih pravila, što u slučaju ovog jezika predstavlja jedinstvenost imenovanja, isključivanje petlji, povezanost svih pinova i postojanje ulaznih i izlaznih varijabli. Validaciju je moguće vršiti samo za čvorove ili samo za veze. Takođe, u opciji je i potpuna validacija koja uključuje prethodno navedene validacije.

Još jednu funkcionalnost predstavlja opciono generisanje koda na osnovu šeme. Preduslov za generisanje predstavlja kompletnu ispravnost šeme na osnovu koje se želi generisati kod. Prilikom generisanja koda, vrši se odabir ciljnog programskog jezika.

U toku generisanja vrši se popunjavanje predefinisanih *Jinja2* šablona i kreira se odgovarajuća struktura direktorijuma i paketa. Generisani kod je spreman za upotrebu u nekom programu, ili za direktno izvršavanje na nekoj mašini uz neke preduslove, poput postojanja kompajlera ili interpretera za ciljni jezik.



Slika 3. Prikaz aplikacije

Čuvanje i učitavanje šeme omogućava njenu ponovnu upotrebu. U planu za dalji razvoj predstavlja učitavanje šeme kao podšeme glavne šeme, čime će biti omogućeno dekomponovanje i mogućnost ponovne upotrebe.

Za svaku operaciju ili tip elementa koji je moguće dodati na šemu postoji i odgovarajuća dokumentacija, koja dodatno objašnjava dati element, primer njegove upotrebe i njegovu strukturu. U ovoj verziji dokumentacija predstavlja veb pretraživač, čime je omogućeno linkovanje na druge relevantne elemente i korišćenje javascript-a, kao npr. za bolji prikaz matematičkih formula.

5. ZAKLJUČAK

Prednost kreiranja šema uz pomoć vizuelizacije jeste bolje razumevanje i jednostavnost dodavanja, jer postoji već prethodno definisana paleta alata.

U pogledu vremena potrebnog za kreiranje koda ili potrebnog za kreiranje šeme, često je rezultat da je manje vremena neophodno za kreiranje šeme, pogotovo kod složenijih šema. FLCT se pokazao kao dobro rešenje za generisanje koda šema fazi logičkih kontrolera i samim tim je postao dobar izbor za dalja proširenja i unapređivanja.

Dodavanje DSL-a za neuronske mreže je pokrilo još jednu oblast koja se često koristi.

Generisanje koda predstavlja samo opcioni deo procesiranja ovog DSL-a. Šema se može upotrebiti i za skiciranje arhitekture mreže, koja može biti deljena drugim korisnicima, koji na osnovu nje mogu izvršiti svoju implementaciju. Validacija šeme predstavlja korisnu opciju koja će otkloniti potrebu za validaciju šeme od strane korisnika, i time napraviti još jednu uštedu vremena u kreiranju šeme.

Takođe, moguće je vršiti upotrebu oba DSL-a u jednoj šemi. To može biti korisno da se npr. neuronska mreža koristi za izbor parametara funkcija pripadnosti u fazi logičkim kontrolerima. Drugi primer bi bio upotreba fazi logike za preprocesiranje i postprocesiranje podataka za obučavanje neuronskih mreža.

Dalja proširenja jezika bi mogla da omoguće i proces obučavanja neuronskih mreža, testiranje neuronskih mreža, kao i dodavanje komponente za prikaz rezultata, poput *TensorBoard*-a [6].

Pored toga, kreiranje i drugih reprezentacija DSL-a bi omogućila veći izbor kreiranja šema u skladu sa korisnikovom naklonjenošću za određene DSL-ove.

6. LITERATURA

- [1] A. Mitrović, "Razvojno okruženje za projektovanje i generisanje koda fazi logičkih kontrolera," Bachelor thesis, University of Novi Sad, Novi Sad, 2017.
- [2] "PyTorch," 2018. [Na mreži]. Dostupno na: <https://pytorch.org/>. [Pristupljeno: 11-Sep-2018].
- [3] "Qt for Python - Qt Wiki," 2018. [Na mreži]. Dostupno na: https://wiki.qt.io/Qt_for_Python. [Pristupljeno: 11-Sep-2018].
- [4] "Model/View Programming | An editable model," 2018. [Na mreži]. Dostupno na: <http://doc.qt.io/qt-5/model-view-programming.html#an-editable-model>. [Pristupljeno: 18-Sep-2018].
- [5] "Welcome to Jinja2 — Jinja2 Documentation (2.10)," 2018. [Na mreži]. Dostupno na: <http://jinja.pocoo.org/docs/2.10/>. [Pristupljeno: 11-Sep-2018].
- [6] "TensorBoard: Graph Visualization," *TensorFlow*, 2018. [Na mreži]. Dostupno na: https://www.tensorflow.org/guide/graph_viz. [Pristupljeno: 19-Sep-2018].

Kratka biografija:



Aleksandra Mitrović rođena je u Loznici 1994. god. Osnovne studije iz oblasti Elektrotehnika i računarstvo – softversko inženjerstvo i informacione tehnologije je završila 2017. godine kada je upisala i master studije iz iste oblasti.

VIZUELIZACIJA OBJEKTNOG MODELA U MEMORIJSKIM BAZAMA PODATAKA**VISUALIZATION OF THE OBJECT MODEL IN MEMORY DATABASES**Borka Samardžić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – *Ovaj rad predstavlja moguće rešenje vizuelizacije objektnog modela podataka u sistemu za upravljanje distribucijom elektroenergetske mreže. Istražena su postojeća rešenja za vizuelizaciju i opisana je implementacija rešenja.*

Abstract – *This paper presents a possible solution for visualization of the object data model in distribution management system of the electrical network. Existing solutions for data visualization were explored and implementation of the solution was described.*

Ključne reči: *Vizuelizacija, Distribution Management System*

1. UVOD

Kompanije se već dugo oslanjaju na tabelarni prikaz podataka, ne samo za njihovu obradu već i za prikaz drugim ljudima. Ali to se menja, zato što kompanije generišu sve veće količine podataka koji su pretežno u *cloud*-u i lista izvora podataka je takođe proširena. Zbog tih trendova tabelarni prikaz i aplikacije za poslovnu inteligenciju nisu dovoljne za obradu podataka, već se koriste alati koji podatke obrađuju i vizualizuju da bi ljudima olakšali shvatanje podataka i otkrili nove informacije.

U slučaju aplikacija za poslovnu inteligenciju, vizuelizacija podataka je pretvaranje numeričkih algoritamskih izlaza u slike. Svrha tih slika je da se informacije sa mašine na ljude prenesu brzo, što znači da je jasnost poruke koja se prenosi važna, ali jasnost ne isključuje složenost poruke. Vizuelizacije su konfigurisane kao samostalne i u grupama kako bi prenele više informacija kroz slike.

Ono čime se ovaj rad bavi je vizuelizacija objektnog modela u memorijskim bazama podataka, odnosno jedan način implementacije vizuelizacije podataka iz keša industrijskog sistema koji upravlja radom elektroenergetske mreže.

2. SISTEM UPRAVLJANJA DISTRIBUCIJOM

Distribution Management System (DMS) je skup aplikacija koje su dizajnirane da efikasno i pouzdano nadgledaju i kontrolišu celu distributivnu mrežu. Aplikacije su sposobne za prikupljanje, organizovanje, prikaz i analizu informacija o elektroenergetskoj mreži u realnom vremenu ili blisko realnom vremenu.

NAPOMENA:

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

DMS deluje kao sistem podrške u odlučivanju za kontrolnu sobu i osoblje na terenu koje nadzire i upravlja elektrodistributivnim sistemom.

Ono što *DMS* omogućava jeste poboljšanje pouzdanosti i kvaliteta usluge, u smislu smanjenja broja ispada, smanjenja trajanja ispada, održavanje nivoa frekvencije i napona sistema.

Sistemi upravljanja distribucijom pristupaju podacima u realnom vremenu iz različitih sistema, i pružaju integrisane informacije na jednoj konzoli u kontrolnom centru. Zbog toga je obim podataka koji se obrađuje veoma velik, i potrebno ih je prikazati tako da je lako izvesti odgovarajuće zaključke.

IEC 61968 je serija standarda koji definišu razmenu informacija između elektrodistributivnih sistema. Standard je namenjen podršci integracije aplikacija među korisnicima koji trebaju da prikupljaju podatke iz različitih aplikacija koje imaju različite interfejsne i okruženja za izvršavanje. *IEC 61968* definiše interfejs za sve glavne elemente za sistem upravljanja distribucijom [1].

Model podataka koji se koristi u *DMS*-u je baziran na *Common Information Model*-u (*CIM*), koji je zvanično usvojila Međunarodna elektrotehnička komisija (*International Electrotechnical Commission*).

CIM je otvoren industrijski standard za definisanje karakteristika uređaja i aplikacija tako da sistemski administratori i programi za upravljanje mogu na isti način da kontrolišu uređaje i aplikacije razvijene od strane različitih proizvođača. To znači da su definisane klase koje se koriste u sistemu upravljanja distribucijom, njihovi atributi, hijerarhija nasleđivanja i međusobne veze.

Podaci kojima se pristupa se nalaze u kešu sistema, i da bi se lakše uočile veze između njih potrebno je vizualizovati objektni model podataka kao i njihove međusobne veze.

3. POSTOJEĆA REŠENJA ZA VIZUELIZACIJU

Na tržištu postoji velik broj alata koji služe za vizuelizaciju objektnog modela i eventualno za neko izvođenje zaključaka o podacima iz modela. Svaki od tih alata ima konektore za različite izvore podataka, neku vrstu inteligencije da analizira podatke i skup nekoliko osnovnih vizuelizacija koje se mogu izabrati.

Primeri nekih najpopularnijih alata za vizuelizaciju su: *Microsoft Power BI*, *Google Analytics*, *IBM Watson Analytics*, *Tableau Desktop*, *Salesforce Einstein Analytic Platform*.

3.1. Microsoft Power BI

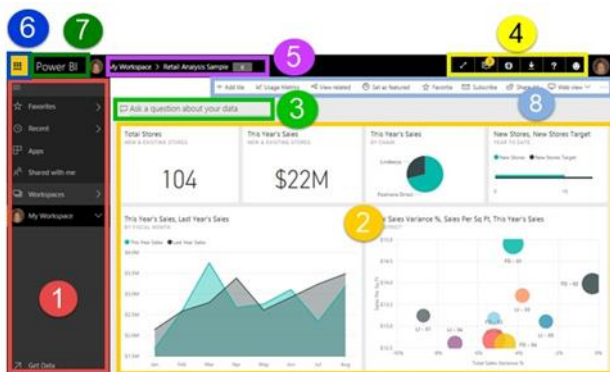
Power BI je skup softverskih usluga, aplikacija i konektora koji rade zajedno kako bi svoje nepovezane izvore podataka pretvorili u koherentne, vizuelno impresivne i interaktivne uvide. Može biti jednostavan i brz, sposoban da kreira brza shvatanja iz lokalne baze podataka ili Excel tabele. Takođe je i robustan i korporativan, spreman da ekstenzivno modeliranje i analitiku u realnom vremenu, kao i za prilagođen razvoj. To znači da *Power BI* može biti lični izveštaj i alat za vizuelizaciju, a takođe može poslužiti kao analitički i odlučujući mehanizam iza grupnih projekata, odeljenja ili celih korporacija. [2]

Microsoft Power BI trenutno ima nekoliko desetina konektora za podatke, koji omogućavaju korisniku da usmeri svoj *Business Intelligence* alat u određenu aplikaciju ili skup podataka i uključi te podatke u upite. Pomoću lokalnog gateway-a je moguće upućivati upite iz *Power BI* u lokalne izvore podataka. Kada se intereaguje sa vizuelizacijom, upiti iz *Power BI* se direktno šalju u bazu podataka. Ažurirani podaci se zatim vraćaju i vizuelizacije se ažuriraju.

Što se tiče same vizuelizacije, u *Power BI* postoji standardan set formata vizuelizacije koji korisnik može izabrati u zavisnosti od toga šta treba da prikaže, a postoje i prilagođene vizuelizacije u *Office Store*-u. Glavne karakteristike korisničkog interfejsa *Power BI* su [2]:

- navigaciono okno;
- platno (kontrolna tabla);
- blok za pitanja;
- ikone (uključujući pomoć i povratne informacije);
- naslov kontrolne table;
- pokretač aplikacije *Office 365*;
- dugme za *Power BI* početnu stranicu;
- označene ikone,

što je prikazano na slici 1.



Slika 1: *Microsoft Power BI* korisnički interfejs [2]

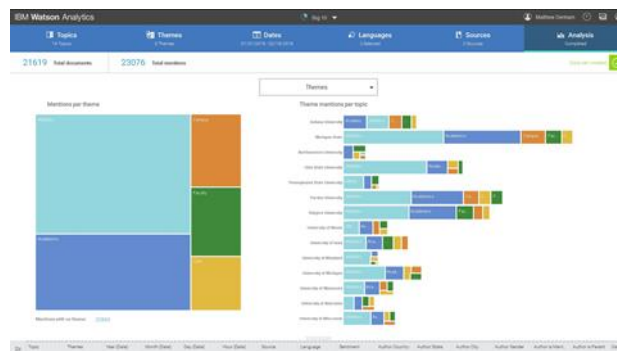
3.2. IBM Watson Analytics

Watson Analytics je softver za poslovnu analizu, koji omogućava pametno otkrivanje podataka, automatsku prediktivnu analitiku i kognitivne sposobnosti pomoću kojih je moguća interakcija sa podacima. [3]

Softver omogućava korisnicima da istražuju i postavljaju pitanja o svojim skupovima podataka i efikasno prenose rezultate unutar kompanije ili tima kroz vizuelizaciju podataka. Korisnik može da interpretira podatke, razume zašto se pojavljuju obrasci i trendovi i razvijaju ključne

obuke koje objašnjavaju podatke i doprinose prediktivnim funkcijama analitike. *Watson Analytics* omogućava kompanijama da komuniciraju sa svojim podacima na inovativan način, povećavajući efikasnost i preciznost kako bi uspeali.

Korisnički interfejs *IBM Watson Analytics* je prikazan na slici 2.

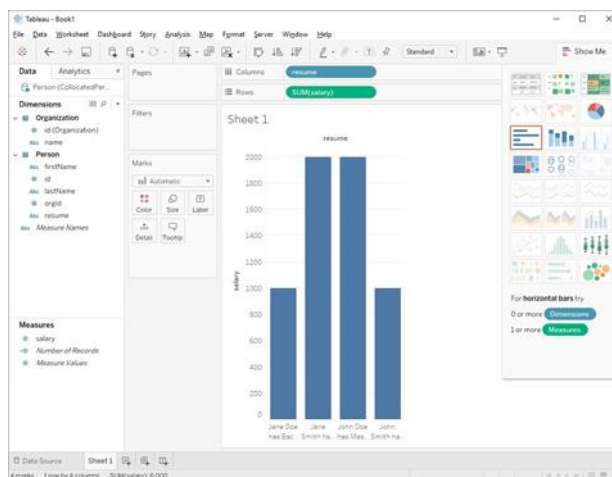


Slika 2: *IBM Watson Analytics* korisnički interfejs [3]

3.3 Tableau Desktop

Tableau je sistem poslovne inteligencije koji pomaže kompanijama da vizuelizuju i razumeju svoje podatke. *Tableau Desktop* je rešenje samouslužnog analitičara koje omogućava brzo pregledanje podataka i odgovore na pitanja. [4] Namenjen je za lako kreiranje i distribuciju interaktivnih kontrolnih tabli podataka koji pružaju upečatljiv prikaz dinamike, trendova promena i raspodele gustine podataka preko jednostavnih i efikasnih vizuelnih prikaza.

Tableau omogućava povezivanje sa većim brojem izvora podataka, uvoz i vizuelizaciju podataka za nekoliko minuta. Sam korisnički interfejs je intuitivan i jednostavan čak i za korisnike koji nisu obučeni za njegovo korišćenje. Ono što *Tableau* izdvaja od ostalih ovakvih rešenja je mogućnost da se podaci analiziraju na različite načine, a na kraju se svi zaključci prikazuju sveobuhvatnom vizuelizacijom podataka, koja može da se podeli sa ostalim korisnicima. Takođe nudi uputstva o tome kako najbolje iskoristiti svaku vizuelizaciju, što je korisno kada korisnik dobije neočekivan vizuelni izgled. Na slici 3 je prikazan izgled *Tableau* korisničkog interfejsa.



Slika 3: *Tableau Desktop* korisnički interfejs [4]

4. PREDLOG REŠENJA VIZUELIZACIJE

Sistem upravljanja distribucijom koristi podatke koje nakon startovanja učitava sa diska u RAM memoriju. Podaci sa kojima se radi se nalaze u *SQLite* bazi podataka. Nad podacima trebaju da se obavljaju operacije replikacije, perzistencije, vršenje upita, čuvanje za istoriju, transakcije, prikaz, itd.

Zbog toga mora postojati objektni model nad kojim će se raditi. Model je zasnovan na enumeraciji (u daljem tekstu kod) koja opisuje hijerarhiju nasleđivanja i entitete koji postoje u sistemu. Tako je moguće jednoznačno identifikovati svaki tip resursa u modelu – svaka klasa i svaki atribut svake klase dobijaju svoju vrednost enumeracije.

Da bi se jedinstveno identifikovao svaki entitet u sistemu, koristi se globalni identifikator (GID) koji je tipa *long*, i kreira se na osnovu tipa entiteta i brojača entiteta.

Reference između entiteta se modeluju globalnim identifikatorom. To znači da entitet koji referencira neki drugi entitet sadrži globalni identifikator tog entiteta.

Postojeća rešenja za vizuelizaciju podataka nisu razvijena za rad sa velikom količinom kritičnih podataka koji se menjaju u realnom vremenu i koji prikazuju sve veze između objekata.

Zbog toga je implementirano rešenje koje će biti predstavljeno.

4.1 Opis rešenja

U sistemu upravljanja distribucijom, koji je korišćen za potrebe ovog rada, postoji aplikacija koja prikazuje trenutne podatke iz sistema, ali u tabelarnom obliku. Aplikacija se konektuje na servise koji rade u sistemu, i za izabrani tip servisa i tip entiteta prikazuje sve vrednosti entiteta, odnosno njihove globalne identifikatore.

Mana sa ovakvim tabelarnim prikazom je to što se ne uočavaju jasne veze između postojećih objekata, jer su u tabeli upisani globalni identifikatori koji predstavljaju reference.

Ako korisnik želi da vidi koje atribute imaju ti elementi koji su reference, treba da ih pronade u tabeli od tipa reference, što nije praktično.

Ulazi u vizuelizaciju su:

- globalni identifikator elementa čiji atributi se prikazuju;
- TCP adresa;
- ime procesa;
- kodovi tipova elemenata koji mogu da postoje na izabranom servisu.

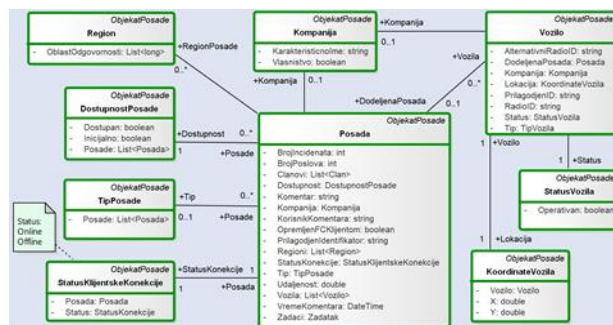
Izlaz iz vizuelizacije je dijagram na kome su prikazani početni element i objekti na koje on pokazuje.

Koraci procesa vizuelizacije su:

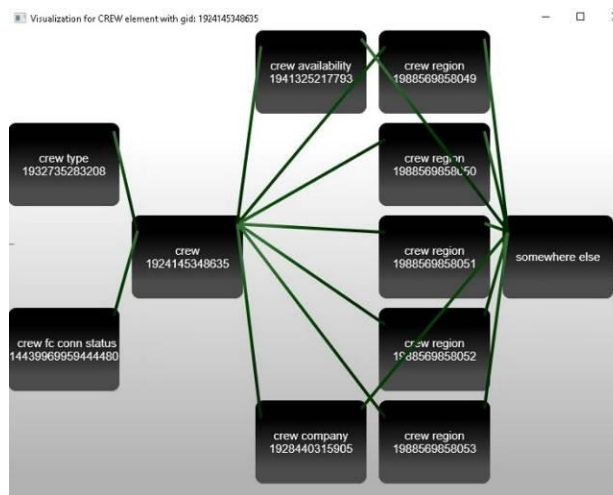
- pravljenje modela;
- vizuelna reprezentacija;
- prikaz detalja.

Kao primer za predstavljanje rešenja je korišćen keš servisa koji čuva podatke o posadi, koji su bazirani na CIM standardu. Na slici 4 prikazan je UML dijagram klasa sa potrebnim podacima o modelu posade. Na slici 5 prikazan je izgled vizuelne reprezentacije objektnog modela posade iz keša sistema za upravljanje

distribucijom, sa svim entitetima koji postoje na servisu i njihovim međusobnim vezama.



Slika 4: UML dijagram modela posade



Slika 5: Vizuelizacija objektnog modela iz keša Distribution Management System-a

Na slici 6 predstavljen je izgled prozora za prikaz detalja o elementu sa slike 4, za koji se radila vizuelizacija njegovih veza.

Property code	Property value
crew customid	CR
crew comment	
crew assignments	null
crew crewcompany	1928440315905
crew crewtype	1932735283208
crew crewregions	1988569858049 1988569858050 1988569858051
crew crewavailability	1941325217793
crew crewvehicles	null
crew crewmembers	null
crew equipped with fieldcl	False
crew comment user	
crew comment time	0001-01-01 00:00:00.000
crew jobs count	False
crew distance	0
crew work plans count	False
crew incidents count	False
crew opt disp index	False
crew has required skills	False
crew crewconnstatus	144399699594444802
crew shift ref	0
crew dars show count	False
crew excluded from shift	False
crew working period	13630
crew resting period	0
crew work limit column	False
crewobj gid	1924145348635
crewobj timestamp	2018-10-01 13:39:27.999
crewobj name	Control room
crewobj updated properties	-4463067230694795514 -4467570830351531512

Slika 6: Prozor za prikaz atributa objekta iz keša Distribution Management System-a

4.2 Rezultati merenja performansi

Za implementirano rešenje vizuelizacije objekata je mereno koliko je potrebno vremena za iscrtavanje objekata elemenata i njihovih međusobnih veza. U tabeli 1 su prikazani dobijeni rezultati.

Tabela 1: Rezultati merenja vremena iscrtavanja modela

broj elemenata	broj veza	vreme
2	1	~1,21775 ms
10	12	~17,40655 ms
50	99	~223,46045 ms
150	393	~622,79951 ms
1000	2620	~2742,68392 ms

Na osnovu dobijenih rezultata se može zaključiti da sa porastom broja elemenata koji se vizuealizuju vreme potrebno za njihovo iscrtavanje raste kvadratno.

5. ZAKLJUČAK

Vizuelizacija podataka pomaže tako što organizuje podatke u forme koje su lakše za razumevanje, označava trendove i odstupanja u ponašanju. U sistemima sa velikim količinama kritičnih podataka koji stalno pristižu, alati i tehnike za vizuelizaciju podataka su od suštinskog značaja za analizu velikih količina informacija i donošenje odluka na osnovu podataka.

U sistemu upravljanja distribucijom ovakav način prikaza podataka, u odnosu na tabelarni prikaz, znatno povećava preglednost entiteta u sistemu i lakše je uočavanje nekonzistentnosti unutar memorije sistema.

6. LITERATURA

- [1] International Electrotechnical Commission, IEC 61968–6:2015, Application integration at electric utilities – System interfaces for distribution management – Part 6: Interfaces for maintenance and construction, July 2015
- [2] Microsoft, Power BI – basic concepts for Power BI service, July 2018, <https://docs.microsoft.com/en-us/power-bi/consumer/end-user-basic-concepts>
- [3] IBM, What is Watson Analytics?, 2018, <https://www.ibm.com/watson-analytics>
- [4] Tableau, Tableau Desktop, 2018, <https://www.tableau.com>

Kratka biografija

Borka Samardžić rođena je 1994. godine u Novom Sadu. Završila je gimnaziju „Svetožar Marković“ u Novom Sadu 2013. godine. Osnovne akademske studije na smeru Elektroenergetski softverski inženjering na Fakultetu tehničkih nauka u Novom Sad je upisala 2013. godine i završila 2017. godine. Iste godine je upisala master studije na smeru Primenjeno softversko inženjerstvo na Fakultetu tehničkih nauka. Ispunila je sve obaveze i položila je sve ispite predviđene studijskim programom.

MODELOVANJE DISTRIBUTIVNE ELEKTROENERGETSKE MREŽE UPOTREBOM NEO4J BAZE PODATAKA**MODELING POWER DISTRIBUTION NETWORK USING NEO4J DATABASE***Milica Plavšić, Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu opisane su graf baze podataka i aplikaciono rešenje koje ima za cilj analizu i pretragu distribuirane elektroenergetske mreže upotrebom Neo4j baze podataka.

Ključne reči: Neo4j, Cypher, elektroenergetska mreža

Abstract – This work is about graph databases and application solution and its aim was the analyses and search of power distribution network by using Neo4J database.

Key words: Neo4j, Cypher, power distribution network.

1. UVOD

Za ovaj rad su najznačajniji softverski modeli podataka koji su neophodni da bi se proces modelovanja elektroenergetskih sistema obavio uspešno. Naime, elektroenergetski sistem je jedan od najvećih i najuticajnijih upravo zbog toga što se bez njega život ne može zamisliti. Cilj ovog rada je analiza i pretraživanje upravo tih modela, modela elektroenergetske mreže, pri čemu je čuvanje podataka realizovano pomoću Neo4J baze podataka. Osim uvoda i zaključka rad sadrži još 5 sekcija. U sekciji 2 opisane su relacije i NoSQL baze podataka, a u sekciji 3 su graf orijentisane baze podataka. U sekciji 4 opisana je Neo4j baza podataka i Cypher upitni jezik. Sekcija 5 opisuje CIM model podataka, dok je u sekciji 6 prikazano aplikativno rešenje i prikaz rezultata istog. U poslednjoj sekciji izvedeni su najvažniji zaključci.

2. BAZA PODATAKA

Baza podataka [2] kao deljiva kolekcija međusobno organizovanih i uređenih podataka, danas je najčešće korišćena metoda za čuvanje istih.

Šema baze podataka je apstrakcija na koju se naslanja konkretna baza podataka. Model podataka je apstrakcija koja se koristi za definisanje šeme baze podataka.

2.1. Relacioni model podataka

Relacioni model podataka [2] se zasniva na matematičkom pojmu relacija. Relacija, predstavlja skup n -torki. Relacija je pojava nad šemom relacije. Šema relacija predstavlja apstrakciju odnosno okvir za relaciju. Šema relacije definiše atribut i ograničenja nad njima, koja će zatim relacija (konkretni podaci) poštovati. Relacija predstavlja osnovni tip strukturu gde se podaci čuvaju. Njena tipična reprezentacija jeste tabelarni oblik, gde je svaka vrsta u tabeli jedna torka.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Čapko vanr.prof.

Pomoću torke se svakom obeležju iz nekog skupa obeležja, dodeljuje konkretna vrednost. Svaki podatak je zapisan samo jednom, odnosno u slučaju pravilnog projektovanja, nema redundantnih podataka. Upitni jezik je deklarativan i napredniji od upitnih jezika baza koje nisu relacione. Sistemi za upravljanje bazama podataka su značajno napredniji softveri nego u slučaju nerelacionih baza. Omogućuju bolju validaciju šeme, odnosno forsiranja ograničenja nad podacima. Najnapredniji transakcioni režim nude sistemi za upravljanje relacionim bazama. Velika količina informacija dostupna je na internetu o relacionim bazama.

Skupo je uspostavljanje i održavanje ovakvih baza. Pri projektovanju mora se odrediti količina podataka koja može stati u polje da ne dođe do gubitka podataka. U relacionim bazama, da bi se dobavili podaci iz više tabela, potrebno je vršiti spajanje datih tabela (JOIN), što se može vrlo negativno odraziti na performance ovakvih sistema. Nije pogodna za čuvanje nestruktuiranih podataka.

2.2. NoSQL baze podataka

U odnosu na relacione baze podataka, koje podatke predstavljaju putem tabličnih relacija, kod NoSQL baza podataka [1] podaci se smeštaju u različite objekte u zavisnosti od njihove vrste.

NoSQL baze omogućavaju dodavanje podataka bez prethodnog saznanja baze podataka o strukturi podataka koje skladišti. Takođe, NoSQL baze omogućavaju lakši rad na više računara, čime se postiže distribuiranost NoSQL podataka.

Vrste NoSQL baza podataka

Key-value orijentisane: podaci su organizovani u asocijativne nizove ili mape koje koriste jednostavnu metodu ključa i vrednosti za skladištenje podataka.

Dokumentima orijentisane: podaci se čuvaju u dokumentima i smeštaju se u kolekcije. Dokumenti su nezavisni i svaki od njih ima svoj jedinstveni ključ koji služi za dobavljanje podataka.

Column-family orijentisane: ovaj tip baze predstavlja tabelarnu strukturu u kojoj redovi predstavljaju entitete a različiti redovi mogu sadržati različite kolone i one se mogu dodati nezavisno bilo kom redu u bilo kod trenutku. I graf orijentisane o kojima će biti reči u narednoj sekciji.

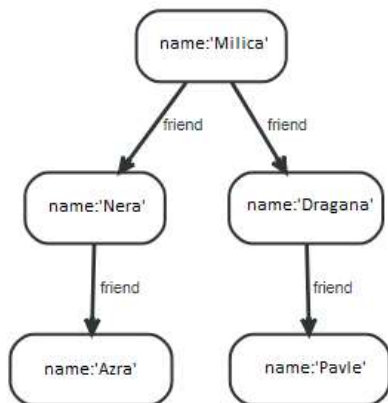
3. GRAF ORIJENTISANE BAZE PODATAKA

Graf baze [1] su kategorija NoSQL baza podataka čija je upotreba sve veća. Graf baze je najadekvatnije koristiti za reprezentaciju i pretraživanje povezanih podataka

značajnijih veličina. Danas se ova vrsta baza veoma često upotrebljava upravo zbog toga što je, zbog prirode podataka, graf najbolji oblik prikaza istih.

3.1. Graf kao model podataka

Graf baze podataka koriste strukturu podataka tipa graf [3] za predstavljanje i skladištenje podatka. Čvor u grafu predstavlja određeni podatak (pojavu entiteta) koji predstavlja objekte u stvarnom svetu. Svaki čvor ima svoj identifikator i svojstva u obliku parova ključ-vrednost, dok više čvorova može imati istu oznaku. Svojstva opisuju čvorove. Veze predstavljaju relaciju odnosno povezanost između dva čvora. Pored toga, sadrže i identifikator, smer veze i početni kao i odredišni čvor.



Slika 1: Primer grafa sa 5 čvorova i 4 veze

Predstavljanje grafa

Uobičajena su dva načina predstavljanja grafa [3]:

Matrica povezanosti - Za graf $G=(V,E)$, neka je $V=\{v_1, v_2, \dots, v_n\}$, matrica povezanosti grafa je kvadratna matrica $A=[a_{ij}]$ reda n , sa elementima $a_{ij}=1$ ako (v_i, v_j) pripada E , a ostali elementi matrice A su nule.

Lista povezanosti - Predstavljanje grafa $G=(V,E)$ pomoću liste povezanosti sastoji se od vektora lista, po jednu za svaki čvor iz skupa V . Za svaki čvor u koji pripada V , lista povezanosti $A_{adj}[u]$ sadrži sve čvorove v takve da postoji veza (u,v) koja pripada E .

3.2. Modeliranje graf baza podataka

Da bi model grafa bio ispravan i tačan, potrebno je da bude čitljiv i smislen, da baza omogućava lako dodavanje veza i čvorova kao i brisanje postojećih bez narušavanja strukture grafa. Skladište mora biti optimizovano za smeštanje podataka u obliku grafa, a i potrebno je da se mogu izvoditi upiti koji su od značaja i koji daju ispravne i željene rezultate.

3.3. Razlika između graf baze i relacione baze podataka

Kod relacionih baza svaka tabela je kolekcija fiksnih atributa, dok je kod graf baza promena znatno jednostavnija. Vreme izvršavanja postavljenog upita u graf bazi je konstantno, dok u relacionim bazama ono zavisi od broja zapisa u tabeli [4]. Graf baze su

optimizovane za veze između podataka dok su relacione primenljivije za agregaciju podataka. Graf baze su adekvatnije za složene strukture podataka, dok su relacione za relativno jednostavne strukture.

3.4. Prednosti graf baze podataka

Brže se može izvesti upit i to nezavisno od količine podataka. Mogućnost reprezentacije povezanih podataka na prirodan način kao skup objekata (čvorova) povezanih skupom objekata (veza). Model domene može se izraditi u vrlo kratkom roku, pa programer može odmah početi s izradom aplikacije, što ubrzava a i olakšava postupak. Postoji izrazita fleksibilnost na moguće promene tokom razvoja ili nakon isporuke aplikacije.

3.5. Nedostaci graf baze podataka

Niska stopa zrelosti i podrške, usled toga što je ova kategorija relativno nova. Nedostatak podrške za upravljanje većim brojem korisnika zbog čega upravljanje korisnicima mora biti implementirano izravno u aplikaciji. Veći su troškovi implementacije u praksi jer, da bi se sprečila smanjenja u performansama u slučaju više korisnika, graf baze podataka moraju biti implementirane i skladištene na jednom serveru.

4. Neo4j

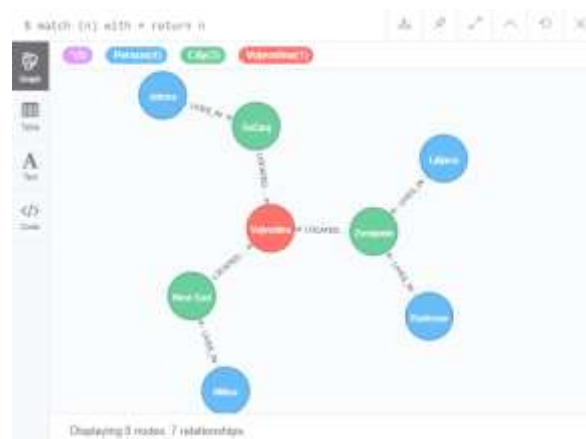
Neo4j [7] je jedna od najpopularnijih baza podataka u okviru graf baza podataka i veoma je jednostavna i za korišćenje a i za implementaciju. Najjednostavniji način korišćenja Neo4j baze podataka jeste preko Neo4j pretraživača za izvršavanje upita koji su napisani u Cypher upitnom jeziku.

4.1. Cypher upitni jezik

Cypher [1] je jednostavan deklarativni upitni jezik za graf baze, koji se ujedno i najčešće koristi. Po strukturi je veoma sličan SQL upitnom jeziku.

Mogućnosti Cypher upitnog jezika

Najznačajnije naredbe za upravljanje podacima u Cypher-u su naredbe za projekciju rezultata, naredbe za čitanje, pod-naredbe za čitanje, naredbe za pisanje, naredbe za čitanje/pisanje.



Slika 2: Primer grafa u Neo4j bazi podataka

5. CIM

CIM [5] je apstraktni model kojim se mogu predstaviti najvažniji entiteti elektroenergetske mreže. Svi objekti iz realnog sistema poput generatora, transformatora, prekidača, vodova i slično predstavljeni su klasama u CIM modelu. CIM predstavlja skup paketa definisanih korišćenjem objektno orijentisanih tehnika modelovanja.

5.1. Osnovne klase CIM modela

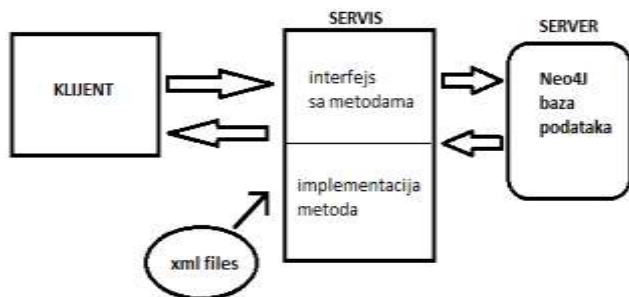
Većina CIM klasa zasniva na *IdentifiedObject* klasi te se ista može uzeti kao osnova i proglasiti glavnom u hijerarhiji. Pored *IdentifiedObject* klase, u okviru elektroenergetskog sistema izdvajaju se i *PowerSystemResource*, *Equipment*, *EquipmentContainer*, *ConductingEquipment*, *Company* kao i *PSRType*.

6. MODELOVANJE ELEKTROENERGETSKE MREŽE

Bazične komponente CIM modela su transformatorske stanice i fideri odnosno izvodi iz istih. U ovom radu unutrašnjost transformatorskih stanica i fidera biće modelovana u graf bazu podataka. Svrha rada je brzo pretraživanje graf baze podataka za određene upite koji su od značaja. Deo sistema za prenos i distribuciju električne energije predstavlja transformatorsku stanicu. Fideri polaze iz transformatorske stanice. Fideri opisuju opremu i elektroenergetske veze na koje se priključuju potrošači. Fideri sa svojim grananjem čine distributivnu mrežu elektroenergetskog sistema. U obrađivanom modelu postoji 88 transformatorskih stanica i 243 fidera (izvoda) koji su sa svojim karakteristikama izdvojeni u xml fajl [6].

6.1. Realizacija rešenja

Aplikativno rešenje rađeno je u integrisanom razvojnom Visual Studio okruženju, u C# programskom jeziku. Korišćena je Microsoft tehnologija koja se naziva WCF(Windows Communication Foundation)[4]. Osnovni delovi WCF modela u ovom aplikativnom rešenju su klijent, servis i server. Klijent preko servisa šalje zahtev za dobijanje određenih podataka i da onda servis komunicira sa bazom koja dobavi željene informacije servisu, koji nakon toga odgovara klijentu (slika 3.).



Slika 3: Šema wcf komunikacije

Potrebno je obezbediti komunikaciju između klijenta i servisa, takođe između servisa i servera. Adresa pristupne tačke na kojoj servis sluša dolazne upite predstavlja se u standardnom url formatu.

Klijent preko krajnje tačke(endpointa) pristupa servisu kako bi prihvatio podatke od servisa. Nakon toga kreira se

proxy koji preslikava potpise endpoint operacija kako bi na klijentskoj strani mogao da se pozove taj endpoint, odnosno omogućava klijentu da poziva servis operacije, a da sakrije detalje implementacije od istog. Ugovor (contract) definiše funkcionalnosti koje pristupna tačka pruža i format poruka koje implementirane metode očekuju.

Ovo rešenje podeljeno je u 3 projekta u visual studio-u, a svaki će biti objašnjen u sledećim pod-sekcijama.

WCFServis

Servis će se povezati sa Neo4j bazom podataka koja predstavlja server u ovoj aplikaciji. Povezivanje se vrši preko binarnog protokola. GraphService projektom je predstavljena klasa wcf servisa i naredne klase:

ConnectToNeo4j: u ovoj klasi se vrši konekcija na bazu, a pored konekcije, u ovoj klasi implementirane su i metode koje će se slati graf bazi, za izvršavanje upita u Cypher-u. *LoadXmlToDictionary*: najvažnija klasa jeste upravo ova, u kojoj se vrši učitavanje xml-ova u određenu kolekciju, a zatim i kreiranje upita za pravljenje grafa u Neo4j bazi podataka na osnovu učitanih podataka. Na osnovu izgleda i strukture xml-a, zaključuje se da su čvorovi u grafu svi elementi koji opisuju jednu distributivnu mrežu (transformatori, prekidači, sekcije, bay-ovi, signali, pristupne tačke i slično). Suština jeste da se kroz učitani xml fajl prođe samo jednom i tom prilikom se napravi string koji sadrži naredbe za kreiranje čvorova u Cypher upitnom jeziku, kao i da se napuni kolekcija koja će za svaki čvor čuvati gid-ove čvorova koji su povezani na njega.

WCFServis: u klasi *WCFServis* kreira se objekat klase *ConnectToNeo4j* čime se vrši konekcija na bazu, a potom se preko reference objekta pozivaju metode koje su implementirane za slanje upita u Neo4j bazu. Sam servis se implementira kao klasa koja nasleđuje interfejs servisa, i ugovor koji se iz interfejsa definiše. Takođe se definiše pristupna tačka sa svim informacijama potrebnim klijentu da bi bio u mogućnosti da komunicira sa servisom.

GraphInterfaces

Ova klasa predstavlja ugovor i ovde su prikazane deklaracije metoda, koje su dostupne korisniku, a implementirane su u *WCFServis* klasi

GraphClient

Kao što je objašnjeno u prethodnom delu rada, klijent preko endpointa pristupa servisu, što se izvrši u podešavanjima. Proxy omogućava da pozovemo metode koje su dostupne na servisu. Stoga, najpre se mora otvoriti kanal za komunikaciju sa servisom. Takođe se na klijentu jednostavno biraju opcije, odnosno izborom na određeni broj, određen zahtev se šalje dalje servisu, koji prikupi tražene podatke od Neo4j baze i vrati odgovor klijentu, koji dalje biva prikazan na konzoli

6.2. Analiza rezultata

Prilikom pokretanja aplikacije, potrebno je pokrenuti servis koji će se nakon toga konektovati na bazu i klijenta koji će slati zahteve servisu. Klijent može da izabere jednu od sedam ponuđenih metoda, dok osma predstavlja izlazak iz program. Metode su poredane po rednim brojevima i pored svakog je opisana metoda koja

predstavlja taj određeni broj. U zavisnosti da li klijent želi da upiše neke podatke u bazu, ili da dobavi podatke iz iste, ili pak da izbriše neki određeni čvor, bira određeni broj. Izborom na određeni broj servis šalje određeni cypher upit ka bazi podataka i javlja se poruka na servisu da je zahtev poslat serveru. Nakon dobijenog odgovora od servera, servis ga prosleđuje klijentu i na klijentu se prikazuju traženi rezultati na konzoli.

Metode koje su dostupne klijentu su: upis svih transformatorskih stanica i njenih fidera u bazu podataka, pronalaženje čvora sa zadatim gid-om, pronalaženje svih čvorova koji su istog tipa, pronalaženje svih čvorova koji su istog tipa koji pripadaju istom kontejneru, gde se kao parametar prosleđuje ime labela kontejnera, pronalaženje svih čvorova koji su istog tipa koji pripadaju istom kontejneru, gde se kao parametar prosleđuje gid kontejnera, dodavanje novog čvora u bazu i brisanje postojećeg čvora u bazi.

Ako klijent pozove metodu koja kao povratnu vrednost vraća sve attribute o određenom čvoru iz baze podataka nakon pritiska na broj dva, od klijenta se traži da unese gid čvora za koji želi da mu se dobave podaci. Nakon upisanog gid-a rezultati se prikazuju na konzoli klijenta, što se može videti sa slike 4.

```
2
enter gid
0x0227007100000001
node ID : 22318
node type : BUSBAR
node container : S77
properties:
name : BUS_155093197498875905
busbar_type : MainBusbar
GID : 0x0227007100000001
Time in milliseconds 74.5925
```

Slika 4: Prikaz rezultata – test1

Takođe se može videti i vreme odziva koje u ovom slučaju iznosi 74.5925ms. Ista metoda pozvana je 20 puta za drugačije zadate gid-ove, a prosečno vreme iznosi 102.48ms.

Prosečno vreme traženja čvora tipa BREAKER pomoću zadate transformatorske stanice pomoću labela, metodom broj četiri je 96.44ms.

Na slici 5. prikazan je rezultat metode koja kao povratnu vrednost vraća sve čvorove koji su određenog tipa, a deo su određenog kontejnera, odnosno određene transformatorske stanice ili fidera.

```
5
write type, press enter and write GID of container
LOADBREAKSWITCH
0x0201004400000410
Total elements 5
ID:15956, CON:F11, name:loadbreaker_352187350925,
D:0x0000005200007f8d
ID:15958, CON:F11, name:loadbreaker_352187350929,
D:0x0000005200007f91
ID:15963, CON:F11, name:loadbreaker_352187350933,
D:0x0000005200007f95
ID:15965, CON:F11, name:loadbreaker_352187350937,
D:0x0000005200007f99
ID:15974, CON:F11, name:loadbreaker_352187350941,
D:0x0000005200007f9d
Time in milliseconds 159.248
```

Slika 5: Prikaz rezultata – test 2

Klijent preko konzole unosi tip čvora, u ovom slučaju je to je LOADBREAKSWITCH, a potom i gid kontejnera, odnosno fidera sa gid-om: 0x0201004400000410. Na slici 5. se takođe može videti i tačan broj određenih elemenata u zadatom kontejneru, u ovom slučaju ih je pet. Isto tako se vidi i broj fidera kome pripada čvor, u ovom slučaju je to F11 (fider broj 11), a vreme odziva iznosi 159.248 ms. Na isti način beležena su vremena odziva za pronalazak čvorova tipa LOADBREAKSWITCH koji pripadaju različitim kontejnerima čiji je gid zadat. Prosečno vreme pretrage iznosi 174 ms.

Vreme odziva metode šest, gde klijent unosi podatke za kreiranje novog čvora u bazu je 158,6737ms. Nakon unošenja novog čvora u bazu, pomoću metode dva koja vraća određeni tip čvora sa zadatim gid-om pronalazi se upravo napravljen čvor za 186.3621ms.

Vreme odziva metode sedam, koja briše čvor sa zadatim gid-om čvora iznosi 249.6253 ms.

7. ZAKLJUČAK

Zadatak ovog rada je bio da se, na osnovu elektroenergetske mreže odnosno transformatorskih stanica i fidera, napravi graf u Neo4J bazi podataka kao i da se izvrše upiti nad tom bazom. Podaci su bili zasnovani na CIM standardu a važno je napomenuti da su bili dati u xml fajlovima. Razvijena je aplikacija koja vrši obradu modela podataka distributivne mreže i kreira upite koji se šalju Neo4J bazi. Na taj način se formira model kome se pristupa od strane razvijenog klijenta u skladu sa specifikacijama za pristup CIM podacima. Dati projekat bi se mogao proširiti u smislu daljih aktivnosti koje bi doprinele da pretraga dostigne veću brzinu. Jedno od rešenja bi moglo biti to da se unapredi algoritam indeksiranjem čvorova u Neo4J bazi jer bi se time ubrzala pretraga grafa a samim tim i pretraga podataka.

LITERATURA

- [1] Ian Robinson, Jim Webber, Emil Eifrem: Graph Databases Second Edition, O'Reilly Media, June 2015.
- [2] Pavle Mogin, Ivan Luković: Principi baza podataka, Fakultet tehničkih nauka, Novi Sad, 1996.
- [3] Milo Tomašević: Algoritmi i strukture podataka, Akademski misao, 2008.
- [4] John Sharp: Windows Communication Foundation 4 Step by Step, Microsoft, 2014.
- [5] J.Simmins: Common Information Model Primer, EPRI-Final Report, Palo Alto, California, 2015.
- [6] Elliotte Rusty Harold: XML™ Bible, IDG Books Worldwide, 1999.
- [7] Adam Fowler: NoSQL For Dummies, John Wiley & Sons, New Jersey, 2015.

Kratka biografija:



Milica Plavšić rođena je u Zrenjaninu 1993. god. Fakultet tehničkih nauka upisala je 2012. god. Diplomski rad iz oblasti Elektrotehnike i računarstva – automatika i upravljanje sistemima, odbranila je 2016. god.

ARTS test okruženje za M2B Wi-Fi HaLow™

ARTS test environment for M2B Wi-Fi HaLow™

Robert Šafer, Željen Trpovski, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – ARTS predstavlja automatizovano test okruženje za testiranje M2B Wi-Fi HaLow™. U ovom radu je dat opis okruženja, sastavni elementi i način implementacije.

Ključne reči: Automatizovano testiranje, Wi-Fi HaLow™, test okruženje

Abstract –ARTS environment presents automated test environment for testing M2B Wi-Fi HaLow™ . This paper describes environment, its components and the way environment is implemented.

Keywords: Automated testing, Wi-Fi HaLow™, test environment

1. UVOD

Ideja za projekat nastala je iz potrebe za kompletno automatizovanim test okruženjem. Zbog brzog razvoja projekta javila se potreba za kontrolom prethodno napisanih delova koda za mnoge komponente sistema, da se osigura ispravnost i valjanost postojećeg dela projekta kao i da se osposobi automatizovano testiranje performansi i stabilnosti za rad u buduću.

Internacionalna kompanija Methods2Business osnovana je 2010. godine sa sedištem u Eindhoven, u Holandiji, dok se tim inženjera nalazi u Novom Sadu. Projekat kompanije je Wi-Fi HaLow™ koji je namenjen za primenu u IoT oblasti upotrebe. Ovaj Wi-Fi standard je zasnovan na IEEE 802.11 grupi standarda i zvanična oznaka amandmana je 802.11ah.

Za razliku od ostalih Wi-Fi standarda iz ove grupe, HaLow™ radi na frekvencijama od 900MHz što omogućuje povećanje dometa bežične lokalne mreže. Dodatne prednosti ovog standarda su ušteda energije kojom je omogućena dugovečnost uređaja, poboljšanje protoka i iskorišćenosti kanala korišćenjem RAW algoritma, skalabilnost i mogućnost priključivanja značajno većeg broja stanica na jednu pristupnu tačku.

2. ARTS OKRUŽENJE

Kako bi se omogućilo testiranje sistema u realnim uslovima korišćenja potrebno je izgraditi okruženje sa uređajima za komunikaciju i kontrolu koji simuliraju upotrebu. ARTS okruženje se sastoji od četiri FPGA razvojne ploče i jednog računara.

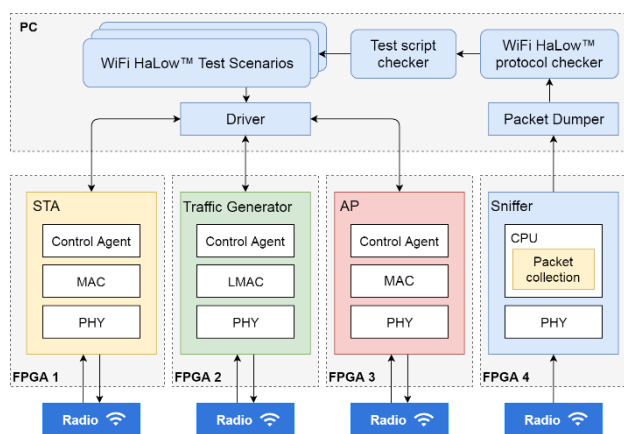
NAPOMENA:

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

Računar predstavlja centar sistema sa kog se upravlja svim ostalim uređajima i na kom se vrši kontrola i provera ispravnosti komunikacije.

Slika 1 prikazuje elemente okruženja i komunikacione puteve između njih. FPGA 1 uređaj konfigurisan je da radi kao stanica, a FPGA 3 da radi kao pristupna tačka. Ova dva uređaja vrše komunikaciju propisanu standardom i simuliraju uređaje iz realne upotrebe.

Uz njih, na FPGA 4 razvojnoj ploči radi Sniffer komponenta koja ima ulogu da prikuplja saobraćaj koji se razmenjuje između stanice i pristupne tačke, da ga skladišti i prosleđuje računaru na dalju obradu.



Slika 1. Osnova ARTS okruženja

Četvrti uređaj, na FPGA 2 razvojnoj proči, predstavlja generator saobraćaja koji služi za simulaciju komunikacije, ubacivanje neodgovarajućih paketa da se proverí robustnost, provera sigurnosti komunikacije, provera enkripcije i slično.

Na računaru postoji više modula koji rade paralelno u sinhronizmu kako bi se cela komunikacija kontrolisala i proveravala. Wi-Fi HaLow™ test scenariji predstavljaju testove realizovane da se izvršavaju korak po korak.

Komande se prosleđuju Drajveru i dalje FPGA pločama. Kada se komande izvrše i započne komunikacija Sniffer prikuplja saobraćaj i prosleđuje ga modulu za proveru usklađenosti sa standardom. Ovaj modul takođe učestvuje i u proverí upita koji su zadati u test skriptama, a ta provera se vrši pomoću Test script checker-a.

3. DRAJVER I KONTROLNI AGENTI

Drajver i kontrolni agenti su moduli koji služe za uspostavljanje komunikacije između računara i FPGA ploča prikazanih na Slici 2. Drajver je deo koji radi na računaru. Unatar test skripti potrebno je inicijalizovati

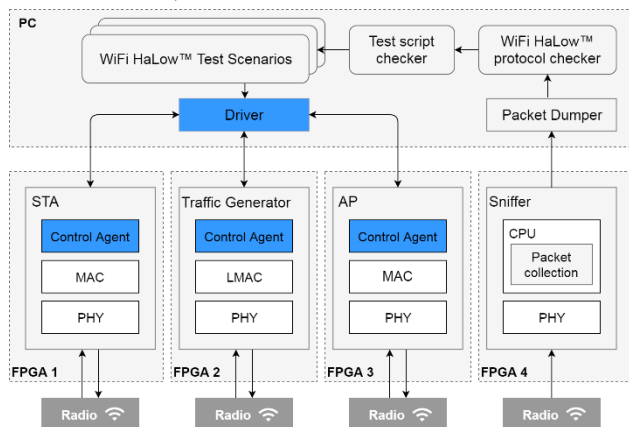
onoliko drajver objekata koliko ima uređaja za komunikaciju. Za potrebe inicijalizacije objekta argumenti koje driver zahteva su:

- host – IP adresa FPGA razvojne ploče
- port – TCP/IP port za komunikaciju
- konfiguracija – vrsta konfiguracije za određenu ploču, da li ima ulogu stanice ili pristupne tačke
- konekcija – predstavlja “pipe” komunikaciju između test skripti i drajvera

Komande se iz test skripti šalju preko “pipe” sistema komunikacije drajveru gde se one obrađuju i proverava se njihova ispravnost, a zatim se preko “socket” komunikacije i TCP/IP protokola šalju na FPGA proču gde ih prima kontrolni agent. Komanda je formirana na sledeći način:

```
sta_associate,interface,m2bwah0,ssid,M2BWIFI,password,""
```

Kontrolni agent ima ulogu da primi komandu, proveriti da li je komandu moguće izvršiti na trenutnoj konfiguraciji ploče, parsira argumente komande i proverava njihovu vrednost. Ukoliko je komanda ispravna poziva se jedan od CAPI-ja (Control Application Programming Interface) koji izvršavaju primljenu komandu u sistemu. Kao sredstvo komunikacije između drajvera i kontrolnih ageneta koriste se četiri sistemske poruke “RUNNING”, “COMPLETE”, “ERROR” i “INVALID”.



Slika 2. Drajver i kontrolni agenti

4. TESTOVI

Testiranje softvera predstavlja istraživanje koje pruža informaciju o kvalitetu sistema ili uređaja koji se testira. Testiranje takođe pruža i nezavisan i objektivan uvid u stanje softvera iz poslovnog ugla i rizik koji dolazi sa implementacijom tog softvera. Proces testiranja vrši se na način da se pokušava izazvati greška u radu programa ili aplikacije, i da se verifikuje spremnost programa za upotrebu. Za novu tehnologiju kao što je M2B Wi-Fi HaLow™ bilo je potrebno razviti novo okruženje kako bi se mogle testirati performance sistema, stabilnost rada i komunikacije i da se osigura da process komunikacije podleže standardu.

4.1 Regresiono, funkcionalno i stres testiranje

Regresiono testiranje podrazumeva izvršavanje funkcionalnih i nefunkcionalnih testova sa ciljem da se osigura kvalitet i ispravnost prethodno napisanog koda. Izmene

koje mogu zahtevati regresiono testiranje su ispravke grašaka u kodu, unapređenje softvera, promena konfiguracije, a potencijalno i promena harverskih elemenata okruženja. Zbog neprestanog rasta i napretka projekta, ARTS okruženje je razvijeno sa ciljem da omogući proširivanje kako hadrverskih elemenata tako i funkcionalnosti koje poseduje okruženje.

Funkcionalno testiranje je tip “black box” testiranja koje je koncentrisano na testiranje određene funkcionalnosti sistema. Ono se vrši pozivanjem funkcionalnosti sa varijacijom ulaznih podataka, a kao rezultat se analizira samo izlaz testirane funkcionalnosti, ne ulazi se u detalje realizacije i konfiguracije (za razliku od “white box” testiranja). Funkcionalno testiranje za cilj ima da proveriti da li su postojeće funkcionalnosti realizovane u skladu sa postojećom dokumentacijom.

Tipovi funkcionalnog testiranja su:

- Smoke testiranje
- Sanity testiranje
- Regresiono testiranje
- Usability testiranje

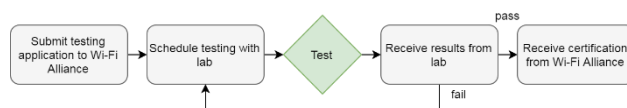
Stres testiranje je način testiranja kojim se sistem dovodi u stanje intenzivne upotrebe i detaljnog testiranja kako bi se otkrile potencijalne nestabilnosti u sistemu. Ono podrazumeva simuliranje upotrebe iznad predviđenih kapaciteta, često do kritične tačke, i posmatranje rezultata naknadno. Razlozi za stres testiranje:

- da bi se odredila kritična tačka do koje sistem pravilno funkcioniše
- za potvrđivanje preciznosti matematičkih modela
- potvrđivanje usklađenosti sa specifikacijama
- ustanoviti način na koji nastaju greške pri opterećenju sistema
- proceniti stabilnost izvan propisanih specifikacija sistema

4.2 Testovi Wi-Fi Alianse

Wi-Fi Aliansa je globalna neprofitabilna organizacija posvećena proliferaciji Wi-Fi tehnologije u mnoštvu uređaja i na raznim segmentima tržišta. Program Wi-Fi Certified™ predstavlja program sertifikacije kojim se dobija široko poznata oznaka koja garantuje interoperabilnost sa drugim uređajima koji su sertifikovani i kvalitet koji ovaj sertifikat garantuje. Dobijanjem ovog sertifikata otvara se mogućnost za prolaznost uređaja sa novim tehnologijama na već postojeće Wi-Fi tržište.

Da bi nova tehnologija stekla pravo na dobijanje sertifikata mora da prođe seriju testova koji su propisani od strane Wi-Fi Alianse. Ovi testovi za cilj imaju da osiguraju usklađenost rada sistema sa propisima zadatim IEEE 802.11ah standardom u slučaju M2B Wi-Fi HaLow™, i interoperabilnost sa drugim uređajima na tržištu koji podržavaju ovu tehnologiju.



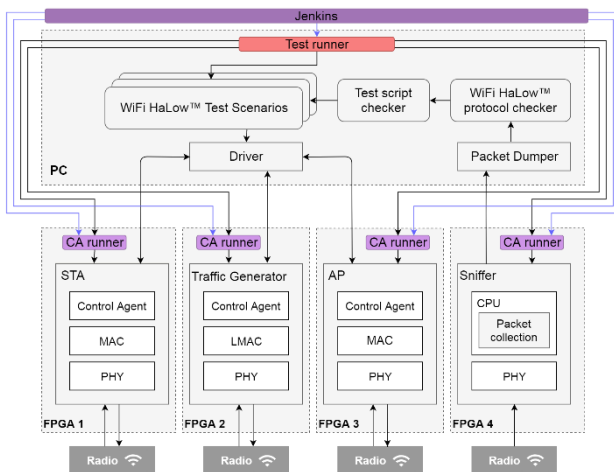
Slika 3. Proces sertifikacije u Wi-Fi alijansi

Slika 3 prikazuje proces sertifikacije:

- Član podnosi zahtev za testiranje Wi-Fi Aliansi
- Podnosilac zahteva koordinira sa akreditovanom laboratorijom oko datuma i vremena testiranja
- Vrš se testovi u autorizovanoj test laboratoriji i izveštaj o testiranju se prosleđuje Wi-Fi Aliansi i podnosiocu zahteva
- Ukoliko je testiranje prošlo neuspešno, podnosioc zahteva ima pravo da ponovi procedure
- Članovi alianse pregledaju rezultate. Ukoliko su svi uslovi ispunjeni i testiranje uspešno izvršeno podnosioc zahteva dobija sertifikat za proizvod koji je podlegao testiranju.

5. EGZEKUTOR I JENKINS ALAT

Razvoj okruženja doveo je do tačke kada korisnik može da izvršava automatizovane testove, ali samo pojedinačno. Da bi se pojedinačni test pokrenuo prvo je potrebno da korisnik pokrene kontrolne agente na FPGA pločama, a zatim pokrene test skriptu. Nova tehnologija kao što je M2B Wi-Fi HaLow™ zahteva puno testiranja, pa ovakav način rada nije odgovarajući. Kako bi se problem prevazišao, novi modul je dodat okruženju. Egzekutor je modul koji radi na nivou iznad osnovnog okruženja i sastoji se iz dva dela, jedan je Test runner, a drugi CA runner prikazani na slici 4. Ova dva modula započnu komunikaciju preko soketa na početku test sesije, a završavaju na kraju cele sesije. Uloga CA runnere je da pokrene kontrolne agente pre svakog testa, nakon što primi odgovarajuću komandu od Test runner modula. U Test runneru se nakon slanja komande pokreće svaki test, a nakon završetka testa ceo ciklus se ponavlja. Ceo tok izvršavanja se prati i ispisi na konzoli se zapisuju u log fajl koji kasnije može biti korišćen za otkrivanje grešaka, naknadnu obradu i slično.



Slika 4. ARTS okruženje sa egzekutorom i Jenkins alatom

Nakon izvršene serije testiranja vrši se postprocesiranje podataka za potrebe dokumentovanja test sesije.

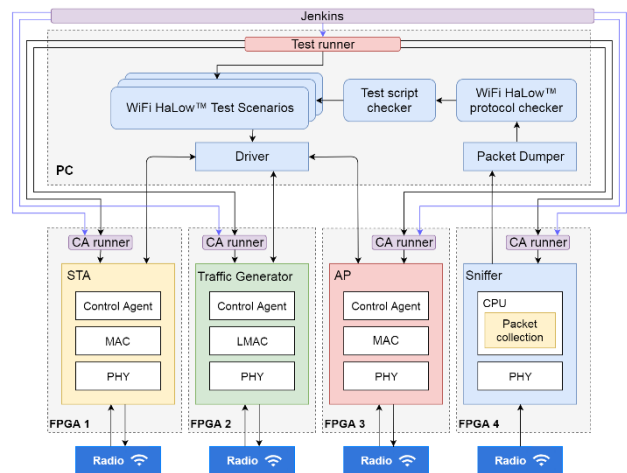
Poslednji alat koji je dodat i za cilj ima kompletnu automatizaciju okruženja je Jenkins alat. Korišćenjem Jenkinsa ceo proces postaje kompletno automatizovan bez potrebe za naknadnim intervencijama od strane korisnika. Na svim uređajima okruženja, pri pokretanju se pokreće i

Jenkins agent koji uređaj stavlja pod kontrolu Jenkins alata na serveru.

Za svaki od uređaja napravljen je Jenkins posao koji izvršava komande na sistemu i automatski pokreće seriju testiranja na celom okruženju. Pomoću posebne vrste Jenkins posla više poslova se može izvršavati paralelno što je odgovarajuće rešenje za kontrolu svih uređaja u okruženju istovremeno.

Izvršavanje ovakvih serija testiranja može biti podešeno vremenski, npr. jednom nedeljno ili da pokretanje bude izazvano nekim bitnim promenama na repozitorijumu. Alat poseduje pregledan interfejs za praćenje rada. Unutar jedne test sesije mogu se pratiti stanja izvršavanja na svim uređajima i ukoliko dodje do greške proces otrivanja uzroka je jednostavan. Praćenje istorije izvršavanja posla je moguće jer se svi logovi unutar Jenkinsa pamte.

Dodavanje ovih modula upotpunjuje funkcionalnost i kompletira finalni izgled okruženja. Na slici 5 prikazano je kompletirano okruženje sa svim komponentama i linijama za prikaz komunikacije.



Slika 6. Finalni izgled ARTS okruženja

6. REZULTATI

Prikupljanje rezultata se vrši na više nivoa. Osnovni način prikupljanja rezultata je beleženje svih ispisa na konzolu tokom izvršavanja svih testova u sesiji. Ovaj dokument postaje relativno obiman u zavisnosti od tipa test sesije, pa je potrebno izvršiti postobradu kako bi se izvukle ključne informacije.

Drugi način beleženja rezultata vrši se unutar test skripte gde se rezultati pamte u rečniku rezultata koji sadrži informaciju o vrsti funkcije, koraku u kom se funkcija izvršavala i o rezultatu izvršavanja. Ovi rezultati se beleže kroz upite postavljene u određenim koracima unutar testa. Nakon završene sesije vrši se postobrada i generišu se dva tipa dokumenta.

Na slici 6 prikazan je pregled rezultata po upitima u svakom testu. Ovaj document pruža mogućnost da se locira mesto nastanka greške kako bi dalja analiza bila olakšana i precizirana.

Na slici 7 prikazan je izgled dokumenta koji pruža generalni pregled test sesije sa rezultatima svih testova

koji su izvršavani. Da bi se test proglasio za uspešno izvršen neophodno je da svi upiti unutar testa budu uspešno izvršeni.

===== Test steps summary =====

Test step: Status:

APUT_5_13.py

Association in step 2: **PASS**

Ping in step 3: **PASS**

Ping in step 4: **PASS**

Ping in step 5: **FAIL**

Ping in step 6: **FAIL**

Ping in step 7: **FAIL**

Association in step 8: **PASS**

APUT_5_14.py

Association in step 2: **PASS**

Ping in step 3: **FAIL**

Association in step 5: **PASS**

Ping in step 6: **FAIL**

Slika 6. Pregled pojedinačnih upita u seriji testiranja

Radi bolje preglednosti ovi dokumenti su generisani u .html formatu koji omogućava bojanje i formatiranje teksta. Završni korak sesije je slanje generisanih dokumenata na listu e-mail adresa. Ovo zaključuje test sesiju koja je pokrenuta i izvršena automatski, dok korisnik na kraju dobije izveštaj o statusu sistema i postignutim rezultatima.

01.08.2018. 11:23:49

This is summary of all tests in this regression:

Test name: Status:

APUT_5_16.py final status: **PASS**

APUT_5_17.py final status: **PASS**

APUT_5_18.py final status: **PASS**

APUT_5_19.py final status: **PASS**

APUT_5_20.py final status: **PASS**

Number of PASSED tests: 5

Number of FAILED tests: 0

Slika 7. Generalni pregled rezultata na nivou test sesije

7. ZAKLJUČAK

Razvijanje softvera je komplikovan proces i mora biti dobro organizovan, naročito ukoliko se razvija nova tehnologija kao što je M2B Wi-Fi HaLow™. Kako vreme odmiče i projekat raste nastaje potreba za proverom ispravnosti ranije napisanog koda kako bi se garantovala stabilnost sistema, performanse i usklađenost sa

standardom. ARTS okruženje predstavlja kompletno automatizovano rešenje dizajnirano da osigura stabilnost sistema kroz regresiono testiranje, performanse stres testiranjem i simulira razne scenarije upotrebe iz realnog sveta.

Ovo kompleksno okruženje sačinjeno od četiri FPGA razvojne ploče, jednog računara i Jenkins alata na serveru može simulirati lepezu događaja i načina upotrebe sa kojima se sistem može sresti u realnoj upotrebi kada se nađe na tržištu.

Postepenim proširivanjem okruženja i objašnjenjem uloge svih uređaja i softverskih komponenata ovaj rad prezentuje način funkcionisanja i logiku iza svakog od njih. Na kompetitivnom tržištu kao što je današnje, vreme, novac i veštine ljudi moraju biti efikasno upotrebljavani.

Korišćenje oružja kao što je ARTS iziskuje vreme za razvoj, ali kada sistem postane funkcionalan, proces testiranja je jednostavniji, bez potrebe za konstantnim intervencijama od strane korisnika i gubljenjem vremena.

8. LITERATURA

[1] <http://www.methods2business.com/>

[2] https://en.wikipedia.org/wiki/IEEE_802.11

[3] IEEE 802.11-2016, Dec 2016; Standard for Information technology -Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

[4] IEEE 802.11ah-2016; Dec 2016; Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation

[5] https://en.wikipedia.org/wiki/Software_testing

[6] https://en.wikipedia.org/wiki/Functional_testing

[7] https://en.wikipedia.org/wiki/Stress_testing

Kratka biografija:



Robert Šafer rođen je 19.05.1992. u Subotici. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu, na Katedri za telekomunikacije i obradu signala u oktobru 2016. godine.



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

RAZVOJ OPTIČKOG KOMUNIKACIONOG UREĐAJA ZA RAD U SLOBODNOM PROSTORU**DEVELOPMENT OF FREE SPACE OPTICAL COMMUNICATION DEVICE**Vojislav Milivojević, *Fakultet tehničkih nauka, Novi Sad***Oblast: ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu opisan je proces razvoja *free space optics* uređaja. Uz pomoć kolimiranog snopa svetlosti dva projektovana uređaja prenose podatke bežično na razdaljini do 150 m i protokom od 10 Gbps. Detaljno je opisan razvoj hardvera i softvera, kao i sva istraživanja koja su omogućila realizaciju ovakvog uređaja.

Ključne reči: *Free space optics, telekomunikacije, PCB projektovanje, embeded softver*

Abstract – This paper describes the process of developing a free space optics device. With the help of a collimated light beam, two devices transmit data wirelessly at a distance of up to 150 m and speed of 10 Gbps. Detailed development of hardware and software, as well as all the research that enabled the realization of such a device is described in the paper.

Keywords: *Free space optics, telecommunications, PCB development, embeded software*

1. UVOD

Brzi i jeftin pristup internetu je jedan od primarnih ciljeva ovog vremena. Instalacija informacionih mreža u bilo kom okruženju ključna je za premošćavanje digitalne podele, kao i za jačanje ekonomskog razvoja. "Digitalna agenda za Evropu 2020" ima za cilj da omogući ultra brzi širokopolasni pristup internetu (100 Mbps) za najmanje polovinu domaćinstava u Evropi, koji je u ovom trenutku dostupan samo za neke korisnike, zbog vrlo visokih troškova infrastrukture.

Internet infrastruktura pristupne mreže zaostaje u nadogradnjama usled visokih troškova i sporog postavljanja novih žičnih instalacija. U većini urbanih područja je korišćenje bežičnog pristupa internet otežano pre svega zbog zagušenja RF spektra.

Zbog svega navedenog, *Free space* optička komunikacija je izuzetno održivo rešenje.

Sistemi *Free-space* optičke komunikacije (FSO) imaju veliki potencijal za ublažavanje ovih problema i ograničenja. Svetlosni spektar nije regulisan, njegova upotreba ograničena je samo propisima o maksimalnoj snazi lasera tako da on ne ugrožava ljudski vid.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Ivan Mezei, vanr. prof.

Svetlost može biti vrlo dobro kontrolisana i fokusirana, pa je mnogo manja verovatnoća da će izazvati zagušenje spektra. Trenutno, na tržištu postoji nekoliko FSO sistema, međutim svi oni su namenjeni za industriju ili telekomunikacione operatore i veoma su skupi za potrošačko tržište. U ovom projektu razvijena je jeftinija FSO tehnologija koja može da se poredi sa Wi-Fi-om u kapacitetu, jednostavnosti korišćenja i troškova, sa osnovnim ciljem da omogući bežičnu mrežu velikog kapaciteta između objekata koji imaju međusobnu optičku vidljivost, sa brzinom od 1 Gbps i 10 Gbps, i to bez postavljanja podzemnih i nadzemnih vodova u dometu od 200m na otvorenom, gde je zagušenje RF najveće.

Ovaj projekat se zasniva na istraživačkom projektu Koruza 1.0 (5. generacija), koji je integrisao naučne pronalaskeske u nisko budžetni optički sistem kroz slobodan prostor za premošćavanje 100m razdaljina sa protokom od 1Gbps. Cilj projekta Koruza bio je razvoj jeftinih otvorenih sistema za primarnu upotrebu u bežičnim mrežama manjih zajednica i istraživačko okruženje za razvoj FSO-a, podržavajući eksperimentisanje i istraživanje na tu temu. Taj cilj je postignut sa sistemom Koruza 1.0.

2. FSO SISTEMI I TEHNOLOGIJA

Sisteme *free-space* optičke komunikacije karakteriše laka propagacija kroz slobodan prostor za prenošenje informacija i ovi sistemi se smatraju bežičnom verzijom tehnologije optičkih vlakana. Smatra se da osobine komunikacionog medija slobodnog prostora utvrđuju uslove rada i pružaju dovoljno informacija za dizajniranje FSO sistema. Pregled komercijalno dostupnih FSO mrežnih sistema, poređenje njihovih karakteristika i ukupna procena postojećih rešenja, postavlja reper kvaliteta sistem razvijenog u ovom projektu. Snop svetlosti prolazi kroz atmosferu, tako da su atmosferski uslovi ključni faktor koji utiče na kvalitet prenosa. Ovi uticaji se mogu prikazati odnosom signal/šum (SNR) i BER (Bit Error Ratio). Neke od prednosti FSO-a su [1].

- Komunikacioni čvorovi nisu fizički povezani (bežični komunikacioni sistem)
- Pristupačnije od sistema sa optičkim vlaknima
- FSO sistem je otporan na RF smetnje
- Sigurnost signala
- Prenos podataka velike brzine
- Jeftiniji od RF sistema (ne zahteva licenciranje za RF spektar)

3. KORUZA 2.0 SPECIFIKACIJA

Svrha ovog rada je prikaz projekta koji je uređen prilikom redizajniranja svih aspekata projekta Koruza1.0. Polazeći od koncepta i ideje projekta, sve do hardvera i softvera koji su realizovani i implementirani u krajnji proizvod. Ono što će ostati isto je funkcija uređaja, tako da će početna tačka tehničke specifikacije biti funkcionalna specifikacija bazirana na prethodnoj verziji sistema. Ovaj rad se bavi samo implementacijom hardvera i softvera unutar projekta. S obzirom na to da je Koruza uređaj koji se može okarakterisati kao telekomunikaciona oprema, potrebno je bude profesionalno izrađen i da budu ispunjeni svi standardi koji su vezani za telekomunikacionu industriju. Planira se da na kraju redizajna uređaj bude ekonomičan za proizvodnju i jednostavan za održavanje.

3.1. Zahtevi projekta

Povratne informacije koje su dobijene od korisnika prethodne verzije uređaja koja je instalirana i operativna širom sveta, su bile od izuzetnog značaja za razvoj nove verzije uređaja. Promene u novoj verziji uređaja su izvršene kako na softveru tako i na hardveru.

Jedan od nedostataka primećenih od strane korisnika je vezan za instalaciju sistema. Glavni problem je bio u tome što je inicijalno podešavanje zasnovano na vidljivom zelenom laseru koji se koristi za određivanje pravca snopa svetlosti koju emitije uređaj. Prilikom postavljanja uređaja na većim rastojanjima, laser nije bio vidljiv tokom dana kada je intenzitet sunčeve svetlosti visok. To je značilo da se instalacija morala odvijati tokom večeri ili noćni, što utiče na bezbednost onoga koji postavlja uređaj.

3.2. Specifikacija hardvera

Na osnovu projektnih zahteva i funkcionalne specifikacije, napravljena je detaljna specifikacija hardvera. Jedan od pomenutih zahteva bio je da uređaj ima samo jedan kabl za upravljanje i korisničke podatke. Zbog toga je postojala potreba za nekom vrstom platforme za rutiranje unutar uređaja.

Ponovnom revizijom funkcionalne specifikacije utvrđeno je da je samo jedna stvar prepreka na putu ka implementaciji jeftine i jednostavne platforme. To je bio zahtev za spajanje upravljačkih podataka i korisničkih podataka u jedan Ethernet kabl.

Nova specifikacija je bazirana na prethodnoj, bez jedne karakteristike koja je unosila dodatnu kompleksnost uređaju Koruza. Nova glavna ploča imaće sledeće karakteristike:

- Glavni CPU koji podržava više Linux distribucija
- Ethernet 10/100 + PoE
- Ethernet Bandwidth: 10Base-T/100Base-TX
- 24VDC pasivno PoE napajanje
- SFP konektor
- 2x USB 2.0 konektor
- GPIO konektor
- Kamera
- Izlaz velike snage
- Drajver za stepper motore

3.3. Specifikacija softvera

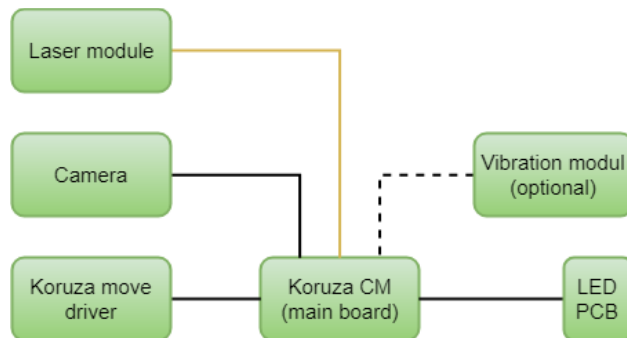
Ovakav proizvod zahteva više različitih softvera koji međusobno komuniciraju. Na osnovu povratnih informacija korisnika iz prethodne verzije, bile su potrebne su neke izmene. Svaki Koruza uređaj je uređaj za sebe i treba ga posebno konfigurisati. Korisnici moraju imati pristup svakom uređaju. Najbolji način pristupa uređaju je putem web pretraživača. Nakon što se uloguje u uređaj, korisnik će imati pristup konfiguraciji i statistici uređaja. Svi uređaji će slati svoj status na jedno centralizovano mesto, cloud server. Tu korisnik može jednostavno da prati celokupno stanje uređaja. Dodatni zahtev je update-ovanje OTA (na daljinu), tako da se softver uređaja može nadograditi u bilo kom trenutku bez dodatnih troškova osoblja na terenu.

Pored korisničkog interfejsa i softvera za praćenje parametara, Koruza ima svoje interne funkcije tako da može ispravno da funkcioniše u svim uslovima. Softver za kontrolu motora, softver za automatsko poravnanje, indikator intenziteta signala itd. [2].

4. IMPLEMENTACIJA HARDVERA

Unutar Koruza uređaja postoji nekoliko elektronskih blokova. Prva i glavna je Koruza CM ploča. Ova ploča sadrži glavnu procesnu jedinicu i dodatne komponente za različite funkcije uređaja. Druga je Koruza drajver motora, mala ploča zadužena za pomeranje dva mala stepper motora. Osim ove dve glavne ploče, jedinica ima nekoliko manjih PCB-ova za obavljanje jednostavnih zadataka.

Kao što je prikazano u blok dijagramu na slici 1, sve ploče su povezane sa glavnom pločom Koruza CM. Ploče koje su povezane sa crnom linijom na dijagramu imaju žičnu vezu, a jedna povezana sa žutom linijom je spojena fleksibilnim PCB-om, dok isprekidana linija predstavlja vezu koja je nije obavezna.



Slika 1. Koruza 2.0 blok dijagram

Koruza CM je glavna ploča koja je zadužena za većinu zadataka unutar uređaja. Kao što je opisano u specifikaciji ova ploča ima nekoliko modula, ali najvažniji je glavni procesor. Izbor ovog CPU-a nije bio lak zadatak.

Uslovi koje je trebalo zadovoljiti su mogućnost lakog modifikovanja, korišćenje poznatih modula kako bi se smanjilo vreme razvoja, i dostupnost u prodaji svih velikih distributera u narednih pet godina. Cilj je bio da se ispune makar dva od tri uslova.

Modul koje je ispunjavao ove uslove i koji je odabran je zasnovan na *Raspberry Pi* platformi. Ovaj modul je bilo lako implementirati u dizajn, a proizvođač je garantovao da će ga proizvoditi do 2025. godine. Pored ovog navedenog, prednost je bila i to što modul bio je dovoljno malih dimenzija i posedovao je sve potrebne funkcije. Naziv ovog modula je *Raspberry Pi Compute Module*, ili skraćeno *Raspberry Pi CM*.

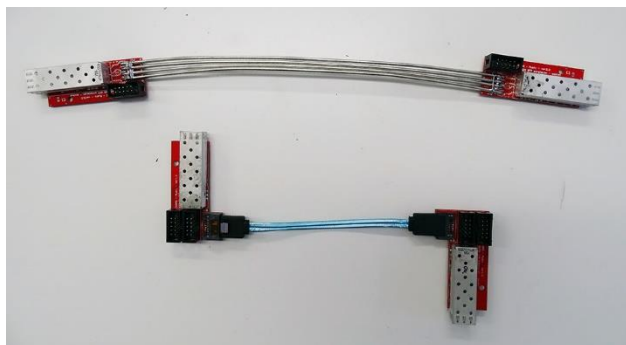
Nakon istraživanja, došli smo do zaključka da je najbolji interfejs za korisničke podatke *small form-factor pluggable* (SFP). To je kompaktan primopredajni optički modul koji se koristi za telekomunikacione aplikacije. Oblik modula i električni interfejs određeni su MSA sporazumom koji je odredio *Small Form Factor Committee*. To je popularan format industrije razvijen i podržan od strane mnogih proizvođača mrežnih komponenti [3].

SFP interfejs na mrežnom hardveru je modularni (*plug-and-play*) slot za varijabilni primopredajnik specifičan za mediji koji prenosi, kako bi se povezao optički kabl ili ponekad bakarni kabl. Ovo je posebno zanimljivo jer na ovaj način možemo obezbediti ili optičku ili *Ethernet* kablovsku vezu sa korisnicima.

4.1. Koruza fleksibilni PCB

Koruza fleksibilni PCB je bio jedan od najizazovnijih delova ovog projekta. Prva verzija sistema, Koruza 1.0, je koristila SATA kabl za povezivanje SFP linija podataka između dva SFP porta, a to je bilo dovoljno za stariju verziju jer je maksimalni protok podataka bio 1 Gbps. Sa novom verzijom, želeli smo da postignemo maksimalni protok od 10 Gbps koji nije bio izvodljiv sa SATA kablom.

Prvi eksperiment urađen je uz pomoć stare verzije sistema. Pokušano je da se postigne brzina od 10 Gbps. Ključno poboljšanje hardvera koje je potrebno za operaciju od 10 G je bilo uklanjanje medija pretvarača i stvaranje direktne konekcije između dva SFP-a. Verzija 1 G koristi SATA kabl koji je dovoljan za tu brzinu, međutim, za 10 Gbps potrebno je bolje rešenje. Tako je ploča od 10 G SFP dizajnirana pomoću koaksijalnih kablova za povezivanje velike brzine. Svaka linija koja se koristi za prenos podataka, RX +, RX-, TX + i TX- bila je povezana sa odvojenim koaksijalnim kablom između laserskog modula uređaja Koruza i SFP porta za korisnika. U ovoj konfiguraciji, operacije do 40 Gbps trebalo bi da budu podržane. Eksperiment je sproveden sa dva modifikovana Koruza 1.0 uređaja.

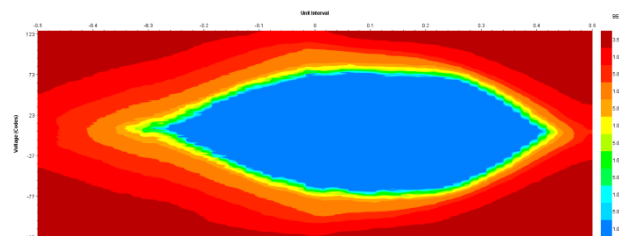


Slika 2. Koaksijalna konekcija (gore) i SATA konekcija (dole)

Nakon ovog eksperimenta, zaključeno je da je Koruza sa brzinom od 10 Gbps izvodljiva. Trebalo je da se napravi uređaj koji je robusan i da se dizajnira za proizvodnju. Koaksijalni kablovi, koji su korišćeni u eksperimentu, su bili skupi i nisu bili dovoljno fleksibilni. Uređaj ima unutrašnji deo koji se pomera zbog laserskog modula, i stoga je neophodno da veza između SFP porta i laserskog modula bude fleksibilna, tako da ne primenjuje dodatno opterećenje na sistem motora.

Posle nekoliko različitih pristupa pri eksperimentima, pronađeno je rešenje problema. Umesto uspostavljanja veze između krutog i fleksibilnog PCB-a za povezivanje *high-speed* vodova, postavljeni su SMD konektor na fleksibilni PCB i montirano je metalno kućište za SFP konektor preko njega. Ovaj pristup je bio dovoljno jeftin i zadržao istu diferencijalnu impedancu između vodova.

Međutim, problem verifikacije *high-speed* primopredajnika je rešen od strane FPGA proizvođača pre nekoliko godina i ovde je iskorišćena funkcionalnost koju su dodali na čipove da bi se utvrdila ukupna margina dizajna u JESD204B rešenju bez kupovine skupe opreme za testiranje [4].



Slika 3. Eye diagram fleksibilnog PCB-a 10Gbps

Otvoreni UI je bio 66,15%, ali uprkos tome, nije bilo grešaka u prenosu podataka. Ovo je sve urađeno sa pseudorandom binarnom sekvencom (PRBS) od 7 bita. U cilju dodatnog opterećenja ploče koja je bila pod testom, PRBS je povećan na 8, a zatim na 9 i tako dalje do 31 bita, što je bio maksimum. Dok se povećavao PRBS, primećeno je i povećanje stope greške. Analizirajući podatke sa dijagrama očiju i rezultate pod testom, došlo se do zaključka da fleksibilni PCB treba promeniti, tako da parazitne kapacitivnosti budu još niže. Ovo se može uraditi sa "rupičastim" (*hatched*) referentnim slojem umesto sa jednodelnom površinom.

4.2. Koruza drajver motora

Koruza drajver za motore je mali upravljački kontroler. Glavni zadatak ove ploče je kontrola steper motora. Ovi motori su zaduženi za pomeranje i usmeravanje laserskog dela unutar Koruza uređaja.

Polazna tačka dizajna su bili motori koja su već izabrani. Ovo su dva unipolarna steper motora 28BYJ-48. Unipolarni steper motor radi sa jednim navojem pomoću centralne slavine po fazi. Svaki se deo namotaja uključuje za svaki pravac magnetskog polja. Svako namotavanje je relativno jednostavno sa komutacijom. Ovo se radi zato što aranžman ima magnetni pol, koji se može preokrenuti bez promene smera struje [5].

5. IMPLEMENTACIJA SOFTVERA

Projekat Koruza ima nekoliko softverskih celina. Glavna komponenta je softver koji se izvršava na ploči *Raspberry Pi Compute Module*-a. Osim toga, postoje i druge manje celine poput *firmware*-a za drajver motora i *firmware*-a za akcelerometar modul. U ovom poglavlju će biti opisane sve softverske komponente pojedinačno. *Firmware* za drajver motora i akcelerometar će biti detaljno opisani, dok će za *Compute Module*-a biti opisana samo inicijalna podešavanja, zbog toga što ovim radom nije obuhvaćen aplikativni softver.

Koruza drajver za motore je mala *Atmel ATmega328p* ploča sa magnetnim enkoderima, tasterima za detekciju krajnjih pozicija i unipolarnim stepper drajverom.

Razmena podataka između Koruza CM ploče i Koruza drajvera za motore se obavlja preko UART komunikacije. Prenosa podataka i nadogradnja *firmware*-a moraju raditi na istom UART protokolu. *ATmega328P* mikrokontroler ne dozvoljava direktno sistemsko programiranje preko UART komunikacije, tako da stvari moraju biti unapred programirane kako bi sve funkcionalno.

Da bi bi omogućen ovakav rad potreban je poseban ISP programator.

Koruza-akcelerometer-module je dodatna ploča/modul koji je ugrađen unutar Koruza uređaja. Ovaj modul se povezuje sa USB portom Koruza CM ploče, a njegova glavna namena je sakupljanje i obrada podataka o vibracijama Koruza uređaja. Postoji nekoliko načina za merenje mehaničkih vibracija. Merenja koja se rade zasnovana su na akcelerometru koji je napravljen pomoću MEMS tehnologije - MPU-6050 akcelerometar. MPU-6050 akcelerometar je uređaj tipa MEMS sa tri ose, koji se može programirati da meri od 2 g do 16 g. Senzor MPU-6050 je montiran na ploču *Flip32 All In One (Pro) Flight Controller V1.03*.

6. ZAKLJUČAK

Koruza bežični optički sistem postavio je pred nas niz interesantnih problem koje smo analizirali u ovom projektu i razvili tehnička rešenja kako bi ih rešili.

Najznačajniji doprinos ovog projekta jeste implementacija fleksibilnog PCB-a koja je obezbedila jeftiniji sistem za međusobno povezivanje dva SFP modula od kojih je jedan montiran na glavnu PCB ploču, a drugi nije.

Sledeći koraci na unapređenju ovog projekta će biti poboljšanje fleksibilnog PCB-a, u cilju poboljšanja stabilnost i kvalitet signala. Još jedan ključni deo je redizajniranje Koruza drajvera za motore, tako da je ploča nadograđena novim mikrokontrolerom koji može da upravlja sa stepper motorima koristeći povratnu spregu magnetnog senzora.

7. LITERATURA

- [1] Z. Ghassemlooy and W. Popoola, "Terrestrial free-space optical communications," in *Mobile and Wireless Communications Network Layer*, 2010, pp. 355–392.
- [2] Koruza Compute Module board Manual, Online: https://github.com/IRNAS/koruza-compute-module/blob/board_1.x/koruza_CM_Manual.md
- [3] Small form-factor pluggable transceiver, Online: https://en.wikipedia.org/wiki/Small_form-factor_pluggable_transceiver
- [4] JESD204B Eye Scan, Online: https://wiki.analog.com/resources/tools-software/linux-software/jesd_eye_scan
- [5] Unipolar Stepper Motor, Online: <https://www.circuitspecialists.com/blog/unipolar-stepper-motor-vs-bipolar-stepper-motors/>

Kratka biografija:



Vojislav Milivojević rođen je u Lazarevcu 1991. god. Diplomirao je na Fakultetu tehničkih nauka 2015. godine sa temom "Implementacija kompjuterske vizije na Xilinx Zync platformi". Iste godine upisuje Master akademske studije na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Embedded sistemi i algoritmi.

kontakt: milivojevicvoja@gmail.com

ASINHRONI PRINCIPI U JAVASCRIPT PROGRAMSKOM JEZIKU**ASYNCHRONOUS PRINCIPLES IN JAVASCRIPT PROGRAMMING LANGUAGE**Marko Striško, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSTVO I AUTOMATIKA**

Kratak sadržaj – Tema rada je proučavanje asinhronih principa i definisanih šablona prilikom rada sa asinhronošću u JavaScript programskom jeziku. Tokom rada su prikazani različiti principi koji omogućavaju rad sa asinhronim podacima, kako onih osnovnih, tako i onih za čije razumevanje je potrebno čvrsto znanje prethodnih osnovnih principa. Svaki princip sa sobom nosi svoje prednosti i mane, koje su adekvatno objašnjene primerima u obliku programskog koda.

Ključne reči: Asinhroni principi, JavaScript

Abstract – The subject of this paper is the study of asynchronous principles and defined patterns when working with asynchronism in the JavaScript programming language. Different principles that allow working with asynchronous data, both basic and those whose understanding requires firm knowledge of the previous basic principles, are presented during the work. Each principle carries with it its advantages and disadvantages, which are adequately explained by examples in the form of a program code.

Keywords: Asynchronous principles, JavaScript

1. UVOD

Asinhron (eng. *asynchronous* = vremenski neusklađeno, neistovremeno) princip je već duži period prisutan u svetu programiranja. Ustaljena je navika da se kod u najvećem broju slučajeva izvršava sinhrono (eng. *synchronous* = istovremen, jednakovremen), međutim postoje scenarija gde je asinhron princip mnogo efikasniji i primenljiviji, koji proističe iz svoje mogućnosti da se u isto vreme izvršava veći broj operacija. Suština problema je u tome što su sinhroni zahtevi potencijalno blokirajući, tj. postoji mogućnost čekanja određeni vremenski period da se operacija izvrši.

2. ASINHRONI I SINHRONI MODELI

U okviru ovog poglavlja, biće ukratko predstavljene glavne razlike između sinhronog i asinhronog modela.

2.1. Sinhroni programski model

Kod sinhronog modela, procesi se izvršavaju jedan za drugim. Kada se pozove izvršavanje određene funkcije, koja izvršava „dugoročni“ process, rezultat će biti vraćen onog trenutka kada proces završi svoj posao.

NAPOMENA:

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

Na ovaj način, sistem koji izvršava navedeni proces biva zaustavljen sve do onog momenta dok se ne vrati krajnji rezultat izvršavanja. Ovo kao posledicu ima slučaj da će sledeći proces ili zadatak započeti tek po završetku prethodnog, što se može videti sa slike 1.

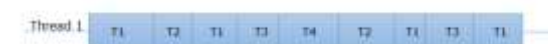


Slika 1. Sinhrono izvršavanje, jedan tok izvršavanja

Ukupno vreme izvršavanja biće suma vremena koje je trebalo da se ovi zadaci redom završe, što potencijalno predstavlja problem, u zavisnosti od toga kolika je suma kao krajnja vrednost [1].

2.2. Asinhroni programski model

Asinhroni model omogućava da se više procesa dešavaju u isto vreme. Glavna ideja asinhronosti je omogućavanje da korišćeni *thread* ne stoji blokirano, tj. „besposlen“ čekajući da se zadatak izvrši, niti da se sama asinhrona aktivnost dešava u tom trenutno korišćenom *thread-u*. Kada započne izvršavanje izabranog zadatka, sistem će nastaviti svoje izvršavanje potpuno nesmetano, bez blokiranja ili čekanja da se započeti zadatak završi. Kada dođe trenutak uspešnog završetka zadatka, sistem biva obavešten o statusu i dobija pristup rezultatu izvršavanja. Slika 2. prikazuje sam redosled operacija u asinhronom programskom modelu [1].



Slika 2. Asinhrono izvršavanje zadataka

2.3. Razlike između asinhronosti i paralelizma.

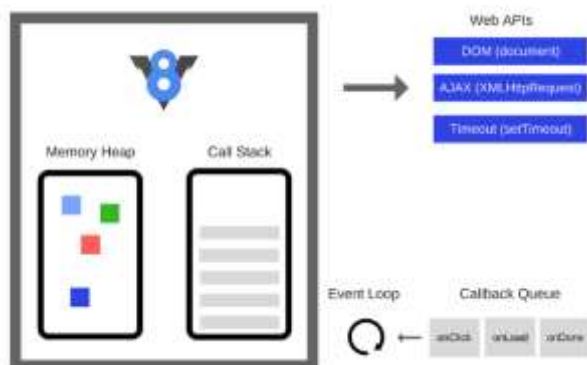
Česta je situacija da se zajedno spominju termini asinhronost i paralelno programiranje, ali su oni u osnovi prilično drugačiji. Suštinski gledano, predstavljaju dva načina implementacije konkurentnosti. JavaScript bazira svoju implementaciju konkurentnosti na jednom *thread-u*. Dok sa druge strane, ranije napomenuti *thread-ovi*, zajedno sa procesima, predstavljaju glavno „oruđe“ prilikom paralelnog programiranja. Navedena asinhronost, koja se postiže procesom koji se zove *event loop* deli posao na manje zadatke i izvršava ih serijski, tj. jedno za drugim, sprečavajući na taj način paralelni pristup i promenu vrednosti u deljenoj memoriji. Preklapanje izvršavanja paralelnih niti i preklapanje asinhronih događaja dešavaju se na totalno drugačijim nivoima. U okruženju koje se izvršava upotrebom samo jednog *thread-a*, uopšte nije bitno koji tip operacija se izvršava, jer ništa ne može da prekine taj *thread*. Dok u paralelnim sistemima, gde su u optičaju veliki broj različitih *thread-ova* u istom programu, postoji velika šansa da se dese neplanirana scenarija [3].

3. ASINHRONOST I JAVASCRIPT

Pored nekoliko načina manipulacije izvršavanja asinhronih operacija korišćenjem *JavaScript* jezika, biće prikazana i struktura *JavaScript* okruženja, koja je od primarnog značaja za načine funkcionisanja asinhronih operacija.

3.1. JavaScript okruženje

JavaScript okruženje se sastoji od nekoliko komponenti. Glavna komponenta nosi naziv *JavaScript Engine* koja je najčešće ugrađena u pretraživač i web server, koji omogućava **runtime compilation** i nakon toga izvršavanje *JavaScript* koda. Navedena komponenta se sastoji od *Memory Heap-a* i *Call Stack-a*. U preostali deo spadaju *WEB APIs* i *Event Loop* zajedno sa *Callback/Event Queue-om* (slika 3).



Slika 3. Celokupno JavaScript okruženje

3.1.1. Call Stack

Call Stack predstavlja linearno struktuiranu memoriju, poseduje veliku brzinu, služi za privremeno čuvanje podataka (prostih primitiva, funkcija) i čuva informacije o tome gde se trenutno nalazimo u kodu. Ova struktura podataka koristi LIFO (*Last In First Out*) princip. Kad funkcija bude pozvana, tada će biti stavljena na vrh *stack-a*. Kad funkcija vrati rezultat, ista biva izbačena iz *stack-a*.

3.1.2. Event Loop

JavaScript za izvršavanje asinhronih operacija koristi strukturu koja nosi naziv *Event Loop*. Svaki put kad se želi izvršiti neka od asinhronih operacija, prvo što se dešava jeste da se ta operacija dodaje u **Event Table**. *Event table* je struktura podataka u kojoj je definisano koja funkcija je vezana za koji događaj. Kada se određeni događaj izvrši, funkcija čiji se događaj aktivirao biva poslata u **Event Queue** [4]. *Event queue* je takođe struktura podataka, opet se novi podaci dodaju na kraj, ali se samo mogu izbaciti spređa (FIFO). Na ovaj način se čuva tačan redosled ubacivanja u *queue*, da bi se znalo koja funkcija je naredna za obrađivanje. Da bi *event queue* poslao funkciju koja je prva na redu u *call stack*, koristi se prethodno spomenuti *event loop* [4]. *Event Loop* predstavlja proces koji se konstantno izvršava u pozadini i proverava da li je *call stack* prazan. Najlakše ga je zamisliti kao sat, koji kad god otkuca, uradi se provera da li je *call stack* prazan. Ako jeste, onda se izvršava provera stanja u *event queue*. Ako *event queue* sadrži funkcije u sebi, prva na redu se automatski prebacuje u *call stack* [4].

3.2. Callbacks

Asynchronous callbacks ili samo *callbacks* je princip za koji se može slobodno reći da je najčešći način na koji se asinhronost u *JavaScript-u* izražava i upravlja, tj. predstavlja prvobitni i najosnovniji asinhroni patern u *JavaScript* programskom jeziku.

3.2.1. Osnove callback principa

Callback nije ništa drugo do obična *JavaScript* funkcija prosleđena kao argument drugoj funkciji, za koju se očekuje da bude izvršena u nekom trenutku unutar funkcije kojoj je prosleđena. *Callback* funkcije su nasleđene od strane programske paradigme koja se zove funkcionalno programiranje (eng. **functional programming**). Glavni razlog postojanja *callback* funkcija je u tome što je *JavaScript* jezik vođen događajima (eng. **event-driven language**) sa jednim izvršnim *thread-om*. Asinhron *callback* može biti iskorišćen kad se ne želi čekati završetak izvršavanja određene operacije. Ta operacija će na ovaj način postati neblokirajuća, ako se definiše kao *callback* [2].

3.2.2. Nedostaci callback principa

Callback princip zaista predstavlja osnovnu jedinicu asinhronosti u *JavaScriptu*. Ipak, praksa pokazuje da ovaj princip nije dovoljan za evolutivni napredak asinhronog programiranja.

3.2.2.1. Callback hell

Listing 1. prikazuje *setTimeout()* funkciju koja odlaže izvršenje logike za određen vremenski period. Prvi deo programa predstavljaju A i B delovi koda, dok drugi deo predstavlja C. Prvi deo se odmah izvršava i nakon toga dolazi do pauze neodređene dužine. U nekom trenutku u budućnosti, program će nastaviti gde je stao i nastaviti sa drugim delom [5].

```
1. // A deo koda
2. setTimeout(function () {
3.     // C deo koda
4. }, 2000);
5. // B deo koda
```

Listing 1. Sinhrono razmišljanje tokom asinhronosti

Iako je na operativnom nivou mozak predstavljen kao asinhron, ljudi ipak planiraju izvršavanje svojih svakodnevnih zadataka na sekvencijalni, tj. sinhroni način. Stoga, ako se sinhroni mozak dobro mapira na sinhrono izvršenje programske logike, koliko dobro mozak radi kad planira izvršenje asinhronog koda? Ljudsko razmišljanje se odvija korak po korak, ali alati (*callback* funkcije) dostupne u kodu se ne izražavaju po principu korak po korak onog trenutka kad se programski tok prebaci sa sinhronog na asinhrono.

Navedena isključenost je suština samog nedostatka *callback* principa prilikom rada i upravljanja asinhronim operacijama. Onog trenutka kad se predstavi kontinualna tj. upravljanje tokom programa u obliku *callback* funkcija, omogućili smo razilaženje načina kako možak funkcioniše i način na koji će program funkcionisati [3].

3.2.2.2. Inversion of Control

Zasniva se na ideji da druga strana kojoj je predana *callback* kontinuiranost nije funkcija napisana od strane programera koji razvija trenutni sistem, niti pod njegovom kontrolom, već je to funkcionalnost dostavljena od neke druge spoljašnje strane. Ovakva situacija nosi naziv „inverzija kontrole“ (eng. ***inversion of control***) gde se uzme deo programa i njegova izvršna kontrola se preda drugoj strani.

Glavni razlog zašto se ne preporučuje upotreba *callback* funkcija je to što ovaj patern ne pruža nikakvu zaštitu od potencijalnih grešaka dok je kontrola na drugoj strani. Sva neophodna mašinerija za takvu vrstu zaštite se mora lično izgraditi.

3.3. Promises

Najveći izazov koji nastaje prilikom rada sa *callback* funkcijama jeste nemogućnost izražavanja asinhronosti na što je moguće više sinhroni način, čime bi se što lakše ispratio tok programa. Rešenje za navedene probleme se pronalazi u obrascu koja nosi naziv **Promises** (eng. **Promise** = obećanje). Umesto predavanja kontinuiranosti programa drugoj strani, ovde se očekuje, kao povratni rezultat, mogućnost da se sazna u kom trenutku će zadatak, koji obavlja druga strana, biti završen i onda implementacija može da odluči šta će se dalje raditi [3].

Promise princip predstavlja vrednost koju možemo da iskoristimo u nekom trenutku u budućnosti i u odnosu na *callback* funkcije, pružaju garanciju za buduću vrednost. Jedna od najbitnijih koncepata *Promise* principa predstavlja činjenica da nijedna registrovana strana koja posmatra i čeka razrešenje *Promise* objekta ne može da promeni vrednost rezultata (slučajno ili namerno) nakon razrešenja (*Promise* objekti su nepromenljivi (eng. immutable)).

3.3.1. Stvaranje Promise objekta

Konstruktor *Promise*-a kao argumente prima dve funkcije. Prva funkcija (**resolve**) biće pozvana kada se asinhrona operacija izvršila uspešno. Njena povratna vrednost biće rezultat asinhronne operacije. Druga funkcija (**rejected**) poziva se kada se asinhrona operacija nije izvršila uspešno. Povratna vrednost *rejected* funkcije sadrži *error* objekat [4]. Listing 2. prikazuje primer pravljenja *Promise* objekta.

```
1. var p = new Promise(function(resolve, reject) {
2.   if ( /* uslov */ ) {
3.     resolve( /* vrednost */ ); // uspešno razrešen
4.   } else {
5.     reject( /* razlog */ ); // greška, odbijen
6.   }
7. });
```

Listing 2. Prikaz stvaranja *Promise* objekta

Na osnovu navedenog listinga, *Promise* objekti poseduju tri moguća stanja:

- **Pending**: Predstavlja početno stanje pre nego što operacija započne.
- **Fulfilled**: Predstavlja stanje uspešno izvršene operacije
- **Rejected**: Operacija se nije uspešno završila, greška je vraćena kao rezultat.

3.3.2. Korišćenje Promise principa

Da bi se iskoristio *Promise* objekat, potrebno je postaviti *handler* na *Promise* korišćenjem *then()* funkcije. Ova funkcija uzima kao parametar funkciju koja će biti prosleđena kao *resolve* vrednost, tj. ona će biti vraćena kao vrednost ako se *Promise* uspešno razreši, dok je drugi parametar funkcija koja će biti pozvana kada *Promise* ne bude uspešno razrešena.

3.3.3. Ulančavanje Promise-a

Promise ne predstavlja isključivo mehanizam za jednostavne pojedinačne korake u obliku *this-then-that*. Naravno, to predstavlja osnovu, ali postoji mogućnost povezivanja više *Promise* objekata zajedno. Svaki poziv *then()* funkcije na *Promise* objektu, stvara novi *Promise* objekat i vraća se kao povratna vrednost, na koji se može dalje ulančavati.

3.3.4. Nedostaci Promise principa

Prilikom ulančavanja *Promise* objekata, javlja se mogućnost da greška koja se javi u bilo kom razrešenju *Promise* objekta bude ignorisana, tj. propuštena. Ako se napravi lanac *Promise* objekata koji nema obradu grešaka uključen u lanac, bilo koja greška bilo gde u lancu biće propagirana dole do dna lanca.

Promise po definiciji uvek imaju samo jednu uspešno razrešenu vrednost ili samo jedan neuspešni razlog za odbijanje. Onog trenutka kada se napravi *Promise* objekat i kada se registruju uspešna ispunjena ili odbijanja samog objekta, ne postoji način da bi se zaustavio započeti proces. Ova ideja narušava verodostojnost buduće vrednosti i ideju o nepromenljivosti na kojoj je originalno *Promise* princip zasnovan [3].

3.4. Async / Await princip

Async/Await princip ili *Async Functions* kako još nosi naziv predstavlja sintaksu za kontrolisanje asinhronog programskog toka u *JavaScript*-u. Inicijalno je započeo u okviru *C#* i *F#* programskog jezika, nakon čega se pojavljuje kao standard u *EcmaScript2017*, tj. verziji 8. Ovaj princip u potpunost omogućava pisanje asinhronog koda na sinhroni način. Takođe raspolaze sa veoma jasnim i intuitivnim načinom za obradu grešaka, jer u sebe uključuje *try..catch* sintaksu.

```
1. async function getValueWithAsync() {
2.   const value = await this.resolveAfter2Seconds(20);
3.   console.log(`Async rezultat: ${value}`);
4.   return value;
5. }
```

Listing 3. Primer upotrebe *async / await* sintakse

Obavezna je upotreba *async* ključne reči ako funkcija sadrži *await* ključnu reč, u suprotnom kompajler će prijaviti *SyntaxError* grešku. Svaka *async* funkcija implicitno vraća *Promise* objekat, dok će rezultat razrešenja tog objekta biti ono što je vraćeno iz *async* funkcije.

3.5. Observables

Glavni nedostaci *Promise* principa ugledaju se pre svega u tome da će se kao rezultat uvek vraćati samo jedna vrednost, tj. samo jedna vrednost će biti vraćena nakon uspešnog (ili neuspešnog) razrešenja *Promise* objekta. Još jedan nedostatak predstavlja nemogućnost prekidanja započetog izvršavanja *Promise* objekta. Navedene nedostatke nadoknađuje **Observables** princip.

Observable princip (eng. *Observable* = vidljiv, opažen) zasnovan je na *Observer* paternu. Ovaj patern funkcioniše tako što postoji *subjekat*, koji sadrži listu *objekata* koji zavise od njega. Prilikom bilo koje promene stanja, *subjekat* poziva *posmatrača* (eng. **observer** = posmatrač) koji automatski obaveštavaju zavisne *objekte* o promeni stanja *subjekta*. *Observables* princip predstavljen je kao deo *EcmaScript-a 2016*. Ovaj princip potiče iz **RxJS** (*Reactive Extensions Java Script*) biblioteke, koja predstavlja *Observables* princip kao novi *push* sistem za *JavaScript* [6].

Observables se može uporediti sa funkcijama. Sličnost se zasniva na tome da pozivanje *subscribe()* kod *Observables* principa analogno je pozivanju funkcije. Razlike između funkcije i *Observables* principa leži u tome da funkcije mogu da vrate jednu i samo jednu vrednost. Dok to ne važi za *Observables*, koji mogu da vrate koliko god vrednosti je potrebno [6].

3.5.1. Struktura Observables-a

Postoje četiri glavna aspekta koja se nalaze unutar *Observables* instanci, mada su neki od aspekata vezani za druge tipove, kao što su *Observer* i *Subscription*. Spomenuta četiri aspekta su:

- Stvaranje *Observable* instance.
- Pretplaćivanje (*subscribe*) na *Observable* instance.
- Izvršavanje *Observable* instance.
- Odlaganje *Observable* instance.

4. ZONE.JS

Ranije je spomenuto da je neophodno okruženje da bi se *JavaScript* mogao izvršavati, bilo unutar pretraživača ili unutar *Node.js runtime-a*. Navedena okruženja direktno su odgovorna za zakazivanje izvršavanja *JavaScript* operacija. Problem koji se javlja je efikasno praćenje izvršavanja svih zadataka, kako sinhronih, tako i asinhronih. *Zone.js* upravo pruža takvu mogućnost, stvarajući konteksts, tj. celinu u kome se izvršavaju asinhronne operacije, nad kojom stvaraoac takvog konteksta ima mogućnost da posmatra i kontroliše izvršavanje koda unutar navedene zone [7].

5. ZAKLJUČAK

JavaScript nesumnjivo predstavlja jedan od najpopularnijih jezika u programerskoj delatnosti. *JavaScript* se ne koristi više samo kao klijentski deo, već je sposoban i za rad nad serverskim delom, što omogućava stvaranje *full-stack* aplikacije potpuno pod kontrolom *JavaScript-a*. Međutim, ispostavlja se da njegova prisutna ekscentričnost čini potpuno ovladavanje svim principima koje poseduje kao izuzetno težak podvig. Jedan od glavnih principa prilikom rada sa *JavaScript-om* leži u njegovoj sposobnosti u izvršavanju više različitih zadatak u isto vreme, upotrebom asinhronosti. Upravo u tome leži sama ekscentričnost samog jezika, koja ovim omogućava izuzetne performanse. Iako je opšte poznato da se *JavaScript* može koristiti bez razumevanja, to isto razumevanje se često jako teško postiže.

6. LITERATURA

- [1] https://eloquentjavascript.net/11_async.html (pristupljeno u junu 2018.)
- [2] K. Gallaba, A. Mesbah and I. Beschastnikh, “Don’t call us, we’ll call you: Characterizing Callbacks in JavaScript”, Canada 2015.
- [3] Kyle Simpson, “You Don’t Know JS: Async & Performance.”, USA 2015.
- [4] Daniel Parker, “JavaScript with Promises”, USA 2015
- [5] Keheliya Gallaba, “Characterizing and refactoring asynchronous JavaScript callbacks”, University of Moratuwa, 2011
- [6] Mark Clow, “Observers, Reactive Programming and RxJS”, California 2018.
- [7] <https://docs.google.com/document/d/-1F5Ug0jcrm031vhSMJEOgp1l-Is-Vf0UCNDY-LsQtAIY> (pristupljeno u avgustu 2018.)

Kratka biografija:



Marko Striško, rođen je u Beogradu 1993. god. Osnovnu i srednju školu završio u Staroj Pazovi. Diplomirao 2016. god. na Fakultetu tehničkih nauka, smer računarstvo i automatika, na kojem iste godine upisuje master studije.

4G LTE STANDARD MOBILNE TELEFONIJE I NJEGOVA OPENAIR INTERFACE (OAI) REALIZACIJA**4G LTE STANDARD OF MOBILE PHONES AND ITS OPENAIR INTERFACE (OAI) REALISATION**

Srđan Šobot, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**2. LTE ARHITEKTURA**

Kratak sadržaj – U ovom radu opisan je 4G standard mobilne telefonije, kao i OpenAir Interface - platforma otvorenog koda koja pruža potpunu implementaciju 3GPP standarda, kroz celokupnu realizaciju svih slojeva protokola.

Ključne reči: OpenAir Interface, 4G mobilne mreže, LTE

Abstract – This paper describes 4G LTE standard of mobile communications, and also OpenAir interface platform that provides open-source software-based implementation of the LTE system spanning the full protocol stack of 3GPP standard.

Keywords: OpenAir Interface, 4G mobile networks, LTE

1. UVOD

Brz rast količine podataka koji se prenose mobilnom mrežom, podstaknut sve većim brojem mobilnih uređaja, otvara široko polje za inovacije u pogledu arhitekture, protokola, servisa i algoritama koji se koriste u mobilnoj mreži. Dok se 4G LTE (Long-Term Evolution) standard mobilne telefonije širi svetom, uporedo se radi na istraživanju i razvoju sledeće generacije mobilne mreže koja bi predstavljala proširenu verziju LTE sa novim i naprednim karakteristikama. Inovacije i razvoj u sledećem standardu zahtevaju otvorenu i fleksibilnu platformu koja će omogućiti kompleksne simulacije realnih sistema. Ograničenja nametnuta od strane operatora i velikih proizvođača onemogućuju istraživanje sa komercijalnom opremom, te se stvara dodatni razlog za razvoj ovakve platforme. S druge strane, korišćenje SDR (Software Defined Radio) uređaja, koji predstavljaju univerzalnu hardversku platformu, daje mogućnost da se svi komunikacioni moduli realizuju putem softvera.

U ovom radu dat je pregled 4G LTE standarda mobilne telefonije, kao i njegova arhitektura i radio interfejs. Takođe, predstavljena je OpenAirInterface (OAI) platforma koja omogućava istraživanje LTE sistema jer pruža kompletnu implementaciju LTE standarda, kroz celokupnu realizaciju svih slojeva protokola. OAI povezuje dve glavne komponente LTE arhitekture: E-UTRAN (Evolved Universal Terrestrial Radio Access Network) i EPC (Evolved Packet Core), ali takođe može da funkcioniše kao korisnička oprema.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dejan Vukobratović.

LTE radio-pristupna mreža u 3GPP sistemu bazira se na novoj arhitekturi - SAE (System Architecture Evolution), koja je osmišljena sa ciljem da pojednostavi LTE mrežu i uspostavi ravnu arhitekturu sličnu drugim IP-baziranim mrežama. SAE se sastoji iz radio pristupne mreže (Radio-Access Network - RAN) koja je nazvana Evolved Universal Terrestrial Radio Access Network (E-UTRAN) i jezgra mreže (Core Network - CN) nazvane Evolved Packet Core (EPC). Zajedno, RAN i EPC čine Evolved Packet System (EPS).

Evolved Packet Core (EPC) sastoji se od nekoliko vrsta čvorova:

- MME (Mobility Management Entity) predstavlja kontrolni čvor koji je odgovoran za povezivanje nosioca sa terminalom, rukovanje načinom rada (idle-active), autentifikacijom korisnika uz pomoć HSS (Home Subscriber Server), inicijalna selekcija S-GW za UE (User Equipment). Jedna mreža može da sadrži nekoliko MME čvorova, gde svaki opslužuje jedan geografski region. Svakom terminalu dodeljen je jedan MME čvor, ali ukoliko terminal napusti region, automatski mu se dodeljuje drugi MME čvor;

- P-GW (Packet Data Network Gateway) je ključni čvor za povezivanje EPC sa spoljašnjim svetom. Putem SGI interfejsa svaki P-GW razmenjuje podatke sa drugim mrežnim operatorima, serverima ili internetom. Svaki P-GW može da se indentifikuje putem APN (Access Point Name). Takođe, P-GW je poveznica ka ne-3GPP tehnologijama, kao što je CDMA2000;

- S-GW (Serving Gateway) služi kao ruter, prosleđuje podatke između bazne stanice i P-GW, odnosno povezuje EPC sa RAN. Odgovoran je za mobilnost terminala kada prelaze iz jednog eNodeB u drugi, kao i za upravljanje procesima ka drugim 3GPP tehnologijama (GSM/GPRS, HSPA). Kao u slučaju sa MME čvorom, jedna mreža može da sadrži više S-GW čvorova koji opslužuju različite regione. Svakom mobilnom terminalu dodeljen je jedan S-GW čvor.

LTE radio-pristupna mreža koristi ravnu arhitekturu sa jednom vrstom čvora - eNodeB, koji je odgovoran za sve radio orijentisane funkcije u jednoj ili više ćelija. Važno je napomenuti da je eNodeB logički, a ne fizički čvor. ENodeB je povezan sa EPC uz pomoć S1 interfejsa, tačnije povezan je na S-GW uz pomoć S1 user plane (S1-u), i sa MME preko S1-c. Jedan eNodeB može da se poveže na nekoliko MME/S-GW, sa ciljem smanjenja

opterećenja. X2 interfejs se koristi za međusobno povezivanje eNB čvorova [1,2].

3. ARHITEKTURA RADIO PROTOKOLA

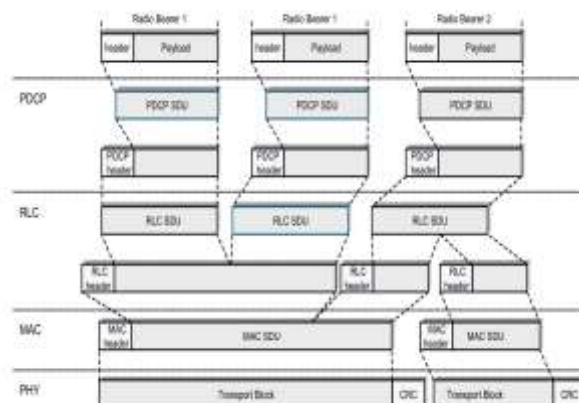
Arhitektura radio protokola u LTE može da se podeli na dva dela: kontrolnu ravan i korisničku ravan. Mnogi slojevi radio protokola zajednički su za ove ravni. Kontrolna ravan, između ostalog, odgovorna je za povezivanje, mobilnost i sigurnost u LTE mreži. Kontrolne poruke koje se šalju ka terminalima mogu da dolaze i od MME čvora koji je smešten u jezgri mreže, ali i od RRC (Radio Resource Control) čvora smeštenog u eNB. RRC čvor koji se nalazi u eNB odgovoran je sve radio orijentisane funkcije: emitovanje sistemskih informacija potrebnih terminalu da bi komunicirao u ćeliji; prenos paging poruka koje šalje MME čvor da bi obavestio terminal o nadolazećem zahtevu za povezivanje; upravljanje povezivanjem, što uključuje podešavanje parametara potrebnih za komunikaciju između terminala i RAN; mobilnost, što omogućava terminalu izbor ćelije; rukovanje mogućnostima terminala (nisu svi terminali u mogućnosti da podrže sve funkcije koje nudi LTE specifikacija).

Poruke od strane RRC čvora ka terminalu šalju se uz pomoć SRB (Signaling Radio Bearer) nosioca. Ovi nosioci mapiraju se na CCCH kanal tokom uspostavljanja konekcije, a nakon što je konekcija uspostavljena mapiraju se na DCCH kanal. U LTE, terminal može da bude u dva različita stanja: RRC_CONNECTED (aktivan) i RRC_IDLE (neaktivan). Kada je terminal u aktivnom stanju, parametri potrebni za komunikaciju između terminala i RAN poznati su obema stranama. Aktivno stanje namenjeno je za prenos podataka ka terminalu, pri čemu postoji mogućnost isprekidanog prijema da bi se smanjila potrošnja baterije u terminalu. U neaktivnom stanju, terminal ne pripada ni jednoj ćeliji, ali povremeno proverava da li postoje paging poruke [1].

Radio protokol korisničke ravni se sastoji iz sledećih slojeva:

- PDCP (Packet Data Convergence Protocol) vrši kompresiju headera IP paketa s ciljem da smanji ukupan broj bita koji se prenose radio interfejsom. ROHC (Robust Header Compression) je standardizovan algoritam koji se koristi za kompresiju kako u LTE, tako i u ostalim tehnologijama radio prenosa. PDCP je, takođe, odgovoran za šifrovanje i integritet podataka, kao i za isporuku paketa određenim redosledom. Na prijemnoj strani PDCD protokol obavlja dešifrovanje podataka i dekompresiju header-a.
- RLC (Radio-Link Control) je odgovoran za segmentaciju/spajanje, retransmisiju, uklanjanje duplikata i za redosled dostave paketa višim slojevima.
- MAC (Medium-Access Control) rukovodi multipleksiranjem logičkih kanala, vrši hibridnu ARQ retransmisiju i upravlja raspoređivanjem za uplink i downlink. Funkcija planiranja smeštena je u eNodeB. Hibridni ARQ protokol nalazi se i u predajnoj i u prijemnoj strani MAC sloja.
- PHY (Physical Layer) rukovodi kodovanjem/dekodovanjem, modulacijom/demodulacijom, MIMO. Fizički sloj obezbeđuje MAC sloju servise u vidu transportnih kanala [1].

Na slici 1. prikazan je downlink tok podataka kroz sve slojeve protokola. Ovaj primer prikazuje tri IP paketa, od kojih se dva mapiraju na jedan radio nosilac (bearer), a jedan paket na drugi radio nosilac.



Slika 1, LTE tok podataka kroz slojeve RAN protokola [1]

Prema primeru na slici 1, PDCP sloj vrši kompresiju IP zaglavlja, zatim šifrovanje. Dodaje PDCD zaglavlje (header) u koje je upisana informacija potrebna za dešifrovanje podataka u terminalu. Izlaz iz PDCP sloja proleđuje se RLC sloju. RLC sloj vrši spajanje i / ili segmentaciju SDU-a, koje je dobio od gornjeg sloja. Zatim dodaje RLC zaglavlje koje se koristi za pravilan redosled dostave u terminalu, kao i za identifikaciju u slučaju retransmisije. RLC PDU se zatim šalje u MAC sloj, gde se vrši multipleksiranje primljenih PDU-a i dodaje MAC zaglavlje da bi se formirao transportni blok podataka. Konačno, na fizičkom sloju dodaje se CRC transportnom bloku u svrhu detekcije greške, vrši se kodovanje i modulacija te prenos signala uz pomoć nekoliko antena [1].

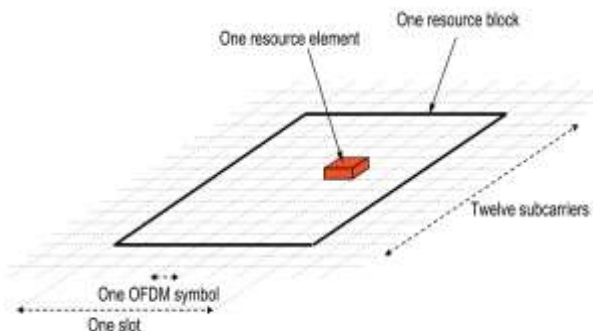
4. LTE RADIO INTERFEJS

LTE radio interfejs bazira se na upotrebi OFDM (Orthogonal Frequency Division Multiplexing), koji nudi visok stepen otpornosti na frekvencijsku selektivnost kanala, omogućava fleksibilnu alokaciju opsega i broadcast /multicast prenos od više baznih stanica. Za downlink kanal koristi se OFDMA (Orthogonal Frequency-Division Multiple Access) tehnika prenosa sa višestrukim pristupom, dok se za uplink koristi SC-FDMA (Single Carrier-Frequency Division Multiple Access) koji omogućava korištenje efikasnijeg pojačavača snage, što je za mobilni terminal veoma vežno. OFDM ima i nekoliko nedostataka, ukoliko se izgubi ortogonalnost između nosioca dolazi do interferencije između potkanala, ali ovaj nedostatak se otklanja uvođenjem cikličnog prefiksa, tj. poslednji deo OFDM simbola se kopira i umeće na početak simbola. Drugi nedostatak se odnosi na trenutne amplitude OFDM signala koje imaju veliki dinamički opseg, što zahteva linearne pojačavače ili dodatnu obradu signala pri modulaciji. Ipak, ovo nije veliki problem u baznoj stanici.

U vremenskom domenu, LTE prenos je organizovan u ramove (frame) dužine 10 ms, pri čemu je svaki ram podeljen na 10 jednakih podramova trajanja 1 ms. Svaki podram sastoji se od dva slotova, trajanja 0.5 ms. Svaki slot sastoji se od 7 ili 6 OFDM simbol, uključujući i ciklički

prefiks. Broj OFDM simbola zavisi od toga da li se koristi normalan ili produženi ciklički prefiks. U slučaju da se koristi normalan ciklički prefiks, tj. 7 OFDM simbola, tada je dužina trajanja prvog OFDM simbola nešto duža od ostalih simbola, da bi se ispunio celi slot od 0.5 ms. Produženi ciklički prefiks može biti potreban u slučaju otežanih uslova u kanalu. Resurs element, najmanji fizički resurs u LTE, sastoji se od jednog OFDM nosioca tokom trajanja jednog OFDM simbola.

Resurs elementi grupisani su u resurs blokove, kao što je prikazano na slici 2. Svaki resurs blok sadrži 12 uzastopnih OFDM nosioca u frekventijskom domenu, odnosno jedan slot (trajanja 0.5 ms) u vremenskom domenu. Kako je razmak između OFDM nosioca 15 kHz, a minimalan broj OFDM nosioca koji se dodeljuju jednom korisniku je 12, minimalan opseg koji se može dodeliti jednom korisniku je $12 \cdot 15 \text{ kHz} = 180 \text{ kHz}$. Svaki resurs blok sadrži $7 \cdot 12 = 84$ resurs elementa, ili u slučaju produženog cikličkog prefiksa: $6 \cdot 12 = 72$ resurs elementa. Naravno, ne nose svi resurs elementi korisničke informacije, neki su namenjeni za prenos neophodnih informacija za funkcionisanje sistema, npr. postoje referentni simboli (ili pilot simboli) koji služe za procenu stanja potkanala.



Slika 2, Resurs blok u LTE [1]

LTE fizički sloj omogućava da nosilac može da sadrži različit broj resurs blokova, počevši od minimalnih 6 resurs blokova pa do 100 resurs blokova, što odgovara opsegu od 1.4 MHz do 20 MHz. Parametri različitih propusnih opsega prikazani su u tabeli 1.

Tabela1. Parametri različitih propusnih opsega u LTE [1]

Opseg (MHz)	1.4	3	5	10	15	20
Broj resurs blokova (N_{RB})	6	15	25	50	75	100
Broj podnosioca ($N_{SC} = 12 \cdot N_{RB}$)	72	180	300	600	900	1200

Iako je resurs blok definisan nad jednom slotom, osnovna jedinica koja se koristi pri dinamičkom raspoređivanju resursa je podram, odnosno dva međusobno povezana resurs bloka.

Kako je fleksibilnost spektra jedan od ključnih parametara, LTE podržava FDD (Frequency Division Duplex) i TDD (Time Division Duplex). Upravo primena OFDM to omogućava, jer je raspoloživ frekvencijski spektar podeljen na više potkanala, širine 180 kHz. Zbog toga LTE ne mora koristiti kontinualan spektar. Struktura vremen-

skog domena je slična za FDD i TDD, s razlikom da u TDD postoji namenski podram koji služi kao zaštitni interval između downlink i uplink veze. U slučaju FDD postoje dve frekvencije, jedna za uplink prenos (fUL) i jedna za downlink prenos (fDL). Tokom svakog rama postoji 10 uplink podramova i 10 downlink podramova, što znači da se uplink-downlink prenos odvija istovremeno. Izolacija između uplink i downlink prenosa ostvarena je uz pomoć tzv. dupleks filtera, kao i dovoljnim razmakom u frekvencijskom domenu. Treba napomenuti da iako se prenos uplink-downlink podataka u slučaju FDD odvija istovremeno, terminal mora podržavati full-duplex prenos za datu frekvenciju da bi se istovremeni prenos ostvario. Ukoliko terminal podržava samo half-duplex prenos istovremeni tok podataka nije moguć. Bazna stanica radi kao full-duplex, bez obzira na mogućnosti terminala.

U slučaju TDD prenosa postoji samo jedna frekvencija gde su tokovi podataka za uplink i downlink podeljeni u vremenu. Prelaz između dva toka podataka osigurani su posebnim podramom, koji je podeljen na tri dela: downlink deo (DwPTS), zaštitni interval (GP) i uplink deo (UpPTS). DwPTS se koristi kao klasičan podram, ali zbog njegovog kraćeg trajanja prenosi se manja količina podataka nego podramom normalne dužine. UpPTS ne koristi se za prenos podataka jer traje veoma kratko, ali može da se koristi za slučajni pristup ili da ostane prazan i u tom slučaju da služi kao dodatni zaštitni interval. Cilj svakog TDD sistema je da omogući dovoljno velik zaštitni interval koji omogućava bezbedan prelaz između uplink-downlink toka podataka [1].

5.1. Obrada na fizičkom sloju

Obrada na fizičkom sloju primenjuje se na DL-SCH, glavnom downlink prenosnom kanalu u LTE. Obrada uključuje i mapiranje fizičkih resursa na resurs elemente OFDM vremensko-frekvencijske strukture. U svakom TTI (Transmission Time Interval), koji odgovara jednom podramu trajanja 1 ms, dostavi se do dva transportna bloka fizičkom sloju, te se dalje prenose preko radio interfejsa. Broj transportnih blokova koji mogu da se prenesu u okviru jednog TTI zavisi od više-antenskog sistema. U slučaju da se ne koristi prostorno multipleksiranje, može da se pošalje najviše jedan transportni blok u okviru TTI. Ukoliko se koristi prostorno multipleksiranje, gde se prenos vrši preko više paralelnih slojeva do istog terminala, može da se pošalje do dva transportna bloka u okviru jednog TTI.

U prvom koraku obrade svakom transportnom bloku dodaje se CRC dužine 24 bita. CRC omogućava prijemoj strani da detektuje greške pri prenosu i time omogući hibridnom ARQ protokolu da zatraži retransmisiju pogrešno prenetih transportnih blokova. Ukoliko dužina transportnog bloka, uključujući i CRC, prevazilazi maksimalnu dužinu bloka koju može da obradi interleaver u sklopu turbo-kodera, vrši se segmentacija. Segmentacija predstavlja podelu transportnog bloka na više manjih blokova odgovarajuće dužine, ne veće od 6144 bita.

U toku segmentacije, svakom bloku dodaje se dodatni CRC, takođe dužine 24 bita, što omogućava ranu detekciju greške pri prenosu. Segmentacija se vrši samo nad transportnim blokovima velike dužine. Kodovanje u DL-SCH kanalu bazira se na turbo-koderu. Zadatak

RM+HARQ bloka je da iz blokova bita dostavljenih iz kodera izdvoji tačan set bitova koji su poslani u datom TTI podramu. Blokovi bita se zatim skrembluju, tj. vrši se xor operacija sa slučajnom sekvencom. Bez skremblovanja, dekodier u terminalu mogao bi pomešati originalni signal sa interferirajućim signalom.

Downlink skremblovanje vrši se nad svim transportnim kanalima kao i nad L1/L2 kontrolnim kanalima. U modulacionom bloku vrši se transformacija blokova skremblovanih bita u odgovarajuće blokove kompleksnih modulacionih simbola. Modulacione šeme koje se koriste u LTE su: QPSK, 16QAM, 64QAM, što odgovara 2, 4 i 6 bita, respektivno, po modulacionom simbolu. U poslednjem koraku vrši se antensko mapiranje modulacionih simbola na odgovarajuće portove [1].

6. OPENAIR INTERFACE

OpenAirInterface™ (OAI) je platforma otvorenog koda (open-source) koja nudi softversku implementaciju LTE sistema, tj. potpunu realizaciju 3GPP standarda. OAI je razvijen pri departmanu za mobilne komunikacije u sklopu EURECOM istraživačkog centra, s ciljem da omogućiti inovacije i razvoj u oblasti mobilnih/bežičnih komunikacija. Pisan je u standardnom C jeziku za nekoliko verzija Linux operativnih sistema, i optimizovan za rad sa Intel™ i ARM™ procesorima. OAI pruža bogato razvojno okruženje sa širokim spektrom ugrađenih alata. Omogućava izgradnju i uređivanje LTE bazne stanice

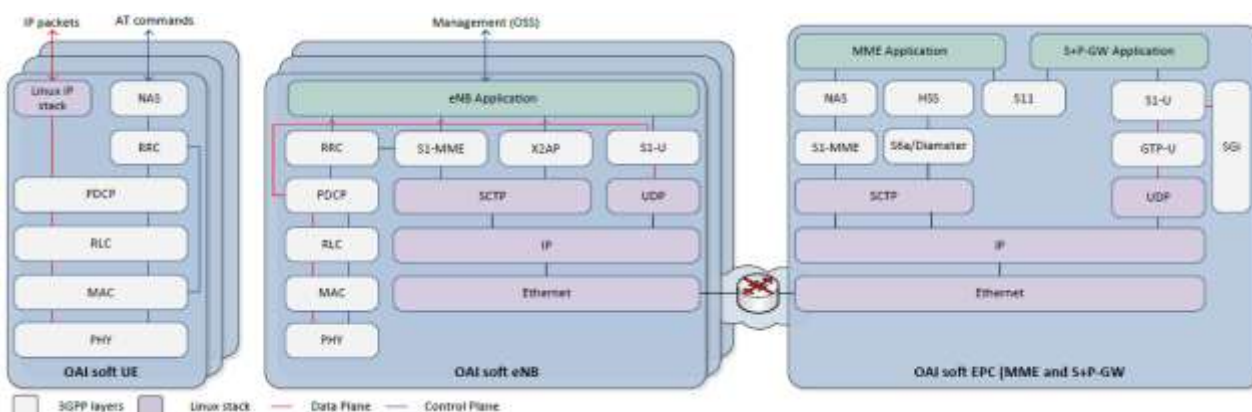
(OAI eNB), korisničke opreme (OAI UE) i jezgra mreže (OAI EPC). OAI bazna stanica može da se poveže sa komercijalnom korisničkom opremom ali i sa OAI UE radi testiranja različitih mrežnih postavki i konfiguracija, takođe omogućava nadgledanje mrežnih i mobilnih uređaja u realnom vremenu.

Trenutno, OAI platforma pruža potpunu softversku implementaciju četvrte generacije mobilnih ćelijskih sistema, u skladu sa 3GPP LTE standardom. Na slici 3. belom bojom šematski su prikazani svi slojevi protokola koji su implementirani u sklopu OAI. OAI platforma omogućava nekoliko različitih konfiguracija, uključujući korištenje komercijalnih uređaja:

- komercijalni UE + komercijalni eNB + OAI EPC,
- komercijalni UE + OAI eNB + komercijalni EPC,
- komercijalni UE + eNB + OAI EPC,
- OAI UE + OAI eNB + OAI EPC,
- OAI UE + OAI eNB + komercijalni EPC,
- OAI UE + komercijalni eNB + komercijalni EPC.

OAI je dizajniran da bude nezavisan od hardverske RF platforme. Lako je prilagodljiv komercijalnim SDR RF uređajima. OAI zvanično podržava:

- EURECOM™ EXMIMO2,
- USRP X-series/B-series [3].



Slika 3, OpenAirInterface LTE softver stek

7. ZAKLJUČAK

Mrežna arhitektura koja se koristi u LTE nudi mnoge prednosti u odnosu na prethodne mrežne arhitekture korištene u ćelijskim sistemima. Novom arhitekturom uvećan je kapacitet mreže, čime se postižu veće brzine prenosa podataka. Uspostavljena je All-IP arhitektura, što dovodi do jednostavnije mreže te smanjenja troškova. Smanjeno je vreme odziva na oko 10 ms (latency), što omogućava rad mnogih aplikacija koje zahtevaju brz odziv. S druge strane, OAI pruža mogućnost izvođenja različitih eksperimenata u sklopu LTE sistema. Platforme kao što je OAI pružaju ubrzan razvojni proces budućih tehnologija, naslednica LTE. Kroz jednostavnu izmenu softverskog koda, bez potrebe za menjanjem hardvera, moguće je izvršiti različite eksperimente, te uveliko smanjiti troškove i vreme potrebne za realizaciju jednog realnog sistema. Ovome doprinosi i korištenje SDR uređaja, univerzalne hardverske platforme.

8. LITERATURA

- [1] Erik Dahlman, Stefan Parkvall, Johan Sköld, "4G LTE/LTE-Advanced for Mobile Broadband", Academic Press, 2011.
- [2] Christopher Cox, "An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications", Wiley Publishing, 2012.
- [3] http://www.openairinterface.org/?page_id=864 (pristupljeno u avgustu 2018.)

Kratka biografija:



Srdan Šobot rođen je u Loznici 1993. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva - Telekomunikacioni sistemi odbranio je 2018.godine.

MATEMATIČKI MODEL TRANSFORMATORA SA NAMOTAJIMA SPREGNUTIM U SLOMLJENU ZVEZDU**MATHEMATICAL MODEL OF TRANSFORMER WITH ZIGZAG CONNECTED WINDINGS**Ognjen Kundačina, Marko Obrenić, Predrag Vidović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U okviru rada objašnjen je način modelovanja transformatora sa namotajima spregnutim u slomljenu zvezdu za potrebe osnovnih proračuna u elektroenergetskim sistemima. U teorijskom delu rada izvedene su generalne ekvivalentne šeme transformatora sprege Yz1, Yz5 i Yz9, a zatim su izvedene i konkretne ekvivalentne šeme transformatora sa spregom Yz5. U praktičnom delu rada izvršena je verifikacija izvedenih modela vršeći proračune tokova snaga za jednostavne test mreže.

Ključne reči: Transformatori, Konekcije namotaja transformatora, Slomljena zvezda

Abstract – This paper presents the method for modeling a power transformer with zigzag connected windings for the purpose of basic calculations in electrical power systems. In the theoretical part of the work, general equivalent schemes for Yz1, Yz5 and Yz9 connected transformers are derived, and then specific equivalent schemes for Yz5 transformer connection are derived. In the practical part of the work, verification of derived models is performed using the Load Flow calculations on simple test networks.

Keywords: Transformers, Transformer winding connections, Zigzag

1. UVOD

Transformatori su statički elektroenergetski uređaji pomoću kojih se vrši unutrašnja transformacija električne energije (u električnu energiju) [1]. Pri tome se menjaju napon i struja, sa težnjom da se snaga (energija) i napon kvalitativno ne degradiraju. Budući da su transformatori osnovni elementi elektroenergetskog sistema (EES), bitan zadatak je odrediti njihov matematički model za potrebe osnovnih proračuna u EES-u poput tokova snaga i kratkih spojeva.

Cilj ovog rada jeste da se izvedu ekvivalentne šeme trofaznih transformatora sa niženaponskim namotajem spregnutim u slomljenu zvezdu, u stacionarnom režimu, kao i da se iste numerički verifikuju na prostoj test mreži. Teorijska izvođenja koja se nalaze u ovom radu zasnovana su na neobjavljenom radu navedenom u pregledu literature [2]. Prema tome namotaj transformatora može biti spregnut u trougao, zvezdu i slomljenu zvezdu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Predrag Vidović.

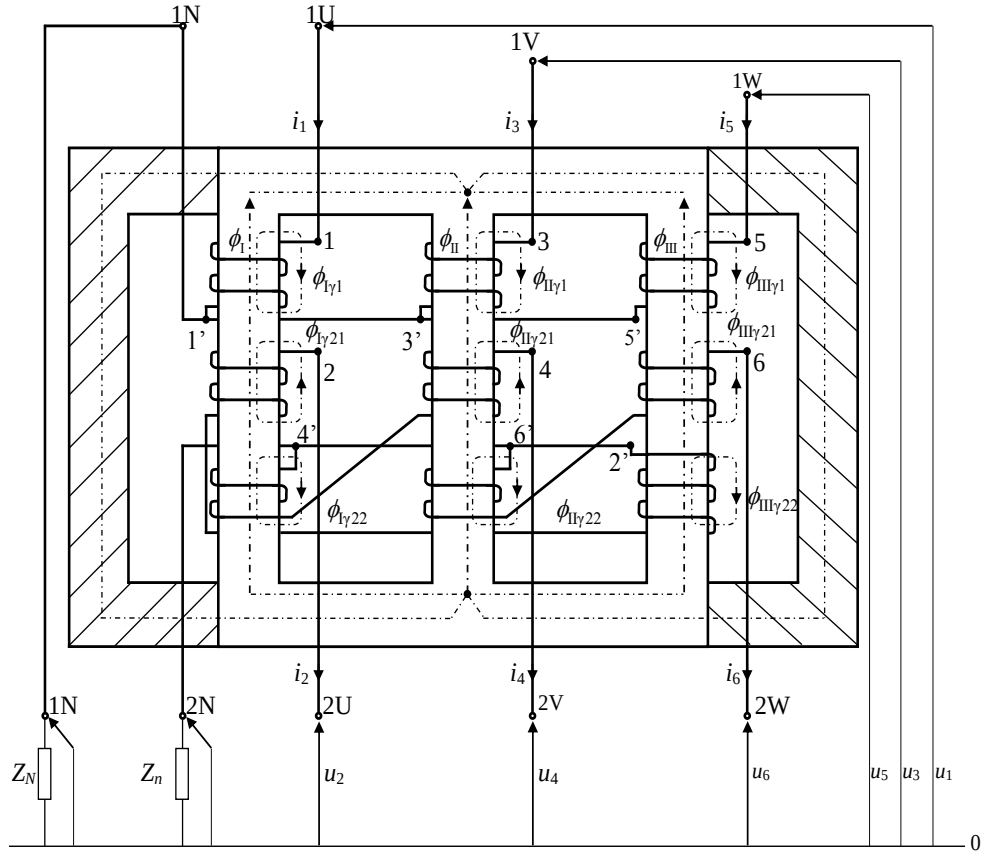
U trećoj glavi se izveden matematički model verifikuje pomoću računarskog programa napisanog u programskom jeziku C++, koji vrši proračun tokova snaga metodom sumiranja struja na jednostavnim test mrežama [3].

2. MATEMATIČKI MODEL TRANSFORMATORA SA SPREGOM Yz1, Yz5 i Yz9

U ovoj glavi su prvo izvedene ekvivalentne šeme transformatora čiji je primarni namotaj spregnut u zvezdu, a sekundarni u slomljenu zvezdu, na taj način da sprežni broj iznosi 1, 5 ili 9. Zatim su izvedene konkretne ekvivalentne šeme za transformator sa spregom Yz5.

2.1. Sprege Yz1, Yz5 i Yz9

U svrhu izvođenja ekvivalentnih šema razmatra se trofazni (trostubni, četvorostubni ili) petostubni, uravnotežen transformator prikazan na slici 1. On je doveden u stacionaran režim. Brojevi navojaka (N) i otpornosti namotaja (R) svakog od trojke namotaja na istoj strani transformatora jednaki su međusobno, pa nisu označeni uz svaki namotaj. Treba napomenuti da se niženaponski (NN) namotaj, koji je spregnut u slomljenu zvezdu, sastoji od dva polunamotaja smeštena na različite stubove. Transformator se sastoji od tri namotaja po fazi na višenaponskoj (VN) strani: 1-1', 3-3' i 5-5', i šest namotaja na NN strani (po dva električno povezana): 2-2', 4-4' i 6-6', koji su međusobno magnetski spregnuti. Električne veličine koje karakterišu režim transformatora su: naponi u_1, u_2, u_3, u_4, u_5 i u_6 i struje i_1, i_2, i_3, i_4, i_5 i i_6 . U magnetske veličine spadaju: fluksevi rasipanja $\phi_{IY1}, \phi_{IY21}, \phi_{IY22}, \phi_{IY1}, \phi_{IY21}, \phi_{IY22}, \phi_{IIY1}, \phi_{IIY21}, \phi_{IIY22}, \phi_{IIY1}, \phi_{IIY21}, \phi_{IIY22}$ i zajednički fluksevi po svakom stubu transformatora ϕ_I, ϕ_{II} i ϕ_{III} , kao i njihov zbir ϕ . Nesimetričan režim razmatranog uravnoteženog transformatora se može prikazati s tri jednofazna reprezentata – direktnog (d), inverznog (i) i nultog redosleda (o). Svakom od njih se asocira po jedna pogonska (jednofazna) ekvivalentna šema, koje su međusobno raspregnute. Svaka od njih se može izvesti razmatranjem trofaznog transformatora u simetričnom režimu odgovarajućeg redosleda. To se može učiniti "nametanjem" trofaznih simetričnih napona (izabrane vrste simetrije) na oba kraja transformatora, čime je i trofazni režim uravnoteženog transformatora simetričan iste vrste simetrije kao i pobuda. U simetričnim režimima direktnog i inverznog redosleda, suma flukseva prvog, drugog i trećeg stuba transformatora jednaka je nuli. Međutim, u simetričnom režimu nultog redosleda, fluksevi tih stubova međusobno su u fazi, pa je njihov zbir jednak trostrukoj vrednosti fluksa jednog stuba.



Slika 1. Principalska šema trofaznog, dvonamotajnog transformatora sprege Yz1, Yz5, Yz9

Matematički opis režima tri namotaja, u simetričnom režimu sastoji se od dve relacije saglasne s naponskim Kirhofovom zakonom (drugi i treći namotaj su električno spojeni pa čine jednu konturu), četiri relacije za četiri magnetne konture, kao i od jedne relacije s kojom se izražava kontinuitet flukseva u transformatoru:

$$\begin{aligned}
 u_1 &= R_1 i_1 + N_1 \frac{d\phi_{IY1}}{dt} + N_1 \frac{d\phi_I}{dt}, \\
 u_2 &= -\frac{R_2}{2} i_2 - \frac{R_2}{2} i_2 + \frac{N_2}{2} \frac{d\phi_I}{dt} - \frac{N_2}{2} \frac{d\phi_{III}}{dt} \\
 &\quad - \frac{N_2}{2} \frac{d\phi_{IY21}}{dt} - \frac{N_2}{2} \frac{d\phi_{IIY22}}{dt}, \\
 N_1 i_1 - \frac{N_2}{2} i_2 + \frac{N_2}{2} i_4 &= \mathfrak{R}_s \phi_I + \alpha_{Fe} \frac{d\phi_I}{dt} + \mathfrak{R}_u \phi + \alpha_u \frac{d\phi}{dt}, \quad (1) \\
 N_1 i_1 &= \mathfrak{R}_{IY1} \phi_{IY1}, \\
 \frac{N_2}{2} i_2 &= \mathfrak{R}_{IY21} \phi_{IY21}, \\
 \frac{N_2}{2} i_4 &= \mathfrak{R}_{IY22} \phi_{IY22}, \\
 \phi &= \phi_I + \phi_{II} + \phi_{III},
 \end{aligned}$$

gde su $\mathfrak{R}_\gamma$ magnetni otpori kontura rasipnih flukseva, $\mathfrak{R}_s$ magnetni otpori stubova, $\mathfrak{R}_u$ magnetni otpor "povratnog puta" fluksa (četvrtog i petog stuba, odnosno okoline), α_{Fe} magnetna karakteristika gvožđa kojom se modeluje

aktivna snaga gubitaka u gvožđu (magnetnom kolu) i α_u magnetna karakteristika sredine. Prelaskom iz vremenskog u kompleksni domen, zamenom proizvoda tipa $\omega N^2 / \mathfrak{R}_\gamma$ dobijenih uvrštavanjem izraza za flukseve rasipanja u električni deo modela za par namotaja na istom stubu, dobija se:

$$\begin{aligned}
 \hat{U}_1 &= (R_1 + jX_{\gamma1}) \hat{I}_1 + N_1 \hat{E}_I, \\
 \hat{U}_2 &= -(R_2 + jX_{\gamma2}) \hat{I}_2 + \frac{N_2}{2} \hat{E}_I - \frac{N_2}{2} \hat{E}_{III}, \\
 \hat{I}_m^1 &= \hat{I}_1 - \frac{N_2}{2N_1} \hat{I}_2 + \frac{N_2}{2N_1} \hat{I}_4, \\
 N_1 \hat{E}_I &= \hat{Z}_m^1 \hat{I}_m^1,
 \end{aligned} \quad (2)$$

gde je $\hat{Z}_m^1$ impedansa magnećenja svedena na stranu 1, za namotaje na istom stubu:

$$\hat{Z}_m^1 = \frac{\omega^2 (N_1)^2 \alpha_m}{(\mathfrak{R}_m)^2 + \omega^2 (\alpha_m)^2} + j \frac{\omega (N_1)^2 \mathfrak{R}_m}{(\mathfrak{R}_m)^2 + \omega^2 (\alpha_m)^2} = R_m^1 + jX_m^1. \quad (3)$$

Shodno vezama između elektromotornih sila na različitim stubovima transformatora i struja namotaja, model transformatora za simetrične režime direktnog, inverznog i nultog redosleda predstavljen je relacijama 4, 5 i 6. Ekvivalentna šema, kojom se opisuje režim namotaja trofaznog transformatora, ima oblik prikazan na slici 2, a oznake korištene na slici su definisane relacijama 7.

$$\begin{aligned}\hat{U}_1^d &= (R_1 + jX_{\gamma 1})\hat{I}_1^d + N_1\hat{E}_1^d, \\ \hat{U}_2^d &= -(R_2 + jX_{\gamma 2})\hat{I}_2^d + \sqrt{3}\frac{N_2}{2}e^{-j\pi/6}\hat{E}_1^d,\end{aligned}\quad (4)$$

$$\hat{I}_m^{1d} = \hat{I}_1^d - \sqrt{3}\frac{N_2}{2N_1}e^{j\pi/6}\hat{I}_2^d,$$

$$N_1\hat{E}_1^d = \hat{Z}_m^{1d}\hat{I}_m^{1d}.$$

$$\begin{aligned}\hat{U}_1^i &= (R_1 + jX_{\gamma 1})\hat{I}_1^i + N_1\hat{E}_1^i, \\ \hat{U}_2^i &= -(R_2 + jX_{\gamma 2})\hat{I}_2^i + \sqrt{3}\frac{N_2}{2}e^{j\pi/6}\hat{E}_1^i,\end{aligned}\quad (5)$$

$$\hat{I}_m^{1i} = \hat{I}_1^i - \sqrt{3}\frac{N_2}{2N_1}e^{-j\pi/6}\hat{I}_2^i,$$

$$N_1\hat{E}_1^i = \hat{Z}_m^{1i}\hat{I}_m^{1i}.$$

$$\begin{aligned}\hat{U}_1^o &= (R_1 + jX_{\gamma 1})\hat{I}_1^o + N_1\hat{E}_1^o, \\ \hat{U}_2^o &= -(R_2 + jX_{\gamma 2})\hat{I}_2^o,\end{aligned}\quad (6)$$

$$\hat{I}_m^{1o} = \hat{I}_1^o,$$

$$N_1\hat{E}_1^o = \hat{Z}_m^{1o}\hat{I}_m^{1o}.$$

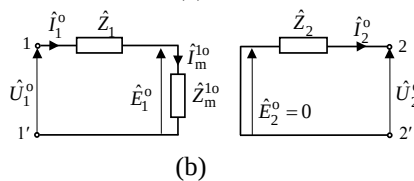
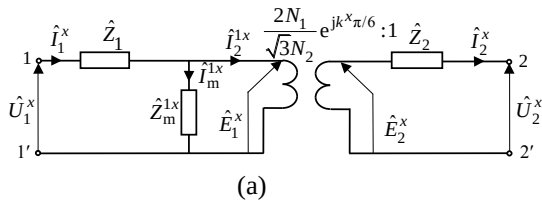
$$\hat{Z}_1 = R_1 + jX_{\gamma 1},$$

$$\hat{Z}_2 = R_2 + jX_{\gamma 2},$$

$$N_1\hat{E}_1^x = \hat{E}_1^x, \quad x = d, i, o,$$

$$\sqrt{3}\frac{N_2}{2}e^{-jk^x\pi/6}\hat{E}_1^x = \hat{E}_2^x, \quad x = d, i, \quad (7)$$

$$\hat{I}_m^{1x} = \hat{I}_1^x - \sqrt{3}\frac{N_2}{2N_1}e^{jk^x\pi/6}\hat{I}_2^x, \quad x = d, i.$$

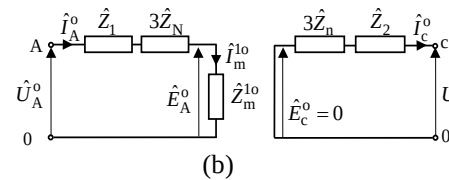
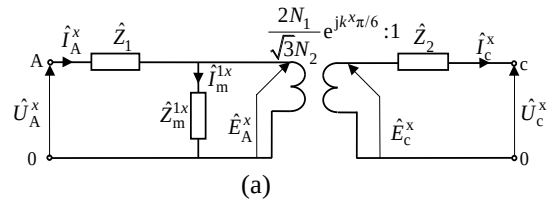


Slika 2. Ekvivalentna šema trofaznog transformatora sprege Yz1, Yz 5 i Yz 9; a – za simetričan režim direktnog ($x=d$), inverznog ($x=i$); b – nultog redosleda

2.2. Sprega Yz5

Da bi se realizovala sprega Yz5, transformator sa slike 1 (njegovi priključci – 1U, 1V, 1W i 1N, odnosno 2W, 2U, 2V i 2N) se priključuju na elektroenergetski sistem na faze A, B i C, odnosno c, a i b (VN i NN strana, respektivno). Transformator se nalazi u okviru elektroenergetskog sistema, da bi se sagledao i s aspekta simetričnih režima nultog redosleda. U tim režimima struje po fazama čine konture kojima su obuhvaćene i veze transformatora sa zemljom, kao i sama zemlja. Neka se razmatrani transformator nalazi u simetričnom režimu

bilo koje vrste simetrije „x“. Tada su režimi svakog stuba, sa svojim parom namotaja, isti, sa odgovarajućim pomerajem u vremenu (zavisnim od vrste simetrije razmatranog režima). Kod ovog transformatora između tačaka 1-1' priključen je napon faze A, a između tačaka 2-2' napon faze c. Pošto je suma faznih struja impedansi uzemljenja, na VN i NN strani transformatora, u simetričnim režimima direktnog i inverznog redosleda jednaka nuli i naponi, $\hat{U}_{0N}^x$ i $\hat{U}_{0n}^x$, $x = d, i$, takođe su jednaki nuli, te tačke N, n i 0 predstavljaju istu tačku. Tako se šema razmatranog transformatora, za simetrične režime direktnog ($x=d$) i inverznog redosleda ($x=i$), može prikazati kao na slici 3.



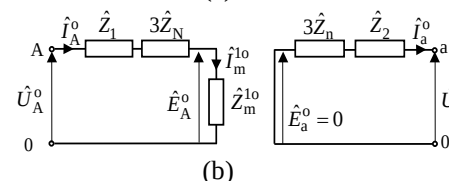
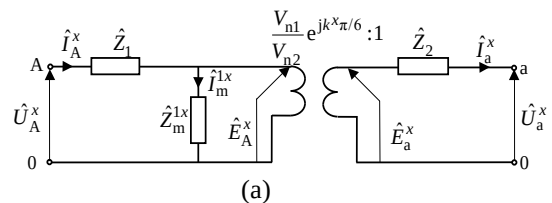
Slika 3. Ekvivalentna šema trofaznog transformatora sprege Yz5; a – za simetričan režim direktnog ($x=d$; $k^d=5$), inverznog ($x=i$; $k^i=7$); b – nultog redosleda

Za simetrične režime direktnog, inverznog i nultog redosleda važe sledeće relacije:

$$\begin{aligned}\hat{E}_c^d &= \hat{E}_a^d e^{j2\pi/3}, \\ \hat{U}_c^d &= \hat{U}_a^d e^{j2\pi/3}, \\ \hat{I}_c^d &= \hat{I}_a^d e^{j2\pi/3}.\end{aligned}\quad (8)$$

$$\begin{aligned}\hat{E}_c^i &= \hat{E}_a^i e^{j4\pi/3}, \\ \hat{U}_c^i &= \hat{U}_a^i e^{j4\pi/3}, \\ \hat{I}_c^i &= \hat{I}_a^i e^{j4\pi/3}.\end{aligned}\quad (9)$$

$$\begin{aligned}\hat{E}_c^o &= \hat{E}_a^o, \\ \hat{U}_c^o &= \hat{U}_a^o, \\ \hat{I}_c^o &= \hat{I}_a^o.\end{aligned}\quad (10)$$

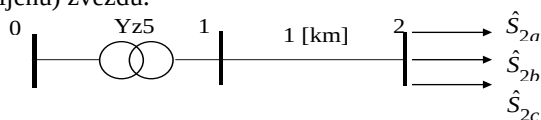


Slika 4. Pogonska šema trofaznog transformatora sprege Yz5; a – za simetričan režim direktnog ($x=d$; $k^d=5$), inverznog ($x=i$; $k^i=7$); b – nultog redosleda

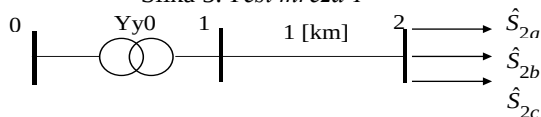
Relacije 8, 9 i 10 omogućavaju da se ekvivalentne šeme razmatranog trofaznog transformatora, za simetričan režim direktnog, inverznog i nultog redosleda, prikažu kao na slici 4. Šeme na slici 4a i b, definitivno jesu pogonske. Njima su povezani samo fazni naponi i struje iste faze sa obe strane transformatora.

3. NUMERIČKA VERIFIKACIJA IZVEDENOG MATEMATIČKOG MODELA

U okviru ove glave je na primeru jednostavnih test mreža verifikovan matematički model transformatora sa namotajima spregnutim u slomljenu zvezdu. Model se verifikuje proračunom tokova snaga u domenu simetričnih komponenti i relativnih vrednosti. Test mreže, koje su prikazane na slikama 5 i 6, se razlikuju samo po sprezi NN namotaja transformatora. U prvoj test mreži NN namotaj transformatora je spregnut u (direktno uzemljenu) slomljenu zvezdu, a u drugoj u (direktno uzemljenu) zvezdu.



Slika 5. Test mreža 1



Slika 6. Test mreža 2

Potrošnja u čvoru 2 za obe test mreže iznosi: $\hat{S}_{2a} = (30 - j15)$ kVA, $\hat{S}_{2b} = (20 - j10)$ kVA, $\hat{S}_{2c} = (10 - j5)$ kVA. Naponi čvora 0 iznose: $\hat{U}^d = 5.774e^{j0.0^\circ}$ kV, $\hat{U}^i = 1.020e^{j0.0^\circ}$ kV, $\hat{U}^o = 1.020e^{j0.0^\circ}$ kV. Simetrične komponente napona čvorova test mreža 1 i 2, u domenu relativnih vrednosti, prikazane su u tabelama 1 i 2.

Tabela 1. Naponi čvorova test mreže 1

Čvor	$\hat{U}^d$	$\hat{U}^i$	$\hat{U}^o$
0	$1.000e^{j0.0^\circ}$	$0.177e^{j0.0^\circ}$	$0.177e^{j0.0^\circ}$
1	$0.985e^{-j150.609^\circ}$	$0.180e^{j148.972^\circ}$	$0.003e^{j3.447^\circ}$
2	$0.970e^{-j150.852^\circ}$	$0.182e^{j147.816^\circ}$	$0.020e^{-j12.855^\circ}$

Tabela 2. Naponi čvorova test mreže 2

Čvor	$\hat{U}^d$	$\hat{U}^i$	$\hat{U}^o$
0	$1.000e^{j0.0^\circ}$	$0.177e^{j0.0^\circ}$	$0.177e^{j0.0^\circ}$
1	$0.986e^{-j0.626^\circ}$	$0.178e^{-j1.150^\circ}$	$0.174e^{j0.670^\circ}$
2	$0.971e^{-j0.898^\circ}$	$0.178e^{-j2.146^\circ}$	$0.169e^{j3.361^\circ}$

Nulta komponenta napona na primaru transformatora sa spregom Yz se ne prenosi na sekundar transformatora, za razliku od transformatora sa spregom Yy. U tabelama 3 i 4 prikazane su vrednosti struja grana test mreža 1 i 2, takođe izraženih u domenu simetričnih komponenti i relativnih vrednosti. Nulta komponenta struje koja teče kroz sekundarni namotaj transformatora sa spregom Yz se ne prenosi na primarni namotaj transformatora, za razliku od transformatora sa spregom Yy.

Tabela 3. Struje grana test mreže 1

Grana	$\hat{I}^d$	$\hat{I}^i$	$\hat{I}^o$
0-1	$0.234e^{-j30.259^\circ}$	$0.059e^{j70.792^\circ}$	$0.001e^{-j80.401^\circ}$
1-2	$0.234e^{-j179.740^\circ}$	$0.059e^{-j139.208^\circ}$	$0.078e^{j119.232^\circ}$

Tabela 4. Struje grana test mreže 2

Grana	$\hat{I}^d$	$\hat{I}^i$	$\hat{I}^o$
0-1	$0.226e^{-j27.651^\circ}$	$0.046e^{j42.537^\circ}$	$0.043e^{-j100.106^\circ}$
1-2	$0.226e^{-j27.651^\circ}$	$0.046e^{j42.537^\circ}$	$0.043e^{-j100.106^\circ}$

4. ZAKLJUČAK

U okviru ovog rada prikazan je model transformatora sa NN namotajem spregnutim u slomljenu zvezdu. U rezultatima verifikacionog proračuna tokova snaga za dve test mreže ističu se razlike u simetričnim režimima nultog redosleda dva transformatora, sprega Yz5 i Yy0. Kao posledica prekida u pogonskoj šemi za nulti redosled kod transformatora sprega Yz5, nulta komponenta napona se ne prenosi sa VN na NN stranu transformatora, a nulta komponenta struje se ne prenosi sa NN na VN stranu, što nije slučaj kod transformatora sprega Yy0.

5. LITERATURA

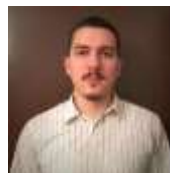
- [1] V.C.Strezoski, S.R.Milakovic, "Ekvivalentne šeme elektroenergetskih transformatora u analizi stacionarnih režima", Srbija, Novi Sad, Fakultet tehničkih nauka Novi Sad, EPS-JP "Elektrovojvodina", MP "STYLOS", 1998.
- [2] P.Vidović, "Transformatori sa niskonaponskim namotajima povezanim u slomljenu zvezdu", skripta, Fakultet tehničkih nauka, Novi Sad, 2018.
- [3] D. Popović, D. Bekut, V. Treskanica, "Specijalizovani DMS algoritmi", Srbija: Novi Sad, DMS Grupa DOO za elektroenergetski inženjering, 2004.

Kratka biografija:



Ognjen Kundačina rođen je u Nevesinju, Bosna i Hercegovina, 1994. god. Osnovne studije završio je na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi 2017. god. i 2017. god. upisao master studije na istom fakultetu.

kontakt: ognjenkundacina94@gmail.com



Marko Obrenić rođen je u Beogradu, Srbija, 1990. god. Osnovne i master studije završio je na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi 2013. i 2014. god. i 2014. god. upisao doktorske studije na istom fakultetu.



Predrag Vidović rođen u Vlasenici, Bosna i Hercegovina, 1981. god. Diplomirao je, magistrirao i doktorirao na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi 2005, 2008. i 2015. god, respektivno.

**RAZVOJ DISTRIBUIRANE, VISOKO SKALABILNE APLIKACIJE
UZ MICROSOFT ORLEANS****DEVELOPMENT OF DISTRIBUTED, HIGHLY SCALABLE APPLICATION WITH
MICROSOFT ORLEANS**Dušan Jeftić, *Fakultet tehničkih nauka, Novi Sad***Oblast – Primenjeno softversko inženjerstvo**

Kratak sadržaj – Pružen je opis kako uz pomoć jedinstvenog pristupa Microsoft Orleans radnog okvira da se na potpuno novi način modeluje distribuirana, visoko skalabilna aplikacija. Ovo je postignuto objašnjenjem glavnih koncepata ovog radnog okvira.

Ključne reči: radni okvir Microsoft Orleans, distribuirana aplikacija, skalabilnost, virtuelni akter model, pouzdanost

Abstract – A description of how to use Microsoft's framework Orleans, with its unique approach, to model a distributed, highly scalable application is given. This was provided by explaining key features and concepts of this framework.

Keywords: Microsoft Orleans framework, distributed application, scalability, virtual actor model, reliability

1. UVOD

Distribuirane aplikacije su u današnje vreme od ključnog značaja, usled naglog razvoja tehnologije. Zajedno sa razvojem tehnologije javlja se i ogromna potreba za podrškom velikog broja korisnika. Sistemi koji nisu distribuirani, teško mogu da pruže podršku od na primer milion korisnika. Distribuirani sistemi predstavljaju mrežu autonomnih, računarskih sistema, koji međusobno razmenjuju informacije i resurse od značaja, kako bi se omogućilo odražavanje velikih sistema [2]. Ovi sistemi pružaju mogućnost skalabilnosti, pa je takvo rešenje lako proširiti dodavanjem novih komponenti. Kako uvek postoji više mašina koje mogu da rade isti posao, ukoliko je jedna u nemogućnosti da izvršava potreban posao, lako se dolazi do zamene [1]. Sve prethodno navedeno dovodi do pozitivnog efekta pouzdanosti sistema i poboljšanju performansi.

**2. MICROSOFT ORLEANS I NJEGOV PRISTUP
MODELIRANJU DISTRIBUIRANE APLIKACIJE**

Microsoft Orleans predstavlja radni okvir sa jedinstvenim konceptom čija je svrha da omogući lakše i efikasnije pravljenje skalabilnih, pouzdanih i održivih distribuiranih aplikacija [3]. Kreiran je 2011. od strane Microsoftove istraživačke organizacije za korišćenje u oblaku (*cloud*)

NAPOMENA:

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

sistemima [3]. Ono što je dobra strana Orleansa jeste što se lako dolazi do željenih rezultata, bez potrebnog znanja o pisanju konkurentnih i skalabilnih programa. Ovaj radni okvir je javno dostupan za uređivanje od Januara 2015. godine, što znači da je javnosti dostupan za javne kolaboracije, ispravke i dopune od strane svih zainteresovanih [4].

U toku prošlosti postojalo je više pokušaja da se napravi adekvatan model distribuirane aplikacije. Svaki od pristupa iz prošlosti su uspevali da reše problem prethodnog, ali su kao posledicu stvorili novi.

Orleans pokušava da objedini pozitivne stvari starih modela i da uvede sopstvene novine uvođenjem modela virtuelnog aktera [3]. Ono što Orleansov pristup čini jedinstvenim, je što srednji sloj, koji se nalazi između krajnjeg korisnika i skladišta, koji će činiti mnoštvo nezavisnih, jedinstvenih objekata, koji postoje u svakom trenutku. Ovakvi objekti iako izolovani, u mogućnosti su da komuniciraju međusobno, a njihovom izolovanosti od strane programera, ne mora se voditi računa o tome da li su objekti na istim ili različitim serverima [3]. Takođe je od značaja napomenuti da se nad ovakvim objektima radi asinhrono.



Slika 1. Isti primer slike sa manjim dimenzijama

Ovaj pristup takođe može da se zamisli kao gomilu malih specifikiranih radnika, gde svaki radnik ima svoja zaduženja i koja je u mogućnosti da izvršava zavisno ili nezavisno od drugih radnika. Virtuelni akteri predstavljaju one činioce sistema, koji su svakog momenta postojani u memoriji, a predstavljaju podskup svih ikada kreiranih učesnika [4]. Sa stanovišta programera, na ovaj način više nije na njemu da vodi računa o životnom veku i fizičkoj aktivaciji aktera, već to radi sam Orleans automatski, zahvaljujući ovom vidu apstrakcije aktera [4]. Akteri predstavljaju jedinice izolacije i distribucije u samom Orleansu. Svaki akter ima 128-bitni jedinstveni GUID, koji biva kreiran na osnovu

njegovog tipa i primarnog ključa [10]. Kako su akteri izolovani, oni međusobno ne mogu da dele memoriju, odakle sledi da jedini način na koji dva aktera mogu da komuniciraju je putem poruka [10].

3. ZRNA

Zrno predstavlja glavnu egzekucionu jedinicu radnog okvira Orleans, i ono je zapravo predstavnik samog virtuelnog aktera. Dva zrna međusobno mogu da interaguju tako što će pozivati metode zrna sa kojim žele da ostvare komunikaciju [6]. Takođe je moguće da skladište reference ka drugim zrnima [6]. Jedan od najvećih izazova je obezbediti konzistentnost podataka unutar jednog velikog sistema sa velikim brojem korisnika.

Ovo se postiže zabranom deljenja podataka između zrna, osim putem poruka [6]. Takođe, Orleans garantuje da će svaka akcija nad zrnom uvek biti izvršana nekonkurentno [6].

3.1. Kreiranje zrna

Identitet i funkcionalnost svakog zrna postiže se kroz interfejs, koje će zrno da implementira [7]. Kao što je prethodno pomenuto, jedno zrno će komunicirati sa drugim zrnima ukoliko pozove metode drugog zrna, koje su mu od koristi. Odatle sledi, da je za kreiranje zrna potrebno kreirati klasu zrna, i implementirati interfejs, koji će obuhvatati željeno ponašanje [7].

Pre nego što budu objašnjeni sami detalji pravljenja interfejsa zrna, bitno je objasniti šta je to klasa *Task*, koja pripada Microsoftovom .NET radnom okviru. Klasa *Task* predstavlja izolovanu operaciju, koja će se izvršavati asinhrono, nezavisno od glavne niti [7]. Ova klasa zbog svog asinhronog ponašanja, omogućava pristup stanja određenog zadatka (*Task*), kako bi se videlo da li je stanje otkazano, završeno ili se desila greška [7]. Ono što je bitno napomenuti, jeste da sve metode interfejsa, koje će zrno da implementira, moraju da vraćaju objekat tipa *Task* za sve metode koje nemaju povratni tip vrednosti [7]. Za metode koje imaju proizvoljan povratni tip *T*, očekuje se da vraćaju objekat tipa *Task<T>* [7]. Slika 2 pokazuje primer jednog zrna koji predstavlja mušteriju.

```
public interface ICustomerGrain : IGrainWithGuidKey
{
    Task Login(IShopGrain shop);
    Task Logout(IShopGrain shop);
    Task BuyProduct(IShopGrain shop, IItemGrain item);
    Task<IShopGrain> GetCurrentShop();
    Task<IItemGrain> GetCurrentItem();
}
```

Slika 2. Interfejs mušterije zrna

Nakon što je kreiran željeni interfejs ponašanja, sledeći korak je da se kreira klasa koja će da implementira ovaj interfejs. Pored toga što će ova klasa morati da nasledi prethodno kreiran interfejs, ona će takođe morati da nasledi osnovnu klasu zrna, koja se zove *Grain*. Posle toga ove iste metode je potrebno implementirati.

3.2. Životni ciklus zrna

Kao i svaki objekti u objektno orijentisanom programiranju, tako i zrna imaju svoj životni vek. Svako zrno prolazi kroz 4 stanja unutar svog životnog ciklusa.

Ova 4 stanja su: aktivno u memoriji, deaktivirano, perzistirano i aktivirano [6]. Zrno prolazi kroz sva 4 stanja u toku jednog ciklusa i to prelazi iz jednog stanja u drugo, redom koji je naveden prethodno a prikazan na slici 3.



Slika 3. Životni ciklus zrna

Sva zrna žive u egzekucionim kontejnerima koji se nazivaju silosi i predstavljaju Orleansove servere. Ovi silosi formiraju klastere, kombinujući mnoštvo fizičkih ili virtualnih mašina i na taj način pružaju skalabilnost i stabilnost distribuirane aplikacije [6]. Kada dođe do određenog zahteva, potrebno je aktivirati zrno. Da bi se to obavilo, zrno se prvo traži na nekom od silosa unutar klastera, ukoliko potraga bude neuspešna, napraviće se nova instanca tog zrna, koja će posle biti postojana na nekom od silosa [6].

Dokle god to zrno obavlja određene zahteve, ili sam šalje svoje zahteve drugim zrnima, to zrno će biti aktivno u memoriji, i biće spremno za obradu [6]. Ukoliko nakon dužeg, unapred konfigurisanog vremena, neko zrno ne bude niti slalo, niti primalo, niti obavljalo ikakve zadatke, pokreće se proces deaktiviranja zrna, gde se oslobađa memorija koju je to zrno zauzelo, a nakon toga zrno prelazi u perzistentno stanje [6].

Ovo perzistentno stanje, znači da će instance tog zrna postojati negde, na nekom od silosa, ali što se tiče same programerske strane, može se uvek smatrati kao da je zrno uvek aktivno i spremno za korišćenje, jer ostale stvari Orleans radi sam u pozadini [6].

4. KLIJENT

Klijentom se smatra deo sistema, koji će obavljati interakciju sa zrnima radi ostvarivanja određenih ciljeva, ali sam klijent nije deo logike zrna [8]. Ono što se misli pod time, da taj deo sistema nije deo logike zrna, jeste da se kod klijenta neće izvršavati na silosima, kao što je to slučaj kod zrna [8]. Klijent je uglavnom web servis, koji može biti napisan u proizvoljnoj tehnologiji, ali tako da je sposoban da se konektuje na Orleansov servis tj. silose, kako bi bio u mogućnosti da ostvaruje komunikaciju sa željenim zrnima.

4.1. Inicijalizacija klijenta

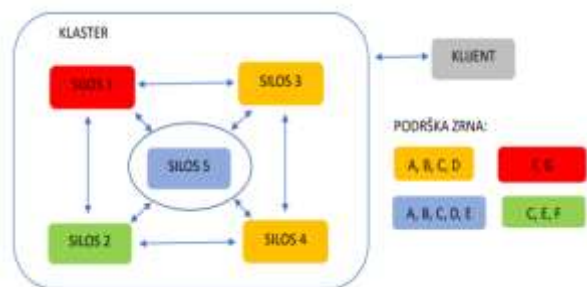
Kao što je prethodno spomenuto, neophodno je ostvariti komunikaciju između klijentskog i serverskog dela, kako bi klijent bio u stanju da obavlja interakciju sa potrebnim zrnima sistema. Konfiguracija klijenta se vrši pomoću

objekta `ClientConfiguration`, koji sadrži mnoštvo podešavanja, koji omogućavaju pravilnu konfiguraciju klijenta [8].

Prilikom razvoja aplikacije, smatraće se da će se aplikacija izvršavati na toj mašini pa će zbog toga koristiti localhost za pokretanje silosa. Nakon toga, potrebno je izgraditi `IClusterClient` objekat pomoću prethodno kreirane konfiguracije i klase `ClientBuilder` [8]. Kada je sve prethodno odrađeno, potrebno je konektovati klijenta sa klasterom, i takođe je potrebno sačekati potvrdu izvršavanja ovog zadatka.

5. KLASITERI I SILOSI

Sva zrna žive i egzistiraju unutar silosa, a mnoštvo silosa, čini jedan klaster. Sami silosi su heterogeni, i mogu podržavati više različitih tipova zrna, ali je onda neophodno naglasiti koja zrna želimo da smestimo na koji silos [9]. Svi tipovi zrna koja se definišu na silosima, zapravo definišu i koje tipove zrna sam klaster podržava.



Slika 4. Heterogeni silosi

Iz priloženog primera može da se vidi da tipovi zrna A i B mogu da se nalaze na silosima 3, 4 i 2. Tipovi zrna C mogu da se nađu na silosima 2, 3, 4 i 5. Tipovi zrna D mogu da se nađu na silosima 3, 4 i 5. Tipovi zrna E mogu da se nađu na silosima 2 i 5. Tipovi zrna F mogu da se nađu na silosima 1 i 2. Na kraju, tipovi zrna G mogu da se nađu samo na klasteru 1. Ono što je ovde bitno napomenuti, jeste da definicije, koji tipovi zrna smeju da se nalaze na kom silosi, govori samo o njihovom životu, ali to ne znači da silosi ne mogu da dobiju informacije o drugom tipu zrna od drugog silosa, radi izvršavanja određene operacije.

Takođe se vidi da sam klijent nije svestan ove podele, već on izvršava samo komunikaciju sa klasterom, dok se ostale stvari dešavaju u pozadini [9]. Izuzetno je važno znati, da svi silosi treba da referenciraju interfejse svih tipova zrna, ali konkretne implementacije klase tih zrna treba da referenciraju samo silosi, koji su predviđeni da će ih skladištiti [9]. Jedna od ključnih stvari gde mogu nastati greške jesu same implementacije tipova zrna na silosima. Naime, ukoliko dva silosa podržavaju određeni tip zrna, onda implementacija tog tipa zrna na silosima mora biti ista na oba silosa, inače će doći do greške [9]. Ograničenja ovakvog pristupa ogledaju se u tome što u toku klijentovog rada sa klasterom, ukoliko je samo jedan silos bio zadužen za određen tip zrna, korisnik će dobiti `OrleansException` [9]. Greška takođe može da se desi ukoliko se u toku klijentove komunikacije doda novi silos, koji je zadužen za korisnika od interesa tip zrna, doći će do greške `ArgumentException` [9].

6. ORLEANSOV TOK PODATAKA

Orleansov tok ili tunel, predstavlja mehanizam za slanje različitih tipova informacija ka različitim tipovima zrna, sa jednom velikom prednošću da omogućava da jedan isti tok obrađuje različite stvari u zavisnosti od potrebe konkretnih zrna, što znači da se ovim drastično može smanjiti sama topologija slanja tokova podataka [10]. Ono što Orleansov tok izdvaja od drugih jeste da ispunjava 4 glavna zahteva postavljena od strane potreba programerske zajednice [10]:

- Fleksibilna logika za obradu toka,
- Podrška za visoko dinamičke topologije,
- Dobra granuliranost toka,
- Distribucija.

Pod fleksibilnošću logike za obradu toka smatra se da se pruži podrška za bilo koji od načina za izražavanje logike obrade toka kao na primer: deklarativni upitnici, funkcionalno programiranje, protok podataka itd. [10]. Podrška visoko dinamičke topologije podrazumeva, da će tok biti sposoban da se dinamički menja i prilagođava potrebama na primer menjanjem logike [10].

Treći zahtev podrazumeva da postoje što manje jedinice kontrole radi boljeg upravljanja [10]. Četvrti zahtev predstavlja zapravo ciljeve svakog distribuiranog sistema a to su: skalabilnost, elastičnost, stabilnost, efikasnost i odzivnost [10]. Ove tokove podataka 7 osobina izdvaja u odnosu na druge [10]:

- Orleansov tok podataka se nikad ne kreira i ne uništava eksplicitno, već on uvek postoji jer je virtuelan, a shodno tome ne može nikada ni da otkaze.
- Ovi tokovi podataka su jedinstveno identifikovani sa kombinacijom GUID-a i proizvoljnog stringa
- Omogućenost razdvajanja samog procesiranja podataka od vremena i mesta, što pruža stabilnost prilikom razmene podataka između vremenski i geografski udaljenih servera
- Orleansovo okruženje za tokove omogućava održavanje velikog broja tokova
- Povezivanje zrna na tokove je dinamičko, što omogućava podršku čestih uključenja i isključenja zrna
- Orleansovo okruženje za tokove vodi računa o životnom veku tokova i njihovoj upotrebi
- Orleansov tok podataka radi uniformno sa svim zrnima i Orleans klijentima.

7. ZAKLJUČAK

Iz svega priloženog, može se doći do zaključka da Microsoft Orleans framework pruža potpuno jedinstveno rešenje za distribuirane aplikacije. Ne samo to, već novi principi, koji su ovom tehnologijom uvedeni pružaju potpuno novi pogled na celokupan razvoj jedne takve aplikacije.

Prilikom samog razvoja ovakve aplikacije, verovatno će trebati vremena prihvatiti ovakav pristup, ali nakon toga postaje potpuno prirodno razmišljanje kroz spektar zrna i silosa.

Kao što su i sami ljudi koji razvijaju ovaj framework primetili, celokupan pogled virtuelnog aktera je neophodno sada na ostale delove sistema primeniti, kako bi se postigle pune performanse ovog modela.

Pored toga što se trenutno radi na distribuiranoj transakciji, geo-distribuciji i indeksiranoj bazi virtuelnih aktera, potrebno je takođe dodatno raditi na sigurnosti i bezbednosti ovakvih podataka, jer je to ključna stvar kod sistema, koji za cilj imaju pravljenje aplikacije sa millionskim korisnicima. Ovo je posebno bitno s obzirom da je ova tehnologija namenjena za cloud platforme, koje su i dalje u razvoju, i još uvek sa dosta propusta.

8. LITERATURA

- [1]https://www.ibm.com/support/knowledgecenter/en/SSAL2T_9.1.0/com.ibm.cics.tx.doc/concepts/c_wht_is_dlstd_comptg.html (pristupljeno u avgustu 2018.)
- [2]<https://www.techopedia.com/definition/18909/distributed-system> (pristupljeno u avgustu 2018.)
- [3]<https://dotnet.github.io/orleans/Documentation/index.html#sidetoggle> (pristupljeno u avgustu 2018.)
- [4] George Pallis, "Developing Cloud Services Using the Orleans Virtual Actor Model" septembar-oktobar 2016
- [5] Philip A. Bernstein, Sergez Bzkov, Alan Geller, Gabriel Kliot, Jorgen Thelin, "Orleans: Distributed Virtual Actors for Programmability and Scalability", 2014
- [6]https://dotnet.github.io/orleans/Documentation/core_concepts/index.html (pristupljeno u septembru 2018.)
- [7]<https://dotnet.github.io/orleans/Documentation/gains/index.html> (pristupljeno u septembru 2018.)
- [8]https://dotnet.github.io/orleans/Documentation/cluster_and_clients/index.html (pristupljeno u septembru 2018.)
- [9]https://dotnet.github.io/orleans/Documentation/cluster_and_clients/heterogeneous_silos.html (pristupljeno u septembru 2018.)
- [10]https://dotnet.github.io/orleans/Documentation/streaming/streams_why.html (pristupljeno u septembru 2018.)

Kratka biografija:



Dušan Jeftić je rođen je 21.06.1994. godine u Novom Sadu. Završio je srednju školu Gimnazija Jovan Jovanović Zmaj u Novom Sadu 2013. godine. Fakultet tehničkih nauka u Novom Sadu je upisao 2013. godine. Ispunio je sve obaveze i položio je sve ispite predviđene studentskim programom, nakon čega je sa Diplomskim radom "Obfuskacija programskog koda i detekcija obfuskovanog malvera" ostvario titulu Diplomiranog inženjera Elektrotehnike i računarstva. Ubrzo posle toga je upisao master studije takođe na Fakultetu tehničkih nauka u Novom Sadu.

MODELOVANJE I SIMULACIJA MODULARNOG PRETVARAČA SA VIŠE NIVOA (MMC) U REALNOM VREMENU KORISTEĆI PREKIDAČKU FUNKCIJU**MODELING OF MODULAR MULTI-LEVEL CONVERTER (MMC) USING SWITCHING FUNCTION APPROACH SUITABLE FOR REAL-TIME SIMULATIONS.**Simiša Simić, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad***Oblast –ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu su objašnjeni neki od osnovnih topologija pretvarača sa više nivoa. Takođe urađen je matematički model ovog pretvarača. Prikazani su načini modelovanja kao i rezultati simulacije modela u realnom vremenu korišćenjem Typhoon HIL uređaja. Prikazana su tri primera primene ovog modela. Prva simulacija pokazuje rad pretvarača priključenog na „jaku“ mrežu. Druga simulacija prikazuje rad pri iznenadnoj promeni potrošnje, a treća pri linearnoj promeni snage potrošača.

Ključne reči: Modularni pretvarač sa više nivoa, Simulacija sistema, Modelovanje u realnom vremenu

Abstract – In this paper some of the basic topologies of multi-level converters (MMC) are explained, and MMC mathematical model has been developed. Modeling methods as well as the simulation results in real-time using the Typhoon HIL emulator has been shown. Three examples of application of the model are given. The first simulation shows operation of the MMC converter connected to a strong grid. The second simulation shows operation under instantaneous change of power demand, while the third under a linear change of power demand.

Keywords: Modular Multi-Level Converter, Simulation, Real-time modelling

1. UVOD

Razvojem novih tehnologija i uređaja tokom XX veka u centru pažnje je došla racionalna konverzija električne energije putem pretvarača energetske elektronike. Kada je u pitanju transport energije od velikih elektrana do samih potrošača i to na velike udaljenosti, sve više pažnje se posvećuje visoko-naponskom prenosu jednosmernom strujom (eng. *High-Voltage Direct Current - HVDC*). Osnovne prednosti HVDC prenosa električne energije su:

- Manji gubici,
- Lakša kontrola toka aktivne snage,
- Mogućnost kontrolisanog transfera energije,
- Ograničenje struje kratkog spoja,
- Manji uticaj na okolinu i životnu sredinu,
- Manji troškovi.

Prilikom primene konvencionalnih pretvarača za HVDC javljaju se izvesni problem, kao što su niske vrednosti napreznja, veliki broj komponenti, mala pouzdanost, složena konstrukcija, nepouzdana pogonska kola itd.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. drVladimir Katić.

Da bi se ovi problem umanjili predložena je primena modularnih pretvarača sa više nivoa (eng. *Modular Multi-level Converters - MMC*). Oni donose određene prednosti kao što su:

- Mogućnost postizanja višeg naponskog nivoa,
- Modularna konstrukcija,
- Duži vek trajanja,
- Povećanje pouzdanosti samog pretvarača,
- Niža cena,
- Jednostavno balansiranje napona,
- Mala vrednost harmonijskog izobličenja naona.

Nedostaci su veliki broj DC kondenzatora i manja pouzdanost zbog većeg broja komponenti.

U ovom radu su razmotreni MMC pretvarači za primenu u HVDC sistemima. Napravljen je matematički i simulacioni model korišćenjem Tajfun HIL emulatora. Prikazana su tri primera primene ovog modela, tj. rešena su tri zadatka. Prvo rešenje, odnosno simulacija pokazuje rad pretvarača priključenog na „jaku“ mrežu. Druga simulacija prikazuje rad pri iznenadnoj promeni potrošnje, a treća pri linearnoj promeni snage potrošača.

2. MODULARNI PRETVARAČ SA VIŠE NIVOA

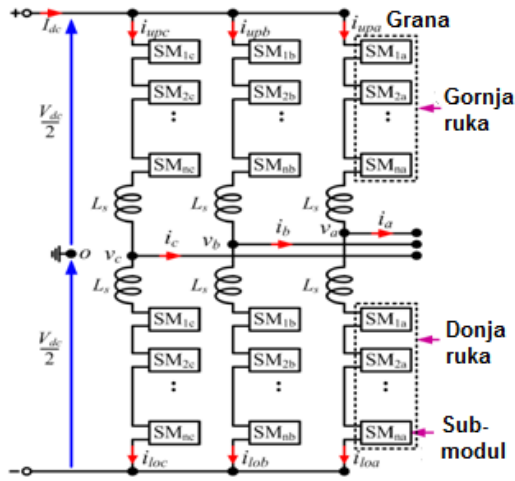
MMC je nova topologija prvi put predstavljena 2002 [1]. Ova topologija je sastavljena od više submodula kao na slici 1 [2]. Sličan koncept prikazan je i u [3]. Svaki submodul se sastoji od polu-mostne konfiguracije sastavljene od dva IGBT-a i kondenzatora (slika 2 [2]). Submoduli su uključeni ili isključeni u zavisnosti od prekidačke funkcije, koja određuje stanje svakog submodula.

U svakom submodulu nalazi se jedan DC kondenzator, koji je napunjen sa naponom:

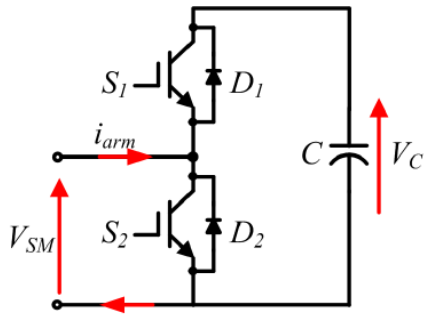
$$V_c = \frac{V_{DC}}{m-1}, \quad (1)$$

gde je V_c napon na kondenzatoru, V_{DC} napon na DC strani, a m – broj nivoa. Svaka ruka može da generiše najviše $\pm n * V_c = \pm V_{DC}$, gde su moduli uključeni ili isključeni tako da kreiraju AC izlazni napon.

Za m - naponskih nivoa potrebno je $m - 1 = n$ submodula po grani, dok za celu granu invertora je potrebno $2(m - 1) = 2n$ submodula po grani. U poređenju sa sličnom CMCI modulima, kod MMC modul može da ima samo dva naponska nivoa V_c ili 0. Ovo objašnjava zašto dve ruke se nalaze unutar jedne grane. Dve induktivnosti, u svakoj ruci grane su tu da bi preuzeli razliku napona prilikom uključanja i isključanja submodula.



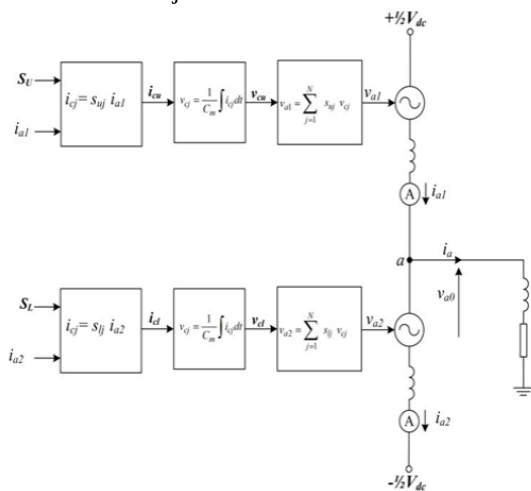
Slika 1. Uprošćen izgled MMC pretvarača [2]



Slika 2. Izgled jednog SM u MMC pretvaraču [2]

2.1. Matematički model prekidačke funkcije

Na slici 3 [4] prikazana je reprezentacija jedne grane MMC-a sa N submodula po grani, čija blok šema je data na slici 1, koristeći upravljačku (prekidačku) funkciju. Ovaj prikaz je pogodan za postavljanje matematičkog, odnosno simulacionog modela. Svaki kondenzator je opterećen $\frac{V_{dc}}{N}$ i ta vrednost mora ostati približno ista tokom cele simulacije.



Slika 3. Reprezentacija jedne noge MMC koristeći prekidačku funkciju [4]

Matematički model izlaznog napona v_{a0} za fazu a može se predstaviti preko gornje ili preko donje grane u kolu. Ako se predstavi preko gornje grane dobija se izraz (2), a ako preko donje grane izraz (3):

$$v_{a0} = \frac{1}{2}V_{dc} - \sum_{j=1}^N S_{uj}V_{cj} - R_d i_{a1} - L_d \frac{di_{a1}}{dt} \quad (2)$$

$$v_{a0} = -\frac{1}{2}V_{dc} + \sum_{j=1}^N S_{lj}V_{cj} + R_d i_{a2} + L_d \frac{di_{a2}}{dt} \quad (3)$$

gde su: $[v_{a0}]$ – izlazni AC napon, $[V_{dc}]$ – napon DC – sabirnice, $[S_{uj}]$ – prekidačka funkcija gornje grane j -og kondenzatora, $[S_{lj}]$ – prekidačka funkcija donje grane j -og kondenzatora, $[V_{cj}]$ – napon j -og kondenzatora, $[R_d]$ – otpornost celokupne grane (gornje i donje), $[L_d]$ – induktivnost celokupne grane (gornje i donje), $[i_{a1}]$ – struja kroz gornju granu, $[i_{a2}]$ – struja kroz donju granu. Naponi na svakom od kondenzatora u grani se mogu prikazati koristeći prekidačku funkciju za gornju granu:

$$v_{cj} = \frac{1}{C_m} \int S_{uj} i_{a1}(t) dt \quad (4)$$

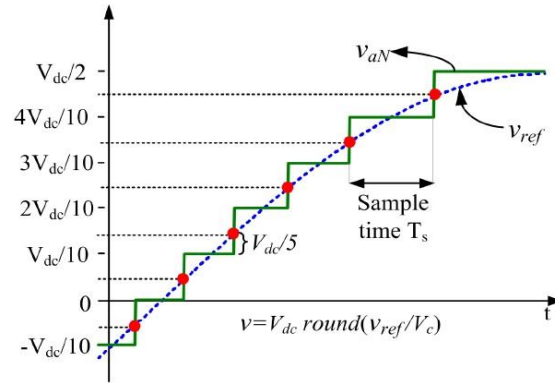
odnosno sledeću prekidačku funkciju za donju granu:

$$v_{cj} = \frac{1}{C_m} \int S_{lj} i_{a2}(t) dt \quad (5)$$

gde su: $[V_{cj}]$ – napon j -og kondenzatora, $[i_{a1}]$ – struja kroz gornju granu, $[i_{a2}]$ – struja kroz donju granu, $[S_{uj}]$ – prekidačka funkcija gornje grane j -og kondenzatora, $[S_{lj}]$ – prekidačka funkcija donje grane j -og kondenzatora, $[C_m]$ – vrednost kondenzatora.

2.2. Kontrola i balansiranje MMC-a

Da bi se dobila referenca za odabir naponskog nivoa koristi se metoda najbližeg naponskog nivoa (*eng. Nearest Level Control - NLC*) [2]. Naponi u tri faze se kontrolišu nezavisno. Slika 4 prikazuje način na koji se kreiraju nivoi u poređenju sa referentnim naponom.



Slika 4. NLC metoda [2]

Korišćenjem v_{ref} signal, preko relacije (6) dobija se naponski nivo n_{nl} .

$$n_{nl} = \frac{1}{v_c} * \text{round}(v_{ref}) \quad (6)$$

gde je $[v_c]$ – napon na kondenzatorima dat sa:

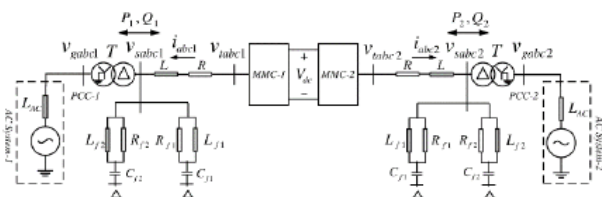
$$v_c = \frac{v_{dc}}{n} \quad (7)$$

gde je n broj kondenzatora po grani, $[v_{ref}]$ – referentni signal, $[n_{nl}]$ – broj naponskog nivoa, odabiraju se oni kondenzatori na kojima je napon najveći. Ovim algoritmom se postiže da kondenzatori “plutaju”

(“plivaju”) oko $\frac{v_{dc}}{n}$ vrednosti, i to tako da se prazne oni sa najvećim naponom, a pune oni sa najmanjim naponom. Kondenzatori, koji imaju najmanju količinu naelektrisanja na njihovim pločama su oni sa najmanjim naponom, ukoliko je struja u gornjoj ruci manja od nule. Ukoliko je struja u gornjoj grani veća od nule odabiraju se dvadeset submodula. Način nakoji se bira koji će od submodula biti uključen zavisi isključivo od smera struje i od količine elektriciteta na pločama kondenzatora.

3. REZULTATI SIMULACIJE

Posmatra se MMC u HVDC sistemu, koji se sastoji od MMC-1 (AC/DC pretvarača) na jednoj (ulaznoj) strani i MMC-2 (DC/AC pretvarača) na drugoj (izlaznoj) strani (slika 5 [5]). Sistem povezuje dve AC mreže (AC system 1 i AC system 2).



Slika 5. Primena MMC u HVDC sistemu [5]

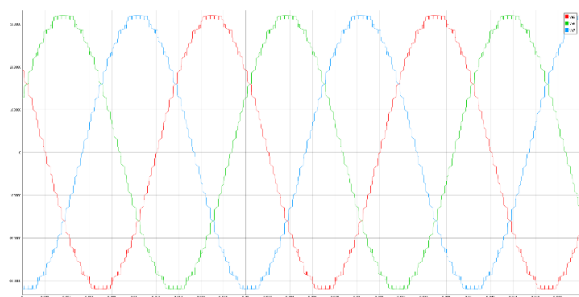
Zadatak 1: Potrebno je modelovati i simulirati rad MMC pretvarača, koji je priključen na javnu mrežu u stabilnim (balansiranim) uslovima.

Zadatak 2: Modelovati i simulirati rad pretvarača u slučaju da se na izlaznoj strani (u AC sistemu 2) naglo priključi i isključi opterećenje od $S = 250 \text{ MVA}$ sa faktorom snage $\cos \varphi = 0.85$. Pratiti promene talasnih oblika struje i napona.

Zadatak 3: Modelovati i simulirati rad pretvarača u slučaju da se na izlaznoj strani (u AC sistemu 2) opterećenje linearno menja aktivna snaga u opsegu od $P = 500 \text{ MW}$ do $P = 935 \text{ MW}$.

3.1 Rezultati simulacije pretvarača na mreži u ustaljenom stanju (Zadatak 1)

Prvi zadatak je da se simulira model povezan na krutu mrežu. Pretvarač u ovom slučaju treba da predstavlja idealni izvor sinusnog napona. Model MMC pretvarača sadrži dvadeset i jedan naponski nivo. Model koristi prekidačku funkciju za opis submodula.



Slika 6. Izgled izlaznog napona iz pretvarača

Na slici 6 se vidi da svaki od napona je kreiran koristeći MMC sa dvadeset i jednim naponskim nivoom. Vrednost ukupnog harmonijskog izobličenja napona (*eng. Total Harmonic Distortion - THD*) iznosi $\text{THDU} = 2,43 \%$. Amplituda napona iznosi $\frac{v_{dc}}{2} = 320.000 \text{ [V]}$ sa

frekvencijom od 50 Hz. Vidi se da je izgled napona na izlazu iz pretvarača skoro pa idealan sinus sa jako malom procentualnom vrednosti harmonijskog izobličenja.

3.2 Rezultati simulacije za promenu potrošnje u mreži (Zadatak 2)

U drugom zadatku je potrebno pratiti ponašanje pretvarača za naglo povećanje potrošnje. Model je napravljen tako da upravljanjem prekidača iz stanja 1 u 0 predstavlja prelaz iz stanja manje potrošnje u stanje veće potrošnje. Povećanje potrošnje se postiže tako što se naglo smanjuje otpornost grane, što je adekvatno dodavanju novih potrošača, odnosno naglom povećanju njihove snage. U ovom primeru prate se vrednosti napona i struje po fazama, kao i snage koje se prenose od pretvarača do potrošnje.

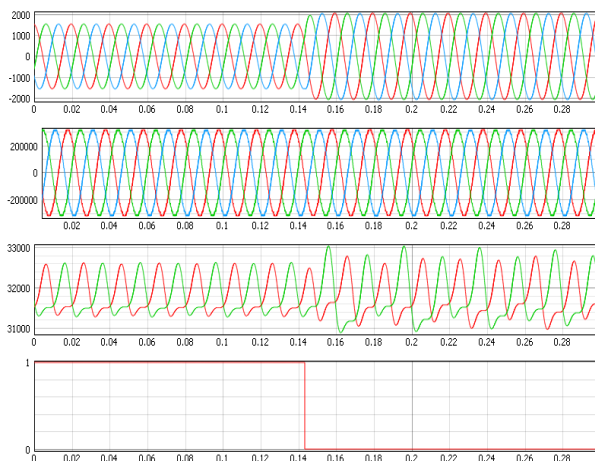
U periodu od početka simulacije do reagovanja prekidača vrednost očekivane struje se računa na sledeći način. Potrošnja iznosi $S=750 \text{ MVA}$ sa $\cos \varphi=0.85$ za nominalni napon DC-sabirnice $V_{dc}=650 \text{ kV}$. Potrošnja se predstavlja preko konstantne impedance po fazi i struje grane.

$$Z_{pot} = \frac{V_a^2}{3 * S} \quad (8)$$

$$I_a = \frac{V_n}{Z_{pot} + Z_d} \quad (9)$$

$$I_{a_{eff}} = |I_a| \quad (10)$$

gde su: V_a – efektivna fazna vrednost napona na izlazu iz pretvarača, S – fazna snaga potrošnje, Z_{pot} – impedansa potrošača, I_a – fazna struja potrošača, $I_{a_{eff}}$ – efektivna fazna vrednost struje potrošnje, Z_d – impedansa pretvarača. Na slici 6 prikazani su talasni oblici odziva u slučaju naglog povećanja (odnosno smanjivanja otpornosti). Karakteristične vrednosti (očekivane i dobijene) za nivo potrošnje od 750 kVA i povećane od 1000 kVA, date su u tabelama 1 i 2, respektivno.



Slika 7. Prikaz odziva modela u slučaju naglog povećanja potrošnje

Na slici 7 su na prvom grafiku prikazane tri fazne struje, na drugom tri fazna napona, na trećem trenutne vrednosti napona na kondenzatorima, a na četvrtom preklopno stanje prekidača sa 1 na 0 kada dolazi do povećanja

potrošnje. Vidi se da dolazi do povećanje struje pri većoj snazi potrošnje. Takođe se može videti da se pretvarač ponaša stabilno u slučaju naglog povećanja potrošnje. Zbog veće struje potrošnje, javlja se i veći napon na kondenzatorima.

Tabela 1. Očekivane i dobijene vrednosti za potrošnju od 750 kVA

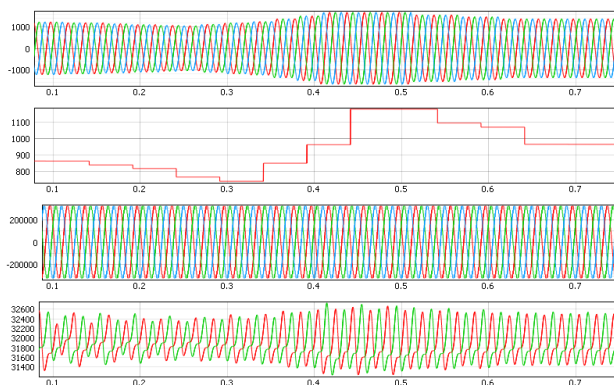
Naziv signala	Očekivane vrednosti	Izmerene vrednosti	Jedinica
Z_{pot}	174.33 + i117.3	174.33 + i117.3	Ω
P	637.500	636.692	MW
Q	395,087	394.563	MVA _r
k	0.850	0.851	-
S	750	749.0	MVA
$I_{a_{eff}} = I_a $	1.0769	1104.138	kA

Tabela 2. Očekivane i dobijene vrednosti za potrošnju od 1000 kVA

Naziv signala	Očekivane vrednosti	Izmerene vrednosti	Jedinica
Z_{pot}	138.490 + i76.377	138.490 + i76.377	Ω
P	900	897.585	MW
Q	435.889	434.986	MVA _r
k	0.9	0.901	-
S	1000	997.432	MVA
$I_{a_{eff}} = I_a $	1.4307	1.468365	kA

3.3 Rezultati simulacije za linearnu promenu potrošnje (Zadatak 3)

Za samu promenu se koristio klizač, koji menja otpornost mreže od 150 Ω do 300 Ω . Snaga se kreće u opsegu od 500 MVA do 935 MVA.



Slika 8. Rezultati simulacije za linearnu promenu potrošnje

Na slici 8 se vide rezultati linearne promene snage. Zbog velike rezolucije simulacije linearna promena izgleda kao stepenasta. Na prvom grafiku se vidi kako se vrednosti struje povećavaju pri povećanju potrošnje mreže, na drugom vrednosti efektivne struje u fazi a, na trećem fazne vrednosti napona, i na četvrtom vrednosti napona na kondenzatorima. Tokom ove simulacije vrednosti napona se ne menjaju što dovodi do zaključka da se ovaj model MMC-a ponaša kao idealni naponski izvor. Vrednosti napona na kondenzatorima i dalje se kreću oko očekivane vrednosti

4. ZAKLJUČAK

Rad opisuje način modelovanja i simulacije u realnom vremenu modularnog pretvarača sa više nivoa (MMC). Objasnjene su osnove MMC pretvarača, napravljen matematički model i urađene simulacije u realnom vremenu jednog od najčešće korišćenih pretvarača. Pokazane su mogućnosti MMC pretvarača kao jednog od osnovnih topologija za buduće HVDC prenosne sisteme.

Simulacija nam omogućava da sa velikom preciznosti pratimo vrednosti svih važnih signala. Jedna od osnovnih prednosti jeste da se može sa sigurnosti vršiti odabir prekidača i kondenzatora unutar submodula. Uz povećanje broja naponskih nivoa smanjuje se naponsko naprezanje prekidača, čime se dolazi do zaključka da je moguće koristiti prekidače koji mogu da izdrže manje naponsko opterećenje dok je izlazni naponom znatno veći.

Model MMC-a na zavidnom nivou reprezentuje inverter HVDC postrojenja. Očekivani rezultati se podudaraju sa onim očekivanim što dovodi do zaključka da se ovakav model pretvarača može koristiti za verifikaciju HVDC postrojenja.

5. LITERATURA

- [1] R. Marquardt, A. Lesnicar, J. Hildinger, "Modulares stromrichterkonzept fur netzkupplungsanwendung bei hohen spannungen," *ETG-Fachtagung, Bad Nauheim, Germany*, 2002.
- [2] P.M. Meshram, V.B. Borghate "A Simplified Nearest level control (NLC) Voltage Balancing Method for Modular Multilevel Converter (MMC)", *IEEE Trans. on Power Electronics*, Vol.30, No.1, Jan.2015.
- [3] D. Siemaszko, et al, "Evaluation of control and modulation methods for modular multilevel converters," *Inter. Power Electronics Conf. (IPEC)*, pp.746-753, 21-24 June 2010,
- [4] G.P. Adam, P. Li, I.A. Gowaed, B.W. Williams, "Generalized switching function model of modular multilevel converter", *ICIT, Sevilla*, 2015.
- [5] M. Saeedifard, R. Iravani, "Dynamic Performance of a Modular Multilevel Back-to-Back HVDC System", *IEEE Trans. on Power Delivery*, Vol. 25, No. 4, Oct. 2010

Kratka biografija:



Simiša Simić rođen je u Bijeljini 1994. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2018.god.
kontakt: simicsimisa@gmail.com



Vladimir Katić rođen je 1954. godine u Novom Sadu. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu 1978. god., a magistrirao i doktorirao na Univerzitetu u Beogradu 1981. i 1991. godine, respektivno. Oblast interesovanja su energetska elektronika, kvaliteta električne energije, obnovljivi izvori električne energije i električna vozila.

OSETLJIVOST REŽIMA DISTRIBUTIVNE MREŽE NA PROMENU OPTEREĆENJA I VREDNOSTI PARAMETARA

SENSITIVITY ANALYSIS OF DISTRIBUTION NETWORK

Zoran Sikirica, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je analiziran problem netehničkih gubitaka u distributivnim mrežama kao i osetljivost distributivnih mreža na promenu parametara i opterećenja. Izvršena je klasifikacija uzroka i posledica tih gubitaka. Analiza osetljivosti režima distributivne mreže na promene njenog opterećenja i parametara mreže izvršena je na osnovu proračuna tokova snaga. Program za proračun tokova snaga realizovan je u programskom jeziku Fortran.

Ključne reči: Distributivne mreže, proračun tokova snaga, netehnički gubici, osetljivost.

Abstract – This paper analyzes the problem of non-technical losses in distribution networks as well as the sensitivity of distribution networks to the change of parameters and load. Furthermore, the classification of causes and consequences of those losses have been done. Analysis of the distribution network regime sensitivity to the change of its load and network parameters has been done based on power flow calculations. Power flow calculation program is realized in Fortran programming language.

Keywords: Distributive networks, Power flow calculation, non-technical losses, sensitivity.

1. UVOD

U radu je izvršena analiza promene režima radikalne distributivne mreže (DM) usled promene vrednosti opterećenja i parametara DM. Pritom, razmatrane su promene vrednosti napona električno najudaljenijeg čvora DM, vrednosti ukupnih gubitaka aktivne i reaktivne snage i granice opterećenja DM. Proračuni su izvršeni za različite modele potrošnje. U skladu sa tim, u radu je razmatran uticaj netehničkih gubitaka (pre svega kao posledica krađe električne energije [8]). Ukazano je na značaj poznavanja tipa potrošnje i vrednosti parametara prilikom proračuna i planiranja DM.

Nakon uvoda, u drugoj glavi rada biće prikazana struktura DM, tj. modelovanje njenih elemenata. Treća glava je posvećena problematici tokova snaga, načinu klasifikacije čvorova, kao i formiranju strukture mreže kako bi se što efikasnije izvršili proračuni. Pored ovog u treće glava je posvećena proračunu tokova snaga Rajičićevom metodom. Osetljivosti distributivne mreže pri promeni njenih parametara i opterećenja prikazana je u četvrtoj glavi.

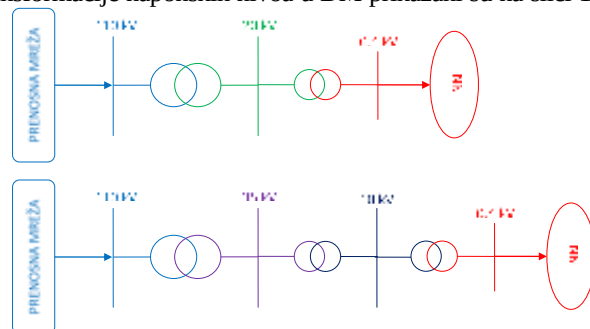
NAPOMENA:

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

Peta glava je posvećena detaljnoj podeli svih gubitaka u DM, kao i njihova klasifikacija i uticaj na mrežu. U šestoj glavi je prikazana test DM na kojoj će se vršiti proračuni, kao i analiza i detekcija netehničkih gubitaka sa matematičkim modelom tokova snaga. U sedmoj glavi kroz niz primera predstavljene su promene koje se dešavaju u DM kada se promeni opterećenje ili neki od parametara DM. Zaključak je prikazan u osmoj glavi, i potom, u devetoj glavi sledi spisak literature korišćene pri izradi rada.

2. TOPOLOGIJA DISTRIBUTIVNE MREŽE

DM preko transformatorskih stanica VN/SN preuzima električnu energiju iz prenosne mreže i distribuira je do krajnjih (srednjih i malih) potrošača. Potrošači električnu energiju preuzimaju preko transformatorskih stanica SN/NN i iz manjih elektrana koje su priključene u DM [2]. Primeri transformacije naponskih nivoa u DM prikazani su na slici 1.



Slika 1 – Primer transformacije naponskih nivoa u DM

2.2.1. Model potrošača

Snaga potrošnje potrošača (potrošačkog područja) je funkcija modula napona i učestanosti. U proračunima stacionarnih tokova snaga, vrednost učestanosti je unapred specificirana (poznata). U skladu sa tim kompleksna snaga potrošača je funkcija samo modula napona čvora na koji je priključen. One se obično iskazuju preko tri komponente [3]:

$$S_x(U_x) = k_{sp} P_x^{spec} - jk_{sq} Q_x^{spec} + k_{ipx} \frac{U_x}{U_{nNN}} P_x^{spec} - \quad (1)$$

$$jk_{iqx} \frac{U_x}{U_{nNN}} Q_x^{spec} + k_{ypx} \left(\frac{U_x}{U_{nNN}} \right)^2 P_x^{spec} - jk_{yqx} \left(\frac{U_x}{U_{nNN}} \right)^2 Q_x^{spec}$$

2.2.2. Vodovi

Vodovi su elementi elektroenergetskog sistema za prenos i distribuciju napona i električne energije, pri čemu se razlikuju nadzemne i kablovske deonice [4]. Za njihovo modelovanje najčešće se koristi π ekvivalentna šema.

2.2.3. Transformatori

Transformator je osnovni uređaj za transformaciju električne energije sa jednog na drugi naponski nivo (posredstvom magnetne sprege, bez pokretnih delova). Trofazni energetski transformator moguće je ekvivalentirati pasivnim četvorokrajnikom („Π“ šemom) [10].

3. TOKOVI SNAGA

Velika dimenzionalnost realnih DM i njihova slaba potencijalna upetljanost, kao i relativno visok odnos R/X, prouzrokovali su vrlo slabu uslovljenost matrica kojim se opisuju takvi problemi. Na taj način, svi navedeni numerički postupci, koji su bazirani na matričnom pristupu, postaju neefikasni u rešavanju problema proračuna tokova snaga u DM.

Da bi se ovi problem prevazišli, kao i da bi se ovi proračuni maksimalno ubrzali korišćenjem poznavanja karakteristika DM, razvijeni su specijalizovani algoritmi za proračun tokova snaga u DM. U ovim postupcima proračun se vrši po granama DM, čime se potiskuje potreba za rešavanjem slabo uslovljenih matričnih sistema jednačina. Na taj način, značajno se povećava brzina proračuna i opadaju memorijski zahtevi [1].

U cilju proračuna napona i tokova snaga u ovom radu je primenjena Rajić metoda [11]. Proračuni su realizovani na primeru jednostavne radialne DM.

4. OSETLJIVOST REŽIMA DM

U ovom delu je predstavljena osetljivost DM na promenu opterećenja i promenu parametara DM.

4.1. Uvodna razmatranja

Jedan deo rada analizira rezultate proračuna DM pri različitim parametrima potrošnje. Tako je kroz različite primere prikazana promena karakterističnih veličina DM: struje najopterećenije deonice, napona električno najudaljenijeg čvora, ukupnih gubitaka aktivne i reaktivne snage pri različitim napojnim naponima i različitim tipovima potrošnje itd.

4.2. Osetljivost DM na promenu opterećenja

U ovom delu su analizirane promene napona električno najudaljenijeg čvora DM, ukupnih gubitaka aktivne snage i granice mogućeg opterećenja sa promenom tipa potrošnje, faktora snage i preraspodelom reaktivne snage potrošnje po čvorovima mreže. Iz tabela 1, 2, 3, 4 utvrđeno je da, ako tip potrošnje i faktor snage nisu poznati, za opterećenje duplo veće od zadatog može doći do greške u proceni napona od 2% i ukupnih gubitaka snage od 32%.

Tabela 1 – Vrednosti u pri promeni tipa potrošnje i p za $\cos\varphi = 0,85$

$k_p=k_q$	$u[\%]$	
	$p=1$	$p=2$
0	-0,43122	-2,05649
1	0,023852	0,240748
2	0,407368	1,815741

Ako je faktor snage nepoznat, a tip potrošnje poznat

moguće greške su 4.2% i 46%, respektivno. Sa porastom kvaliteta informacija o faktoru snage rastu granice mogućeg opterećenja mreže za sve tipove potrošnje.

Tabela 2 – Odstupanje u pri promeni faktora snage i p

$k_p=k_q$	$\cos\varphi$	$u[\%]$	
		$p=1$	$p=2$
0	0.7	-1,61313	-4,20728
	0.85	-0,27612	-0,57553
	1.0	1,889249	4,782811
1	0.7	-1,41693	-3,04903
	0.85	-0,26003	-0,52615
	1.0	1,67696	3,575175
2	0.7	-1,26368	-2,41424
	0.85	-0,31539	-0,76122
	1.0	1,579073	3,175458

Tabela 3 – Vrednosti Δp pri promeni tipa potrošnje i p za $\cos\varphi = 0,85$

$k_p=k_q$	$\Delta p[\%]$	
	$p=1$	$p=2$
0	14,51973	32,14334
1	-1,19775	-5,36949
2	-13,322	-26,7739

Tabela 4 – Odstupanje Δp pri promeni $\cos\varphi$ i p

$k_p=k_q$	$\cos\varphi$	$\Delta p[\%]$	
		$p=1$	$p=2$
0	0.7	40,65547	46,04159
	0.85	-4,92168	-5,85348
	1.0	-35,7338	-40,1881
1	0.7	37,33021	37,33499
	0.85	-4,6266	-4,62667
	1.0	-32,7036	-32,7083
2	0.7	34,73897	32,40959
	0.85	-4,61535	-4,91265
	1.0	-30,1236	-27,4969

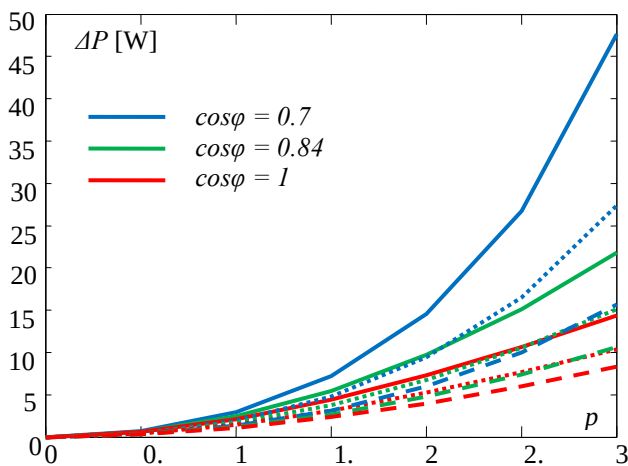
Tabela 4 – Odstupanje Δp pri promeni $\cos\varphi$ i p

Iz tabele 5 je moguće videti da neravnomerna raspodela reaktivne snage potrošnje može dovesti do povećanja gubitaka aktivne snage od desetak procenata i samim tim smanjenja granice mogućeg opterećenja DM.

Tabela 4 – Smanjenje u i povećanje Δp pri neravnomernoj, u odnosu na ravnomernu raspodelu reaktivnih snaga potrošnje

$k_p=k_q$	$p=1$	$p=2$
$u[\%]$	0,179614	0,467402
$\Delta p[\%]$	10,93201	10,35854

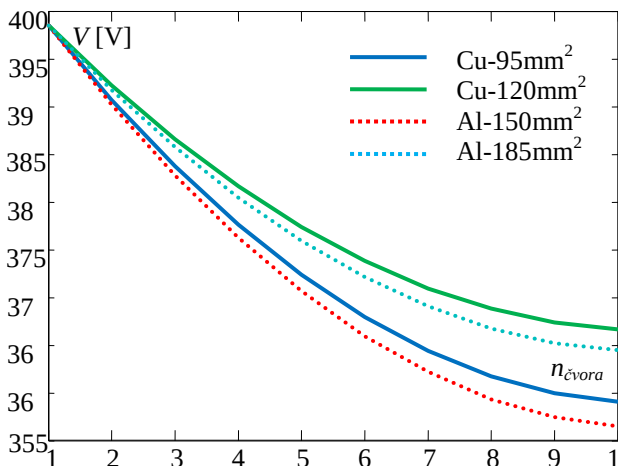
Na slici 2. prikazane su krive gubitaka aktivne snage za $k_p=k_q=0, 1$ i 2 , i faktore snage $0.7, 0.84$ i 1 pri čemu p predstavlja vrednost sa kojom je množeno osnovno opterećenje u čvorovima test DM.



Slika 2 – Ukupni gubici aktivne snage za tip potrošnje konstantne snage (—), struje (·····) i impedanse (— · — ·) pri različitim $\cos \varphi$

4.3. Osetljivost DM na promenu parametara voda

Osetljivost vrednosti napona na razmatranoj DM u zavisnosti od parametara deonice (tipa i preseka kabla) prikazana je na slici 3.



Slika 3 – Dijagram pada napona za različite preseke kablova i različite materijale (potrošač konstantne snage)

5. VRSTE GUBITAKA U DISTRIBUTIVNIM MREŽAMA

Vrednost gubitaka jedan je od direktnih pokazatelja efikasnosti prenosa energije od proizvođača do potrošača. Najveći deo gubitaka električne energije u procesu prenosa električne energije od proizvođača do potrošača čine gubici u DM[1]. Analiza i određivanje mesta nastanka ovih gubitaka je ujedno i preduslov za njihovo smanjenje do nivoa prihvatljivih vrednosti. Iz tog razloga došlo je do razvika novih i savremenih metoda za kvalitetnu analizu gubitaka električne energije u mreži, koje će koristiti ograničen broj podataka. Funkcija estimacije stanja sistema je proces u kome se na osnovu telemetrisanih veličina merenja i istorijskih podataka u sistemu, uz korišćenje optimizacionih kriterijuma, određuju nepoznate vrednosti promenljivih veličina. Ova funkcija se pokazala kao pogodna za određivanje nepoznatih veličina u sistemu, međutim, osnovni problem je skromna automatizacija, što otežava proces

prikupljanja telemetrisanih (SCADA) podataka [1,6].

Gubici električne energije u DM mogu se svrstati u dve osnovne grupe (prema načinu nastanka):

- Tehnički gubici
- Netehnički (komercijalni) gubici.

5.1. Tehnički gubici

Pod tehničkim gubicima smatraju se gubici koji nastaju kao posledica stavljanja postrojenja pod napon i proticanja struje kroz električnu mrežu. Veličina ovih gubitaka zavisi od električnih karakteristika elemenata instaliranih u mreži i režima proticanja struje kroz sva postrojenja koja se nalaze između proizvođača i potrošača električne energije [7,8,9].

Tehnički gubici sa stanovišta uzroka nastanka dele se na [6,9]:

- gubitke nezavisne od opterećenja (fiksne gubitke)
- gubitke zavisne od opterećenja (varijabilne gubitke).

5.2. Netehnički gubici

Druga grupa obuhvata gubitke koji nastaju na osnovu razlike evidentiranih veličina nabavljene i isporučene (naplaćene) električne energije u određenom vremenskom periodu. Ovi gubici se javljaju kao posledica nesavršenosti organizacije eksploatacije sistema, nesavršenosti uređaja za registraciju protoka električne energije, kvarova na postrojenjima, kao i neovlašćenog korišćenja električne energije [6,8,9].

Na osnovu iskustva komercijalni gubici [7,8,9]:

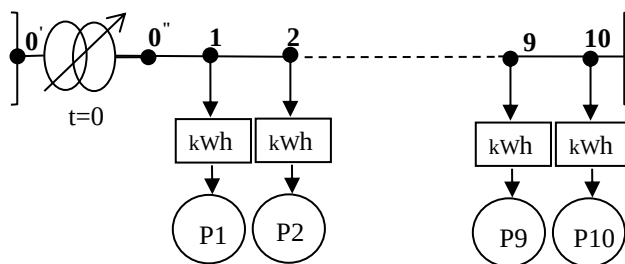
- egzistiraju u obliku veoma varijabilnih veličina,
- praktično ih je nemoguće odrediti računskim putem,
- mogu biti pozitivni i negativni.

6. ANALIZA DETEKCIJE NETEHNIČKIH GUBITAKA

Analiza uticaja neovlašćene upotrebe električne energije na stanje sistema izvršena je na test DM, slika 5. Osetljivost rešenja i brojila, analizirana je na osnovu simulacije dodatnog, neregistrovanog opterećenja u jednom, dva i više čvorova DM.

6.1. Opis test distributivne mreže

Radialna test DM se sastoji se od regulacionog Tr $m_r = 20 \pm 2 \times 5\% / 0.4 \text{ kV/kV}$ i niskonaponskog dela sa deset jednakih deonica i potrošača. Snaga Tr iznosi $S_n = 1 \text{ MVA}$. Pozicija teretnog menjača je $t=0$. Svaki potrošač je opremljen brojilom klase tačnosti 0.5, 1 i 1.5. Smatra se da su vrednosti potrošnje za svaki od potrošača snimane.

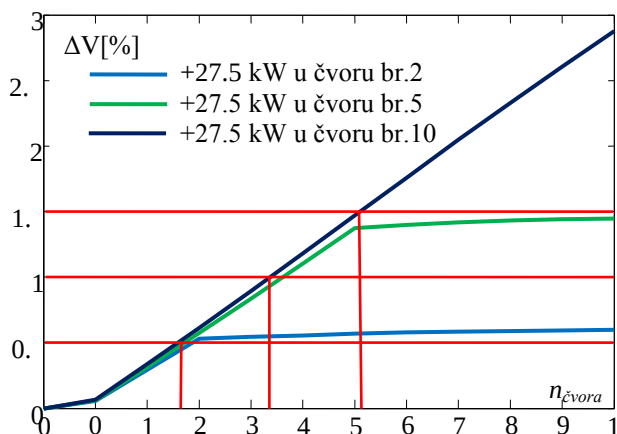


Slika 4 – Test distributivna mreže

6.2. Simulacija dodatnog opterećenja u mreži

U ovom delu razmatra se mogućnosti detekcije dodatnog opterećenja: Dodatno opterećenje je simulirano u čvorovima br. 2, 5 i 10, pri tom faktor snage ima konstantnu vrednost 0.84 (ind).

Jedan od ciljeva rada je da se pokaže za koje će vrednosti dodatnog opterećenja, promena modula napona biti veća ili jednaka 0.5%, 1.0% i 1.5%, odnosno da se analizira osetljivost brojila klase tačnosti 0.5, 1.0 i 1.5. Takođe, jedan od ciljeva rada je da se pokaže na kom mestu i kolika dodatna potrošnja može da se detektuje primenom takvih brojila.

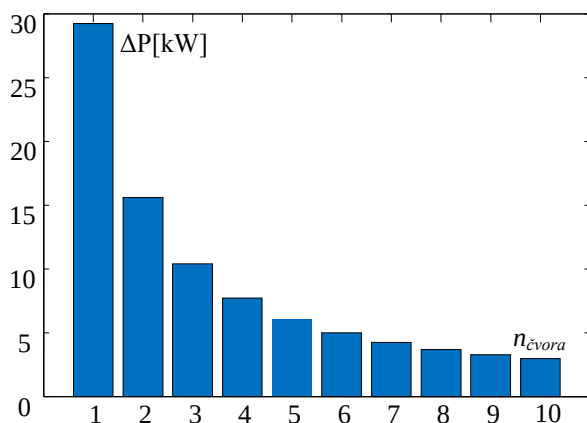


Slika 5 – Promena vrednosti modula napona čvorova usled dodatnog opterećenja od 27.5 kW u čvorovima 2, 5, 10

Na slici 6 prikazane su promene vrednosti modula napona za različita opterećenja u čvorovima br.2, 5, 10. Na slici se vidi koje se vrednosti opterećenja mogu detektovati sa instrumentima različite klase tačnosti.

6.3. Rezultati analize

U ovom delu prikazani su rezultati analize ispitivanja osetljivosti mogućnosti primene pametnih brojila različitih klasa za razmatranu test DM.



Slika 6 – Minimalne vrednosti dodatne potrošnje, koja može da se detektuje primenom brojila klase tačnosti 0.5

Na slici 7 prikazane su minimalne vrednosti dodatne potrošnje u potrošačkim čvorovima mreže, koje mogu da se detektuju primenom brojila klase tačnosti 0.5.

8. ZAKLJUČAK

Tema ovog rada jeste osetljivost režima DM na promenu opterećenja i vrednosti parametara kao i netehnički gubici koji se javljaju u DM. Na primeru jedne DM simulirane su različite promene parametara i pri tome su praćene promene.

U radu je prikazan uticaj neovlašćene upotrebe električne energije na početku, na sredini i na kraju test mreže na ostatak mreže. Cilj je bio da se pokaže za koje će vrednosti dodatnog opterećenja promena modula napona biti veća ili jednaka 0,5%, 1% i 1,5%, tj. da se pokaže osetljivost pametnih brojila čija je klasa tačnosti 0,5, 1 i 1,5. Rezultati pokazuju da se prilikom neovlašćene upotrebe električne energije uspešno mogu detektovati, pre svega, promene vrednosti modula napona čvorova sa dodatnom potrošnjom koji se nalaze u drugoj polovini test mreže, tj. da mesto dodatne potrošnje u test mreži u velikoj meri utiče na mogućnost detekcije dodatne potrošnje. Na početku mreže dodatna potrošnja u realnosti ne može biti detektovana jer bi vrednosti minimalne dodatke potrošnje koju bi napredno brojilo moglo da detektuje nerealno visoka.

9. LITERATURA

1. D.Bekut, V.Dabić D.Popović, Specijalizovani DMS Algoritmi; Prosveta, Novi Sad, 2011.
2. D.Jakus, I.Penović, R.Goić: Distribucija električne energije, Fakultet elektrotehnike i strojarstva, Split, 2008.
3. P.Vidović: Nesimetrični tokovi snaga distributivnih mreža; Novi Sad, magistarska teza, Fakultet Tehničkih Nauka, novembar 2008.
4. V.Strezorski: Osnovi elektroenergetike (elektroenergetski sistemi); radni materijal, školska 2002/2003. god.
5. D.Stojanović, L.Korunović: Analiza režima distributivnih mreža pri promeni opterećenja, Proc. 49th ETRAN Conference, Budva, June 5-10, Vol. I: Elektronski fakultet u Nišu, 2005.
6. D.Stojković, Tehnički i komercijalni gubici u distributivnim mrežama, FTN, diplomski rad, 2006.
7. J.Samardžija, Netečnički gubici u distributivnim mrežama, FTN, master rad, 2016. J. Samardžija.
8. S. Žutobradić, Ž. Rajić, L. Wagmann, H. Miličić: Analiza problematike gubitaka električne energije u distribucijskim mrežama članica EU, Umag, 2010.
9. D. Radojčić, Netečnički gubici, FTN, master rad, 2015.
10. Zadaci iz energetskih transformatora
11. D.Rajičić, R.Taleski: Two Novel Methods for Radial and Weakly Meshed Network Analysis; Electric Power Systems Research 48, pp. 79-87, December, 1998.

Kratka biografija:



Zoran Sikirica rođen je u Karlovcu (Republika Hrvatska), 27.05.1989. god. Srednju tehničku školu (Elektrotehničar automatike) završio je 2008. god. u Rumi. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranio je 2015. god. Iste godine upisao se na master studije.

ПРЕГЛЕД И ИСПИТИВАЊЕ ГРОМОБРАНСКИХ ИНСТАЛАЦИЈА**INSPECTION OF LIGHTNING PROTECTION SYSTEMS**

Боривој Јовановић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У раду су презентовани основни елементи који чине једну громобранску инсталацију, као и важност редовне контроле истих. Приказане су неке од најчешћих метода мерења отпора распростирања уземљивача, са својим предностима и манам, као и начини на који се могу избећи нетачни резултати мерења. У једном примеру дати су резултати мерења отпорности уземљивача, са тумачењем добијених вредности и предлогом мера за отклањање наведених недостатака.

Кључне речи: Громобранска инсталација, испитивање, отпорност уземљивача

Abstract – This Study presents the key elements that form a lightning protection system, as well as importance of it's regular inspection. Some of the most common earthing resistance measurment methods are shown, with their advantages and disadvantages, including the ways of avoiding wrong measuring results. With one example, earthing resistance measuring results are given along with their interpretation and suggested set of actions to improve the system condition.

Keywords: Lightning protection system, inspection, earthing resistance

1. УВОД

За заштиту зграда и других објеката од удара грома користи се громобран односно громобранска инсталација. Громобранска инсталација изведена у складу са стандардом, не може пружити апсолутне гаранције за заштиту објеката, људи или предмета, међутим, њено постојање знатно смањује ризик оштећења изазваних ударом грома уштићени објекат [1].

Улога громобрана јесте да заштити објекат од директног удара грома прихватајући га и што безбедније га спроводећи у земљу преко прихватног система, спусних проводника и уземљивача, чиме се умањује или потпуно елиминише потенцијална штета настала од директног удара грома у објекат.

Објекат са лоше изведеном, оштећеном, односно нефункционалном громобранском инсталацијом може бити у већој опасности него објекат који громобранску инсталацију и не поседује.

НАПОМЕНА:

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

Из овога произилази важност контроле громобранских инсталација, од момента извођења првих грађевинских радова на објекту, увидом у елементе инсталације који касније неће бити видљиви и приступачни, затим након измена на громобранској инсталацији, удара грома у исту, и периодично, ради утврђивања општег стања у ком се громобранска инсталација налази.

Приликом испитивања, елементарну важност представља правилно спровођење изабране методе мерења, као и исправно тумачење добијених резултата, из ког ће проистећи коначна оцена стања громобранске инсталације.

2. ЕЛЕМЕНТИ ГРОМОБРАНСКЕ ИНСТАЛАЦИЈЕ

Заштитни систем објекта који треба да омогући заштиту од директног атмосферског пражњења састоји се од:

- Прихватног система (који треба да прихвати атмосферско пражњење),
- Спусних проводника (који треба да спроведу струју грома ка земљи у уземљивач),
- Система уземљења (уземљивача),
- Система изједначења потенцијала и заштите од пренапона.

Прве три набројане компоненте громобранске инсталације припадају спољашњој громобранској инсталацији, док четврта (систем изједначења потенцијала и заштите од пренапона) припада унутрашњој громобранској инсталацији. У одређеним случајевима, могуће је постојање само спољашње, односно само унутрашње громобранске инсталације.

Прихватни систем громобранске инсталације може бити једна од, или комбинација следећих изведби:

- Штапна хватаљка,
- Мрежа проводника (Фарадејев кавез),
- Разапете жице,
- „Природна компонента“ (конструктивни елемент објекта, нпр. метални кровни покривач).

Спусни проводници су делови спољашње громобранске инсталације намењени за спровођење струје атмосферског пражњења од прихватног система до система уземљења [2].

Да би се смањиле опасности од појаве опасних прескока, спусни проводници морају бити постављени тако да од места удара грома до земље односно уземљивача:

- Постоји неколико паралелних струјних стаза и
- Дужине струјних стаза буду што краће.

Спусни проводници се најчешће изводе од трака од поцинкованог челика (FeZn), алуминијумских округлих проводника или помоћу „природних компоненти“.

Број спусних проводника громобранске инсталације не треба бити мањи од два и треба бити равномерно распоређен по обиму објекта, у границама могућности.

Уземљивачи су галвански повезане металне електроде, укопане у тло или положене у темељ објекта, које треба да омогуће безбедно одвођење струје атмосферског пражњења у тло.

Распоред типа А. Овај начин распоређивања подразумева радијалне или вертикалне уземљиваче. Сваки од спусних проводника мора се повезати бар на један одвојени уземљивач, који чине један радијални или вертикални (искошени) уземљивач [4].

Морају се поставити најмање два уземљивача овог типа.

Најмања дужина сваког уземљивача мора бити једнака:

- l_1 ако се ради о радијалном хоризонталном уземљивачу,
- $0,5 l_1$ ако се ради о вертикалном (искошеном) уземљивачу,

где је l_1 минимална дужина радијалног уземљивача приказана на слици 1 у функцији нивоа заштите и специфичне отпорности тла ρ .

У случају тла мале специфичне отпорности, може се одступити од минималних дужина приказаних на слици 1 уколико је отпорност распрострањања уземљивача мања од 10Ω .

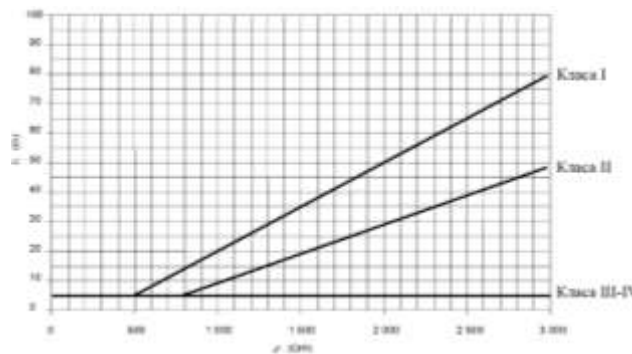
Распоред типа Б. За прстенасти (или темељни) уземљивач средњи геометријски полупречник r прстеног уземљивача не сме бити мањи од вредности l_1 , приказане на слици 1.

Ако је захтевана вредност l_1 већа од усвојене вредности ρ , додају се радијални или вертикални уземљивачи. Дужине l_h (за хоризонталне) и l_v (за вертикалне) добијају се уз помоћ следећих формула:

$$l_h = l_1 - r \quad (1)$$

и

$$l_v = \frac{l_1 - r}{2} \quad (2)$$



Слика 1. Минималне дужине уземљивача у зависности од специфичне отпорности тла [4]

Унутрашња громобранска инсталација има улогу да спречи појаву опасних прескока напона унутар штићеног објекта услед протицања струје пражњења кроз елементе спољашње громобранске инсталације или других проводних делова објекта. Унутрашња громобранска инсталација састоји се од система за изједначавање потенцијала и правилно инсталираних уређаја за заштиту од пренапона.

Ефикасност громобранске инсталације изражава се њеним нивоом заштите, сврстаног у 4 зоне: I, II, III и IV, при чему је највиши ниво заштите I [3].

3. ПРЕГЛЕД И ИСПИТИВАЊЕ ГРОМОБРАНСКИХ ИНСТАЛАЦИЈА

Циљ испитивања громобранске инсталације је провера да њени елементи у потпуности одговарају пројекту и пројектованој намени заштите објекта и то у погледу електричних карактеристика и у погледу физичке очуваности прихватног система, спусних водова и уземљивача [1].

Контролисање и испитивање система громобранске заштите обухвата визуелне контроле употпуњене следећим активностима [4]:

- Испитивање непрекидности, нарочито непрекидности оних делова громобранске инсталације који нису били видљиви за преглед приликом инсталације, а касније неће бити приступачни за визуелни преглед,
- Испитивањем отпорности уземљења система уземљења.

Постоје различите методе мерења отпора распрострањања постојећег уземљивача.

Метода трополног мерења, која захтева употребу две додатне сонде (ако уземљивач рачунамо као посебну сонду), како би се инјектирала струја у мерно коло и обезбедио референтни нулти потенцијал. Позиционирање две додатне сонде у односу на мерени уземљивач је од посебне важности.

Да би се остварило прецизно мерење, „нулта“ сонда мора бити постављена тако, да не упада у зоне утицаја изазване протицањем испитне струје кроз предметни уземљивач и другу додатну сонду.

Мерење се обавља тако што инструмент мери струју која се пропушта кроз коло које укључује предметни уземљивач и другу (крајњу) додатну сонду, а напон се мери између уземљивача и средишње сонде која се налази на нултом потенцијалу, ван зоне утицаја уземљивача и друге сонде.

Метода мерења отпорности петље посебно је погодна тамо где је постављање додатних сонди отежано мањком расположивог простора (урбана подручја), и примењена је за потребе мерења у овом раду, употребом инструмента „GOSSEN METRAWATT – PROFITEST 0100S-II“.

Метода се заснива на мерењу отпорности петље која укључује отпорности фазног проводника, предметног уземљивача и уземљивача дистрибутивног трансформатора.

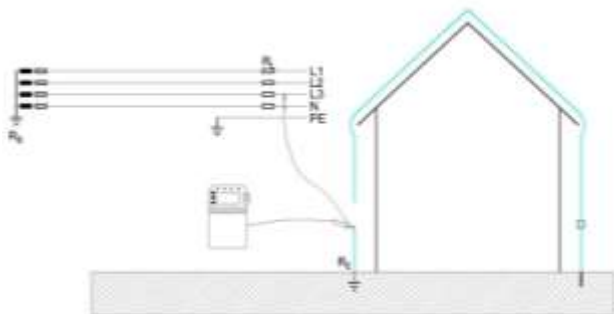
Ако се подразумева да је површински пресек фазног и нултог проводника (L и N) једнак, отпорност фазног проводника је тачно половина импедансе напајања. Импеданса напајања се може мерити помоћу функције инструмента Z_{LOOP} , тако што једну сонду двополног мерног адаптера прикључимо на фазни проводник, а другу на нулти.

Отпорност уземљења може се добити из следеће једначине:

$$R_E = R_{LOOP} - \frac{1}{2}R_L - R_B \quad (3)$$

Вредност отпорности радног/здруженог уземљења дистрибутивног трансформатора R_B мора се занемарити у рачуну отпорности уземљења, зато што је најчешће непозната. На овај начин израчуната отпорност уземљења у себи укључује и отпорност радног уземљивача као безбедносни фактор [5].

Приликом мерења, од кључне важности је одспојити главну сабирницу за изједначење потенцијала од испитиваног уземљивача. Уколико се мерење обавља у TN систему, а заштитни вод је галвански спојен на исту сабирницу као и уземљивач, приликом мерења добићемо ништа друго него вредност отпора петље квара L-PE, а не отпорност уземљивача.



Слика 2. Принцип мерења отпорности уземљења методом отпора петље

Метода мерења клештима за мерење отпорности петље, своди се на обухватање спусног проводника струјним клештима, без претходног растављања проводника. Клешта при том мере струју која протиче кроз спусни проводник.

Струјна клешта се састоје од два намотаја [6]:

- Генераторски намотај који генерише наизменични напон (EMS) константне вредности, услед ког кроз проводник протиче електрична струја јачине $I=E/R$,
- Пријемни (мерни) намотај који мери јачину електричне струје која протиче кроз спусни проводник.

Предност ове методе је што се релативно лако спроводи, али је зато ограничена на примену код система уземљења који се састоје од више паралелно везаних уземљивача.

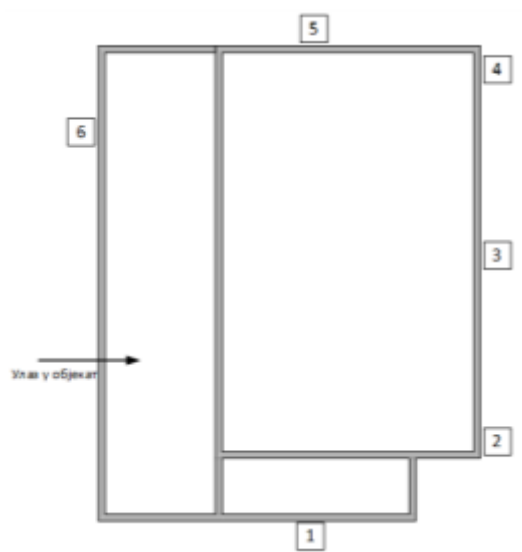
4. ПРИМЕР ТУМАЧЕЊА ДОБИЈЕНИХ РЕЗУЛТАТА МЕРЕЊА

За пример узет је објект Дома Културе у Новим Бановцима.

У овом примеру стављен је фокус само на испитивање отпорности распрострања уземљивача, као битног дела целокупног испитивања громобранске инсталације.

Пројектна документација громобранске заштите предметног објекта не постоји/није дата на увид, што у многоме отежава преглед инсталација.

На слици 3 дата је скица распореда мерних места (мерно раставних спојева) на којима је вршено мерење.



Слика 3. Скица распореда мерних места

Визуелним прегледм установљено је да је прихватни систем громобранске инсталације изведен „природном компонентом“, односно металним кровним покривачем објекта, целом својом површином. Спусни проводници изведени су траком од поцинкованог челика, димензија $25 \times 4 \text{ mm}^2$. Одређени број спусних проводника води се кроз фасаду објекта и самим тим нема изведене мерне спојеве на којима би се вршило мерење. Остали спусни проводници воде се површином фасаде објекта, и причвршћени су за исту.

Ови спусни проводници спајају се са припадајућим земљоводима димензија $25 \times 4 \text{ mm}^2$ помоћу укрсног комада/мерно раставног споја. Без увида у пројектну документацију, тешко је са сигурношћу одредити тип уземљивача, али с обзиром на величину објекта, може се претпоставити да се ради о тракастом/темељном уземљивачу, односно распореду Типа Б.

У табели 1 дати су резултати мерења отпорности распростирања уземљивача.

Табела 1. Резултати мерења

Мерно место	Отпорност уземљивача $R_{uz} [\Omega]$
1	36,11
2	26,02
3	84,51
4	31,85
5	-
6	7,44

Као што се из добијених резултата мерења види, вредности већине мерених спусних проводника превазилазе захтеваних 10Ω . Разлог оваквим вредностима може бити дотрајалост уземљивача, односно оштећење спојева земљовода са уземљивачем (корозија).

Велика одступања у вредностима отпорности такође указују на пропале спојеве земљовода и уземљивача. Велика специфична отпорност тла резултовала би лошим отпорностима простирања на сваком мерном месту, тако да се та могућност одбацује.

Мерењем непрекидности система, добијају се вредности отпора у околини $1,5\Omega$. Ово указује да спусни проводници који пролазе кроз фасаду објекта, који приликом мерења нису растављени, обезбеђују добру везу са уземљивачем. Ови спусни проводници су заштићенији од утицаја влаге и корозије. На основу овога може се додатно потврдити дотрајалост веза на претходно мереним местима.

Препоруке за отклањање недостатака:

- У циљу побољшања отпорности распростирања уземљивача, препоручује се укопавање додатних вертикалних цевних уземљивача, минималне дужине $2,5\text{m}$, и повезивање са проблематичним спусним проводницима (1,2,3 и 4).
- Пожељно је, након изведених преправки обезбедити пројектну документацију изведеног стања громобранских инсталација, у ком ће се поред осталог, одредити и потребан ниво заштите громобранских инсталација,
- Након обављених измена, потребно је поновити мерења.

5. ЗАКЉУЧАК

Овим радом дат је преглед саставних елемената громобранске инсталације, са најважнијим захтевима који једна громобранска инсталација треба да

задовољи, позивајући се притом на тренутно важеће прописе и стандарде.

Приказане су најчешће методе испитивања, дајући притом предност оним најпрактичнијим за примену. Кроз пример, указано је на уобичајене грешке на громобранској инсталацији, које се правилним испитивањем могу открити и уз одговарајуће преправке на инсталацији, отклонити.

На први поглед, поступак контроле громобранских инсталација може се учинити као релативно једноставан задатак. Залазећи дубље у проблематику, наилази се на низ препрека које је потребно премостити како на терену приликом испитивања, тако и приликом израде извештаја.

Прву битну препреку представља недостатак пројектне документације код већине старијих објеката на којима се врши испитивање. Без пројекта громобранске инсталације тешко је стећи јасну слику о елементима који нису видљиви и приступачни. Следећи проблем лежи у мерењу отпорности распростирања уземљивача, односно у одабиру методе мерења и избегавању добијања погрешних резултата који би дали погрешну слику о стању уземљивача.

Неправилним мерењем отпора распростирања и добијањем нетачних вредности, лош уземљивач може се испитивачу врло лако учинити као задовољавајућ, резултујући неоправдано позитивном оценом инсталације.

6. ЛИТЕРАТУРА

- [1] Заштита објеката од атмосферских пражњења, Љубомир Герић, Милан Савић, Чедомир Вујовић, Нови Сад, 2001.
- [2] Основни принципи пројектовања у мрежама средњег и ниског напона, Страхил Ј. Гушавац, Нови Сад, 2014.
- [3] Заштита објеката од атмосферског пражњења, Драган Вићовић, Зоран Хаџић, Београд, 2008.
- [4] Српски стандард SRPS EN 62305-3, Заштита од атмосферског пражњења – Део 3: Физичко оштећење објеката и опасност по живот, Институт за стандардизацију Србије, 2017.
- [5] PROFITEST 0100S-II+ operating instructions, GOSSEN METRAWATT
- [6] Earth/Ground measurement guide, Earth/Ground resistance and soil resistivity testers, Chauvin Arnoux, 2010 – Ed.1

Кратка биографија:



Боривој Јовановић рођен је у Новом Саду 1988. год. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Енергетска електроника и електричне машине одбранио је 2018. год.

REALIZACIJA PAMETNE KLOPKE ZA PRAĆENJE BROJA INSEKATA U POLJU

REALIZATION OF SMART TRAP FOR MONITORING NUMBER OF INSECTS IN THE FIELD

Ivan Hrubik, *Fakultet tehničkih nauka, Novi Sad*

Oblast – Primenjena elektronika

Kratak sadržaj – U ovom radu predstavljen je razvoj uređaja za praćenje broja odraslih jedinki raznih insekata u polju (voćnjaku ili vinogradu). Uređaj se postavlja u odgovarajuće feromonske klopke. Glavni delovi uređaja su solarno napajanje, kontroler, kamera i GSM/GPRS modem. Dobijene slike lepljive ploče iz klopke sa kamere šalju se na server gde se vrši dalja obrada.

Ključne reči: Insekti, feromonska klopka, kamera, solarno napajanje, server.

Abstract – This paper presents the development of a system for monitoring number of insects in the field (orchard or vineyard). The device is installed in appropriate pheromone traps. Main parts of the device are solar power supply, controller, camera and GSM/GPRS modem. Photos of the sticky plate from trap taken by camera are sent to server where further processing is done.

Keywords: Insects, pheromone trap, camera, solar power supply, server.

1. UVOD

Svetska populacija raste iz dana u dan i stoga konstantno raste i potreba za hranom. Jedno od rešenja ovog problema jeste povećanje poljoprivredne proizvodnje. Tržište hrane zahteva proizvode najvišeg kvaliteta, sa smanjenim hemijskim ostacima. Najefikasniji način da se to ostvari jeste smanjenje gubitaka koje nanose štetočine. Aktivnost štetočina izaziva smanjenje prinosa, promenu kvaliteta proizvoda kao i generalno smanjenje produktivnosti uzgoja. Praćenjem razvoja odraslih insekata zaprašivanje insekticidima ograničava se na vreme sazrevanja odraslih jedinki, što smanjuje upotrebu i troškove insekticida kao i količinu insekticida u ubranim usevima.

Vodenje terenskih podataka može biti pravi izazov koji može rezultirati greškama i donošenjem loših odluka. Praćenje brojnog stanja insekata vrši se brojanjem uhvaćenih, odnosno, zalepljenih jedinki u feromonskoj klopki. Feromonske klopke su zamke u kojima se hvataju odrasli mužjaci određenih vrsta insekata privučeni feromonom (supstanca koja se koristi za komunikaciju među jedinkama iste vrste) koji se onda zalepe na lepljivu površinu unutar zamke.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Rajs, docent.

Zbog ovih razloga javlja se potreba za projektovanjem uređaja koji bi automatski pravio snimke sa terena i slao ih do korisnika. Snimci mogu biti sačuvani na serveru, gde bi bili obrađeni i pomoću pametnog alata bi insekti bili prebrojani. Takođe, snimci se uvek mogu proveriti od strane relevantnih stručnjaka.

2. ANALIZA PROBLEMA

Glavni zadatak ovog uređaja je da se primenom modernih tehnologija pruža poljoprivrednom proizvođaču lako dostupna informacija od značaja. Zahtevi koji se postavljaju pri projektovanju jednog ovakvog uređaja uglavnom su diktirani samom primenom istog.

Drugim rečima, s obzirom na to da se uređaj montira na polju potrebna je mogućnost funkcionisanja u širokom opsegu temperatura i pod uticajem različitih vremenskih uslova. Takođe, jedna od najbitnijih stvari jeste da sam uređaj mora biti potpuno neinvazivan, odnosno, da ni na koji način ne utiče na okolinu.

Bitna stavka jeste i autonomnost, odnosno, samostalno napajanje uređaja. Na slici 2.1 prikazana je idejna šema uređaja.



Slika 2.1: Idejna šema uređaja

3. POSTOJEĆA REŠENJA

Projektovanjem ovakvog tipa uređaja rešavamo problem sastavljen od nekoliko celina sa više ili manje stepena slobode u izboru. U zavisnosti od vrste insekta koja se nadgleda uređajem postoje različite klopke sa različitim načinom zavaravanja – prizivanja insekata u klopku, odnosno, različitim feromonom. Dakle, tipom klopke i korišćenim feromon biramo koji tip insekta pratimo. Feromonska delta klopka je najčešći izbor za ovakav način praćenja insekata. Na slici 3.1 prikazana je klopka.



Slika 3.1: Feromonska pametna klopka

Postoje već gotova rešenja sličnih uređaja, koji služe za praćenje broja odraslih jedinki insekata, na stranom tržištu od kojih su detaljnije proučavana ova dva:

- iMetos iScout firme Pessl iz Austrije
- Trapview firme EDOS iz Slovenije

Oba uređaja su kombinacija hardverskog i softverskog rešenja za daljinsko praćenje broja različitih poljoprivrednih i industrijskih insekata. U zavisnosti od korišćene klopke i feromona različiti insekti se mogu pratiti. Kod oba uređaja možemo izdvojiti ove celine:

- Glavna jedinica – unutar vodootpornog kućišta smešteni su kontroler, GSM/GPRS modem i kamera,
- Napajanje – solarni panel sa pratećom punjivom baterijom i kontrolera punjenja,
- Feromonska klopka – slična kao na prethodno prikazanoj slici.

Fotografije ulovljenih (zalepljenih) insekata iz feromonske klopke šalju se jednom dnevno u predodređenom vremenskom trenutku. Oba uređaja imaju dovoljno dobru obradu podataka na prijemnoj strani gde su insekti prebrojani i gde se skupljaju informacije od značaja krajnjem korisniku. Mesečni troškovi samog uređaja ogledaju se u potrošenom internet saobraćaju. U tom pogledu prvi uređaj ima veću potrošnju, zbog većeg kvaliteta slike više podataka se prenese – pošalje na server.

Zbog nedostatka dobavljača pomenutih uređaja kao i nepostojanja sličnih rešenja u našoj državi došlo je do potrebe za projektovanjem uređaja.

4. ELEKTRONSKI SISTEM

Fizički deo uređaja čini feromonska klopka, nosač, kućište i elektronski sklop. Tip feromonske klopke je u suštini isti kao prethodno prikazan. Nosač je stub na

kojem je montiran solarni panel i kućište baterije, najčešće to bude već postojeći stub u polju. Najvažnija osobina kućišta baterije i elektronskog sklopa je da mora biti vodootporna. Koriste se kutije sa IP68 zaštitom.

Elektronski sklop je glavni deo uređaja. Možemo ga podeliti na četiri celine:

- Napajanje
- Kontroler
- Modem
- Kamera

Deo koji napaja uređaj sastoji se iz solarnog panela, punjive olovne baterije i solarnog kontrolera punjenja. Solarni panel snage 20W korišćen je za testiranje rada uređaja. Punjiva olovna baterija napona 12V i kapaciteta 9Ah je upotrebljena. Solarni kontroler punjenja proizvođača Phocos kontroliše punjenje baterije i detektuje eventualni nizak napon baterije te je na taj način štiti od potpunog pražnjenja.

Dve celine elektronskog sklopa zajedno se nalaze na glavnoj ploči, to su kontroler i modem. Glavna ploča predstavlja centralni deo uređaja koji izvršava glavni algoritam, odnosno, prikuplja podatke od kamere i šalje i na udaljenu lokaciju putem GPRS protokola. Na slici 4.1 prikazan je izgled glavne ploče.



Slika 4.1: Glavna ploča

Centar sistema za prikupljanje podataka predstavlja mikrokontroler PIC24FJ256GA106 sa svojim integrisanim blokovima za komunikaciju sa perifernim uređajima. Mikrokontroler je izrađen u CMOS tehnologiji sa ugrađenom programskom memorijom i memorijom za podatke. Pored velike brzine rada, ovaj mikrokontroler je izabran i zbog njegovih ostalih performansi, pre svega postojanja dovoljnog broja serijskih portova za komunikaciju sa kamerom i GSM/GPRS modemom kao i za potrebe razvoja uređaja i rešavanju problema u procesu istog (eng. *debugging*).

GSM/GPRS modem korišćen za prenos podataka je proizvođača Telit. Telit GE864-GPS modem namenjen je korišćenju u bežičnim industrijskim aplikacijama. Malih je dimenzija, lagan i lak za integraciju. Ima malu potrošnju i prošireni opseg radne temperature (od -20°C do +70°C) što ga čini dobrim rešenjem za primenu u ovakvom tipu uređaja.

Pored kontrolera i modema na glavnoj pločici smešten je i stabilizator napona. Stabilizator napona sprečava da

dnevne fluktuacije napona baterije utiču na rad samog uređaja, takođe i filtrira napon i obezbeđuje potrebne nivoe napona za kontroler, modem i kameru. Ulazni napon može biti od 6V do 36V DC.

Deo uređaja koji prikuplja podatke je kamera. Korišćena je μ CAM OV528 kamera. Zbog jednostavnosti rada sa njom, a ipak dovoljnog kvaliteta izabrana je kao optimalno rešenje. Komunikacija kamere i kontrolera je uspostavljena preko serijske komunikacije (UART - universal asynchronous receiver-transmitter). Na slici 4.2 prikazana je kamera.



Slika 4.2: Kamera

5. PROGRAMSKA PODRŠKA

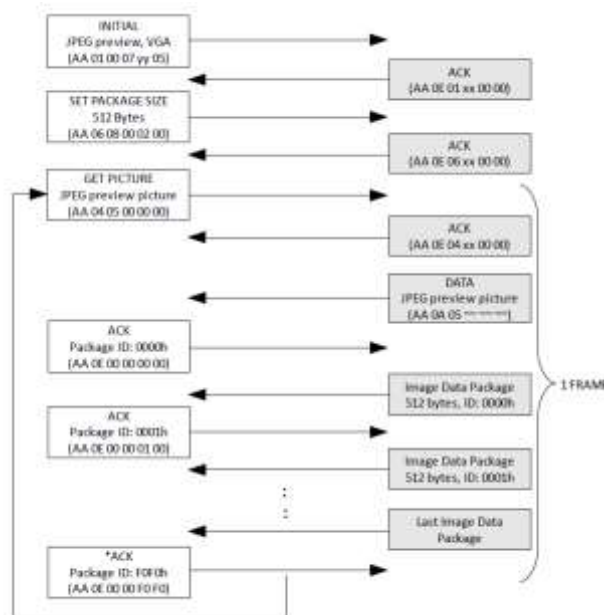
Za potrebe funkcionisanja uređaja razvijena su dva programska koda, jedan za kontroler, a drugi za Telit modem.

Kod za kontroler pisan je u C programskom jeziku prilagođenom za rad sa mikrokontrolerima. Pre početka izvršavanja glavnog dela koda podešavaju se serijski portovi. Koriste se tri serijska porta, jedan za komunikaciju sa kamerom, drugi za komunikaciju sa modemom i treći za *debug*. U kodu razlikujemo 3 stanja:

- IDLE
- TAKING PICTURE
- FORWARDING PICTURE

U *IDLE* stanju kontroler je u stanju štednje energije i praktično se ništa ne izvršava. Iz *IDLE* stanja pokreće ga modem i zadaje mu komandu da počne komunikaciju sa kamerom. Drugo stanje – *TAKING PICTURE* predstavlja deo koda u kom kontroler komunicira sa kamerom i setom komandi podešava parametre slike. Na svaku naredbu kamera vraća *ACK* kao informaciju da je prepoznala komandu. Nakon setovanja potrebnih parametara sledi preuzimanje slike sa kamere. Slika se šalje u paketima, unapred određene veličine. Prilikom preuzimanja slike sa kamere paralelno se izvršava i treća radnja, odnosno slanje slike na modem. Zbog veličine slike istu nije moguće sačuvati na kontroler, koji ima malo memorije, već se paketi prosleđuju modem, tamo spajaju i skladište u memoriju. Kad se završi preuzimanje i prosleđivanje slike kontroler naredi kameri da se

isključi, javi modem da je završio transfer slike i opet prelazi u stanje *IDLE*. Na slici 5.1 prikazan je primer seta komandi kojima kontroler komunicira sa kamerom.



Slika 5.1: Komunikacija između kontrolera i kamere

Kod za modem pisan je u Python programskom jeziku. Modem preko serijske komunikacije razmenjuje podatke i komande sa kontrolerom.

Za normalan rad modema potrebno je staviti SIM karticu u slot za SIM karticu koji se nalazi sa donje strane glavne ploče. Modem je otključan za sve operatore i može da se koristi SIM kartica bilo kog operatera u svetu. Nakon stavljanja SIM kartice potrebno je SMS porukama podesiti APN za operatera koji se koristi. Nakon početnih podešavanja, provere da li je antena dobro postavljena i provere da li na lokaciji postoji pokrivenost signalom mobilnog operatera koji se koristi modem se automatski povezuje na mrežu kad je u modu pune funkcionalnosti. Kod za modem ima 4 stanja:

- IDLE
- START
- DOWNLOADING PICTURE
- UPLOADING PICTURE

U *IDLE* stanju modem je režimu štednje energije. Nakon određenog vremenskog perioda (60 minuta) modem se pokrene iz *IDLE* stanja, poveže se na mrežu i pri tome dobije od mreže trenutno vreme. Ako je vreme za slikanje (svakog dana u 14h) prelazi u drugo stanje, a ako nije vreme za slikanje vrati se u *IDLE* stanje 60 minuta. Kad dođe vreme za slikanje (14h) modem prelazi u stanje *START* i šalje kontroleru komandu za pokretanje i sam modem prelazi u mod pune funkcionalnosti. Nakon slanja komande kontroleru modem čeka da kontroler i kamera završe razmenu komandi. Kontroler nakon toga javlja modem da je spreman za slanje slike i modem prelazi u stanje *DOWNLOADING PICTURE*. U ovom stanju paketi slike se slažu u niz jedan za drugim. Svaki paket ima svoje zaglavlje u kojem je naznačeno koji je to po redu paket. Pri slaganju paketa u niz to zaglavlje je potrebno

ukloniti. Nakon preuzimanja svih paketa zatvara se komunikacija prema kontroleru i modem prelazi u stanje *UPLOADING PICTURE*. Četvrto stanje predstavlja slanje slike na server preko GPRS protokola. Otvori se konekcija i slika se pošalje na željeni server. Nakon potvrde sa servera da su podaci uspešno primljeni konekcija se zatvara, završava se ciklus i modem prelazi u stanje *IDLE*.

Slika u ovom delu obrade je niz heksadecimalnih cifara. Heksadecimalni zapis slike se na serveru pretvara u sliku, koja može vizuelno da se prestavi, pomoću programa koji je deo obrade podataka na serverskoj strani.

6. TESTIRANJE

Projektovan i realizovan uređaj je na kraju testiran. Prilikom testiranja slike veličine oko 20 KB su stizale do servera bez grešaka i gubljenja podataka.

Na slici 6.1 prikazan je deo heksadecimalnog zapisa podataka slike, a na slici 6.2 prikazana je dobijena slika uz tog zapisa.

[illegible]

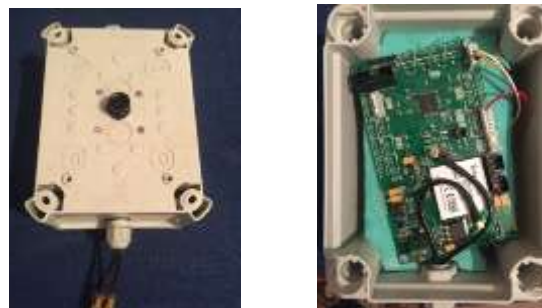
Slika 6.1: Heksadecimalni zapis podataka koji su stigli na server



Slika 6.2: Slika tokom testiranja

U programu može da se podesi vreme slanja podataka (slika), takođe i broj ponavljanja dnevno. Slanje slike traje 2-3 minuta i tada je potrošnja energije najveća, oko 400mA sa napajanjem 12V. Ostatak vremena uređaj je u stanju uštede energije i potrošnja drastično opadne, pa baterija stigne da se dopuni do sledećeg ciklusa.

Na slici 6.3 prikazan je uređaj upakovan u kućište. Na prvom delu slike može se uočiti donji deo kućišta i kamera, a na drugom delu vidi se glavna ploča.



Slika 6.3: Uređaj za praćenje broja insekata u polju

7. ZAKLJUČAK

Problem definisan projektnim zadatkom je uspešno rešen i dobijena slika omogućuje korisniku da odradi potrebno prebrojavanje insekata koji bi se našli na klopoci.

Unapređenje uređaja odnosilo bi se na korišćenje kvalitetnije kamere – veća rezolucija i bolji kvalitet slike. Veći kvalitet slike povećao bi i količinu podataka koji se šalju te bi to zahtevalo korišćenje novijeg modema koji ima 3G ili 4G mogućnosti radi bržeg slanja podataka.

8. LITERATURA

[1] Microchip, PIC24FJ256GA106, dostupno na: <http://pdf1.alldatasheet.com/datasheet-pdf/view/394556/MICROCHIP/PIC24FJ256GA.html> - 15.09.2018.

[2] 4D System, uCAM III, dostupno na : https://www.4dsystems.com.au/productpages/uCAM-III/downloads/uCAM-III_datasheet_R10.pdf - 15.09.2018.

[3] Telit, GE864-GPS, dostupno na: <https://www.telit.com/m2m-iot-products/cellular-modules/standard-industrial-grade/ge864ge865-series/#ge864> – 15.09.2018.

[4] Prof.E.D.Morgan, Feromoni, dostupno na: <http://www.znanje.rs/35.html?FEROMONI> – 15.09.2018.

[5] EFOS d.o.o., Trapview, dostupno na: <http://www.trapview.com/en> - 15.09.2018.

[6] Pessl Instruments GmbH, iMETOS iSCOUT, dostupno na: <http://metos.at/home/iscout/> - 15.09.2018.

Kratka biografija:



Ivan Hrubik rođen je u Novom Sadu 1991. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Primenjene elektronika odbranio je 2015.god. Oblasti interesovanja su mu embeded sistemi, obrada signala, primenjena elektronika i IoT.

**KOMPENZACIJA REAKTIVNE SNAGE I COST BENEFIT ANALIZA NJENOG
ULAGANJA****COMPENSATION OF REACTIVE POWER AND COST BENEFIT ANALYSIS IN OF ITS
INVESTMENT**Marko Nedić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je razmatran pojam kompenzacije reaktivne snage i benefita ulaganja u primenu sistema za kompenzaciju. Pre svega, data je teoretska osnova i pojašnjenje reaktivne snage. Definisana je kompenzacija reaktivne snage, način određivanja potrebne snage kompenzacije, kao i način kompenzacije reaktivne snage kroz sve tipove. Dati su uređaji koji učestvuju u kompenzaciji, što kao analizatori, što kao sami kompenzatori. Teoretski osnov i pojam Cost – benefit analize (CBA) su dati uz detaljne načine izrade kvalitetne CBA. Opisan je instrument Metrel PowerQ4 koji je korišćen za praktični deo u radu. Jasni i detaljni koraci za pripremu i način snimanja i korišćenja instrumenta u eksploataciji, koji su poslužili i kao osnova prilikom korišćenja za samo merenje na predmetnom sistemu. Rezultati merenja na predmetnom sistemu dati su uz CBA u konkretnom slučaju.

Glavne reči: kompenzacija reaktivne snage, Cost-benefit analiza, mrežni analizator.

Abstract – In this paper the term of reactive power compensation and benefits of investment in the application of the system of compensation are considered. First of all, the theoretical basis and a clarification about the power, active and reactive is given. Definition and description of reactive power compensation, a way of determining the power necessary compensations, as well as a way to compensate reactive power through all types is given. Devices that participate in the process, as analyzers and as expansion joints are described. The theoretical foundation of the concept of Cost-benefit analysis is given. An instrument named Metrel PowerQ4 is used for the practical part of the paper. Clear and detailed steps for the preparation and shooting and using the instrument in exploitation are presented as well.

Keywords: Reactive power compensation, Cost-benefit analysis, network analyzer.

1. UVOD

Napon je lokalna karakteristika mreže i može se razlikovati u sistemu od tačke do tačke. Neki od razloga tih razlika mogu biti postojanje više naponskih nivoa unutar sistema, kao i pojava pada napona prouzrokovana tokovima reaktivne snage kroz provodnik.

NAPOMENA:

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

U elektroenergetskom sistemu (EES) dopuštena su minimalna odstupanja, pa se iz tog razloga primenjuju razne metode regulacije. Kako je veza između napona i reaktivne snage očita, oni su obuhvaćeni zajedničkim sistemom regulacije [1].

Od današnjih EES se zahteva racionalizacija proizvodnje i potrošnje, odnosno sve efikasnije funkcionisanje svih njegovih delova. Jedan od efikasnih načina za postizanje tog cilja je i kompenzacija reaktivne snage, odnosno eliminacija nepotrebnog prenosa reaktivne snage kroz sistem [1]. Osnovni zadatak regulacije reaktivnih snaga u normalnom pogonu je održavanje napona u svim čvorovima EES kroz održavanje ravnoteže između proizvodnje i potrošnje reaktivne snage i gubitaka.

Sa stanovišta konkretnog potrošača, ekonomska opravdanost kompenzacije, se ogleda u uštedama koje se postižu kroz neplaćanje reaktivne snage, smanjenje gubitaka energije i poboljšanje naponskih prilika i iste uveliko premašuju troškove investicija u uređaje za kompenzaciju [1]. U tu svrhu elektroprivredne kompanije nastoje da tarifnim sistemima stimulišu potrošače na primenu kompenzacije. U tržišnom okruženju postojeći kupci, koji nepovoljno deluju na EES i koji štete kvalitetu električne energije, moraću dodatno da plaćaju, dok novi kupci moraju da zadovolje odgovarajuće standarde pre samog priključenja na sistem [1].

2. KOMPENZACIJA REAKTIVNE SNAGE

Kompenzacijom se vrši rasterećenje postojećih izvora energije i prenosnih puteva, pa postojeća oprema za proizvodnju i prenos energije može da proizvede i prenese energiju većom aktivnom snagom.

2.1 Kompenzacija reaktivne snage

Za kompenzaciju reaktivne energije, odnosno reaktivne snage u elektroenergetskim postrojenjima se koristi sledeća oprema [1]:

- Obrtni kompenzatori (sinhroni kompenzatori i sinhroni motori);
- Kondenzatori (povezani serijski ili otočno, u pojedinačnoj, grupnoj, centralnoj ili mešovitoj vezi, kao i dinamička kompenzacija u direktnoj ili indirektnoj vezi);
- Pasivni filteri (usklađeni i neusklađeni);
- Aktivni filteri (serijski i otočni).

2.2 Određivanje potrebne snage za kompenzaciju

Potrebnu snagu kompenzacije moguće je odrediti pomoću faznog dijagrama, grafički. Ukoliko koristimo mrežni analizator, onda je moguće odrediti potrebnu snagu računski ili na osnovu pokazivanja postojećih brojeva aktivne i reaktivne energije u određenom vremenskom periodu, ukoliko mrežnim analizator nije na raspolaganju. Relativno lako, potrebna snaga kondenzatora može da se odredi i na osnovu mesečnih računa za utrošenu električnu energiju. Potrebna snaga se može dati i procenom, ukoliko se radi o novoprojektovanom sistemu.

2.3 Sinhroni kompenzatori i sinhroni motori

Kada su industrijski potrošači velikih snaga u pitanju, ekonomski je opravdano da se investira u postrojenje za kompenzaciju čiji je rad baziran na primeni obrtnih mašina. U tom slučaju, potrebno je razlikovati slučaj kada je predviđeno da sinhrona mašina isključivo radi kao izvor reaktivne snage (tada se ona naziva sinhroni motor u praznom hodu) i slučaj kada mašina radi kao normalno opterećen (sinhroni) motor [1].

2.4 Kompenzacija reaktivne snage korišćenjem kondenzatora

Kompenzacija reaktivne snage u EES može se izvesti korišćenjem serijskih i otočnih kondenzatora. Na sistemu koji je predmet ovog rada, primenjena je metoda kompenzacije otočnim kondenzatorskim baterijama. Postoje četiri moguća načina za realizaciju otočne kompenzacije [1]: pojedinačno, grupno, centralno, ili u mešovitoj vezi.

2.5 Savremeni regulatori reaktivne snage

Danas se regulacija reaktivne snage postrojenja za kompenzaciju vrši pomoću savremenih digitalnih regulatora. Oni imaju ugrađenu tehniku za odlučivanje u vezi sa uključivanjem i isključivanjem kondenzatora, tako da su u najkraćem vremenskom roku zadovoljene potrebe za reaktivnom snagom, uz minimiziranje broja uključenja/isključenja kompenzacionih sekveneci i vođenja računa o njihovom ravnomernom korišćenju. Regulatori reaktivne snage definišu broj stepeni regulacije, poredak snaga kompenzacije i broj koraka regulacije.

2.6 Uključivanje i isključivanje kondenzatora

Uključivanje kondenzatora na mrežu je sve do 80-tih godina vršeno pomoću vazdušnih prigušnica i strujno predimenzionisanih elektromehaničkih kontaktora, dok se danas isključivo koriste specijalni kontaktori za uključivanje kondenzatorskih baterija, takozvani kontaktori sa ranim dejstvom i brzim pražnjenjem [1].

2.7 Mrežni analizatori

Pojam mrežni analizator se odnosi na posebne merne instrumente koji su predviđeni za merenje različitih parametara električnih mreža. U suštini, to su digitalni mikroprocesorski instrumenti, koji na osnovu odbiraka napona i struja izračunavaju efektivne vrednosti napona i struja, aktivnu i reaktivnu snagu, ali i višeharmonijske komponente ovih veličina. Instrument je prenosnog tipa i meri, prikazuje i memoriše glavne parametre jednofazne ili trofazne električne mreže.

3. CBA

Da bi se proces investiranja i ocene opravdanosti realizacije jednog investicionog projekta mogli sagledati, potrebno je utvrditi i analizirati ukupne efekte koje realizacija određene investicije donosi. U ovom delu su prikazani osnovni elementi CBA za ocenu investicionih projekata i opravdanosti investiranja u cilju unapređenja predmetnog sistema [2].

3.1 Uvod u CBA

CBA je metoda ekonomske analize kojom se upoređuju i vrednuju sve prednosti i svi nedostaci nekog privrednog poduhvata ili projekta analizom troškova (cost) i koristi (benefit). Izuzetna kompleksnost problematike CBA, kao i njene adekvatne primene u praksi, zahteva poznavanje nekoliko značajnih problema koje će u nastavku biti razmotrene, a to su [3]:

- Utvrđivanje troškova i koristi investicionog projekta;
- Vrednovanje troškova i koristi investicionog projekta;
- Osnovne faze izvođenja CBA;
- Kriterijumi koji se koriste u CBA.

3.2 Utvrđivanje troškova i koristi

Primenu CBA posebno preporučuje Svetska banka. S tim u vezi, ona sugerise da se pri utvrđivanju troškova i koristi, za celo društvo (ekonomska analiza), pođe od efekata koje konkretan projekat daje investitoru (finansijska analiza) i da se uključivanjem ili isključivanjem određenih grupa troškova i koristi dođe do ukupnih rezultata sa aspekta društva u celini. U tom smislu, sugerise se pristup prema sledećim elementima [2]: transferna plaćanja, nepredviđeni troškovi, prethodni troškovi, eksterni efekti, multiplikacioni efekti, međunarodni efekti.

3.3 Vrednovanje troškova i koristi

Nakon utvrđivanja društvenih troškova i koristi koje jedan investicioni projekat donosi, potrebno ih je sve vrednovati i izraziti u novčanom obliku. U tu svrhu primenjuju se metode za prevođenje raznovrsnih efekata u novčane izraze. Postoje dva osnovna metoda u određivanju obračunskih cena, to su [4]:

- Little Mirrleesov metod (LM metod) i
- UNIDO metod.

3.4 Faze procesa CBA

CBA predstavlja veoma složen i najčešće dosta obiman proces, sa mnogo procena, izračunavanja, predviđanja i poređenja. Zbog toga, primena CBA u oceni investicionih projekata, zahteva da se poštuje određena procedura zasnovana na nekoliko osnovnih faza [2]:

- definisanje projekata za analizu,
- definisanje vremenskog perioda na koji se analiza odnosi,
- utvrđivanje svih koristi i troškova za pojedine projekte,
- izračunavanje svih koristi i troškova u novčanom obliku za pojedine projekte,
- određivanje kriterijuma za analizu,
- određivanje diskontne stope koja će se koristiti,

- izračunavanje vrednosti pojedinih kriterijuma za svaki projekat,
- upoređivanje vrednosti kriterijuma za pojedine projekte sa određenom merom,
- određivanje dodatnih kriterijuma analize,
- konačan izbor - donošenje investicionih odluka.

3.5 Kriterijumi CBA

Osnovni princip CBA nalazi da je projekat za realizaciju opravdan jedino ako je ukupna korist koju on donosi, veća od očekivanih troškova. Ovaj princip se koristi i pri definisanju odgovarajućih kriterijuma za ocenu efikasnosti investicionih projekata, naravno uz respektovanje i ostalih elemenata, koje je neophodno, u tim slučajevima, uzeti u obzir.

Kod ocenjivanja investicionih projekata, primenom CBA moguće je koristiti veći broj kriterijuma za ocenu. Ovde će biti prezentovana četiri kriterijuma [4]:

- Kriterijum sadašnje vrednosti neto koristi,
- Kriterijum interne stope rentabilnosti,
- Kriterijum odnosa koristi i troškova,
- Kriterijum roka vraćanja investicije.

4. PRIMER PRAKTIČNE PRIMENE KOMPENZACIJE REAKTIVNE SNAGE

U delu industrijskog pogona jedne od najvećih fabrika za preradu voća i povrća na Balkanu, na računu za električnu energiju, primećena je pojava veće količine reaktivne energije.

Pribavljen je mrežni analizator pomoću kojeg se pristupilo snimanju stanja sistema, praćenju potrošnje aktivne i reaktivne snage, praćenju promena vrednosti faktora snage, napona, struja i harmonika. U toku merenja, minimalna izmerena vrednost faktora snage je bila 0.530 ind.

Usledila je nabavka i ugradnja 12 kondenzatorskih baterija austrijskog proizvođača Schrack nominalne snage 12.5 kVAr, odnosno ukupne snage 150 kVAr. U nastavku je naveden uređaj kojim je izvršena analiza i merenje prilika, analiza ekonomskih prilika, opis opreme koja je ugrađena i krajnji, ekonomski efekat celog procesa.

4.1 Mrežni analizator Metrel PowerQ4 MI 2592

PowerQ4, slika 1, je multifunkcionalni ručni instrument za merenje kvaliteta snage i električne efikasnosti. Tehnički podaci su:

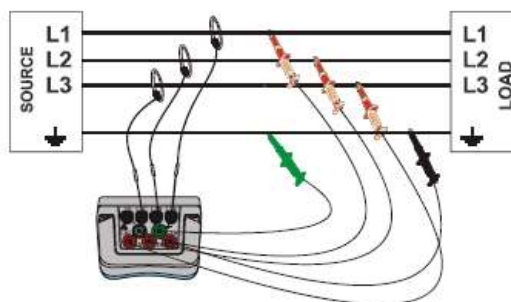
- Broj kanala: 4,
- Dimenzija uređaja, visina: 90 mm, širina 115mm,
- Težina: 650 g.

Koraci za pripremu i način snimanja i korišćenja instrumenta u eksploataciji su: podešavanje instrumenta, podešavanje merenja, provera, merenje on-line, podešavanje snimača i snimanje, zaključak merenja i generisanje izveštaja. Instrument je moguće povezati na monofaznu i trofaznu mrežu.

Predmetni sistem je trofazni sistem sa 3 žice, a na slici 2 je prikazano kako je instrument bio povezan prilikom merenja.



Slika 1 - Instrument PowerQ4 [5]

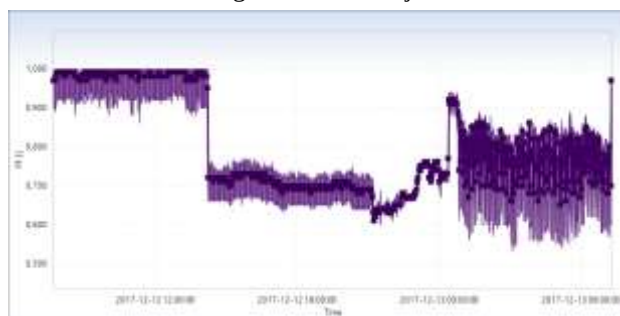


Slika 2 – Povezivanje uređaja na trofazni sistem sa 3 žice

4.2 Merenje u delu proizvodnog pogona

Merenje je vršeno u toku jednog proizvodnog ciklusa. Praćene su vrednosti napona, struje i frekvencije, aktivna i reaktivna snaga, faktor snage i harmonici. Vrednosti su praćene po fazama, ali i međufazno. Merenje je trajalo 23h i 26m, odnosno snimljeno je 708 intervala, sa periodom od 2m, odnosno 120s. Uređaj je podešen na nominalni napon 230V. Merenje možemo podeliti na dva dela. U prvom delu, kompenzacija reaktivne snage je bila aktivna, dok je u drugom delu merenja, kompenzacija bila deaktivirana. Nakon završenog merenja, podaci su preuzeti na računar i očitani uz pomoć softvera PowerView. Vrednosti iz samog softvera za sve parametre moguće je prikazati grafički i tabelarno.

Tokom merenja, vrednosti linijskih napona su bile konstante, tako da nije bilo nepovoljnih naponskih prilika. Isti je slučaj i sa frekvencijom, koja je bila u opsegu od 49.900 Hz do 50.090 Hz. Faktor snage je na samom početku merenja imao vrednost 0.970 ind. U nekoliko momenata, ta vrednost ide i do 1,00 ind., dok je prosečna vrednost 0.990 ind. U drugom delu merenja, vrednost faktora snage značajno pada sa 0.970 ind. na 0.720 ind. Minimalna vrednost faktora snage u toku drugog dela merenja je 0.530 ind. Na slici 3 je dat grafički prikaz vrednosti faktora snage tokom merenja.



Slika 3. Vrednost faktora snage u toku merenja

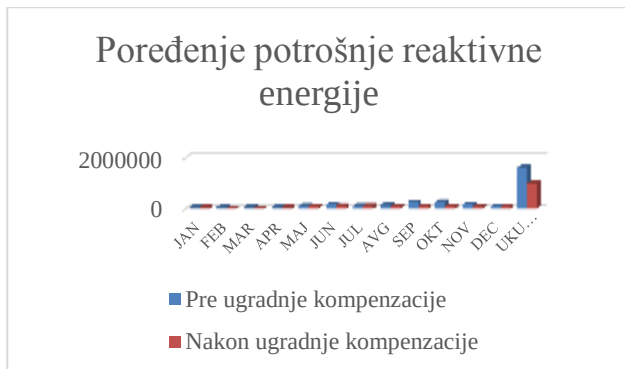
4.3 CBA na konkretnom primeru

Nakon praćenja potrošnje reaktivne energije preko računa za električnu energiju, primećeni su značajni izdaci za istu. Pristupilo se detaljnoj analizi i u toku prvih 6 meseci kalendarske godine praćene su potrošnje i finansijski izdaci. Na računu za električnu energiju za period u kojem se pratilo stanje svih parametara, a posebno stavka potrošene reaktivne snage i faktora snage, dobijeno je da je za taj period potrošeno 399806 kVArh, a faktor snage je u proseku bio 0.987. Odlučeno je da se uđe u investiciju i pokrenut je proces nabavke kondenzatorskih baterija. Nabavljeno je 12 kondenzatorskih baterija, svaka snage 12.5 kVAr. Već u novembru, kada je kompenzacija puštena u rad, primećeno je znatno smanjenje potrošnje reaktivne energije, kao i promena vrednosti faktora snage koji je sa 0.951 ind. smanjen na 0.974 ind.

U ovoj kalendarskoj godini, potrošeno je 1631972 kVArh reaktivne energije. Praćenje je nastavljeno i u toku sledeće kalendarske godine i dobijen je podatak da je u istoj potrošeno 1010308 kVArh reaktivne snage.

Na slici 4 jasno se vidi smanjenje potrošnje reaktivne energije nakon ugradnje kompenzacije. Ukupna potrošena reaktivna energija je sa 1631972 kVArh smanjena na 1010308 kVArh. Ukoliko se uzme da operater naplaćuje 1 kVArh po ceni od 0.428 RSD, dolazimo do podatka da je već u prvoj godini eksploatacije na ime investicije u kompenzaciju reaktivne energije uštedeno 266.072,19 RSD, što je jednako vrednosti celokupne investicije.

Taj podatak je izuzetno uspešan, jer je praksa i nepisano pravilo da su investicije koje se vrte u prvih pet godina eksploatacije izuzetno isplative.



Slika 4 - Poređenje potrošnje reaktivne energije pre i nakon ugradnje kompenzacije reaktivne energije

Bitno je reći da je prosečna vrednost faktora snage u periodu pre ugradnje kompenzacije bila 0.978 ind, dok je u periodu nakon ugradnje, njegova vrednost poboljšana i iznosila je 0.989 ind.

4.4 Opis opreme koja je ugrađena

Primenjena je centralna kompenzacija uz pomoć otočno povezanih kondenzatorskih baterija ukupne snage 150 kVAr. Upravljanje je potpuno automatizovano, regulisano preko reaktivnog regulatora RM9606. Uključivanje i isključivanje kondenzatorskih baterija je obezbeđeno preko kondenzatorskih kontaktora, slika 5.



Slika 5 - Ugrađene kondenzatorske baterije i kondenzatorski kontaktori

5. ZAKLJUČAK

Današnja srednja i velika preduzeća pritisnuta su sve strožijim zahtevima. Sa jedne strane je podizanje kvaliteta proizvodnje, a sa druge minimizacija troškova. Da bi to ostvarili primorani su da kroz „mini“ projekte, u kratkom vremenskom periodu, raspoznaju benefite i isplativost svojih investiranja u cilju da te investicije direktno podstiču napredak preduzeća i podižu kvalitet proizvoda. Investiranjem u opremu koja je opisana, prati se trend i tendencija smanjenja troškova.

To se posebno odnosi na investiciju koja se, kao i ovde opisana, smatra izuzetno isplativom.

6. LITERATURA

- [1] M.B.Kostić: *Teorija i praksa projektovanja električnih instalacija, Drugo, prošireno izdanje*, Akademska misao, Beograd, 2005.
- [2] P.Jovanović: *Upravljanje investicijama*, Fakultet organizacionih nauka, Beograd, 2006.
- [3] B.Mašić: *Strategijski menadžment*, Univerzitet Singidunum, Beograd, 2009.
- [4] M.Ivaniš, S.Slović: *Uloga i značaj cost-benefit analize u investicionom menadžmentu*, Pravno-Ekonomski Pogledi br. 1/2013.
- [5] Metrel, *PowerQ4, MI 2592, Instruction manual, Version 1.2, Code No. 20 751 551*, 2009.

Kratka biografija:



Marko Nedić rođen je u Sremskoj Mitrovici 1991. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranio je 2014. god. Iste godine upisao se na master studije na istom smeru.

ОПТЕРЕЋЕЊЕ ДИЗЕЛ ЕЛЕКТРИЧНОГ АГРЕГАТА**DIESEL ELECTRIC GENERATOR LOAD**

Дејан Старчевић, Веран Васић, Ђура Орос, *Факултет техничких наука, Нови Сад*

Област – ЕНЕРГЕТСКА ЕЛЕКТРОНИКА И ЕЛЕКТРИЧНЕ МАШИНЕ

Кратак садржај – У раду је описан основни принцип рада и функционисања дизел електричног агрегата. У оквиру рада имамо увид у детаљан приказ техничких карактеристика, начина рада као и резултата огледа оптерећења са посебним акцентом на специфични систем побуде синхроног генератора у оквиру дизел електричног агрегата.

Abstract – The basic principle of operation and functioning of diesel electric generator is described. In the framework of the paper, we have an insight into the detailed overview of technical characteristics, working methods and load retrieval results with a special emphasis on the specific system of the synchronous generator initiative within the diesel electric generator.

Кључне речи: Дизел мотор, Синхрони генератор, Оглед оптерећења

1. УВОД

Међународни стандард ISO 8258-5:2013 дефинише термине и специфицира критеријуме који настају комбинацијом (спрегом) мотора са унутрашњим сагоревањем и генератора наизменичне струје као једне оперативне јединице. Стандард се односи на генераторе наизменичне струје погоњене мотором са унутрашњим сагоревањем за примену у копненим и поморским условима искључујући генераторе који се користе унутар авиона или за погон локомотива или пропелера. За одређене специфичне примене (болничке зграде или објекте високих захтева у погледу напајања), могу бити захтевани додатни услови. ISO 8258-5:2013 представља основни стандард на који се додају додатни услови. За агрегатске сетове који су погоњени другим типом погонског горива (парни мотори), овај стандард представља основу на основу које се формирају додатни услови [12].

2. ОПТЕРЕЋЕЊЕ ДИЗЕЛ ЕЛЕКТРИЧНОГ АГРЕГАТА

У овом делу представљени су елементи односно уређаји који су били коришћени приликом спровођења огледа оптерећења дизел електричног агрегата. Осим описа техничких елемената пружен је увид и у амбијенталне услове.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био проф. др Веран Васић.

Сви параметри и компоненте наведене у овој области су неопходне како би имали што релевантнији приступ огледу.

2.1. ОПИС ОПТЕРЕЋЕЊА ДИЗЕЛ ЕЛЕКТРИЧНОГ АГРЕГАТА

У оквиру фабричког испитивања примењен је мотор са унутрашњим сагоревањем произвођача VOLVO серије TAD1642. Примењен је генератор наизменичне струје произвођача WEG серије AG315M160AI са њему одговарајућим моделом аутоматског регулатора напона (у даљем тексту АРН) и специјализовани контролер за мониторинг и заштиту ДЕА произвођача DEIF серије CGC413. Извршени оглед оптерећења вршен је у склопу фабричког тестирања ДЕА са циљем завршне контроле квалитета произведеног сета.

Амбијентална вредност температуре у тренутку реализације огледа износила је $T_a = 25[^\circ\text{C}]$ док је надморска висина износила $h = 85 [\text{mnv}]$. Приликом оваквих амбијенталних услова није потребно извршити никакве додатне мере предострожности приликом извођења огледа оптерећења. Додатне заштитне мере потребно је извршити за агрегате који се испитују или се инсталирају на надморским висинама изнад $1000 [\text{mnv}]$ или су изложени амбијенталним температурама изнад $40[^\circ\text{C}]$.

Временски период огледа оптерећења износи сса. $7 [\text{min}]$. Временско трајање огледа динамичког оптерећења није у складу са поменути стандардом али се тежи ка њиховом усаглашавању. Потребно је нагласити да се временски период огледа оптерећења знатно разликује и продужава током зимског периода. Табеларни подаци на основу којих су формиране функције зависности добијене су директним повезивањем са DEIF CGC413 контролером уз помоћ постојања опције мониторинга (енг. trending) која омогућава приступ и манипулацију добијеним вредностима огледа оптерећења. Подешени период усредњавања добијених вредности износи $1 [\text{s}]$.

Приликом реализације огледа оптерећења обезбеђен је уређај који има улогу резистивног оптерећења (Load Bank Crestich 700kW, $\cos\phi=1$). У складу са тим, оптерећење по фазама као и вредности међуфазних напона су потпуно уравнотежене. Напајање резистивног оптерећења увек мора бити уземљено и напајано путем посебног извора електричне енергије. Потреба за посебним напајањем постоји услед постојања вентилатора у склопу резистивног

оптерећења који има улогу форсираног хлађења отпорничких елемената. Постоје случајеви где се ова врста тестирања спроводи повезивањем резистивног отпора на агрегатско напајање, што потенцијално може представљати технички проблем јер у случају хаваријског стања, кратког споја генератора, при номиналном оптерећењу заштита дизел електричног агрегата ће одрадити док би хлађење резистивног оптерећење тренутно престало са радом. Наглим престанком рада вентилаторског система престаје се са форсираним одвођењем топлоте са резистивног оптерећења што може проузроковати пожар или оштећење уређаја услед прегревања.

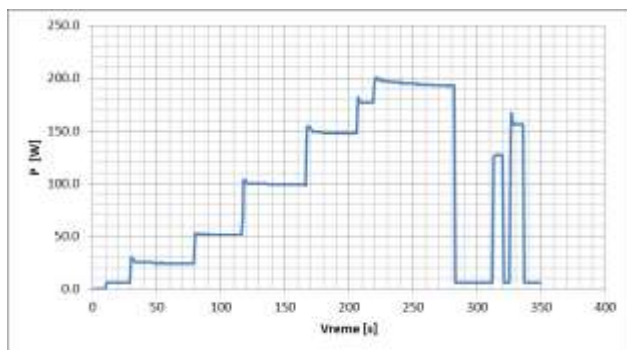
Евентуални несклад у случају оптерећења резистивним елементима, може се очекивати у случају хаваријског стања дизел електричног агрегата или услед недовољно оптимизованих параметара аутоматског регулатора напона. Повезивање овог уређаја и дизел електричног агрегата се остварује путем ширмоване конекције која се завршава USB прикључком.

3. РЕЗУЛТАТИ ОПТЕРЕЋЕЊА ДИЗЕЛ ЕЛЕКТРИЧНОГ АГРЕГАТА

У овом поглављу графички ће бити приказан оглед оптерећења дизел електричног агрегата са акцентом на електрични и неелектричне параметре од интереса.

3.1 ГРАФИЧКИ ПРИКАЗ ОПТЕРЕЋЕЊА ДИЗЕЛ ЕЛЕКТРИЧНОГ АГРЕГАТА

Оглед оптерећења увек почиње без прикључивања резистивног отпора (оглед празног хода) и има намену да прати вредности неелектричних параметара ДЕА (притисак, температуру издувног лонца, температуру блока мотора и др.), механичку повезаност ДЕА (посебно се мора обратити пажња на спојни диск агрегата који у случају неадекватног притезања има тенденцију великог загревања и уништења, издувну грану ДЕА и спојеве агрегатског кућишта) и електричне параметре (ово се посебно односи на вредности међуфазних напона ДЕА, у случају неуравнотежене вредности напона потребно је извршити оптимизацију АРН-а како би се сачували прикључени потрошачи). Важно је нагласити да се било која оптимизација АРН-а врши у безнапонском стању.



Слика 3.1.1 Графички приказ динамике оптерећења у функцији времена

График приказан на слици 3.1.1 даје графички приказ наметнутог оптерећења у складу са дефинисаном номиналном снагом ДЕА. Тестирани агрегат је намењен за рад као подршка при нестанку напајања (*stand by* режим), те стога вредности оптерећења већег од номиналног нису дозвољене приликом његове експлоатације.

Тест динамичког оптерећења се састоји из корака који су дефинисани у складу са приказаним графиком. Временски распон између повећања односно смањења резистивног оптерећења износи, за агрегатско напајање, веома кратак период (ска. 40[s]).

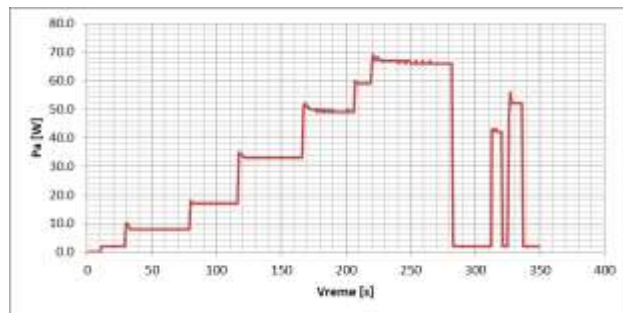
Првобитно, у временски приближно једнаким корацима, приликом којих се тежи да се усагласи са дефинисаним стандардом, повећавамо вредност резистивног отпора до номиналне вредности снаге ДЕА. Затим се врши растерећење до вредности блиске вредностима празног хода а након тога опет се нагло врши оптерећење агрегата до ска. 75% вредности његове номиналне снаге.

Након првобитног шока, опет се понавља исти циклус у још краћем временском периоду (4[s]). Наглим оптерећењем а затим растерећењем симулира се прихват терета у најгорој могућој ситуацији где се агрегат оптерећује са 100% називне снаге.

Препорука од стране произвођача ДЕА на основу емпиријских искустава је да није пожељно оптерећивати агрегат више од 70% његове називне снаге у дужем временском периоду.

Овом врстом испитивања доприноси се испитивању динамичких карактеристика агрегата. Управо из тог разлога овај тест и носи назив динамички оглед оптерећења.

Осим динамичког постоји и статички тест који подразумева веома слично оптерећење ДЕА са много већим временским корацима (ска. 30[min]). Због великих временских корака овај вид испитивања неће бити предмет овог рада.



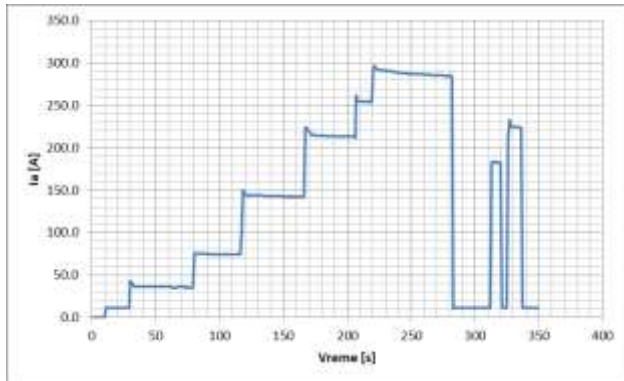
Слика 3.1.2 Оптерећење фазе а

График приказан на слици 3.1.2 репрезентује оптерећење фазе а синхроног генератора. Услед прикључења резистивног терета оптерећење по фазама је потпуно уравнотежено те не постоји додатна потреба за описивањем фазе б и фазе с. Вредност фазних снага је у складу са трофазним уравнотеженим системом синхроног генератора.

Приликом задавања нове референце примећује се да је одзив готово увек већи од задате вредности. Разлог се може пронаћи у подешавању PID регулатора.

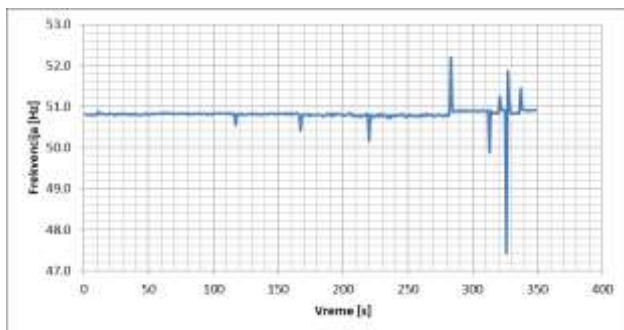
У складу са одабраним вредностима појачања регулатора приликом овог теста одабрана природа одзива је осцилаторна са пригушењем.

Одабир оваквог вида одзива је заступљен у великом броју случајева док је апериодичан одзив PID-а заступљен углавном приликом напајања осетљиве опреме (медицинска опрема, прецизни алати у индустрији и др.).



Слика 3.1.3 Струјно напрезање фазе а

График 3.1.3 представља струјно оптерећење. Струјно оптерећење фазе b и фазе c, услед повезивања резистивног отпора је потпуно исто са вредношћу фазе а. Вредности номиналних струја у огледу оптерећења у складу су са наметнутим резистивним оптерећењем. Релевантним струјним одзивима добија се и одређени вид потврде изабраног прекидача синхроног генератора.

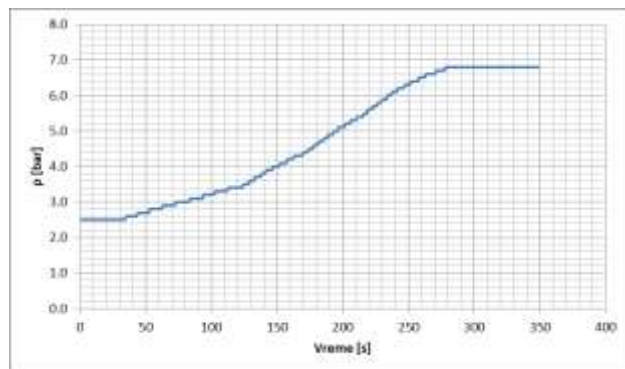


Слика 3.1.4 Фреквенција напона

График приказан на слици 3.1.4 представља фреквенцију напона синхроног генератора. Приликом самог почетка теста може се приметити почетно одступање од $50 \pm 0,8$ [Hz] које је подешено у складу са емпиријским искуством апликационих инжењера. У случају напајања осетљиве опреме потребно је приликом првог пуштања у рад извршити додатна подешавања.

На основу графика са слике 3.1.4 примећује се да вредност фреквенције варира приликом промене терета где се приликом повећања оптерећења има краткотрајан пад фреквенције а приликом растерећења (смањење терета) нагли скок фреквенције.

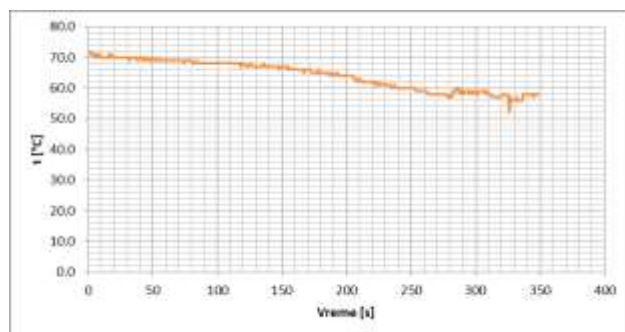
Промене вредности фреквенције су изражене приликом наглих промена оптерећења. Регулацију фреквенције напајања врши ECU (Electronic Control Unit) или већ поменути EDC уређај путем прецизне регулације дотока горива у цилиндри дизел мотора.



Слика 3.1.5 Графички приказ притиска у блоку мотора

За разлику од електричних величина које су од интереса приликом проучавања рада ДЕА график приказан на слици 3.1.5 представља графички приказ пораста притиска у функцији времена. Овај пораст притиска је експоненцијалне природе до тренутка стабилизације вредности притиска. Стабилизација радног притиска се успоставља након окидања прекидача унутар блока мотора (switch eng.) Граничне вредности притиска су дефинисане у склопу параметрисања (програмирања) специјализованог контролера при чему су од посебног значаја доње границе радног притиска приликом којих долази до готово моменталног окидања прекидача унутар блока мотора и моменталног престанка рада ДЕА. Минималне доње границе радног притиска у великом броју случајева износе око 1,5 [bar] и доприносе потенцијалном великом оштећењу дизел мотора.

У поређењу са другом неелектричном величином која је од интереса приликом извођења огледа оптерећења (други неелектрични параметар од интереса нам је температура) притисак је након сваког упуштања у рад временски је променљива по истој функцији. Програмски дефинисан радни притисак у обрађеном огледу износи 6,8 [bar].



Слика 3.1.6 Графички приказ температуре блока мотора

График приказан на слици 3.1.6 представља промену неелектричне величине, тј. температуру у функцији времена. Мерно место у овом случају представља блок мотора, док постоје случајеви где то може бити издувна грана (у том случају у питању су много веће температуре). Приликом старта огледа статичког оптерећења, који није предмет овог рада, вредности температуре су знатно мање и у складу су са амбијенталном вредношћу температуре. Одржавање вредности температуре блока мотора, посебно долази

до изражаја у екстремним условима (зимском и летњем периоду) приликом којих се одговарајућим подешавањем утиче на форсирану циркулацију антифриза кроз канале блока мотора уз помоћ адекватне пумпе у зимском периоду, или уз помоћ додатног хлађења у летњем периоду (што је врло ретко и углавном се реализује уграђивањем вентилатора који је намењен за тропске услове рада). Потребно је нагласити да пумпе које се користе за форсирано загревање блока мотора у хладним зимским данима морају бити повезане на екстерно напајање како би се број неуспелих покушаја стартавања свео на минималан.

Током огледа оптерећења вредност температуре рапидно опада до програмски дефинисане вредности (58 [°C]) посредством уграђеног термостата и великом броју случајева посредством додатне пумпе помоћу које се врши форсирање циркулације расхладног флуида.

4. ЗАКЉУЧАК

Дизел електрични агрегати представљају савремено техничко решење приликом пројектовања било ког система у смислу повећања степена поузданости напајања. У оквиру овог рада извршено је оптерећење дизел електричног агрегата у складу са његовим номиналним оптерећењем при чему се има увид у графички тренд електричних и неелектричних величина на основу којих се може о техничкој исправности дизел електричног агрегата.

На основу оптерећења извршеног у оквиру овог рада има се увид у динамичке карактеристике дизел електричног агрегата које су од интереса приликом наглог оптерећења/растерећења.

Оглед оптерећења дизел електричног агрегата представља врло битну ставку приликом вршења излазне контроле квалитета као и добар показатељ тренутне техничке спремности уређаја приликом спровођења редовног сервиса. Једноставност, робусност, мобилност и техничка способност да одоли и највећим изазовима представља предност у односу на друге видове подршке приликом нестанка или обезбеђивања основног вида напајања

5. ЛИТЕРАТУРА

- [1] др.Зоран Радаковић, др. Милан Јовановић, „Специјалне електричне инсталације“, Академска мисао, Београд, 2008.
- [2] https://en.wikipedia.org/wiki/Electronic_Diesel_Control
- [3] <http://definicije.blogspot.rs/2013/04/>
- [4] <https://www.iso.org/standard/30567.html>
- [5] <https://www.energypower.com.au/products/used-equipment/used-mini-power-stations/default.aspx>
- [6] <http://www.mgmgenerator.com/>
- [7] "Службени лист СФРЈ", бр. 53 од 2. септембра 1988, 54 од 9. септембра 1988 - исправка, "Службени лист СРЈ", број 28 од 16. јуна 1995. „Правилник о техничким нормативима за електричне инсталације ниског напона“
- [8] Simens, Kurs B-42
- [9] <https://sices.eu/en/products/monitoring-systems/si-mo-ne>
- [10] Страхил Ј. Гушавац, „Основни принципи пројектовања у мрежама средњег и ниског напона“, Универзитет у Новом Саду, Факултет Техничких наука, Нови Сад, 2014.
- [11] З.Стојковић, „Пројектовање помоћу рачунара у електроенергетици – примена програмских алата“, Академска мисао, Београд, 2009.
- [12] ISO 8258-5:2013 Генератори наизменичне електричне енергије погоњени мотором са унутрашњим сагоревањем – Одељак 5: Генератори наизменичне струје
- [13] Љ. Рашајски, Г. Дотлић, "Мали електроенергетски приручник", СМЕИТС, Београд, 1997
- [14] Мр. Гојко Дотлић дипл. Инж., „Електроенергетика кроз стандарде, законе, правилнике, одлуке и техничке препоруке“, СМЕИТС, Београд, 2013
- [15] https://www.weg.net/catalog/weg/US/en/Generation%2CTransmission-and-Distribution/Generators/Alternators-for-Generator-Sets/c/GLOBAL_GENSET
- [16] WEG synchronous alternators ag 10 line 12638144 manual english

Кратка биографија:



Дејан Старчевић рођен је у Новом Саду 1992. год. Гимназију „Светозар Марковић“, завршио је у Новом Саду, 2011 год. Факултет техничких наука, студијски програм Енергетика, електроника и телекомуникације уписао је школске 2011/2012. На студијама се определио за смер Електроенергетика-енергетска електроника и електричне машине и дипломирао 20. 09. 2016. год. Ожењен, отац сина Вукана

VEKTORSKO UPRAVLJANJE SINHRONOM MAŠINOM SA STALNIM MAGNETIMA**VECTOR CONTROL OF PERMANENT MAGNET SYNCHRONOUS MACHINE**Slavica Gajić, Boris Dumnić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ENERGETSKA ELEKTRONIKA I
ELEKTRIČNE MAŠINE**

Kratak sadržaj – *Sinhrona mašina sa stalnim magnetima je mašina naizmenične struje koja poseduje trofazni stator iste konstrukcije kao i kod asinhronne mašine, ali rotor je drugačiji i poseduje površinski postavljene stalne magnetne. Karakteristike sinhronne mašine sa stalnim magnetima postaju sve bolje i bolje sa napretkom tehnologije elektromagnetnih materijala. Sada se sinhronne mašine sa stalnim magnetima koriste u raznim oblastima zbog svoje robusnosti, visoke efikasnosti, male težine i dimenzija. Zbog prethodno navedenih prednosti, posebno visoke efikasnosti, sinhronne mašine sa stalnim magnetima se sve više koriste u industriji i ključan su deo opreme koja se koristi u obnovljivim izvorima električne energije. U okviru ovog rada prikazan je matematički model sinhronne mašine sa stalnim magnetima, date su metode za određivanje parametara i opisano je vektorsko upravljanje. Na kraju rada je predstavljen simulacioni model sinhronne mašine sa stalnim magnetima napravljen u Matlab/Simulink programu.*

Ključne reči: *sinhrona mašina sa stalnim magnetima, matematički model, parametri mašine, vektorsko upravljanje, simulacioni model mašine*

Abstract – *The permanent magnet synchronous machine is a rotating electrical machine with a classic 3-phase stator like that of an induction machine, but rotor is different and has surface-mounted permanent magnet. The performance of permanent magnet synchronous machine has improved rapidly by the progress in electromagnetic material technology. Now permanent magnet synchronous machine is widely used because it has many advantages such as robustness against the environment, high efficiency, light weight and small size. Because of advantages above, especially those featuring high efficiency, permanent magnet synchronous machine is frequently used in industry and is a key equipment in renewable energy applications. In this paper mathematical model of permanent magnet synchronous machine is shown, methods for parameter measurements and the concept of vector control are described. At the end a simulation model of permanent magnet synchronous machine made in Matlab/Simulink program is given.*

Keywords: *permanent magnet synchronous machine, mathematical model, parameters of the machine, vector control, simulation model of the machine*

NAPOMENA:

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

1. UVOD

Sinhrona mašina sa stalnim magnetima (*eng. permanent magnet synchronous machine- PMSM*) prvi put se pojavljuje u devetnaestom veku. Nedostaci kao što su nepristupačnost, cena i mala gustina magnetne energije dugo vremena su sprečavali masovniju upotrebu PMSM [1]. U tom periodu su se upotrebljavale sinhronne mašine sa pobudnim namotajem na rotoru. Masovnija upotreba PMSM počinje pojavom AlNiCo magnet. AlNiCo magnet je predstavljao najjači stalni magnet pre otkrića retkih zemalja. Samarijum-Kobalt magnet, napravljen od elemenata retkih zemalja, danas se najčešće koristi u konstrukciji PMSM. U početku su stalni magneti postavljani na statoru, a današnja standardna konstrukcija je takva da se magneti postavljaju na rotoru.

PMSM se dele na mašine sa površinski postavljenim magnetima i mašine sa utisnutim magnetima. PMSM sa površinski postavljenim magnetima nije pogodna za pogone koji zahtevaju veliki raspon brzina. PMSM sa utisnutim magnetima je mehanički robusnija i može biti manjih dimenzija i manje cene od PMSM sa površinski postavljenim magnetima istih karakteristika. PMSM su visokoefikasne električne mašine i upotrebljavaju se u vetroelektranama, hidroelektranama, hibridnim automobilima, električnim sistemima broda i aviona [1].

U nastavku rada biće razvijen matematički model PMSM SOGA koja se nalazi u laboratoriji Fakulteta tehničkih nauka. PMSM SOGA ima površinski postavljene magnetne na rotoru. Na slici 1. prikazana je PMSM SOGA.



Slika 1. PMSM SOGA

Na osnovu matematičkog modela PMSM biće realizovan i simulacioni model u Matlab/Simulink programu pomoću koga se mogu posmatrati odzivi mašine na razne poremećaje. U cilju realizacije simulacionog modela neophodno je odrediti i parametre date mašine, te će metoda za određivanje parametara biti navedena i opisana u nastavku rada. Kako bi se realizovalo što efikasnije

upravljanje sinhronom mašinom u ovom radu korišćen je upravljački koncept poznat kao vektorsko upravljanje.

2. MATEMATIČKI MODEL PMSM

U ovom poglavlju je dat matematički model PMSM sa površinski postavljenim magnetima. Matematički model PMSM u originalnom abc domenu je neefikasan za računarske simulacije u realnom vremenu, jer se sastoji od sistema nelinearnih diferencijalnih jednačina sa vremenski promenljivim koeficijentima i sistema algebarskih jednačina sa vremenski promenljivim koeficijentima [2].

Kako bi se umanjila kompleksnost matematičkog modela u originalnom domenu vrše se matrice transformacije matematičkog modela.

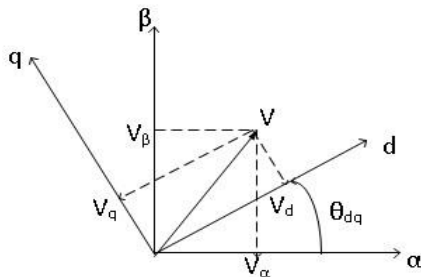
Prvo se na matematički model primenjuje transformacija raspreszanja koja kao efekat ima smanjenje nenulatih elemenata u matrici induktivnosti i značajno pojednostavljenje matematičkog modela. Njome se veličine iz abc domena transformišu u $\alpha\beta$ domen. Veza između promenljivih u $\alpha\beta$ i abc domenu može se zapisati kao u primeru izraza (1):

$$\begin{bmatrix} X_\alpha \\ X_\beta \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \cdot \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \quad (1)$$

Nakon transformacije raspreszanje primenjuje se transformacija obrtanja. Ovom transformacijom se veličine iz $\alpha\beta$ domena transformišu u dq domen. Transformacija veličina iz $\alpha\beta$ u dq domen se vrši zato što je matematički model mašine u dq domenu pogodniji za implementaciju u računarskim simulacijama. Veza između promenljivih u $\alpha\beta$ i dq domenu može se zapisati kao u primeru izraza (2):

$$\begin{bmatrix} X_d \\ X_q \end{bmatrix} = \begin{bmatrix} \cos(\theta_{dq}) & \sin(\theta_{dq}) \\ -\sin(\theta_{dq}) & \cos(\theta_{dq}) \end{bmatrix} \cdot \begin{bmatrix} X_\alpha \\ X_\beta \end{bmatrix} \quad (2)$$

Na slici 2. prikazane su projekcije veličine V u $\alpha\beta$ i dq domenu [2].



Slika 2. Transformacija obrtanja, prelazak iz $\alpha\beta$ u dq domen [1]

Matematički model PMSM u dq domenu nakon transformacije obrtanja se sastoji od jednačine naponske ravnoteže, jednačine fluksnog obuhvata statora, koje su date izrazima (3) i (4) respektivno:

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} R_s & -\omega L_s \\ \omega L_s & R_s \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \psi_m \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_d \\ \psi_q \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} \psi_d \\ \psi_q \end{bmatrix} = \begin{bmatrix} L_s & 0 \\ 0 & L_s \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \psi_{md} \\ \psi_{mq} \end{bmatrix} \quad (4)$$

gde su: $u_d, u_q, i_d, i_q, \psi_d, \psi_q$ - d i q komponente vektora napona, struje i fluksnog obuhvata statora, respektivno; R_s - otpornost namotaja statora; L_s - induktivnost namotaja statora; ω - električna ugaona brzina; ψ_{md}, ψ_{mq} - d i q komponente fluksa stalnog magneta; ψ_m - amplituda fluksa stalnog magneta.

Pored prethodno navedenih izraza u sastav matematičkog modela PMSM ulaze i jednačina momenta konverzije i Njutnova jednačina kretanja, koje su date izrazima (5) i (6) respektivno:

$$m_c = \frac{3}{2} \cdot p \cdot (\psi_d \cdot i_q - \psi_q \cdot i_d) \quad (5)$$

$$m_c - m_m = J \cdot \frac{d\omega_m}{dt} + K_m \cdot \omega_m \quad (6)$$

gde su: m_c - moment konverzije; p - broj pari polova; m_m - mehanički moment; J - moment inercije; K_m - koeficijent gubitaka; ω_m - mehanička ugaona brzina.

U dq domenu sve veličine su jednosmerne u stacionarnom stanju. Matematički model u dq domenu koristi se prilikom formiranja simulacionog modela u Matlab/Simulink okruženju.

3. METODE ZA ODREĐIVANJE PARAMETARA PMSM

Poznavanje parametara mašine je od ključne važnosti za optimalno upravljanje pogonom. Parametri koji su od značaja za PMSM SOGA su: otpornost namotaja statora, fluks stalnog magneta, induktivnost namotaja statora. PMSM SOGA ima 6 pari polova. Natpisna pločica ove mašine prikazana je na slici 3.



Slika 3. Natpisna pločica PMSM SOGA

Za merenje otpornosti namotaja statora mogu se koristiti merni mostovi, UI metoda, univerzalni merni instrumenti kao i druge metode. U ovom radu merenje otpornosti PMSM SOGA izvršeno je pomoću univerzalnog mernog instrumenta. Izmerena otpornost pomoću datog uređaja je otpornost između dva priključka PMSM. Da bi se dobila otpornost faznih namotaja, u slučaju namotaja statora povezanih u zvezdu, potrebno je vrednost otpora izmerenu na ovaj način podeliti sa dva. Otpornost jednog faznog namotaja statora PMSM SOGA iznosi 12.02 Ω .

Poznavanje induktivnosti PMSM je veoma važno, pogotovo ako se koriste u pogonima visokih performansi. Induktivnost namotaja statora može da se odredi pomoću vremenske konstante kola. Nakon što se rotor mehanički

učvrsti u položaju za koji važi da je $\theta = 0$ između priključaka statorskog namotaja se priključi izvor napajanja promenljive amplitude. Tada se induktivnost namotaja statora može odrediti pomoću izraza (7):

$$L_s = \frac{2}{3} \cdot L = \frac{2}{3} \cdot \tau \cdot R \quad (7)$$

U izrazu (7) L predstavlja ukupnu induktivnost serijsko-paralelne veze statorskih namotaja, a τ predstavlja vremensku konstantu kola [3]. Nakon što je određena vremenska konstanta kola koristeći se izrazom (7) dobijeno je da induktivnost namotaja statora PMSM SOGA iznosi 98.4 mH. Kod PMSM sa površinski postavljenim magnetima induktivnost namotaja statora po d i q osi su jednake.

Još jedan od parametara koji je od važnosti za PMSM jeste fluks stalnih magneta. Kako bi se odredio fluks stalnih magneta PMSM se mehanički spregne sa pogonskom mašinom kojom se upravlja po brzini. U brzinskom režimu rada zadaje se i pomoću regulatora održava konstantna mehanička brzina. Zatim se izmeri efektivna vrednost faznog napona, pa se fluks stalnog magneta određuje pomoću izraza (8):

$$\psi_m = \sqrt{\frac{2}{3}} \cdot U_{rms} / \omega \quad (8)$$

U izrazu (8) U_{rms} predstavlja efektivnu vrednost faznog napona. Za PMSM SOGA fluks stalnog magneta iznosi 0,7907 Wb.

4. VEKTORSKO UPRAVLJANJE PMSM SOGA U MATLAB/SIMULINK OKRUŽENJU

Vektorsko upravljanje je upravljački koncept koji se pojavljuje još 1960-tih, ali glavni razvoj i primena vektorskog upravljanja počela je tek pojavljivanjem bržih i jeftinijih mikroprocesora. Pošto je za vektorsko upravljanje pogodniji model u dq domenu u drugom poglavlju ovog rada izvršene su transformacije matematičkog modela i na kraju je dat matematički model PMSM u dq domenu. Ideja vektorskog upravljanja jeste da se pronađu dve veličine kojima će se nezavisno upravljati fluksom i momentom. Te dve upravljačke veličine su poprečna i podužna komponenta struje statora. Podužna komponenta

struje statora je zadužena za upravljanje fluksom, a poprečna za upravljanje momentom [4]. Regulacija vektora struje u dq domenu je realizovana akcijom dva nezavisna strujna regulatora [3]. Nezavisan rad dva strujna regulatora omogućen je zahvaljujući dekuplovanju. Ideja dekuplovanja je da se komponente koje ne zavise od struje i_d eliminišu iz jednačine naponske ravnoteže d ose. Isto je urađeno i sa komponentama koje ne zavise od struje i_q u jednačini naponske ravnoteže q ose. Te komponente se dodaju na izlaz PI regulatora struje. Regulator u d osi kontrolom u_d reguliše i_d . Na sličan način radi i regulator u q osi.

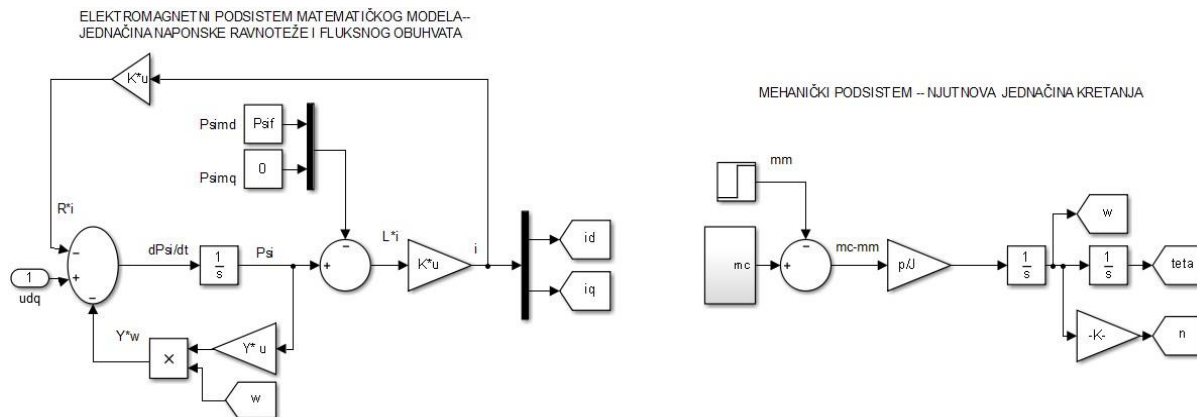
Kod PMSM sa površinski postavljenim magnetima ne realizuje se slabljenje polja, jer zbog male induktivnosti namotaja statora potrebna bi bila velika negativna vrednost struje i_d . Zato se referenca struje i_d održava na nuli. S obzirom da je $i_d=0$, a $\psi_d = \psi_m$ sledi da se sa i_q linearno upravlja momentom. Brzina se upravlja pomoću linearnog PI regulatora koji na izlazu zadaje referencu struje i_q takvu da se razlika zadate i ostvarene vrednosti brzine svede na nulu.

4.1. Simulacioni model i rezultati simulacija

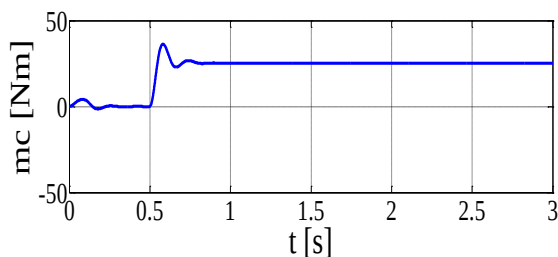
Skupo eksperimentalno testiranje se može izbeći u ranoj fazi istraživanja upotrebom adekvatnih simulacionih modela. U simulacionom modelu se tastaturom izazivaju poremećaji i na ekranu računara se posmatraju odzivi. U ovom radu razvijeni matematički model u dq domenu je iskorišćen za formiranje simulacionog modela. Na slici 4. predstavljen je izgled simulacionog modela PMSM SOGA u Matlab/Simulink okruženju.

Na osnovu realizovanog modela izvršeno je više simulacija a u ovom radu prikazani su rezultati simulacije u kojoj je najpre izvršen zalet neopterećene PMSM, a zatim se u 0,5 s nametne mehanički moment koji iznosi 25 Nm.

Na slici 5. prikazana je zavisnost momenta konverzije od vremena, gde se može primetiti da nakon izvršenog zaleta vrednost momenta pada na nulu. U trenutku kada se mašina opteretiti sa 25 Nm dolazi do momentne neravnoteže i potom mašina razvija momenat koji je jednak momentu opterećenja.

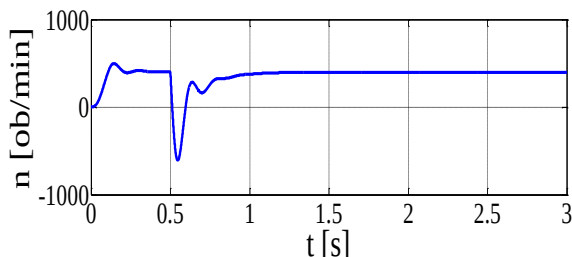


Slika 4. Simulacioni model PMSM SOGA



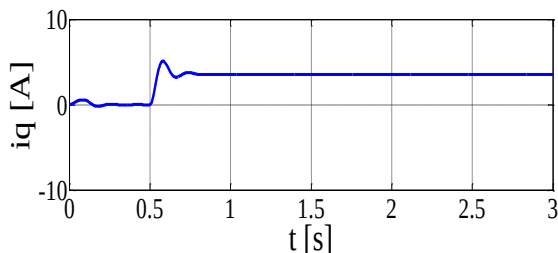
Slika 5. Zavisnost momenta konverzije od vremena

Na slici 6. prikazana je zavisnost brzine obrtanja od vremena. Nakon što je izvršen zalet brzina obrtanja je postala jednaka sinhronoj brzini i iznosi 400 ob/min. U trenutku opterećenja PMSM mehaničkim momentom različitim od nule brzina date PMSM će postati manja od sinhrona, kao što se može primetiti na slici 6. Kada se uspostavi momentna ravnoteža brzina će opet dostići vrednost jednaku sinhronoj brzini. Na osnovu datog odziva brzine evidentna je loša dinamika regulatora brzine (propad brzine ima nedozvoljeno veliku vrednost), što ukazuje na neophodnost pažljivog proračuna i odabira parametara regulatora u regulacionoj strukturi vektorskog upravljanja.



Slika 6. Zavisnost brzine od vremena

Zavisnost q komponente struje od vremena prikazana je na slici 7. Nakon što se izvrši zalet mašine q komponenta struje postaje jednaka nuli. U 0,5 s kada se PMSM optereti sa 25 Nm dolazi do momentne neravnoteže i rasta q komponente struje, kao što se može videti na slici 7.



Slika 7. Zavisnost q komponente struje od vremena

Na osnovu prikazanih rezultata simulacije očigledno je da je uspešno realizovano vektorsko upravljanje datom sinhronom mašinom - PMSM SOGA tj. da je ostvarno nezavisno upravljanje fluksom i momentom mašine.

5. ZAKLJUČAK

Upotreba PMSM raste iz dana u dan i poslednjih decenija jako je važan razvoj u istraživanju karakteristika kao i oblasti primene ove vrste sinhronih mašina. U današnjem savremenom industrijskom svetu potreba za električnom energijom je sve veća i veća i potrebno ju je što efikasnije proizvoditi i koristiti. Elektromotorni pogoni su efikasni ukoliko imaju visok koeficijent korisnog dejstva. Visok koeficijent korisnog dejstva, male dimenzije, jednostavno

održavanje i hlađenje samo su neke od prednosti PMSM u odnosu na druge vrste naizmeničnih mašina. Prethodno navedene prednosti PMSM u odnosu na druge vrste naizmeničnih mašina predstavljaju razlog povećanog broja istraživačkih aktivnosti vezanih za PMSM.

Kako bi se ispitala karakteristike date mašine najpre je napravljen pojednostavljen matematički model u originalnom domenu. Taj matematički model se pomoću transformacija opisanih u radu prebacuje u dq domen. Da bi se dobili tačni rezultati simulacije rada mašine prethodno je izvršeno određivanje nedostajućih parametara PMSM SOGA. Parametri su određeni pomoću metoda izloženih u radu. Pored toga, u radu je predstavljen i simulacioni model vektorski upravljane PMSM kao i rezultati jedne simulacije dobijeni pomoću ovog modela. Prikazani rezultati simulacije jasno ukazuju na činjenicu da je ostvareno nezavisno upravljanje fluksom i momentom sinhrona mašine. Kao zaključak ovoga rada može se navesti i činjenica da je jedna od mana simulacionih modela to što ne postoji univerzalan model za sve mašine, koji bi se samo prilagodio karakterističnoj mašini, nego je za svaku drugu vrstu mašine potrebno napraviti novi simulacioni model.

4. LITERATURA

- [1] Steve Evon, Robbie McElveen and Michael J. Melfi, "Permanent Magnet Motors for Power Density and Energy Savings in Industrial Applications," IEEE Transactions on Industry Applications, vol. 44, no. 5, pp. 1360-1366, 2008.
- [2] Vladan Vučković, Opšta teorija električnih mašina, Nauka Beograd, 1992.
- [3] Darko P. Marčetić, Mikroprocesorsko upravljanje energetskim pretvaračima, FTN Izdavaštvo, 2014.
- [4] Veran Vasić, Đura Oros, Energetska elektronika u pogonu i industriji, FTN Izdavaštvo, 2010.

Kratka biografija:



Slavica Gajić rođena je u Novom Sadu 1994. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranila je 2017. god. Master rad na fakultetu tehničkih nauka iz oblasti Elektrotehnika i računarstvo – Energetska elektronika i električne mašine odbranila je 2018. godine.
kontakt: slavica.gajic994@gmail.com



Boris Dumnić rođen je u Bileći 1976. god. Diplomirao je 2003. godine na Fakultetu tehničkih nauka u Novom Sadu – Elektrotehnika i računarstvo. Od 2004. godine zaposlen je na Fakultetu tehničkih nauka u Novom Sadu. Magistarski rad je odbranio 2007. godine a 2013 godine je odbranio i doktorsku tezu. Oblast interesovanja su mu električne mašine, pogoni, energetska elektronika i obnovljivi izvori električne energije.

ELEKTROMOTORNI POGONI U EKSPLOZIVNIM SREDINAMA**ELECTRIC MOTOR DRIVES IN HAZARDOUS AREAS**Goran Cibula, Veran Vasić, Đura Oros, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu je prezentovana problematika odabira i eksploatacije elektromotornih pogona u eksplozivnim sredinama. Obrađena je klasifikacija eksplozivnih sredina, što predstavlja prvi korak za pravilan izbor opreme. Opisani su različiti tipovi protiveksplozivne zaštite asinhronih elektromotora. Razmotreni su uzroci povećanog zagrevanja elektromotora povećane bezbednosti. Pravilan izbor zaštite elektromotora povećane bezbednosti je jako bitan te je u ovom radu detaljno obrađen. Uticaj pretvarača učestanosti na pogonske parametre elektromotora je detaljno razmotreno ovim radom.

Ključne reči: Asinhroni motori, povećana bezbednost, elektromotorni pogon, pretvarač učestanosti, protiveksplozivna zaštita

Abstract – In this paper problem of selecting and exploiting of induction motors in hazardous areas is presented. Classification of hazardous areas has been processed, which is the first step in proper selection of equipment. Different types of explosion protection induction motors are described. Reasons of increased heating for the increased safety motor. Proper selection of protection device for increased safety motor are very important and that is detailed processed. Effects of frequency converter to the drive parameters of the motor are discussed in detail in this paper.

Keywords: Induction Motors, Increased Safety, Electric Motor Drive, Frequency Converter

1. UVOD

Asinhroni motori po brojnosti prevazilaze sve ostale električne mašine u industriji zbog svoje jednostavne konstrukcije i niske cene. Mnogi od tih motora su ugrađeni u prostorima ugroženim eksplozijom, kao što su naftna, hemijska, farmaceutska industrija i sl. U ovakvim prostorima gde usled pojave varnice ili povišenog zagrevanja kontaktnih površina može doći do eksplozije, tj. do velike materijalne štete ugrožene bezbednosti ljudi, životinja i okoline, kao i teških ekoloških posledica, potrebno je voditi računa oko izbora opreme koja se ugrađuje. Oprema koja se koristi u eksplozivnim sredinama je posebno dizajnirana tako da u bilo kom trenutku eksploatacije ne dovede do ugrožavanja eksplozivne sredine, tj. do eksplozije.

NAPOMENA:

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

2. EKSPLOZIVNE SREDINE I ZONE OPASNOSTI

Elektromotori pogoni (EMP) predstavljaju sklop za konverziju električne u mehaničku energiju. Većina elemenata EMP-a namenjenog za rad u eksplozivnoj sredini mora biti na poseban način konstruisana ispitana i instalirana tako da bude bezbedna za rad u eksplozivnim sredinama. Posebni zahtevi koji se postavljaju pred EMP u eksplozivnim sredinama prvenstveno zavise od eksplozivnih svojstava sredine. U tom smislu se sprovodi klasifikacija eksplozivnih sredina.

Da bi došlo do eksplozije u potencijalno eksplozivnoj sredini potrebno je da budu ispunjena tri uslova:

- prisustvo zapaljivih materija (zapaljivi gasovi i pare)
- prisustvo kiseonika
- prisustvo uzročnika paljenja (električna varnica ili luk, elektrostatičko pražnjenje, zagrejane površine mašina, mehanička varnica itd.)

Kako je nemoguće sprečiti nastanak eksplozivne atmosfere, da bi se stvorio pouzdan sistem, potrebno je eliminisati bar jedan od gore navedenih uslova. Praktično je najjednostavnije eliminisati mogući uzrok paljenja.

Radi sagledavanja problema protiveksplozivne zaštite s obzirom na električne uređaje kao potencijalne uzročnike paljenja potrebno je poznavati i karakteristike eksplozivnih sredina prema kojima se izrađuje protiveksplozivna zaštita. Tako se dolazi do klasifikacije zapaljivih gasova i para u odnosu na električne uređaje. Prema temperaturi paljenja gasovi i pare, prema standardu [1], se razvrstavaju u temperaturne klase T1-T6.

Postoji veoma mali broj kućišta u koja pod normalnim uslovima ne može da uđe gas ili para, jer poseduju procepe (između vratila, poklopaca, itd.) koji omogućavaju ulaz eksplozivne smeše. Posebnom konstrukcijom moguće je sprečiti prolaz plamena i eksplozije iz unutrašnjosti kućišta u okolni prostor. Svi zapaljivi gasovi i pare razvrstavaju se prema vrsti zaštite na "nepropaljivo kućište" gde eksplozivna smeša ne može ući u kućište i u njemu eksplodirati i "povećana bezbednost" gde se može javiti varnica, ali ta eksplozija tj. varnica ne mogu izvršiti paljenje okolne eksplozivne atmosfere, jer je procep dovoljno uzak i dugačak. Prema graničnoj dužini tog procepa kroz koji je nemoguće propaljivanje gasovi i pare se grupišu.

Za klasifikaciju zapaljivih gasova i para prema svojstvu probojnog paljenja uveden je pojam maksimalni eksperimentalni bezbednosni zazor - MEBZ (prema IEC-u [2] MESG - maximum experimental safety gap) u kućištu sa 25mm dugim procepima. Za zaštitu "samosigurnost" gasovi i pare se grupišu prema odnosu

njihove minimalne struje paljenja MSP (prema IEC-u [2] MIC - Minimal ignition current) prema minimalnoj struji paljenja laboratorijskog metana. Grupe u koje su svrstani gasovi i pare prema [2] su A,B,C.

2.1. Klasifikacija eksplozivno ugroženih prostora na zone opasnosti

Ugroženi prostori (u kojima se očekuje pojava eksplozivne atmosfere) pre svega se klasifikuju na one ispod zemlje (rudnici) i one iznad zemlje (industrija). Svaki ugroženi prostor deli se na zone opasnosti prema vrsti, učestanosti i trajanju eksplozivne atmosfere koja se javlja u njemu. Definicije kao i složeni postupak za određivanje zona opasnosti dati su u [3]. Osnovu za klasifikaciju predstavljaju dva osnovna faktora izvor ispuštanja (opasnosti) i brzina ispuštanja.

Prema načinu nastajanja i vremenu trajanja eksplozivnih smeša izvori mogu biti trajni, primarni, sekundarni i višestruki. U odnosu na njih se velikim delom određuju zone opasnosti, međutim ventilacija ima najznačajniji uticaj od svih faktora na zone opasnosti. Ona može da smanji koncentraciju materije koja sa vazduhom čini eksplozivnu smešu i time da zonu opasnosti višeg reda prevedu u zonu nižeg nivoa ili ugroženi prostor prevedu u neugroženi. Klasifikacija se vrši na sledeće zone:

- Zona 0 - prostor u kome je eksplozivna atmosfera prisutna stalno ili duži vremenski period
- Zona 1 - prostor u kome je verovatno da će se eksplozivna atmosfera pojaviti za vreme normalnog pogona odnosno njeno prisustvo je predviđeno tehnološkim postupkom
- Zona 2 - prostor u kome nije verovatno da će se eksplozivna atmosfera pojaviti za vreme normalnog pogona, a ako se ipak pojavi tada traje kratko i to se dešava retko, pre svega u incidentima

Uticaj izvora ispuštanja (opasnosti) i ventilacije na određivanje zona opasnosti prikazan je u tabeli 1. Oznaka NP u tabeli predstavlja neugrožen prostor, dok "+" označava zonu koja se teorijski može javiti u zanemarljivoj meri u normalnim uslovima.

Tabela 1. Zone opasnosti u odnosu na izvor ispuštanja i karakteristike ventilacije

Izvor opasnosti - stepen ispuštanja	Vrsta ventilacije		
	Slaba ventilacija (npr. zatvoren prostor)	Srednja ventilacija (npr. prirodno ventiliran prostor)	Jaka ventilacija (npr. prinudno ventiliran prostor)
Trajni	Zona 0	Zona 0+, Zona 1 ili 2	Zona 1 ili 2 ili NP
Primarni	Zona 1 ili 0	Zona 1+ Zona 2	Zona 2 ili NP
Sekundarni	Zona 2 ili 1 ili 0	Zona 2	NP (eventualno Zona 2)

3. ELEKTROMOTORNI POGONI U EKSPLOZIVNOJ SREDINI

Samo pravilno odabran i korišćen električni uređaj za eksplozivnu sredinu neće izazvati paljenje eksplozivne smeše, tj. eksploziju. Izbor odgovarajućeg električnog uređaja vrši se na osnovu sledećih parametara:

- područje primene (rudnici ili industrija)
- klasifikacija prostora i određivanje zona opasnosti - stepen ugroženosti na predviđenom mestu montaže uređaja (zona 0,1 ili 2)
- minimalna temperatura paljenja gasova ili para, tj. utvrđivanje potrebne temperaturne klase uređaja (T1-T6)
- utvrđivanje grupe gasova ili para u odnosu na MEBS i MSP (I, IIA, IIB, IIC)
- električni parametri uređaja
- karakteristike eksploatacije i održavanje
- cena pojedinog izvođenja protiveksplozivne zaštite

Bez obzira na to u kojoj su vrsti protiveksplozivne zaštite izvedeni, električni uređaji moraju da zadovolje tzv. opšte zahteve (principe) koji garantuju osnovu da se uređaj može izvesti u jednoj od vrsta protiveksplozivne zaštite.

Svi električni uređaji predviđeni za rad u potencijalno eksplozivnom prostoru moraju biti posebno ispitani i na temelju toga atestirani, što je propisano ili standardima regulisano. Ispitivanja se mogu podeliti na tri nivoa:

- tipska ispitivanja ili ispitivanja za sertifikaciju tipa
- pojedinačna ispitivanja ili ispitivanja za svrhu dokaza usaglašenosti pojedinačnog uređaja sa sertifikatom tipa
- ispitivanje posle izvedenih prepravki ili remonta

3.1. Tipovi protiveksplozivne zaštite elektromotora

Električne mašine mogu biti uzročnici paljenja eksplozivne sredine kao rezultat povišene temperature kućišta, elektrostatičkih pražnjenja ili mehaničkih varničenja. Da bi se osigurala protiveksplozivna zaštita kod elektromotora razmatraju se sledeće vrste zaštite od paljenja: povećana bezbednost, nepropaljivo kućište, izvedba koja ne dozvoljava varničenje, zaštita nad pritiskom.

Elektromotori koji su izrađeni u vrsti zaštite nepropaljivog kućišta ("d"), izrađeni su tako da ukoliko dođe do eksplozije unutar kućišta eksplozija se ni u jednom slučaju ne može proširiti na eksplozivnu atmosferu van kućišta, tako da kućište mora izdržati pritisak usled eksplozije. Temperatura kućišta se mora održavati u granicama klase gasa koji se nalazi u atmosferi oko mašine.

Kućišta sa natpritiskom "p" su izrađena tako da se unutar kućišta stvara pritisak koji je viši od ambijentalnog pritiska, tako da se ne dozvoli prodor eksplozivne smeše u samo kućište. Da bi se obezbedila protiveksplozivna zaštita neophodan je konstantan monitoring pritiska u kućištu, kao i da ne dođe do povišenja temperature površine kućišta.

Zaštita elektromotora koja je izvedena tako da ne izaziva varničenje "n" je bazirana na tipu zaštite povećane sigurnosti i namenjena je za upotrebu u zoni 2, zbog nižih verovatnoća pojave prisustva zapaljive atmosfere u ovoj zoni. Motori izvedeni u ovom tipu protiveksplozivne zaštite ne smeju se pokretati ukoliko je prisutna eksplozivna sredina.

Povećana bezbednost "e" je tip sekundarne protiveksplozivne zaštite u kojoj se primenjuju dodatne mere, tako da se postiže povećana bezbednost protiv mogućnosti povećanja temperature i pojave lukova i varnica unutar i van kućišta uređaja koji u normalnom radu ne stvara lukove i varnice. Ova zaštita se zasniva na smanjenju verovatnoće nastanka uzročnika paljenja izborom uređaja koji u normalnom radu ne mogu izazvati paljenje, a koji svojim karakteristikama garantuju malu verovatnoću nastanka kvara, dok se paljenje može javiti samo u slučaju kvara. Ova vrsta protiveksplozivne zaštite elektromotora je samo tehničko-tehnološko unapređenje tzv. motora obične izrade koje sadrži niz dodatnih mera za povećanje sigurnosti.

3.5 Zaštita elektromotora povećane sigurnosti od prekomernog zagrevanja

Najčešći uzrok povećanog zagrevanja elektromotora je preopterećenje, npr. opterećenje elektromotora momentom većim od nominalnog momenta koji može da razvije razmatrani elektromotor. Povećanim opterećenjem elektromotora struje se povećavaju u statorskim i rotorskim namotajima. Povećanjem struja u namotajima dolazi i do povećanja Džulovih gubitaka, koji se povećavaju sa kvadratom struje. Uzrok povećanog zagrevanja elektromotora može biti i rad elektromotora u uslovima gde se javlja odstupanje od osnovnih nazivnih parametara elektromotora.

Rad pri smanjenom naponu napajanja elektromotora dovodi do povećanih struja u statorskom namotaju. Zbog smanjenog napona napajanja dolazi do smanjenja indukovane EMS što dovodi do smanjenja magnetne indukcije u mašini tj. fluksa, što rezultuje povećanom klizanju pri konstantnom opterećenju. Povećanjem klizanja rastu džulovi gubici u rotoru mašine, što dovodi do zagrevanja.

Povećanjem napona napajanja iznad nazivne vrednosti dolazi do povećanja indukcije što može dovesti do zasićenja usled čega dolazi do povećanja gubitaka u gvožđu mašine. Prilikom zasićenja magnetnog kola mašine dolazi do povećanja reaktivne komponente struje statora, što dovodi do povećanja džulovih gubitaka u statorskim namotajima mašine.

Pojavom nesimetričnog napona napajanja, ili nestanka jedna faze ne znači da će doći do zaustavljanja elektromotora. U slučaju malog opterećenja mašine doći će samo do povećanja klizanja što dovodi do povećanja džulovih gubitaka u rotoru mašine. Nestankom jedne od faza doći će do povećanja struja u preostale dve faze što dovodi do povišenja temperature.

Motor u zaštiti povećana sigurnost u pogledu porasta temperature može da izdrži sve očekivane režime rada osim režima ukočenog rotora. Proizvođač garantuje da motor pri struji ukočenog rotora (I_A) iz početnog stanja pri ustaljenom porastu temperature u nominalnom režimu rada, što predstavlja struju polaska, i posle toga dođe do ukočenog rotora pri najvišoj temperaturi okoline, neće premašiti granice temperaturne klase za koju je deklarisan. Vreme t_E predstavlja vreme potrebno da polazna struja (I_A) zagreje namotaj od temperature u radu pri nominalnom opterećenju na najvišoj temperaturi okoline do granične temperature. Pod vremenom

zagrevanja t_E podrazumeva se vreme koje u slučaju da se rotor ukoči, tj. nastanka mehaničkog kratkog spoja, protekne dok se ne dostigne maksimalna dozvoljena granična temperatura statorskog odnosno rotorskog namotaja ili dok ne odreaguje zaštitni uređaj. Ovo je veoma bitno jer motor u zaštiti povećana sigurnost čini protiveksplozivno sigurnu celinu samo u sklopu sa odgovarajućom električnom zaštitom.

Indirektnu zaštitu čine prekostrujni okidači i releji (MZP) koji nadziru zagrevanje namotaja elektromotora posredno preko struje. Upotreba ovih zaštitnih uređaja je uglavnom zadovoljavajuća kod rada elektromotora sa konstantnim opterećenjem u normalnim uslovima. Njihova osnovna mana je što se vremenska konstanta zagrevanja samog releja ne može podešavati u širokom opsegu i sa malom rezolucijom, zbog čega nije moguće postići da vremenska konstanta releja bude uvek jednaka vremenskoj konstanti zagrevanja elektromotora. Bimetal daje neodgovarajuću sliku zagrevanja elektromotora u nekim situacijama kada opterećenje nije konstantno odnosno u intermitentnim režimima rada sa kratkotrajnim prekidima kod kojih se često dešava da je motor preopterećen, a bimetal ne reaguje.

Vrsta greške gde bimetalni relej pruža najpouzdaniju zaštitu jeste upravo zaštita od ukočenog rotora ali u eksplozivnoj sredini kod zaštite povećane bezbednosti motora treba postaviti relej sa funkcijom reseta. Vreme delovanja zaštitnog uređaja zavisi od veličine strujnog preopterećenja elektromotora. Vreme t_E mora biti takvo da se u toku njegovog trajanja dok je rotor motora ukočen mašina može isključiti pomoću strujno zavisnog zaštitnog uređaja pre isteka vremena t_E . Prekostrujna zaštita kao što su bimetali, mikroprocesorski termički releji čine zaštitu elektromotora samo od zagrevanja izazvanog povećanim opterećenjem, tj. strujom. Ovi uređaji ne mogu da štite motore od preopterećenja nastalih zbog pogoršanih uslova hlađenja, povećane ambijentalne temperature i dr.

Zato se koristi direktna termička zaštita elektromotora, što podrazumeva direktan nadzor temperature sondama, tj. termistorima ugrađenim direktno u namotajima motora. Sonda sama za sebe ima samo indikatorski karakter i predstavlja zaštitu samo u sklopu sa nekim prekidačkim uređajem, npr. termistorskim relejom. Direktna zaštita u nekim situacijama je nezamenljiva jer direktno meri temperaturu motora i reaguje na nju nezavisno od uzroka zagrevanja. Ovi uređaji mogu nedovoljno brzo da reaguju u situaciji ukočenog rotora, kada dolazi do naglog povećanja temperature namotaja koja ne mora biti dovoljno brzo prenet na termistor.

Dinamički odziv zavisi od načina ugradnje u namotaj, odnosno brzine prenosa toplote. Zbog toga se za motore sa protiveksplozivnom zaštitom povećana bezbednost zahteva da ovi uređaji ukoliko se koriste za zaštitu od ukočenog rotora budu ispitani zajedno kao sistem, u suprotnom se primenjuje i bimetal koji služi kao zaštita od ukočenog rotora a termistori za direktno praćenje temperature tj. preopterećenja.

4. REGULISANI ELEKTROMOTORNI POGON U EKSPLOZIVNOJ SREDINI

Elektromotorni pogoni napajani iz pretvarača učestanosti pružaju veliki broj prednosti, njihov rad uključuje i neke

mane, u EMP-u u prostorima ugroženim eksplozijama, mogu predstavljati opasnost od eksplozije.

Porast temperature je definisan za rad motora pri nominalnoj brzini, s potpunim hlađenjem, ukoliko nije forsirano hlađenje i uz čist sinusni talasni oblik napona napajanja. Po definiciji čist sinusni talasni oblik napona ne sadrži harmonike višeg reda (samo osnovnu učestanost), ali izlazni napon pretvarača sadrži i više harmonike, koji će se javiti i na priključcima elektromotora. Ovi harmonici u naponu ne daju korisni momenat, već dovode do povećanog zagrevanja samog elektromotora. Sa stanovišta zagrevanja bitno je kod elektromotora napojenih preko pretvarača uzeti u obzir i sledeće elemente [4][5]:

- ventilacija i hlađenje elektromotora kroz kompletan opseg brzine,
- najveće struje preopterećenja i granice trajanja najveće struje (podesivo na pretvaraču),
- U/f karakteristiku uzimajući u obzir U_0 koji kada je većeg iznosa takođe dodatno zagreva motor,
- karakteristiku moment/brzina radne mašine,
- stvarni oblik napona i struje na priključcima elektromotora,
- uticaj uslova okoline i dr.

Ukoliko je hlađenje elektromotora izvedeno ventilatorom koji je montiran na drugu stranu vratila elektromotora, pri smanjenju brzine elektromotora smanjuje se i uticaj hlađenja ventilatora tj. protok vazduha preko kućišta elektromotora. Ukoliko je motor potpuno opterećen, a brzina manja za 50%, od motora se zahteva da daje maksimalni momenat s polovinom potrebne količine hlađenja. Ta redukcija faktora hlađenja sa smanjenjem brzine će dostići tačku na kojoj je dopušteni porast temperature motora premašen, što uzrokuje smanjenje veka eksploatacije izolacije ili grešku koja može biti uzročnik paljenja eksplozivne atmosfere.

U invertorskom bloku pretvarača usled uključenja prekidačkih komponenti (npr. IGBT) nastaju prelazne pojave koje na kablovima, kojim je povezan motor sa pretvaračem, mogu rezultovati pojavom prenapona na priključnim terminalima elektromotora usled rezonantnih putujućih naponskih talasa. Budući da je talasni otpor tj. odnos L/C elektromotora znatno veći od talasnog otpora napojnog kabla motora, na priključnim terminalima motora se javljaju naponi iznosa većeg od napona međukola (i većeg od nazivnog napona motora). Putujući naponski talasi na priključcima motora prodiru u namotaje i mogu dovesti do povećanog naprezanja izolacije.

Takođe ti vršni naponi (U_{peak}), koji moraju biti definisani od proizvođača pretvarača (za različite slučajeve) mogu ugroziti vazdušni zazor i površinske provodne staze u priključnoj kutiji. Ugrožavanje vazdušnog zazora podrazumeva da može doći do preskoka varnice jer je napon veći od nazivnog, dok ugrožavanje površinske provodne staze podrazumeva da u priključnoj kutiji može doći do preskoka između dva provodna dela preko izolacionog materijala koji je projektovan prema nazivnom naponu elektromotora.

Proizvođač pretvarača treba da definiše vreme porasta napona (t_r), kao i vršnu vrednost napona nastalog prekidačkim radom pretvarača (U_{peak}). Definicija vremena porasta napona uzima u obzir tranzitne pojave u namotajima elektromotora.

Napon vratila nastaje zbog nesimetrije magnetnog polja u vazdušnom zazoru. Indukovani napon na vratilu, iako relativno mali, uzrokuje znatne struje u ležajevima, jer je mala otpornost strujnog kruga, koje mogu oštetiti ili čak uništiti ležaj. Zagrevanje se može javiti na dodirnim tačkama tolikog iznosa da se materijal topi. Oštećenja ležajeva mogu dovesti i do mehaničkih kvarova što može izazvati i mehaničko varničenje koja predstavlja uzrok paljenja eksplozivne sredine

5. ZAKLJUČAK

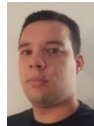
Pred elektromotorne pogone postavljaju se veliki i zahtevni izazovi. Kako osnovu elektromotornih pogona čine električne mašine, koje po svojoj prirodi imaju razne nedostatke koji se između ostalog ogledaju i u nepovoljnom uticaju na eksplozivne sredine, tj. mogu predstavljati uzročnike eksplozije, nameće se potreba za njihovom posebnom konstrukcijom za bezbedan rad u takvim uslovima.

Svakodnevnom razvojem tehnologije, samim tim i napretkom primene električnih mašina uvode se nove metode za ispitivanje električnih mašina čime se ostvaruje značajan napredak na razvoju samih mašina namenjenih za rad u različitim ambijentalnim uslovima, što dovodi do bezbednije i sigurnije primene samih elektromotornih pogona i u eksplozivno ugroženim sredinama.

4. LITERATURA

- [1] IEC 79-4:1975 Method of test for ignition temperature
- [2] IEC 60079-20-1:2010 Material characteristics for gas and vapour classification - Test methods and data
- [3] IEC 60079-10:2002 Classification of hazardous areas
- [4] prof. dr Veran Vasić, prof. dr Đura Oros, "Energetska elektronika u pogonu i industriji"
- [5] IEC 60079-14:2013 Electrical installations design, selection and erection
- [6] Dipl.inž.el. Ivica Gavrančić, "Dijagnostika protuekslozijski zaštićenih elektromotornih pogona", Zagreb, 2002.
- [7] Mr Lidija V.Pavlović, prof. dr Zoran M.Lazarević, "Asinhroni motori u protivekslozivnoj zaštiti "povećana bezbednost", Beograd, 2008.

Kratka biografija:



Goran Cibula rođen je u Novom Sadu 1991. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2014.god

**СОФТВЕРСКИ СИСТЕМ ЗА ПРЕПОРУКУ СТУДИЈСКОГ МОДУЛА НА СМЕРУ
РАЧУНАРСТВО И АУТОМАТИКА****SOFTWARE SYSTEM FOR RECOMMENDING STUDY MODULE ON COURSE
COMPUTER AND CONTROL ENGINEERING**Светозар Стојковић, *Факултет техничких наука, Нови Сад***Област - ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

Кратак садржај - Овај рад бави се истраживањем перформанси два приступа системима за препоруку. Први тип система за препоруку је систем за препоруку заснован на колаборативном филтрирању, а други је заснован на моделу латентних фактора. За складиштење података користе се технологије семантичког веба где се подаци налазе на Фусеки серверу одакле се добављају SPARQL упитима. Тачност система за препоруку заснованом на колаборативном филтрирању износи 83.92% док је тачност система за препоруку засновану на моделу латентних фактора 30.04%. Закључено је да иако примитивнија метода, колаборативно филтрирање, даје боље резултате јер се користи део целокупног скупа студената.

Кључне речи: системи за препоруку, колаборативно филтрирање, модел латентних фактора, семантички веб, RDF, SPARQL, Java Spring

Abstract - This paper contains an performance analysis on two types of recommender systems. First type is a recommender system based on collaborative filtering and the other one is based on latent factor model. Semantic web technologies are used for storing data, data are stored on Fuseki server and they are accessed via SPARQL queries. Recommender system based on collaborative filtering have the accuracy of 83.92% while the accuracy of LFM recommender system is 30.04%. It was concluded that even though collaborative filtering is more primitive method because it uses the part of whole student grades dataset.

Keywords: recommender systems, collaborative filtering, latent factor model, semantic web, RDF, SPARQL, Java Spring

1. УВОД

У овом раду је описан систем за препоруку студијских модула на основу оцена студента. На почетку рада представљени су основни концепти Семантичког Веба: онтологије, RDF, SPARQL и приказани су разни типови система за препоруку. Након тога пружен је увид у организацију наставе у по студијским модулним, где је акценат стављен на модуле на смеру Рачунарство и аутоматика.

НАПОМЕНА:

Овај рад произтекао је из матер рада чији ментор је био др Милан Сегединац.

Потом следи приказ софтверског пројекта, кроз имплементацију две врсте система за препоруку и тестирање тачности појединачних система на реалним подацима са студентске службе Факултета техничких наука.

2. ТЕОРИЈСКЕ ОСНОВЕ

У овој секцији изложени су теоријски концепти потребни за разумевање пројекта, у ком је имплементиран систем за препоруку студијског модула.

2.1 Вокабулари и онтологије на семантичком вебу

У оквиру семантичког веба вокабулари дефинишу концепте и везе које се користе за описивање и представљање одређеног домена.

Вокабулари представљају основне градивне делове Семантичког веба [1].

2.2 Web ontology language (OWL)

W3C Web ontology language (OWL) је језик семантичког веба дизајниран да представи богато и комплексно знање о стварима, групама ствари и њиховим везама. OWL је формални језик заснован на дескриптивној логици. OWL документи - онтологије - могу бити објављени на World Wide Web (WWW) [2].

2.2.1 RDF

RDF стандарни је модел за размену података на вебу који проширује везну структуру веба да користи URI-је за формирање веза међу ставкама као и два краја везе (ово се назива "triple").

Везна структура креира усмерни означен граф, где гране представљају означену везу између два ресурса где се ресурси представљају као чворови графа [3].

2.2.2 SPARQL

SPARQL се може користити за вршење упита у различитим типовима података, без обзира да ли су подаци чувани у RDF или неком другом стандардном формату семантичког веба. SPARQL подржава агрегацију, подупите, негацију, креирање вредности изразима, тестирање вредности и ограничавање упита у RDF графу. Резултати SPARQL упита могу бити сетови или RDF графови [4].

2.3 Системи за препоруку

Системи за препоруку једни су од најпопуларнијих апликација науке над подацима. Користе се да

предвиде “оцену” коју ће корисник дати некој ставци или “преференције” корисника. Скоро свака већа компанија их примењује у неком облику: Amazon их користи да препоручује прозводе корисницима, YouTube их користи да одлучи који ће видео следећи да пусти на autoplay, Facebook их користи да препоручи странице да се “лајкују” или пријатеље са којима би корисници могли да ступе у контакт. Штавише, неке компаније као Netflix и Spotify имају бизнис модел чији успех директно зависи од могућности вршења препорука. Заправо, Netflix је понудио милион долара 2009. године ономе који може да поправи систем за 10% [5].

Системи за препоруку могу бити подељени на три типа: једноставни системи за препоруку; системи за препоруку засновани на садржају; колаборативно филтрирање [5].

2.3.1 Једноставни системи за препоруку:

Ови системи често генерализују препоруке свим корисницима, које су засноване на популарности филма или жанра. Основна идеја иза овог система је да ће филмови који су популарнији и критички прихваћенији имати већу шансу да се свиде просечном кориснику. IMDb Top 250 пример је овог система.

2.3.2 Системи за препоруку засновани на садржају

Овакви системи за одређену ставку препоручују сличне ставке. За препоручивање се користе метаподаци, као што су жанр, режисер, опис, глумци итд. Генерална идеја иза ових система за препоруку је та да ако се особи свиђа одређена ставка, њему или њој ће се свидети слична ставка.

2.3.3 Колаборативно филтрирање

Системи за препоручивање засновани на колаборативном филтрирању покушавају да предвиде оцену коју би корисник могао да да одређеној ставци на основу претходних оцена и преферитета других корисника. Такви системи за препоручивање не захтевају метаподатке о ставкама као системи за препоруку засновани на садржају. [6] Матрица на основу које се врши колаборативно филтрирање као једну димензију има све кориснике, а као другу ставке које корисници оцењују оценама у неком одређеном опсегу [5].

Неке од метода коришћене за прорачунавање сличности су: Jaccard-ова сличност; косинусна сличност; центрирана косинусна сличност (Пирсонова корелација)

Најчешћи алгоритам који се користи за одређивање сличности је Пирсонова корелација (Pearson Correlation).

2.3.3.1 Пирсонова корелација (Pearson Correlation)

Коефицијент Пирсонове корелације је мера снаге линеарне везе између две варијабле.

$$r_{XY} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

Пирсонова корелација може да буде у опсегу од -1 до 1. Ако је -1, онда је то савршена негативна линеарна веза између варијабли; ако је 0, онда значи да не постоји линеарна веза међу варијаблама; ако је корелација 1, онда значи да је савршена позитивна линеарна корелација међу варијаблама [6].

2.3.4 Модел латентних фактора (LFM)

Моделовање латентних фактора мапира како кориснике тако и ставке у заједнички простор латентних фактора R^K (свака димензија енкодира прикривени фактор) [7].

2.4 Студијски модули на смеру Рачунарство и аутоматика

Након што студенти заврше другу годину на Факултету техничких наука одабирају један од наредна три студијска модула: примењене рачунарске науке и информатика (ПРНИ); рачунарска техника и рачунарска комуникација (РТ-РК); аутоматика.

Студенти који су изабрали модул Примењене рачунарске науке и информатика у четвртој години бирају један од четири студијска модула: интернет и електронско пословање; информациони системи; интелигентни системи; софтверско инжењерство.

3. СКУП ПОДАТАКА

Студентске оцене добављене су од Факултета техничких наука у Excel фајлу.

У фајлу се налазе оцене 2420 студената при чему је задржана анонимност студената.

Свака оцена је приказана кроз три атрибута: идентификациони број студента (насумичан број - јединствен за студента); назив предмета; оцена студента из тог предмета. Због мењања назива предмета кроз време, првобитно су се називи предмета морали мапирати на данашње називе предмета. Пример: Пре неколико година назив данашњег предмета “Алгебра” био је “Дискретна математика и линеарна алгебра”. Након тога је креиран .csv фајл у коме редови представљају студенте, а колоне представљају предмете. У том формату се брже добављају оцене одређеног студента, такође боља је и визуелна репрезентација расподељености студентских оцена.

4. ПРЕПОРУКА СТУДИЈСКОГ МОДУЛА

У циљу помоћи студентима на трећој и четвртој години основних академских студија, који морају да бирају наредни студијски модул, креиран је систем за препоруку истог где се врши препорука на основу оцена студента.

4.1 Креирање онтологије

На основу добијеног .csv фајла врши се популисање, односно попуњавање онтологије индивидуалима у Python језику.

Разлог коришћења онтологије је у томе што су складиштени одвојено од било ког система, што значи да ако се направи неки други систем који користи оцене студената може да приступи овој бази података и користи те податке за своје потребе.

4.2 Архитектура система

Архитектура система састоји се од неколико микросервиса који функционишу у кластеру зарад модуларизације система на одвојене целине, које функционишу тако што сваки сервис обавља један “задатак” и они су: *service-registry*; *config-server*; *api-gateway-service*; *module-recommender-service*.

4.3 Портал за препоруку модула

Портал са препоруку модула представља *module recommender service* који се састоји од *frontend* и *backend* дела.

4.3.1 Препоруке

У раду је имплементирано неколико врста система за препоруку.

Све препоруке садрже неке заједничке кораке као што су: учитавање података са семантичког веба; креирање матрице свих студената.

Учитавање података са семантичког веба - подаци о свим студентима су претходно смештени на *Jena Fuseki* сервер, одакле се прикупљају уз помоћ *SPARQL* упита.

Креирање матрице свих студената - након што се добаве подаци, они се морају приказати у неком формату, у овој ситуацији је најадекватнији приказ матрица студент/оцена.

4.3.1.1 Колаборативно филтрирање

Препорука заснована на колаборативном филтрирању вршена је на следећи начин.

Проналазак најсличнијег студента - врши се користећи подкуп матрице свих студената и свих оцена.

Одабир модула - Проналази се најсличнији студент активном студенту уз то да се као додатни фактор узима и његов просек.

$$\text{value} = (\text{value} + 1) * 5 * (\text{sum} / \text{count})$$

- **value** - вредност Пирсонове корелације
- **sum** - сума свих оцена студента
- **count** - број оцена студента

Домен вредности је на овај начин скалиран на опсег од 0 до 100 где 0 (нула) значи да су потпуно различити студенти, а 100 (сто) да је студент вредан препоруке.

Од препорученог студента преузима се модул на ком је он и препоручује се активном студенту као модул који би он требао да одабере на наредном избору модула.

4.3.1.2 Модел латентних фактора (LFM)

Препорука заснована на моделу прикривених фактора вршена је на следећи начин.

Креирање LFM матрице - *LFM* матрица креирана је по узору на препоруку филма [8].

Алгоритам коришћен при креирању излазне матрице је објашњен у области 2.3.4.

Израз алгоритма је матрица, где су све оцене замењене оценама које представљају студентове “препоручене оцене” из свих предмета.

Одабир модула - Модул који има највиши просек оцена се препоручује активном кориснику.

4.3.2 Крајње тачке сервиса

Акцнат пројекта је на *module-recommender* сервису који садржи неколико крајњих тачака које могу да се “тађају” са неког *frontend-a* или из неког програма за тестирање као што је *Postman*.

4.4.2.1 Колаборативно филтрирање (Collaborative filtering)

На основу оцена студента систем врши препоруку тако што проверава оцене свих студената на основу Пирсонове корелације. Модул коме припада најсличнији студент се приказује као препорука активном студенту.

4.3.2.2 Провера колаборативног филтрирања

Тачност система за препоруку, заснованог на колаборативном филтрирању, тестираног на студентима који се већ налазе у бази, износи **83.92%**.

4.3.2.3 Модел латентних фактора (Latent Factor Model - LFM)

На основу оцена студента врши се препорука тако што се примени *LFM* алгоритам на све студенте укључујући и активног, и из добијене матрице се прочита препорука предмета и ком модулу он припада.

4.3.2.4 Провера LFM

Тачност система за препоруку, заснованог на *LFM*, тестираног на студентима који се већ налазе у бази, износи **30.04%**.

Разлог зашто је резултат тако низак је тај што се у рачуницу узима цела матрица за разлику од колаборативног филтрирања где се узимају само оцене активног студента.

5. ЗАКЉУЧАК

У овом истраживању анализирана је могућност развоја и примене система за препоручивање студијских модула на основу студентских постигнућа. Посматрана су два приступа: колаборативно филтрирање и методе латентних фактора.

Истраживање је показало да примена методе колаборативног филтрирања, и ако је “примитивнија” даје боље резултате од методе латентних фактора.

Као најважнија предност методе колаборативног филтрирања издвојена је чињеница да се у обзир узимају све оцене студента, док се у методи латентних фактора врши редукција димензионалности при чему долази до губитка информација и препорука се врши на основу смањеног броја атрибута.

Предност методе латентних фактора је у томе што користи мању матрицу и у огромним системима може да врши препоруке на значајно мањем броју атрибута, што се може користити у случајевима ограничености хардвера.

Систем за препоруку студијског модула на основу колаборативног филтрирања врши препоруку само на основу оцена, али не може да предвиди људске преферитете односно интересовања студената.

Могуће је да би систем за препоручивање студенту препоручио модул на коме би студент остварио виши просек занемаривши интристичку мотивацију и интересовања студента. Стога један од основних праваца даљег истраживања представља укључивање података о интересовањима у систем за препоручивање студијских модула.

Такође, као један од важних праваца даљег истраживања издваја се могућност примене других техника рачунарске интелигенције за развој оваквог система, као што су препоруке вршене уз помоћ неуронских мрежа или чак неке од метода које служе за класификовање података.

Препоруку студијских модула треба посматрати као интегрални део шире слике образовног процеса па се због тога интеграција система предложеног овим истраживањем са другим дигиталним системима који су део образовног процеса издваја као важан правац даљег истраживања.

6. ЛИТЕРАТУРА

- [1] - <https://www.w3.org/standards/semanticweb/ontology>
- [2] - <https://www.w3.org/OWL/>
- [3] - <https://www.w3.org/RDF/>
- [4] - <https://www.w3.org/TR/rdf-sparql-query/>
- [5] - <https://www.datacamp.com/community/tutorials/recommender-systems-python>

[6] - :

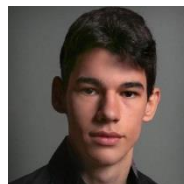
http://onlinestatbook.com/2/describing_bivariate_data/pearson.html

[7] - Learning Personal+Social Latent Factor Model for Social Recommendation; Yelong Shen, Ruoming Jin; Department of Computer Science Kent State University.

[8] - MF-Based-Recommendation

<https://github.com/ShantanuDeshpande/MF-Based-Recommendation>

Кратка биографија:



Светозар Стојковић рођен је у Новом Саду 1994. године. Основне академске студије завршио је 2017. године. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства - Софтверски систем за препоруку студијског модула на смеру рачунарство и аутоматика одбранио је 2018. године.

контакт:

svetozar.cvele.stojkovic@gmail.com

KRATKOROČNA PROGNOZA PROMENE POTROŠNJE USLED PRIMENE SAVREMENIH TEHNIKA REGULACIJE NAPONA U DISTRIBUTIVNIM MREŽAMA**NEAR TERM FORECAST OF DEMAND CHANGE DUE TO USE OF MODERN VOLTAGE REGULATION TECHNIQUES IN DISTRIBUTION NETWORKS**

Srdan Lazarov, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu su predstavljene matematički modeli i algoritmi za proračun prognoze promene potrošnje trofazne aktivne snage usled primene savremenih tehnika regulacije napona u distributivnim mrežama. Cilj prognoziranja ovih vrednosti je u omogućavanju fleksibilnog upravljanja potrošnjom - rezultatima ove prognoze omogućava se distributivnim mrežama da učestvuju u primarnoj i sekundarnoj regulaciji frekvencije i aktivnih snaga, smanjenju vršne potrošnje i eliminaciji dvosmernih tokova snaga. U radu je korišćena simulaciona metoda prognoziranja, koja se bazira na simuliranju primene određene tehnike regulacije napona i proračunu režima za naredni polusatni interval u elementima od interesa u okviru distributivne mreže. Kao ulazni podaci koriste se parametri tih elemenata, kao i trenutna i usrednjena merenja njihovih određenih veličina.

Ključne reči: *Kratkoročna prognoza promene potrošnje, fleksibilno upravljanje potrošnjom, regulacija napona*

Abstract – This paper presents the mathematical models and algorithms for calculating the forecast of three-phase active power demand change due to use of modern voltage regulation techniques in distribution networks. The aim of forecasting these values is to enable the flexible demand management – forecast results allow distribution networks to participate in primary and secondary frequency and active power controls, reduction of peak demand and elimination of two-way load flows. In this paper, a simulation forecasting method is used, which is based on simulating the application of a certain voltage regulation technique and calculating the regime for the next half-hour interval in the elements of interest within the distribution network. The parameters of these elements are used as input data, as well as the currently and averaged measured values of their certain variables.

Keywords: *Near term demand change forecast, flexible demand management, voltage regulation*

1. UVOD

Tokom poslednjih nekoliko decenija koncept pasivne distributivne mreže (DM) značajno se menja. Savremene DM prolaze kroz niz transformacija i unapređenja, a sve u cilju uvođenja tehnologija pametnih mreža (eng. „Smart Grid”). Globalno prognozirani porast potrošnje usled povećanja

upotrebe tehnologija sa niskim emisijama CO₂ (npr. električnih vozila), u kombinaciji sa sve većim udelom obnovljivih i konvencionalnih distributivnih generatora (DG) sa niskim emisijama CO₂ u ukupnoj proizvodnji, predstavlja veliki izazov za distributivna preduzeća u efikasnom planiranju rada i razvoja njihovih mreža. Kao posledica navedenih promena, pogon savremenih DM postaje dinamičniji i podložan brojnim pogonskim izazovima, kao što su: nedostatak pouzdanih resursa za regulaciju frekvencije i aktivnih snaga, preopterećenje u periodima vršne potrošnje, pojava dvosmernih tokova snaga itd. Jedno od rešenja za efikasnije upravljanje DM u navedenim uslovima je integracija sistema za aktivno upravljanje potrošnjom (SAUP, eng. „CLASS – Customer Load Active System Services”) sa (EMS)DMS/SCADA sistemom. SAUP se bazira na osnovnom principu elektroenergetske mreže, koji glasi: „Sa promenom napona menja se i potrošnja u elektroenergetskoj mreži” [1]. Optimalnom primenom savremenih tehnika regulacije napona, uz održavanje kvaliteta električne energije SAUP ima potencijal da:

- omogući DM da bude resurs u primarnoj i sekundarnoj regulaciji frekvencije i aktivnih snaga;
- poveća stepen iskorišćenja postojećeg kapaciteta i odloži ojačanje DM zbog vršnih potrošnji;
- unapredi rešavanje problema dvosmernih tokova snaga u radijalnim delovima DM;
- olakša priključivanje tehnologija sa niskom emisijom CO₂ na DM (npr. električna vozila);
- olakša priključivanje obnovljivih DG na DM;
- smanji emisije CO₂;
- smanji troškove distributivnih preduzeća;
- smanji troškove potrošača.

Da bi se to ostvarilo, neophodno je operatoru DM pružiti uvid u mogućnosti određene SAUP funkcionalnosti, tj. uvid u promenu potrošnje usled njenog aktiviranja u budućem vremenskom periodu (polusatnom intervalu).

2. SAUP – SISTEM AKTIVNOG UPRAVLJANJA POTROŠNJOM

SAUP je niskobudžetno, lako i brzo primenjivo rešenje koje koristi savremene tehnike regulacije napona preko parova regulacionih transformatora (TR) sa mogućnošću regulacije pod opterećenjem, koji rade u paraleli u transformatorskoj stanici (TS) u okviru DM, kako bi se operatoru DM i/ili prenosne mreže (PM) pružio niz mogućnosti upravljanjem potrošnjom, koje je rezultat samoregulacije potrošača usled promene napona [1].

NAPOMENA:

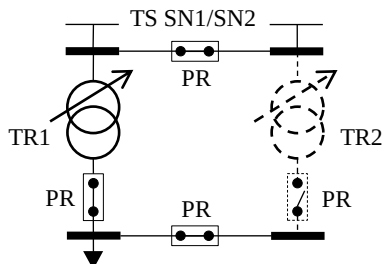
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Andrija Sarić, red. prof.

Cilj SAUP rešenja je da se regulacijom napona upravlja potrošnjom električne energije, a da pri tome potrošači ne primete razliku u kvalitetu i pouzdanosti snabdevanja – njegovom primenom upravlja se potrošnjom, a ne samim potrošačima, čime se izbegava njihovo isključenje. U toku eksploatacije DM, cilj je da se električna energija isporučuje sa naponima unutar definisanih tehnički tolerantnih granica [2]. Obezbeđivanjem takvih napona u svim tačkama DM, problem isporuke kvalitetne električne energije (sa aspekta napona), u klasičnom pristupu regulaciji napona, smatra se rešenim [2]. Očigledno da u takvom rešenju postoji sloboda u izboru napona, na kojoj se i zasniva SAUP rešenje, kao i na osobini potrošača u vidu samoregulacije pri promeni napona. Primenom SAUP rešenja izostavlja se kriterijum regulacije napona u vidu minimiziranje štete koju trpe potrošači, a sve u cilju postizanja željenog upravljanja potrošnjom. Potrošači će se i dalje snabdevati kvalitetnom električnom energijom, tj. naponi će biti unutar dozvoljenih granica, ali se neće obezbeđivati minimiziranje „dozvoljene” štete koju potrošači trpe. Korišćene tehnike regulacije napona su:

- tehnika brze regulacije napona i
- tehnika sporije regulacije napona.

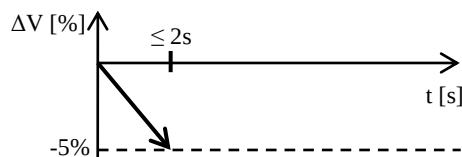
2.1 Tehnika brze regulacije napona

Zasniva se na isključenju iz pogona jednog od TR u TS, otvaranjem prekidača (PR) na niženaponskoj (NN) strani, usled čega se povećava opterećenje TR koji ostaje u pogonu, a to rezultuje većim padom napona i smanjenjem napona na NN strani TS [1]. Na slici 2.1 ilustrovana je primena ove tehnike u TS SN1/SN2 (TR2 isključen).



Slika 2.1 Tehnika brze regulacije napona

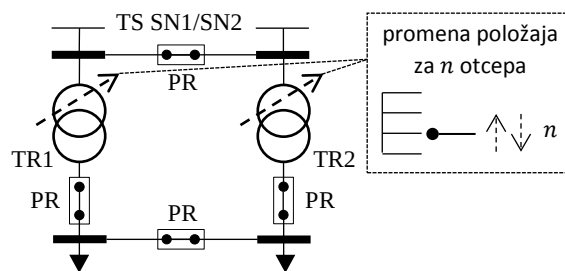
Ovaj postupak pruža kontinualnu i trenutnu promenu napona. Naredba za otvaranje NN prekidača TR se izvršava veoma brzo – za manje od 2 s [1]. Moguće je samo smanjenje napona i ne postoji opcija podešavanja željenog smanjenja, već to zavisi od parametara TR i njihovih opterećenja u trenutku primene. Na slici 2.2 prikazana je osobina kontinualnosti.



Slika 2.2 Kontinualnost tehnike brze regulacije napona

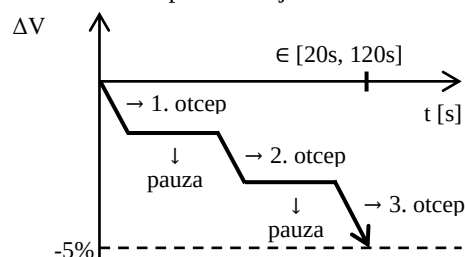
2.2 Tehnika sporije regulacije napona

Zasniva se na promeni položaja regulacionih sklopki oba TR u TS za isti broj otepa u istom smeru, pod opterećenjem, a to rezultuje smanjenjem napona na NN strani TS [1]. Na slici 2.3 ilustrovana je primena ove tehnike u TS SN1/SN2 (položaji regulacionih sklopki se menjaju za n otepa).



Slika 2.3 Tehnika sporije regulacije napona

Ova metoda omogućava diskretnu i sporiju promenu napona. Položaji regulacionih sklopki TR menjaju se otecpo po otecpo, sa pauzama između uzastopnih promena, što zahteva vreme od 20 s do 2 min, zavisno od broja otepa za promenu i tipa regulacionih sklopki [1]. Moguće je smanjenje/povećanje napona i postoji opcija podešavanja željene promene kroz zadavanje broja otepa promene. Na slici 2.4 prikazana je osobina diskretnosti.



Slika 2.4 Diskretnost tehnike sporije regulacije napona

2.3 Funkcionalnosti

Aktivnim upravljanjem potrošnje, upotrebom prethodno opisanih tehnika regulacije napona, SAUP omogućava distributivnim preduzećima da učestvuju u:

- primarnoj regulaciji frekvencije i aktivnih snaga (**PRF** – tehnikom **brze** regulacije napona);
- sekundarnoj regulaciji frekvencije i aktivnih snaga (**SRF** – tehnikom **sporije** regulacije napona);
- smanjenju vršne potrošnje (**SVP** – tehnikom **sporije** regulacije napona);
- eliminaciji dvosmernih tokova snage (**EDTS** – tehnikom **sporije** regulacije napona).

Ovo su ujedno i nazivi SAUP funkcionalnosti koje se obrađuju u ovom radu.

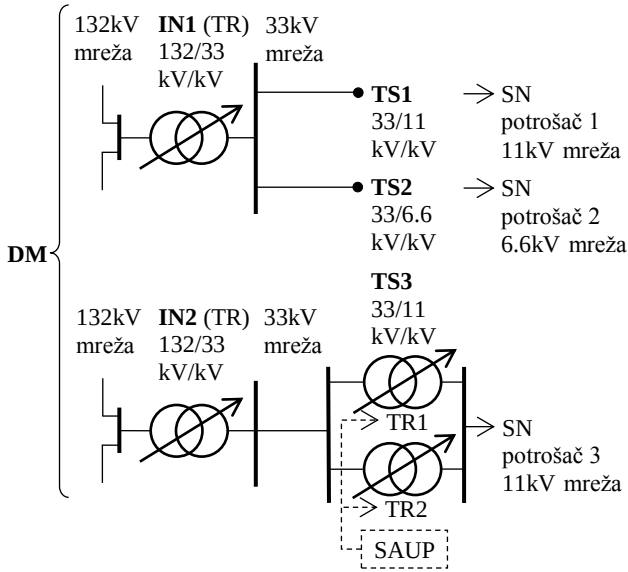
3. OPIS DISTRIBUTIVNE MREŽE

Ispituju se stacionarni efekti regulacije napona na promenu potrošnje trofazne aktivne snage u TS SN1/SN2 u normalnim radnim režimima koji su trofazni, stacionarni i simetrični (direktnog redosleda), i pri tome se smatra da su svi elementi DM uravnoteženi.

Od konstitutivnih elemenata DM od interesa su jedino potrošači i TR, jer se proračuni prognoze vrše na nivou TS, a za više hijerarhijske elemente – izvore napajanja (IN) i DM se vrši sumiranje prognoziranih vrednosti.

Posmatraju se SN potrošači u vidu TS SN1/SN2 koje sadrže dva trofazna dvonamotajna regulaciona TR koji rade u paraleli i imaju mogućnost regulacije napona pod opterećenjem. Potrošnja je svedena i modelovana na NN stranama TS, i predstavlja koncentrisanu potrošnju većeg broja TS SN/NN, koje napajaju veliki broj NN potrošača, a oni se sastoje od različitih elementarnih potrošača (električnih aparata).

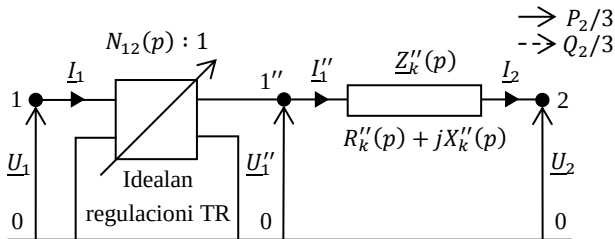
Na slici 3.1. prikazana je radialna test DM, na kojoj je samo jedan SN potrošač (TS3) prikazan detaljnije.



Slika 3.1 Test distributivna mreža

3.1 Model transformatora

Na slici 3.2 prikazana je pogonska šema predstavljenih TR, sprege Yy0, bez grane magnećenja i sa impedansama kratkog spoja (KS) koncentrisanim na NN strani, za opisane radne režime. Regulaciona sklopka TR izvedena je na VN strani, sa direktnom logikom rada – podizanjem ručice (angažovanjem većeg otepa) zahvata se manje navojaka VN namotaja čime se, za isti napon na VN strani, podiže napon na NN strani TR, i obrnuto [2].



Slika 3.2 Pogonska šema transformatora

Svi parametri šeme zavise od angažovanog otepa – p . Izraz za realni odnos transformacije – $N_{12}(p)$ je sledeći:

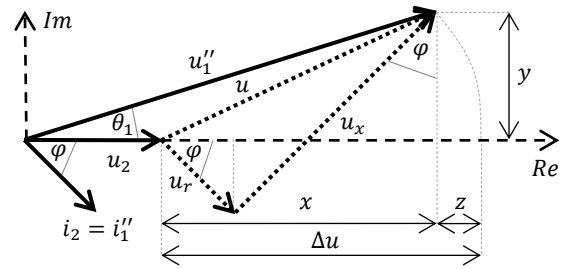
$$N_{12}(p) = \left(1 - (p - p_n) \cdot \frac{\Delta u_{n[\%]}}{100}\right) \cdot \frac{U_{1n}}{U_{2n}}, \quad (1)$$

gde je:

- p_n – nominalni regulacioni otep;
- $\Delta u_{n[\%]}$ – nominalni napon između susednih otepa u %;
- U_{1n} – nominalan fazni napon VN namotaja;
- U_{2n} – nominalan fazni napon NN namotaja.

Za određivanje ostalih parametara šeme, na osnovu dobro poznatih izraza, potrebno je odrediti procentualni napon (impedansu) KS – $u_k[\%](p)$ i trofazne gubitke u bakru TR – $P_{Cu}(p)$ za otep p , pri čemu se to radi linearnom interpolacijom između poznatih vrednosti za minimalni, nominalni i maksimalni otep.

Analitički izrazi za određivanje unutrašnjeg pada napona izvode se polazeći od fazorskog dijagrama u relativnim jedinicama za induktivno opterećenje TR, koji je prikazan na slici 3.3.



Slika 3.3 Fazorski dijagram induktivnog opterećenja TR

Relativna vrednost unutrašnjeg pada napona TR – Δu određuje se dobro poznatim izrazom:

$$\left. \begin{aligned} x &= u_r \cos \varphi + u_x \sin \varphi \\ y &= u_x \cos \varphi - u_r \sin \varphi \end{aligned} \right\} \Delta u = x + \frac{1}{2} \frac{y^2}{u_1''}, \quad (2)$$

gde je:

- u_1'' – relativni napon VN strane sveden na NN stranu;
- x – uzdužna komponenta relativnog pada napona;
- y – poprečna komponenta relativnog pada napona;
- u_r – relativni napon koji se realizuje na rezistansi TR;
- u_x – relativni napon koji se realizuje na reaktansi TR;
- φ – ugao kompleksne snage TR (za sva 4 kvadranta).

3.2 Model potrošača

Koristi se eksponencijalni statički model potrošača, čiji je analitički oblik zavisnosti napona i potrošnje iskazan sa:

$$P = P_0 \cdot \left(\frac{U}{U_0}\right)^{k_{pu}} \quad \wedge \quad Q = Q_0 \cdot \left(\frac{U}{U_0}\right)^{k_{qu}}. \quad (3)$$

U izrazima su U_0 , P_0 i Q_0 uglavnom nominalne vrednosti napona, aktivne i reaktivne snage, respektivno, koje određuju radne tačke (U_0, P_0) i (U_0, Q_0) oko kojih se razvija model. Koeficijenti samoregulacije aktivne i reaktivne snage po naponu (k_{pu} i k_{qu}) praktično se ne mogu izračunati, već se moraju odrediti eksperimentalno, na polusatnom intervalu uz uvažavanje sezone i tipa dana.

4. PROGNOZA PROMENE POTROŠNJE

Svaki TR ima dostupna trenutna i usrednjena merenja sledećih veličina: linijski napon NN strane (V_2), trofazna aktivna/reaktivna snaga NN strane (P_2/Q_2) i položaj regulacione sklopke (angažovani otep) (p).

Posmatraju se dva režima na nivou TR i TS:

1. režim pre simuliranja regulacije napona i
2. režim posle simuliranja regulacije napona.

Poznavanjem ova dva režima za TS i eksponencijalnog statičkog modela SN potrošača, promena potrošnje trofazne aktivne snage TS (SN potrošača) za naredni polusatni interval – $\Delta P_{(TS)}$ može se odrediti izrazima:

$$\Delta u_{reg[\%]} = \frac{U_{2(TS)}^{posle} - U_{2(TS)}^{pre}}{U_{2(TS)}^{pre}} \cdot 100, \quad (4)$$

$$\Delta P_{(TS)} = P_{2(TS)}^{pre} \cdot \left[\left(1 + \frac{\Delta u_{reg[\%]}}{100}\right)^{k_{pu(TS)}} - 1 \right]. \quad (5)$$

4.1 Režim pre simuliranja regulacije

Režim pre simuliranja regulacije napona je većim delom određen dostupnim usrednjenim merenjima u TS. Za potrebe proračuna režima posle simuliranja, neophodno je da se u ovom režimu odrede još i fazni napon na VN strani TS (smatra se konstantnim), kao i ugao kompleksne

snage i struja jednog od TR i cele TS. Postupak za proračun ovih veličina u režimu pre simuliranja je sledeći:

1. Proračun parametara TR za angažovani otecep;
2. Proračun ugla kompleksne snage i struje TR i TS;
3. Proračun pada napona TR (TS) – izraz (2);
4. Proračun napona na VN strani TR (TS).

4.2 Režim posle simuliranja tehnike brze regulacije

Neposredno posle simuliranja ove tehnike, TR koji ostaje u pogonu je opterećen celokupnim snagama TS, tj. aktuelan je režim TS pre simuliranja. Parametri TR se ne menjaju, jer se ne menja angažovani otecep. Poznavanjem režima TS i parametara TR pre simuliranja, određuje se režim TR, odnosno TS posle simuliranja ove tehnike. Postupak za proračun režima posle simuliranja je sledeći:

1. Proračun pada napona TR (TS) – izraz (2);
2. Proračun napona na NN strani TR (TS);
3. Proračun procentualne promene napona – izraz (4);
4. Proračun promene potrošnje TS – izraz (5).

4.3 Režim posle simuliranja tehnike sporije regulacije

Smer promene – Δp i broj otecepa – n za koji je potrebno promeniti položaje ručica regulacionih sklopki TR, kako bi se realizovala željena promena napona usled primene SAUP funkcionalnosti – $\Delta u_{SAUP}[\%]$, određuju se izrazima:

$$\Delta p = \begin{cases} -1, & \Delta u_{SAUP}[\%] < 0, \\ +1, & \Delta u_{SAUP}[\%] > 0, \end{cases} \wedge n = \frac{|\Delta u_{SAUP}[\%]|}{\Delta u_n[\%]} \quad (6)$$

Broj n je ceo i pozitivan, i potrebno ga je validirati i po potrebi korigovati ako nema dovoljno otecepa za promenu.

Diskretnost promene napona se uvažava iterativnim (**n puta**) simuliranjem promene položaja regulacionih sklopki TR za **po jedan otecep** u odgovarajućem smeru.

Neposredno posle simuliranja iteracije ove tehnike, TR su opterećeni snagama iz režima pre simuliranja, tj. u njima je aktuelan prethodni režim. Parametri TR se menjaju, jer se menjaju angažovani regulacioni otecepi. Poznavanjem režima TS pre simuliranja i parametara TR posle simuliranja, određuje se režim TR, odnosno TS posle simuliranja iteracije ove tehnike. Postupak za proračun režima posle simuliranja je sledeći:

1. Određivanje smera i broja otecepa promene – izraz (6);
2. Podešavanje početnih uslova za prvu iteraciju;
3. Iterativni postupak simuliranja, za $h = 1, 2, \dots, n$:
 - 3.1 Određivanje novog angažovanog otecepa;
 - 3.2 Proračun parametara TR za angažovani otecep;
 - 3.3 Proračun pada napona TR (TS) – izraz (2);
 - 3.4 Proračun napona na NN strani TR (TS);
 - 3.5 Proračun procentualne promene napona – izraz (4);
 - 3.6 Proračun promene potrošnje TR i TS – izraz (5);
 - 3.7 Proračun ugla kompleksne snage i struje TR;
4. Proračun ukupne promene potrošnje TS – suma promena potrošnji TS u svakoj iteraciji (korak 3.6).

5. PRIMER PRORAČUNA

Koriste se tri skupa ulaznih podataka: podešenja SAUP funkcionalnosti (tabela 5.1), parametri TR i TS (tabela 5.2) i dostupna merenja u TS (tabela 5.3). Rezultati proračuna prognoze prikazani su u tabeli 5.4.

Tabela 5.1 Podešenja željenih procentualnih promena napona

SAUP funkcionalnost	SRF	SVP	EDTS
$\Delta u_{SAUP}[\%]$	-5	-3	+3

Tabela 5.2 Koeficijenti samoregulacije aktivne i reaktivne snage po naponu za sve TS i parametri svih TR u test DM

	TS1		TS2		TS3	
	TR1	TR2	TR1	TR2	TR1	TR2
k_{pu}	1.29		1.53		1.07	
k_{qu}	6.41		5.74		5.98	
$S_n [MVA]$	19	19	11.5	11.5	16	16
$V_{1n} [kV]$	33	33	33	33	33	33
$V_{2n} [kV]$	11	11	6.6	6.6	11	11
$u_{k_{min}} [\%]$	17.6	17.6	11.2	11.2	12.5	12.5
$u_{k_n} [\%]$	19.1	19.1	12.6	12.6	13.9	13.9
$u_{k_{max}} [\%]$	28.1	28.1	18.2	18.2	20.3	20.3
$P_{Cu_{min}} [kW]$	165.2	165.2	64.1	64.1	98.8	98.8
$P_{Cu_n} [kW]$	179.1	179.1	71.7	71.7	110.5	110.5
$P_{Cu_{max}} [kW]$	264.4	264.4	104.6	104.6	161.1	161.1
p_n	4	4	5	5	5	5
p_{max}	17	17	17	17	17	17
$\Delta u_n [\%]$	1.36	1.36	1.43	1.43	1.45	1.45

Tabela 5.3 Merenja za sve TR u test DM ($0.5S_n$ i $\cos\varphi=0.9$ ind)

	TS1		TS2		TS3	
	TR1	TR2	TR1	TR2	TR1	TR2
$P_2 [MW]$	8.55	8.55	5.17	5.17	7.20	7.20
$Q_2 [MVar]$	4.14	4.14	2.51	2.51	3.50	3.50
$V_2 [kV]$	11	11	6.6	6.6	11	11
p	4	4	5	5	5	5

Tabela 5.4 Rezultati prognoze promene potrošnje

	PRF [MW]	SRF 5% [MW]	SVP 3% [MW]	EDTS 3% [MW]
DM	-3.08	-1.60	-1.11	+0.72
IN1	-2.31	-1.15	-0.80	+0.50
TS1	-1.35	-0.71	-0.50	+0.27
TS2	-0.96	-0.44	-0.30	+0.23
IN2	-0.77	-0.45	-0.31	+0.34
TS3	-0.77	-0.45	-0.31	+0.34

6. ZAKLJUČAK

Na osnovu dobijenih rezultata, primenom određene SAUP funkcionalnosti u narednom polusatnom intervalu može se ostvariti smanjenje potrošnje trofazne aktivne snage od 7.4% (PRF), 3.8% (SRF) ili 2.6% (SVP), ili povećanje od 1.7% (EDTS), u odnosu na ukupnu potrošnju test DM u aktuelnom režimu (41.85MW), uz srednje vrednosti smanjenja napona u TR od 4.7% (PRF), 5.5% (SRF) ili 3.8% (SVP), ili povećanja od 2.8% (EDTS).

7. LITERATURA

- [1] Electricity North West: *Functional Specifications and Voltage Regulation Scheme for ASCs*, Electricity North West Ltd., Manchester, UK, February 2014;
- [2] V. Strezoski, D. Janjić: *Sistemi regulacije napona radialnih distributivnih mreža*, Fakultet Tehničkih Nauka, Novi Sad, Srbija, 1997;

Kratka biografija:



Srđan Lazarov rođen je u Zrenjaninu 1991. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranio je 2018. godine.

PREGLED RASPOLOŽIVIH SOFTVERA ZA SOLARNE ELEKRANE

OVERVIEW OF SOFTWARES FOR PV PLANTS

Andrea Gavrilović, Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad, Srbija*

Oblast – ELEKTROENERGETIKA

Kratak sadržaj – U ovom radu je dat pregled i opisan rad u raspoloživim fotonaponskim softverima. U cilju nastavka prethodnih istraživanja na temu komparacije fotonaponskih softvera zadate su njihove karakteristike na osnovu kojih je po datim kriterijumima izvršena komparacija. Zatim je urađena simulacija u softveru koji se pokazao kao softver sa najvećim dijapazonom opcija.

Ključne reči: softveri, solarni paneli, fotonaponski sistemi, planiranje.

Abstract – In this paper several available photovoltaic (PV) software and criteria for their comparison are being described. In order to continue the previous researches on the topic of the comparison of PV software, their characteristics were compared according to the given criteria. At the end, the simulation in the software that was proved to be the one with largest range of options for design of PV systems was performed.

Keywords: software, solar panels, photovoltaic systems, planning.

1. UVOD

Solarna energija je neograničen i čist izvor, koji svojim korišćenjem u obliku električne ili toplotne energije ne izaziva dodatnu emisiju CO₂ i time ne utiče na klimatske promene. S obzirom na svoj potencijal, ona predstavlja bogat izvor energije za budućnost.

Pretvaranje solarne energije u električnu obavlja se putem fotonaponskog (FN) efekta u FN ćelijama. Zbog malog izlaznog napona i snage FN ćelije, one se grupišu serijskim i paralelnim vezama u FN module, odnosno u jedinice veće snage FN panele. Današnji FN paneli su snaga između 250 W i 300 W standardnih dimenzija i zauzimaju površinu od oko 1,6 m².

Grupisanjem FN panela u nizove (*string-ove*) i njihovim paralelnim vezivanjem formiraju se FN elektrane, koje su snaga od nekoliko kW do više MW i zauzimaju velike površine (i do više desetina hektara).

Kako se FN elektrane ubrzano grade svuda po svetu, FN sistemi se moraju dalje poboljšavati [1]. U tom smislu važna komponenta je razvoj odgovarajućih softverskih alata za određivanje solarnog potencijala i prognoziranje generisanja električne energije.

Prognoziranje je korišćenje modela performansi za procenu očekivane proizvodnje električne energije datog FN sistema. Upotreba simulacionih softverskih paketa je popularna tehnika za prognoziranje, jer ne samo da omogućava korisnicima da modeluju, već i da analiziraju FN sisteme i njihovu proizvodnju. Trenutno, postoji veliki broj FN simulacionih softverskih paketa. Softverska rešenja za izračunavanje proizvodnje energije su najčešće korišćena sredstva u planiranju FN postrojenja. Matematički model FN postrojenja u ambijentalnim uslovima opisuje stvarno stanje FN elektrane, koje se očekuje na izabranoj lokaciji.

Cilj ovog rada jeste da izvrši komparaciju raspoloživih softverskih alata uvažavajući specifične kriterijume njihove upotrebe u FN sistemima niskog budžeta, tj. za projektovanje i planiranje proizvodnje krovnih FN elektrana.

2. PREGLED LITERATURE

U literaturi se pojavljuje niz radova, koji daju pregled i upoređenje karakteristika raspoloživih, komercijalnih softverskih alata. U nastavku će biti predstavljeni neki rezultati, objavljeni u poslednjem periodu.

Veliki doprinos istraživanju u oblasti komparacije softvera za projektovanje FN sistema dali su Sharma, Verna i Singha [2], čiji rad predstavlja pregled FN softvera sa ciljem da korisnik lakše izabere softver koji će obezbediti dobijanje najpreciznijih podataka za simulaciju, ekonomsku evaluaciju, analizu i planiranje FN sistema.

Značajni podaci na temu komparacije FN softvera predstavljeni su u radu Sinha [3] gde su opisani 19 softverskih programa i njihove karakteristike. Glavni cilj rada je da obezbedi trenutni status softverskih proizvoda kako bi istraživač dobio osnovni uvid u korišćenje odgovarajućih alata za istraživanje i razvoj studija hibridnih sistema.

Lalwani, Kothari i Singh [4] ocenili su FN softvere po sledećim kriterijumima: komercijalna raspoloživost i cena, radna platforma, radni kapaciteti, opsezi i izlazi, ažuriranja, od kojih su većina softvera posebno dizajnirani za FN sisteme.

Značajan doprinos istraživanju daju Petrović, Šimić i Vražić [1], koji vrše poređenje FN softvera: PVsyst, PVGIS, HOMER, gde su glavni kriterijumi mesečni i godišnji rezultati proizvodnje i greške procene. Kao softver sa najboljim karakteristikama izabran je HOMER.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je dr Vladimir Katić, red.prof.

Problematika izbora najboljeg softvera za dizajn i projektovanje FN sistema se u poslednje vreme intenzivnije istražuje i na Fakultetu tehničkih nauka u Novom Sadu.

Komparaciju FN softvera su izvršili Petronijević i Katić [5], upoređujući dostupne softvere Sunny Design, Profotonapsis, PVsyst, Homer, PV Sol po kriterijumima cene, izvora meteo podataka, praćenja (tracking), senčenja, prezentiranja izlaznih podataka, raspoloživosti modula i izbora FN invertora u bazi, vremenu učenja i na bazi opšteg utiska. Nakon komparacije, autori su utvrdili da je Sunny Design najpovoljniji softver za planiranje FN elektrana.

Komparaciju softvera: PVGIS, Sunny Design Web, PVSize, SAM, HOMER, PVsyst, izvršila je Repac [6]. Kriterijumi za poređenje su bili biblioteka modula i invertora, baza meteoroloških podataka, efekat senčenja, izlazni podaci, složenost strukture, vreme učenja, dostupnost i cena softvera, opšti utisak. Utvrđeno je da je za dizajn FN sistema najpogodniji PVsyst.

Sličnu temu obradio je Mega [7] u čijem radu je urađena komparacija HOMER, PVsyst, Solarius, BlueSol softvera po kriterijumima: okruženje, baza podataka, jednostavnost, licenca, dodatne f-je, dobijeni rezultati i tačnost. Nakon čega je utvrđeno da se rezultati dobijeni proračunom uz pomoć Solarius-a najbliži referentnim vrenostima sistema.

Savčić [8] je izvršila poređenje softvera SAM, HOMER, PV Planner, RETScreen, PVGIS. Po kriterijumima: funkcije, korisnički interfejs, baza podataka o vremenu, biblioteka modula i invertora, dostupnost i cena softvera. Najbolje ocenjen softver je PVsyst.

Različitost predloženih optimalnih softvera nameće potrebu da se uradi još jedno poređenje, ali sada sa širim dijapazonom korišćenih softvera i sa preciznijim i objektivnijim kriterijumima. U nastavku će biti predstavljeni rezultati poređenja 11 najmodernijih softvera raspoloživih za preuzimanje (*download*) na Internetu.

3. SOFTVERI ZA PLANIRANJE FN SISTEMA

U nastavku će ukratko biti predstavljeno 11 softvera, s tim da se detaljniji prikaz može naći u [9], gde su svi softveri dati kroz najvažnije karakteristike.

3.1. PV*SOL premium 2018 (R6)

PV*SOL je softver za planiranje FN sistema i toplotnih pumpi. Uz pomoć 3D vizualizacije, složene situacije senčenja se lako mogu modelovati, a efekti senčenja lako izračunati u bilo kom trenutku. Integrisana finansijska analiza omogućava korisniku da ispita ekonomski uticaj različitih dizajnova sistema. Rezultati simulacije pokrivaju i pojavu maksimuma, minimuma i odstupanja.

3.2. PVsyst 6

PVsyst se fokusira na modelovanje, simulaciju i analizu FN sistema. Uz dodatne alate nuudi četiri sekcije pomoću kojih se gradi realan FN sistem: preliminarne dizajn gde se bira jedan od 3 sistema (povezan na mrežu, samostalni i pumpni), projektni dizajn i baze podataka.

3.3. RETScreen

RETScreen je softver za analizu projekata obnovljivih izvora električne energije i energetske efikasnosti. Uz pomoć njega se lako utvrđuje tehnička i finansijska održivost potencijalnih projekata obnovljive energije. Program sadrži baze podataka troškova, proizvoda i klimatskih podataka koji se mogu koristiti pri analizama.

3.4. Solmetric

Solmetric je idealan za stambene i male komercijalne sisteme i nudi lako izračunavanje proizvodnje energije za traženi sistem. U kreiranju modela nudi opcije za odabir lokacije, opreme, definisanje područja, kreiranje nizova i odabir invertora.

3.5. BlueSol

BlueSol omogućava kreiranje dizajna sistema koji su povezani na mrežu ili samostalni, sa ili bez akumulatora za skladištenje energije u DC ili AC oblasti. Uz pomoć njega se simulira proizvodnja energije FN sistema i vrši ekonomska analiza povraćaja investicije. Takođe se proverava kompatibilnost veze string-invertor i generiše dokumentacija o projektu.

3.6. HOMER Pro 3.11

HOMER modeluje fizičko ponašanje elektroenergetskog sistema i troškove njegovog životnog ciklusa i omogućava poređenje različitih mogućnosti dizajna. Softver izračunava dostupnu obnovljivu snagu, upoređuje sa elektročnim opterećenjem i odlučuje šta uraditi sa viškom snage u trenutcima njenog viška ili kako najjeftinije iz mreže kupiti dodatnu snagu u trenutku njenog deficita.

3.7. PVSize

PVSize ima korisnički interfejs koji pruža informacije za podršku dizajneru prilikom upoređivanja različitih dizajnova. Pored toga, nudi mogućnost predviđanja performansi sistema sa bazom podataka ugrađenih vremenskih lokacija. Uz pomoć njega se izračunava i predlaže određeni sistem. Izračunavanje uključuje tehnički opis kao i finansijsku ponudu.

3.8. PV-F Chart

PV-F Chart se koristi za analizu potrebnu za projektovanje FN sistema i za procenu dugoročnih performansi interfejs sistema, sistema za skladištenje energije (baterije) i sistema bez interfejsa ili skladištenja energije. Program koristi dugoročno prosečno mesečno sunčevo zračenje i temperaturu okoline kako bi se predvidele godišnje performanse FN niza.

3.9. PVGIS 5

PVGIS služi za procenu proizvodnje električne energije samostalnog FN sistema i FN sistema priključenog na mrežu. Suština softvera je da pokaže kakav uticaj ima izabrana lokacija na planiranje FN sistema, kao i baza podataka o Sunčevom zračenju na toj lokaciji za duži vremenski period. Te informacije su potrebne kako bi se pre izgradnje solarne elektrane procenila njena optimalna proizvodnja.

3.10. Polysun 10.2

Polysun je softver koji korisnicima omogućava simulaciju solarnih, toplotnih, geotermalnih i kombinovanih sistema.

Nudi podršku kada je u pitanju projektovanje, analiza i izračunavanje instalacija u oblasti obnovljivih izvora energije.

3.11. Sunny Design Web

Sunny Design Web je softver za planiranje i projektovanje FN sistema i FN hibridnih sistema. On podržava različite vrste solarnih projekata: solarni sistemi, solarne elektrane velikih razmera, sistemi sa sopstvenom potrošnjom, ostrvski sistemi i solarni hibridni sistemi.

4. KOMPARACIJA RAČUNARSKIH SOFTVERA

Nakon izvršenog pregleda dostupnih softvera poređenjem karakteristika softvera po zadatim kriterijumima utvrđeno je koji od njih je najpogodniji za projektovanje FN sistema.

Karakteristike softvera su opisane u 11 stavki: operativni sistem, jezici, systemske opcije, sistemi za praćenje kretanja Sunca, parametri lokacije sistema, vremenski podaci, mesto montiranja, gubici, senčenje, ekonomska analiza, opcija izbora baterija.

Zatim su karakteristike softvera poređene po kriterijumima ocenjivanja od 0 do 5 bodova. U slučaju da softver nema opciju koja se poredi, on dobija 0 bodova, dok kako softver nudi više opcija tako mu bodovi rastu.

Komparativnom analizom prethodno opisanih softvera, dobijeni su sledeći rezultati (Tab.1.). Analizirajući date kriterijume uočeno je da se softver PV*SOL premium 2018 (R6) izdvaja od ostalih softvera kao softver sa najvećim dijapazonom opcija, gde je od 11 kriterijuma 6 puta ocenjen maksimalnim brojem bodova. Iza njega sa dosta manjim brojem bodova nalazi se BlueSol. Na trećem mestu su dva softvera Sunny Design Web i PVsyst, gde prvi ima više opcija operativnih sistema u kojima radi, i veći dijapazon parametara lokacije sistema, a PVsyst nudi više jezika i odabir mesta montiranja modula van krova objekta. Na petom mestu su HOMER Pro 3.11 i Polysun 10.2, gde prvi nudi veliki izbor sistema za praćenje kretanja Sunca, i vremenskih podataka, a drugi pruža veći izbor broja jezika i sistemskih opcija.

Tabela 1. Rezultati komparativne analize

	Softveri		Bodovi
1	PV*SOL premium 2018 (R6)		46
2	BlueSol		39
3	Sunny Design Web	PVsyst 6	34
4	RETScreen		33
5	HOMER Pro 3.11	Polysun 10.2	29
6	PVGIS 5		27
7	PV-F Chart		24
8	PVSize		17
9	Solmetric		16

5. SIMULACIJA U PV*SOL premium 2018 (R6)

U cilju potvrde softvera sa najvećim izborom opcija izvršena je simulacija u PV*SOL premium 2018 (R6) koji se pokazao kao najbolji od 11 obrađenih softvera.

Simulacija je izvršena izvedbom solarne elektrane na krovu kuće sa lokacijom u Heidelbergu, u Nemačkoj. U cilju uspostavljanja najveće moguće proizvodnje električne energije bilo je potrebno maksimalno iskoristiti površinu krova ali tako da objekti koji uzrokuju senčenje panela (dimnjak, antena) nemaju značajnog uticaja na smanjenje sunčeve radijacije (Sl.1.).



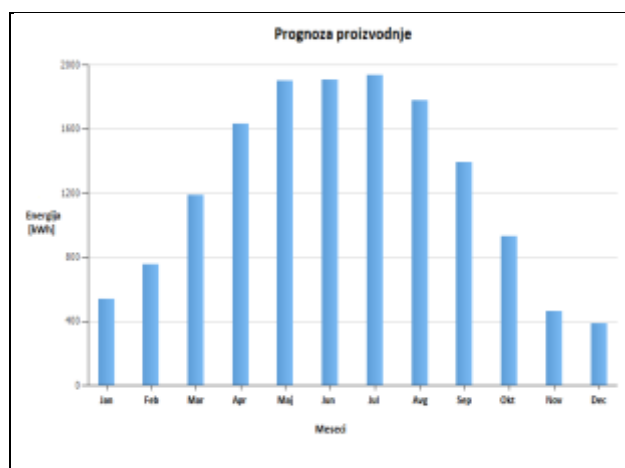
Slika 1. Smanjenje u radijaciji

Izabrani su invertori modela FRONIUS Symo 12.5-3-M. Ugrađena su dva MPPT-a, jedan se sastoji iz dva stringa sa po 12 modula u seriji, a drugi od jednog stringa sa po 22 modula u seriji.

5.1. Rezultati simulacije

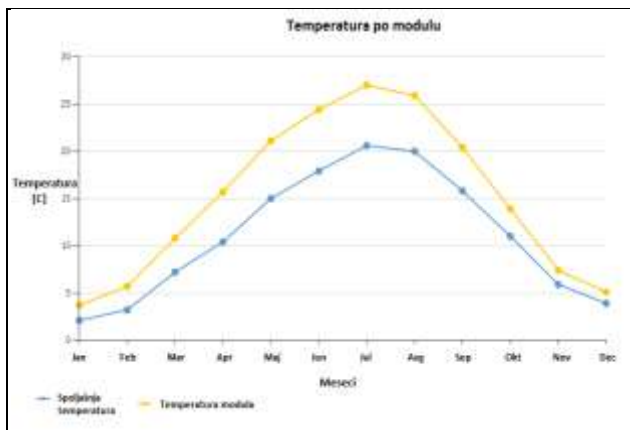
Postavljanjem ove solarne elektrane na površni krova od 75.4 m² emisija CO₂ je na godišnjem nivou smanjena za 8,882 kg/god. Vršna snaga FN panela je procenjena na 13.8 KWp, dok je godišnja proizvodnja 1,072.76 kWh/kWp. Redukcija proizvodnje prouzrokovana senčenjem je 3.5%/god. Standby potrošnja (inverter) je 12 kWh/god, dok je mrežni feed-in 14,804 kWh/god.

Na osnovu datog grafika koji prikazuje godišnju prognozu proizvodnju energije FN sistema po mesecima (Sl.2.) utvrđena je očekivana najveća proizvodnja u prolećnim i letnjim danima kada je sunčeva radijacija najveća.



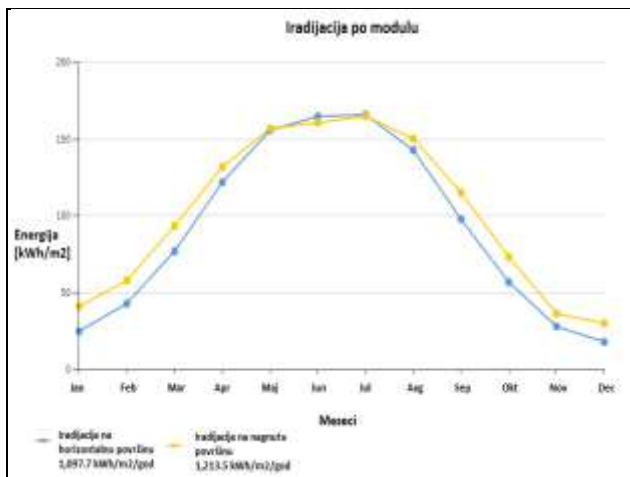
Slika 2. Godišnja prognoza proizvodnje FN sistema

Iako na proizvodnju negativno utiču visoke spoljašnje temperature, Heidelberg je grad sa blagim letnjim temperaturama (Sl.3.) pa se smanjenje proizvodnje skoro i ne primećuje.



Slika 3. Godišnji prikaz temperature po modulu u odnosu na spoljašnju temperature

Primećuje se da je iradijacija na nagnutu površinu modula veća od iradijacije koja bi bila da je modul postavljen na ravnoj površini (Sl.4.). Nagibni ugao koji je izabran u simulaciji je 30°, što se inače smatra najpogodnijim da bi iradijacija, a time i proizvodnja električne energije bila maksimalna.



Slika 4. Godišnja iradijacija po modulu

6. ZAKLJUČAK

U radu je dat opis i pregled trenutno dostupnih softvera koji se koriste za dizajn fotonaponskih sistema. Nakon izvršene komparacije datih softvera izvedeni su sledeći zaključci. Softver PV*SOL premium 2018 (R6) je izabran kao najbolji jer nudi najviše opcija unosa parametara lokacije sistema, ima velik izbor mesta montiranja FN panela, i nudi opcije računanja gubitaka, senčenja, ekonomsku analizu i opcije izbora baterija. U njemu je naknadno urađena simulacija u svrhu prikaza opcija koje softver nudi i što boljeg izbora efikasnog fotonaponskog sistema. Softveri koji su takođe procenjeni kao dobri za planiranje i modelovanje FN sistema sa preko 30 od ukupnih 55 bodova su BlueSol, Sunny Design, PVSyst 6 i RETScreen. Dok su ostali softveri sa ispod 30 bodova (HOMER Pro3.11, Polysun 10.2, PVGIS5, PV-F Chart, PVSize, Solmetric) procenjeni kao pogodni za jednostavne projekte, koji ne zahtevaju puno opcija i detalja. Laki su za shvatanje i prikladni za početnike koji se upoznaju sa tematikom projektovanja fotonaponskih elektrana.

7. LITERATURA

- [1] I. Petrović, Z. Šimić, M. Vražić, "Comparison of PV plant energy generation prediction tools with measured data", *Przeglad Elektrotechniczny*, No 89, pp. 6, 2013.
- [2] D.K. Sharma, V. Verma, AP. Singha, "Review and Analysis of Solar Photovoltaic Softwares", *International Journal of Current Engineering and Technology*, Vol.4, no.2, 2014, Available On-line: <http://inpressco.com/wp-content/uploads/2014/04/Paper48725-731.pdf>
- [3] S. Sinha, S.S. Chandel, "Review of software tools for hybrid renewable energy systems", *Renewable and Sustainable Energy Reviews*, Vol. 32, April 2014, pp.192-205.
- [4] M. Lalwani, M. Kothari, D.P. Singh, "Investigation of Solar Photovoltaic Simulation Softwares", Vol. 1, no. 3, pp. 585–601, 2010.
- [5] M. Petronijević, V. Katić, „Projektovanje fotonaponske elektrane – pregled softverskih alata“, *Zbornik radova Fakulteta tehničkih nauka*, God.27, Br.5, 2012, pp.1025-1028.
- [6] M. Repac, "Pregled softvera za modelovanje FN elektrana", *Diplomski rad, Fakultet tehničkih nauka*, Novi Sad, 2016.
- [7] M. Mega, "Komparativna analiza softvera za projektovanje FN krovnih sistema", *Diplomski rad, Fakultet tehničkih nauka*, Novi Sad, 2018.
- [8] T. Savčić, "Pregled softverskih alata za simulaciju rada fotonaponskih sistema", *Master rad, Fakultet tehničkih nauka*, Novi Sad, 2018.
- [9] A. Gavrilović, „Pregled raspoloživih softvera za solarne elektrane“, *Master rad, Fakultet tehničkih nauka*, Novi Sad, 2018.

Kratka biografija:



Andrea Gavrilović rođena je u Novom Sadu 1994. god. Gimnaziju "Svetozar Marković" završila je u Novom Sadu 2013. god. Fakultet tehničkih nauka, studijski program Energetika, elektronika i telekomunikacije upisala je 2013/2014 god. Diplomirala je 2017. god, nakon čega je upisala master studije na smeru "Elektroenergetika - Distribuirani elektroenergetski resursi", koje je završila 2018. god.



Vladimir Katić rođen je 1954. god. u Novom Sadu. Doktorirao je na Univerzitetu u Beogradu 1991. god. Od 2002. god. je redovni profesor Univerziteta u Novom Sadu. Trenutno je prodekan Fakulteta tehničkih nauka. Oblasti interesovanja su energetska elektronika, obnovljivi izvori električne energije, električna vozila i kvalitet električne energije.

МОДЕЛОВАЊЕ, АНАЛИЗА И ОПТИМИЗАЦИЈА КОНТРОЛЕ БЕЖИЧНОГ ПУЊАЧА БАТЕРИЈЕ ЗА ЕЛЕКТРИЧНИ АУТОМОБИЛ**MODELLING, ANALYSIS AND CONTROL OPTIMIZATION OF WIRELESS BATTERY CHARGER FOR ELECTRIC VEHICLE**

Драгана Шкрбић, *Факултет техничких наука, Нови Сад, Србија*

Хесус А. Оливер, *Политехнички универзитет, Мадрид, Шпанија*

Владимир Катић, *Факултет техничких наука, Нови Сад, Србија*

Област – ЕЛЕКТРОТЕХНИКА

Кратак садржај – У овом раду обрађено је моделовање, анализа и оптимизација контроле бежичног пуњача батерије за електричне аутомобиле. Приказани су резултати симулација и донесени закључци. Анализирана су два начина контроле излазне снаге, која могу да буду примјењена. На крају је приказан дио урађеног експеримента.

Кључне ријечи: електрични аутомобил, бежични пуњач батерије

Abstract – In this paper is processed modelling, analysis and control optimization of wireless battery charger for electric vehicle. The simulation results and conclusions are presented. Two ways of controlling the output power were analyzed, which can be applied. Finally, the part of the experiment that was done is shown.

Key words: electric vehicle, wireless battery charger

1. УВОД

Проблем загађења животне средине и емисије штетних гасова све је актуелнији. Једно од рјешења је и замјена мотора са унутрашњим сагоријевањем (СУС) електричним. Због тога је употреба електричних аутомобила сваке године све већа. Међутим, проблеми њихове употребе су ограничен дomet, мала доступност пуњача и дуго вријеме допуњавања батерија. Пуњење батерија се може вршити кондуктивно (кабелски, жично) и бежично. У овом раду се разматрају само бежични системи.

Бежично пуњење батерија електричног аутомобила функционише на принципу магнетне индуктивности и резонанце [1]. Бежични пренос се практично врши помоћу DC/AC претварача (инвертора) на примарној страни и AC/DC претварача (исправљача) на секундарној страни. Примарна и секундарна страна су повезане преко индуктивних калемова [2].

НАПОМЕНА:

Овај рад проистекао је из мастер рада, који је рађен на Политехничком универзитету у Мадриду и одбрањен на ФТН у Новом Саду у склопу студентске размене Ерасмус+ пројекта, а чији супервизор је био др Хесус А. Оливер, ванр. проф., а ментор др Владимир А. Катић, ред. проф.

Анализа оптималне ефикасности подразумјева израчунавање фактора испуне исправљача. Вриједност се добија помоћу MPPT (eng. *maximum power point tracking*) алгоритма, који израчунава максималну снагу на бази улазних напона и струје. Током анализе оптималне ефикасности неопходно је размотрити три аспекта: фазни став између DC/AC и AC/DC претварача, вриједност напона DC кола и утицај *coupling* фактора.

Управљање може бити реализовано на два начина: 1. контролом излазне снаге помоћу фазног помјераја између DC/AC и AC/DC претварача и 2. контролом излазне снаге помоћу DC напона AC претварача.

У раду је представљен модел овог пуњача, урађена је симулација пуњења батерије и дјелимично реализована експериментална потврда.

2. ЕЛЕКТРИЧНИ АУТОМОБИЛИ

Године 1834-те Томас Девенпорт конструисао је први електрични аутомобил, који је заправо био трицикл. Био је напајан из батерије, која нија имала могућност пуњења и користио се за кратке релације. Данас, после више од 125 година развоја СУС мотора, аутомобилска индустрија постепено прелази на електрични погон. Постоје три главна типа електричних аутомобила: хибридни електрични аутомобили (HEVs), *plug-in* хибридни електрични аутомобили (PHEVs) и потпуни електрични аутомобили (FEVs). Основна разлика између ових типова аутомобила је да посљедња два посједују батерије које могу бити екстерно пуњене [3].

Бежични начин пуњења батерија базира на развијеној регулативи. SAE J2954 је установљена препорука која дефинише критеријуме за електромагнетну компатибилност, минималне перформансе, сигурност и тестирање бежичног пуњења и *plug-in* електричних аутомобила. Стандард за бежични пренос енергије (WPT) заснован је на различитим нивоима, који треба да обезбиједе селекцију нивоа пуњења по тржишним критеријумима. Ово омогућава бољу опрему аутомобила и лакше кориштење.

3. АНАЛИЗА ОПТИМАЛНЕ ЕФИКАСНОСТИ

Приликом анализе оптималне ефикасности бежичног пуњења батерије електричног аутомобила неопходно

је размотрити следеће аспекте: фазни став између DC/AC и AC/DC претварача; вриједност напона DC кола и утицај *coupling* фактора (k).

Прије претходно наведених анализа, потребно је одредити оптималну вриједност фактора испуне d AC/DC претварача за максималну излазну снагу. Прва симулација је урађена са чисто отпорним потрошачем вредности $24,26 \Omega$, у паралели са излазним кондензатором (слика 1) [4].

3.1. Анализа фазног става између претварача

За анализу фазног става између DC/AC и AC/DC претварача искоришћена је симулациона шема приказана на слици 2. Потрошач је константан напонски извор вриједности 300 V , а пренос снаге зависи од фазног помјераја. Резонантна фреквенција за коришћене параметре кола је 85 kHz .

3.2. Анализа вриједности напона DC кола

За анализу напона DC кола потребно је симулацију са слике 2 модификовати тако да напонски извор на излазу буде контролисан, а фазни помјерај између DC/AC и AC/DC претварача фиксан. Изабран је фазни помјерај од 90° , јер се тада обезбеђује најбоља ефикасност. Напонски извор је управљан MPPT алгоритмом, а за почетну вриједност узета је вриједност од 300 V . Вриједности параметара у овој симулацији су: фазни став DC/AC претварача је 90° , фазни став AC/DC претварача је 0° , $k=0,25$ и $d=0,5$. Добијена је вриједност излазне снаге од 7100 W , а резултати су приказани на слици 3.

3.3. Анализа утицаја *coupling* фактора

Coupling фактор (k) је од изузетне важности за бежични пренос енергије. Њиме се изражава међусобна спрегнутост калемова који преносе енергију. Изражава се следећом формулом:

$$k = \frac{M}{\sqrt{L_1 \cdot L_2}} \quad (1)$$

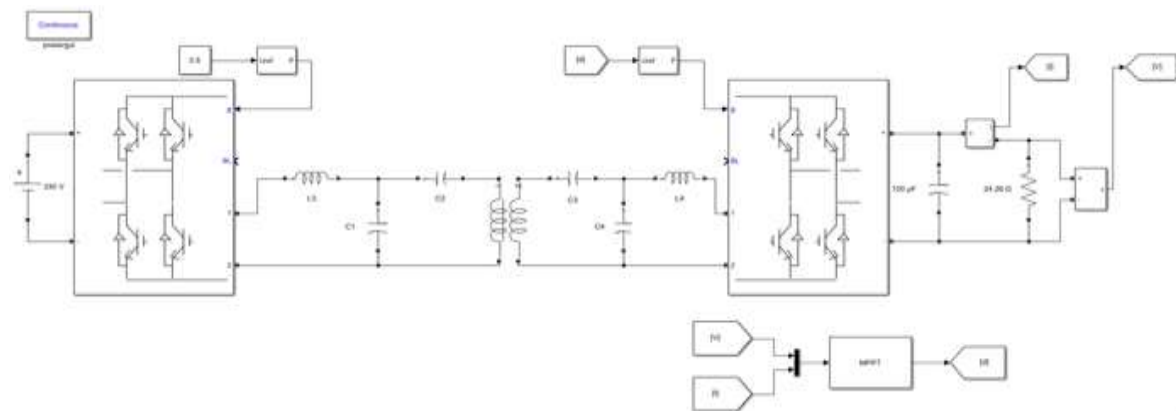
Утицај *coupling* фактора је размотрен на симулацији са константним напонским извором (слика 2). Прво су претпостављени следећи параметри: фазни став DC/AC претварача је 90° , фазни став AC/DC претварача је 0° , $k=0,187$ и $d=0,5$. Добијена је излазна снага 5260 W , а на слици 4 су приказани напон и струја на секундарној страни.

У другом кораку коришћени су исти параметри, само је повећан *coupling* фактор на $k=0,25$. Добијена је излазна снага 7100 W , односно исте вредности као на слици 3.

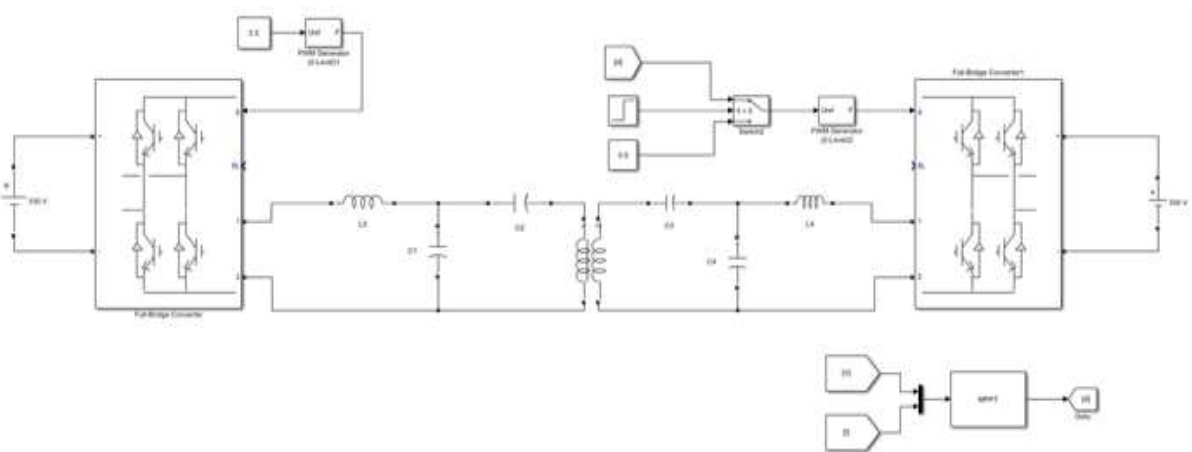
Такође, размотрен је и утицај ниже вредности k . За $k=0,109$ добијена је излазна снага од 3000 W , а напон и струја нису фазно усклађени. Закључено је да мање вриједности *coupling* фактора не одговарају.

4. КОНТРОЛА

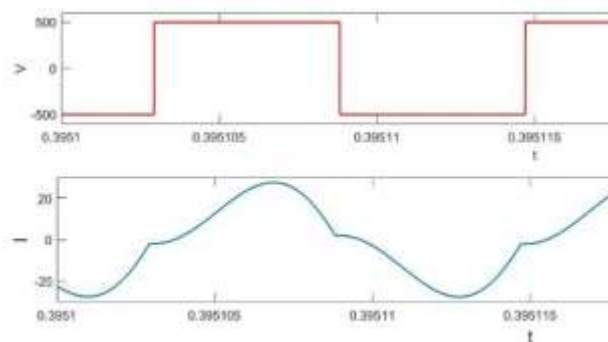
Разматрана су два система управљања (контроле) излазне снаге: 1. помоћу фазног помјераја између DC/AC и AC/DC претварача, и 2. помоћу DC напона AC претварача.



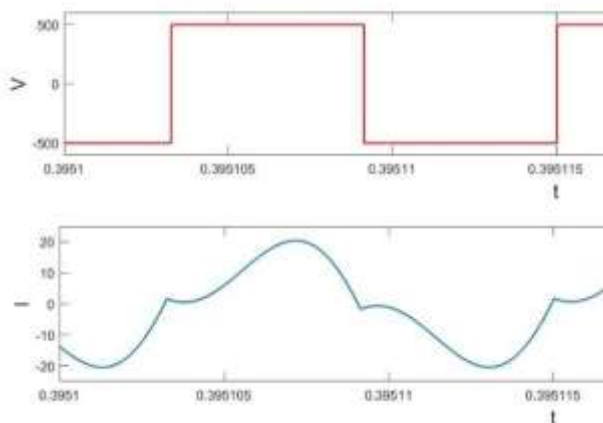
Слика 1. Симулациона шема са отпорним потрошачем.



Слика 2. Симулациона шема са константним напонским извором као потрошачем.



Слика 3. Резултати симулације за $k=0,25$ и $d=0,5$.



Слика 4. Резултати симулације за $k=0,189$ и $d=0,5$.

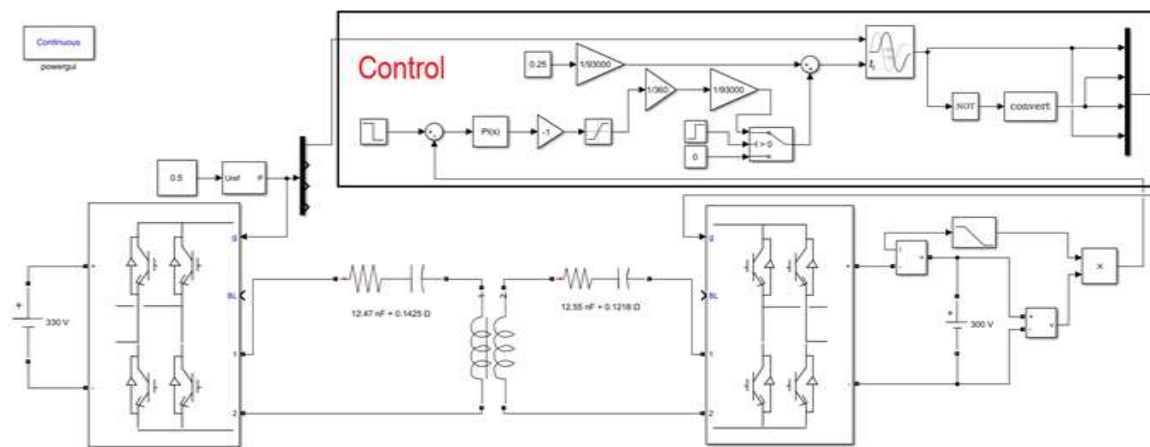
4.1. Контрола излазне снаге помоћу фазног помјераја између DC/AC и AC/DC претварача

Прво је урађена симулација са константним напонским извором од 300 V (слика 5). Ту је контролисана излазна снага помоћу фазног помјераја између претварача. Утврђено је да је потребно испоручити мању снагу, ако је фазни помјерај између DC/AC и AC/DC већи.

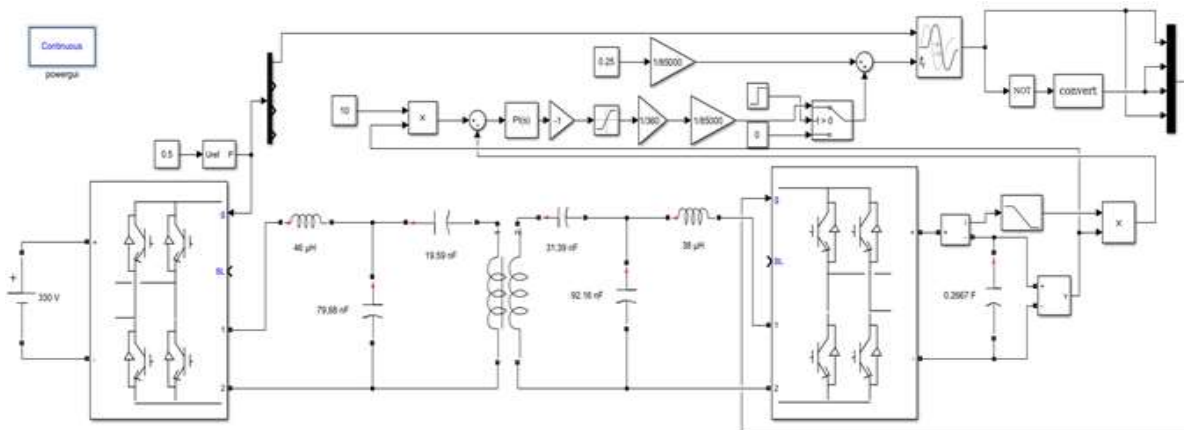
4.2. Контрола излазне снаге помоћу DC напона AC претварача

Током анализа закључено је да је излазни DC напон линеарно повезан са излазном снагом ако је струја константна, што значи да ако повећавамо вриједност излазног напона и излазна снага ће се повећати. Да би то било приказано на мало занимљивији начин одрађена је симулација пуњења батерије.

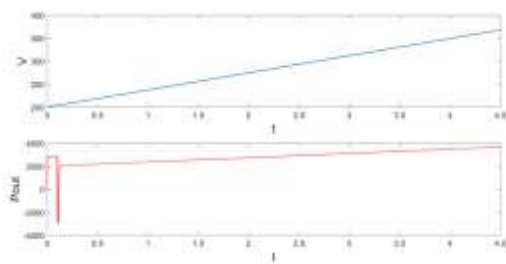
Симулациона шема пуњења батерије приказана је на слици 6. Умјесто константног напонског извора на излазу користи се велики кондензатор, чија капацитивност је 0,2667 F и који заправо представља батерију. Струја на излазу је константна и износи 10 A. Референца за контролу снаге се добија тако што се множе константна струја и измјерени напон на кондензатору. Фазни помјерај је фиксан и износи 45° . Резултати симулације, тј. приказ односа излазне струје и напона, односно снаге пуњења батерије дати су на слици 7.



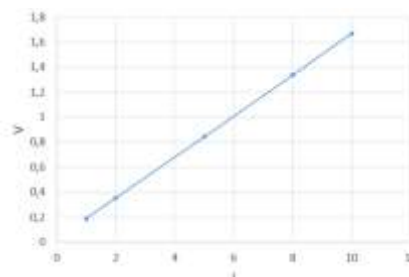
Слика 5. Симулациона шема са константним напонским извором за први начин контроле



Слика 6. Симулациона шема пуњења батерије.



Слика 7. Резултати симулације пуњења батерије.



Слика 10. Резултати струјног тестирања

5. ЕКСПЕРИМЕНТАЛНА ПРОВЈЕРА

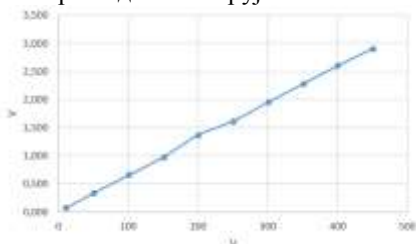
Резултати симулација контроле излазне снаге дјелимично су провјерени на лабораторијском прототипу који има LC топологију (слика 8). Прототип има следеће параметре примарне и секундарне стране бежичног пуњача: $L_1=228,7 \mu\text{H} + 0,415 \Omega$; $L_2=207,3 \mu\text{H} + 0,318 \Omega$ и $k = 0,2$. Резонантни кондензатори су: $C_1=12,47 \text{ nF} + 0,1425 \Omega$ и $C_2=12,55 \text{ nF} + 0,1218 \Omega$. За струјно и напонско мјерење коришћени су већ постојећи сензори, који су прилагођени вриједностима прототипа. На сликама 9 и 10 приказани су резултати напонског и струјног тестирања, респективно. Може се уочити линеарни одзив, односно адекватно прилагођење реалних вредности параметрима аналогних улаза DSP-а.



Слика 8. Приказ експерименталне поставке.

6. ЗАКЉУЧАК

У раду је изложена проблематика бежичног начина пуњења електричних аутомобила. Установљени су најбољи начини подешавања претварача у циљу максималне ефикасности. Такође је показано како вриједност напона у DC колу утиче на излазну снагу. Обрађена су два начина контроле пуњења. Корисник бира у складу са својим захтјевима најбољи начин. На крају је у експерименталном дијелу потврђен линеаран одзив за струјно и напонско мјерење.



Слика 9. Резултати напонског тестирања.

7. ЛИТЕРАТУРА

- [1] D. Maksimović, R.W. Erickson, "Fundamentals of Power Electronics", 2nd Ed., Kluwer Academic Publishers, New York, 2004.
- [2] T.M. Fisher, K.B. Farley, Z. Gao, H. Bai, Z. Ho Tse, "Electric vehicle wireless charging technology: a state-of-the-art review of magnetic coupling systems", Wireless Power Transfer, Vol.1, Issue 2, Sep.2014, pp.87–96.
- [3] A. Berger, M. Agostinelli, S. Vesti, J. A. Oliver, J.A. Cobos, M. Huemer, "A Wireless Charging System Applying Phase-Shift and Amplitude Control to Maximize Efficiency and Extractable Power", IEEE Trans. on Power Electronics, Vol.30, No.11, Nov. 2015, pp.6338 – 6348.
- [4] A. Poullikkas, "Sustainable options for electric vehicle technologies", Renewable and Sustainable Energy Reviews, vol. 41(C), 2015, pp.1277-1287.

Кратка биографија:



Драгана Шкрбић, дипл. инж. рођена је 1994. године у Мркоњић Граду. Факултет техничких наука, студијски програм Енергетика, електроника и телекомуникације (ОАК) уписала је школске 2013/2014. На студијама се определијелила за модул Електро-енергетика – Енергетска електроника и електричне машине и дипломирала 2017. год. Мастер студије истог модула завршила је 2018. год.



Хесус А. Оливер (Jesus A. Oliver), ванр. проф. рођен је у Толеду, Шпанија 1972. год. Мастер и докторске студије завршио је на Политехничком универзитету у Мадриду, Шпанија 1996. и 2007. год., респективно. Области интересовања су моделовање и управљање уређаја енергетске електронике, системи са горивим ћелијама и енергетска ефикасност.



Владимир А. Катић, ред.проф. рођен је 1954. године у Новом Саду. Дипломирао је на Факултету техничких наука у Новом Саду 1978. год., а магистрирао и докторирао на Електротехничком факултету Универзитета у Београду 1981. и 1991. год., респективно. Области интересовања су енергетска електроника, обновљиви извори и квалитет електричне енергије, и електрична возила.

MODELOVANJE SAVREMENIH METODA SKLADIŠTENJA ELEKTRIČNE ENERGIJE
MODELING OF MODERN ELECTRICAL ENERGY STORAGE SYSTEMS

Mirko Macanović, *Fakultet tehničkih nauka, Novi Sad, Srbija*

Octavian Cornea, *Politehnički univerzitet u Temišvaru, Temišvar, Rumunija*

Vladimir Katić, *Fakultet tehničkih nauka, Novi Sad, Srbija*
Oblast – Elektrotehnika i računarstvo

Kratak sadržaj – U radu je obrađeni model olovne baterije. Eksperimentalno su određeni njeni parametri. Prikazani su i upoređeni rezultati dobijeni eksperimentalno i u simulaciji. Takođe su obrađeni uprošćeni modeli litijum jonske baterije i superkondenzatora i prikazani su rezultati. Sve simulacije su pravljene u Matlabu/Simulinku.

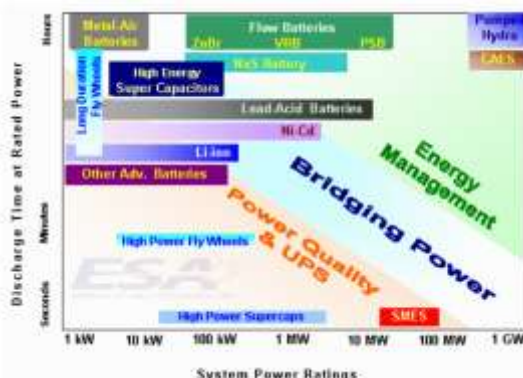
Ključne reči: Olovna baterija, litijum jonska baterija, Superkondenzator, Matlab/Simulink model baterije

Abstract – In this paper model of lead-acid battery in Matlab/Simulink is presented. Parameter identification is performed experimentally. Comparison of simulation and experimental results is done. Simplified model of supercapacitor and lithium ion battery and Matlab/Simulink simulation results are given, also.

Keywords: Lead acid battery, Lithium ion battery, Supercapacitor, Matlab/Simulink model of battery

1. UVOD

U današnje vreme sve više se razmatra upotreba električnih automobila, koji za skladištenje energije koriste baterije. Na slici 1 su prikazane različite vrste ovih sistema [1]. Sortiranje je izvršeno u odnosu na vreme pražnjenja pri nominalnoj snazi i snage. U nastavku će biti obrađene olovne i Li-Ion baterije i superkondenzatori.



Slika 1. Sistemi za skladištenje električne energije [1]

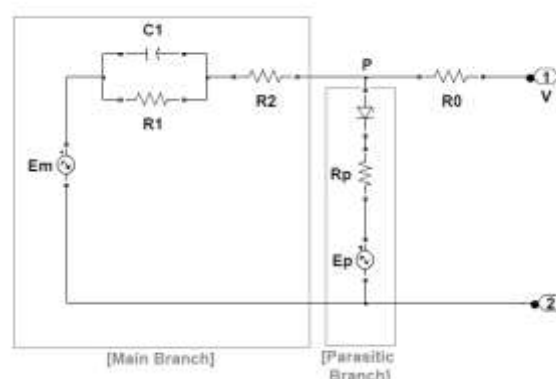
NAPOMENA:

Ovaj rad proistekao je iz master rada, koji je raden na Politehničkom univerzitetu u Temišvaru i odbranjen na FTN u Novom Sadu u sklopu studentske razmene Erasmus+ projekta, a čiji supervizor je bio dr Octavian Cornea, a mentor dr Vladimir A. Katić, red. prof.

Modelovanjem baterija se simulira rad baterija u različitim uslovima i situacijama. U ovom radu prvo će biti dato ekvivalentno kolo olovne baterije, urađen eksperiment za određivanje njenih parametara, kao i neke simulacije. Zatim će biti opisani modeli litijum-jonske (Li-Ion) baterije i superkondenzatora i urađene odgovarajuće simulacije u Simulinku. Na kraju su rezultati i zaključak.

2. EKVIVALENTNO KOLO OLOVNE BATERIJE

Ekvivalentno kolo olovne baterije predstavlja se sa otpornikom na red sa RC paralelnom granom i sa naponskim izvorom koji simulira napon praznog hoda i sa paralelnom granom u vidu redne veze otpornika i naponskog izvora koji predstavlja napon praznog hoda parazitne grane (slika 2).



Slika 2. Ekvivalentno kolo olovne baterije [2]

Naponom glavne grane se računa tako da se pretpostavlja da je elektromotorna sila konstantna kada je baterija napunjena 100%. Napon zavisi od temperature i od stanja napunjenosti baterije (state of charge - SOC):

$$E_m = E_{m0} - K_E(273 + \theta)(1 - SOC) \quad (1)$$

gde je E_m – napon praznog hoda u [V]; E_{m0} – napon praznog hoda pune baterije u [V]; K_E – konstanta u V/°C; θ – temperatura elektrolita u °C.

Unutrašnji otpor baterije se opisuje pomoću (2) gde je pretpostavljeno da je otpor konstantan na svim temperaturama i da zavisi samo od SOC:

$$R_0 = R_{00}[1 + A_0(1 - SOC)] \quad (2)$$

gde je R_0 – otpor u [Ω]; R_{00} – vrednost R_0 za SOC=1 u [Ω]; A_0 – konstanta.

Otpor glavne grane R_1 opisuje se pomoću (3) i zavisi od dubine punjenja (depth of charge - DOC). DOC predstavlja suprotnost od SOC, tj. kada je baterija puna

$DOC=0$, a kada je prazna $DOC=1$. Ovaj parametar se koristi pri diskusiji o životnom veku baterije. Otpor raste eksponencijalno kako se baterija prazni i dat je sa:

$$R_1 = -R_{10} \ln(DOC) \quad (3)$$

gde je R_1 – otpor glavne grane $[\Omega]$; R_{10} – konstanta u $[\Omega]$. Kapacitivnost glavne grane ili vremensko kašnjenje je dato sa (4), s tim da vremenska konstanta modeluje kašnjenje napona kada se promeni struja baterije.

$$C_1 = \tau_1 / R_1 \quad (4)$$

gde je C_1 – kapacitivnost glavne grane u faradima; τ_1 – vremenska konstanta glavne grane u sekundama.

Otpor glavne grane R_2 se računa preko (5) i raste eksponencijalno kako se SOC povećava. Vrednost varira u zavisnosti od struje glavne grane. Ovaj otpor je značajan prilikom punjenja baterije, dok prilikom pražnjenja baterije nije toliko važan i može se zanemariti.

$$R_2 = R_{20} \frac{\exp[A_{21}(1-SOC)]}{1 + \exp(A_{22}I_m/I^*)} \quad (5)$$

gde je R_2 – otpor glavne grane u $[\Omega]$; R_{20} – konstanta u $[\Omega]$; A_{21} – konstanta; A_{22} – konstanta; I_m – struja glavne grane u $[A]$; I^* – nominalna struja baterije u $[A]$.

Strujom parazitne grane se opisuju gubitci struje kada je baterija napunjena. Parazitna struja je data jednačinom (6). Struja zavisi od temperature elektrolita kao i od napona parazitne grane. Ova struja uglavnom ima male vrednosti, osim u slučajevima kada se baterija puni pri visokoj vrednosti SOC :

$$I_p = V_{PN} G_{p0} \exp\left(\frac{V_{PN}}{V_{p0}} \frac{1}{(\tau_p s + 1)} + A_p \left(1 - \frac{\theta}{\theta_f}\right)\right) \quad (6)$$

gde je I_p – struja parazitne grane; V_{PN} – napon parazitne grane; G_{p0} – konstanta u $[s]$; τ_p – vremenska konstanta parazitne grane u $[s]$; V_{p0} – konstanta u $[V]$; A_p – konstanta; θ_f – temperatura zaleđivanja elektrolita u $^{\circ}C$. Količina naelektrisanja koja se oslobodi iz baterije je data izrazom (7). Ona se računa tako što se integriše struja koja protiče kroz ili iz glavne grane.

$$Q_e(t) = Q_{einit} + \int_0^t -I_m(\tau) d\tau \quad (7)$$

gde je Q_e – energija koju je baterija odala; Q_{einit} – energija koja je inicijalno odata; I_m – struja glavne grane u $[A]$; t – vreme simulacije u $[s]$.

Totalni kapacitet baterije zavisi od struje pražnjenja baterije i od temperature elektrolita (jednačina (8)). Zavisnost kapaciteta baterije je samo od struje pražnjenja. Prilikom punjenja struja pražnjenja u (8) jednaka nuli:

$$C(I, \theta) = \frac{K_c C_0^*}{1 + (K_c - 1) \left(\frac{I}{I_f}\right)} \delta \left(1 - \frac{\theta}{\theta_f}\right)^\varepsilon \quad (8)$$

gde je C_0^* – kapacitet neopterećene baterije na $0^{\circ}C$ u $A \cdot s$; I – struja pražnjenja u $[A]$; K_c , δ i ε – konstante.

SOC i DOC se računaju kao količnik dostupne količine napunjenosti baterije i ukupnog kapaciteta baterije. SOC baterije se izražava deo punjenja koji je na raspolaganju u bateriji (izraz (9)). DOC baterije izražava deo korisnog kapaciteta baterije pri nominalnoj struji pražnjenja (izraz (10)). Veće struje pražnjenja čine da se baterija brže prazni, zbog čega je uvek DOC manji ili jednak od SOC :

$$SOC = 1 - \frac{Q_e}{C(0, \theta)} \quad (9)$$

$$DOC = 1 - \frac{Q_e}{C(I_{avg}, \theta)} \quad (10)$$

gde je Q_e – napunjenost baterije u $A \cdot s$; C – kapacitet baterije u $A \cdot s$; I_{avg} – srednja vrednost struje pražnjenja $[A]$. Srednja vrednost struje pražnjenja (I_{avg}) se računa kao:

$$I_{avg} = \frac{I_m}{(\tau_1 s + 1)} \quad (11)$$

gde je τ_1 – vremenska konstanta glavne grane u $[s]$.

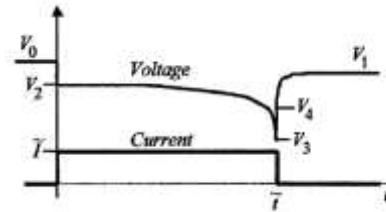
Termalnim modelom se modeluje temperatura elektrolita baterije, odnosno promena temperature elektrolita, usled gubitaka na unutrašnjim otpornostima i usled temperature ambijenta – izraz (12):

$$\theta(t) = \theta_{init} + \int_0^t \frac{(P_s - \frac{\theta - \theta_a}{R_\theta})}{C_\theta} d\tau \quad (12)$$

gde je θ_a – temperatura ambijenta u $^{\circ}C$; θ_{init} – početna temperatura baterije u $^{\circ}C$ (pretpostavljeno $\theta_{init} = \theta_a$); P_s – Džulovi gubitci na R_0 i R_2 u W ; R_θ – termalni otpor u $^{\circ}C/W$; C_θ – termalni kapacitet u $J/^{\circ}C$.

3. POSTAVKA EKSPERIMENTA

Da bi se odredile konstante u prethodno navedenim jednačinama, mora se izvesti eksperiment sa slike 3.



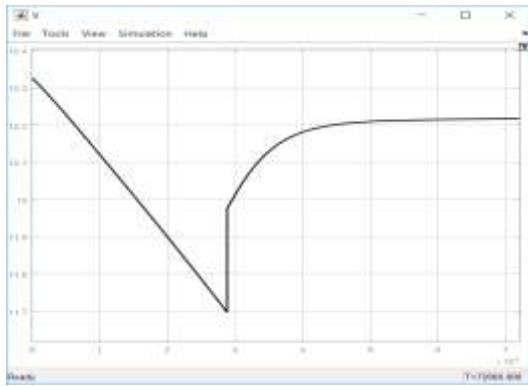
Slika 3. Eksperiment za određivanje parametara [3]

Ovaj test koji se izvodi u prostoriji u kojoj je temperatura ograničena po sledećem redosledu.

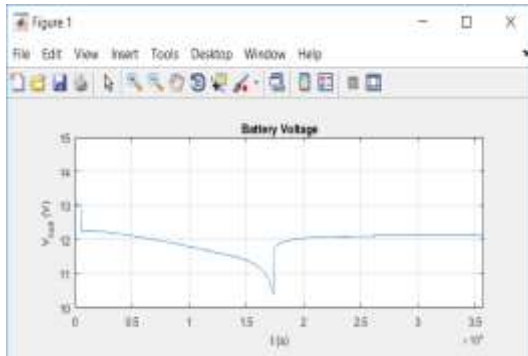
- 1) Prvo, baterija se napuni do 100%.
- 2) Zatim se baterija rasklači sa bilo kakvih spoljnih kola da bi se napon stabilizovao napona (ova faza može da potraje i nekoliko sati).
- 3) Stvarni test se sastoji od merenja početnog napona V_0 (za koji se može pretpostaviti da je jednak elektromotornoj sili koja odgovara stanju punjenja baterije i stvarnoj temperaturi elektrolita).
- 4) Baterija se prazni konstantnom strujom pražnjenja vremenski interval t .
- 5) Registruje se tranzijentni odgovor baterije dok se napon ne stabilizuje na vrednosti V_1 jednakoj elektromotornoj sili baterije [3].

4. REZULTATI

Napon dobijen simulacijom ima slične vrednosti kao napon dobijen eksperimentalnim putem. Razlika se javlja u vrednosti pada napona koja je u simulaciji išla od 12,3 - 11,7 V dok je u eksperimentu bila od 12,2 - 10,5 V. Ta razlika se javlja zbog loših karakteristika baterije prilikom velikog broja punjenja i pražnjenja gde dolazi do smanjenja njenog kapaciteta. Takođe razlike ima i u tranzijentnom periodu baterije, gde se na simuliranom naponu primećuje koleno kojeg nema u eksperimentalnim rezultatima. To je rezultat činjenice da je korišćena samo jedna RC grana, da ih je korišćeno više bili bi bolji rezultati, ali bi i model bio dosta složeniji i teži za modelovanje. Sa ovakvim modelom se postižu dosta dobri rezultati. Nakon uspešno obavljenog eksperimenta dobijaju se parametri baterije prikazani u tabeli 1.



a)



b)

Slika 4. a) rezultati simulacije b) rezultati eksperimenta

Tabela 1. Parametri baterije potrebni za simulaciju rada

Parametri koji se odnose na kapacitet baterije	$I^*=49A$ $C_0^*=261,9 Ah$ $K_c=1,18$ $\theta_f=-40 ^\circ C$ $\varepsilon=1,29$ $\delta=1,4$
Parametri koji se odnose na glavnu granu ekvivalentnog kola baterije	$\tau_1=5000s$ $E_{m0}=2,135V$ $K_E=0,58e^{-3} V/^{\circ}C$ $R_{00}=2 m\Omega$ $R_{10}=0.7 m\Omega$ $A_0=-0.3$ $R_{20}=15 m\Omega$ $A_{21}=-8$ $A_{22}=-8,45$
Parametri koji se odnose na reakciju parazitne grane ekvivalentnog kola baterije	$E_p=1,95V$ $V_{p0}=0,1V$ $G_{p0}=2 \cdot 10^{-12}s$ $A_p=2$
Parametri koji se odnose na termalni model baterije	$C_0=15 Wh/^{\circ}C$ $R_0=0,2 ^{\circ}C/W$

5. MODELOVANJE Li-Ion BATERIJA

Pored modela olovnih baterija, napravljen je model i litijum jonskih baterija. Model se sastoji od jednačina kojima se određuje vrednost elemenata ekvivalentnog kola čime se dobija dobra simulacija rada baterije. Model ekvivalentnog kola je prikazan na slici 5. U simulaciji je korišćena pulsirajuća strija pražnjenja od 1,3 A. Perioda struje je bila 5000 s, a napon je počeo blago da opada, dok je SOC pao na 40%.

Elementi ekvivalentnog kola se modeluju sa polinomskom zavisnošću od stanja napunjenosti baterije. Formula za napon praznog hoda je:

$$OCV(SOC) = K_0 + K_1 \cdot SOC + K_2 \cdot SOC^2 + K_3 \cdot SOC^3 + K_4 \cdot SOC^4 + K_5 \cdot SOC^5 + K_6 \cdot SOC^6 \quad (13)$$

gde su $K_0, K_1, K_2, K_3, K_4, K_5, K_6$ - su konstante. Unutrašnja otpornost baterije se dobija preko formule:

$$R_0(SOC) = A_0 + A_1 \cdot SOC + A_2 \cdot SOC^2 + A_3 \cdot SOC^3 + A_4 \cdot SOC^4 + A_5 \cdot SOC^5 + A_6 \cdot SOC^6 + A_7 \cdot SOC^7 + A_8 \cdot SOC^8 \quad (14)$$

gde su $A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8$ - konstante. Formule za izračunavanje elemenata prve RC paralelne grane su:

$$R_1(SOC) = a_{r1} + b_{r1} \cdot \exp(c_{r1} \cdot SOC) \quad (15)$$

$$C_1(SOC) = a_{c1} \cdot SOC^2 + b_{c1} \cdot SOC + c_{c1} \quad (16)$$

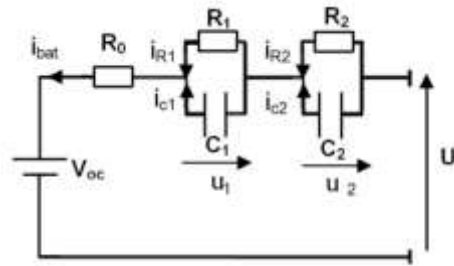
Formule za izračunavanje elemenata druge RC paralelne grane su:

$$R_2(SOC) = a_{r2} + b_{r2} \cdot \exp(c_{r2} \cdot SOC) \quad (17)$$

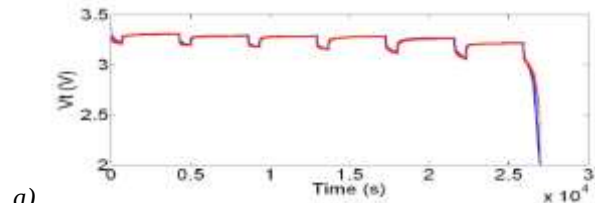
$$C_2(SOC) = a_{c2} \cdot SOC^4 + b_{c2} \cdot SOC^3 + c_{c2} \cdot SOC^2 + d_{c2} \cdot SOC + e_{c2} \quad (18)$$

6. REZULTATI SIMULACIJE

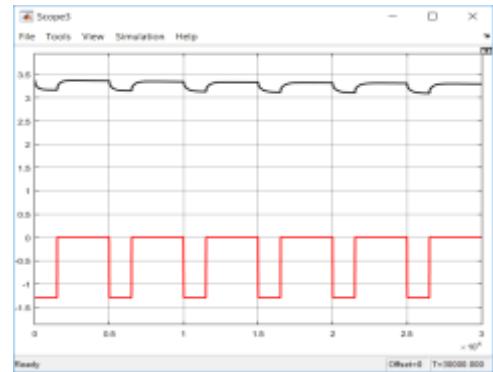
Rezultati simulacije i eksperimenta su prikazani na slici 6. Upoređivanjem rezultata se vidi da eksperimentalno dobijeni rezultati ne odstupaju puno od onih dobijenih simulacijom. Zbog jednostavnosti litijum jonske baterije smo dobili mnogo bolje rezultate nego u slučaju olovne baterije. Razlog za to je da je ekvivalentno kolo litijum jonske baterije jednostavnije i da ga je samim tim i lakše i bolje modelovati u Simulinku.



Slika 5. Ekvivalentno kolo baterije[4]



a)



b)

Slika 6. a) Rezultati eksperimenta [5] b) Rezultati simulacije (struja - crveno, napon crnom bojom)

7. MODELOVANJE SUPERKONDEZATORA

Za modelovanje superkondenzatora se mogu koristiti različita ekvivalentna kola. Ovde je korišćeno kolo koje je osmislio Faranda (slika 7).

Faranda predlaže metodu za određivanje parametara u modelu zasnovanu na merenjima. S obzirom na dve tačke $P_1(t_1, V_1)$ i $P_2(t_2, V_2)$ na krivi napon-vreme tokom punjenja sa konstantnom strujom, C_0 i k_v se mogu odrediti pomoću:

$$C_0 = \left\{ \frac{t_1}{V_1} - \frac{V_1 \cdot t_2 - V_2 \cdot t_1}{V_1^2 - V_1 \cdot V_2} \right\} \cdot I_c \quad (19)$$

$$k_v = 2 \cdot \left\{ \frac{V_1 \cdot t_2 - V_2 \cdot t_1}{V_1 \cdot V_2 - V_1^2} \right\} \cdot I_c \quad (20)$$

Parametri druge RC grane određuju se analizom ponovnog ponašanja napona praznog hoda nakon punjenja do određenog naponskog nivoa. Analizirajući propad napona praznog hoda u srednjem vremenu (do 500 s), mogla bi se naći vremenska konstanta ovog približno eksponencijalnog propada, koja bi trebala biti približno jednaka vremenskoj konstanti srednje grane $\tau_2 = R_2 C_2$. Nakon dve vremenske konstante pretpostavlja se da je napon V_{2f} preko obe kapacitivnosti isti, pri čemu se za izračunavanje C_2 može koristiti ukupno stanje napunjenosti, dato jednačinom:

$$Q_{tot} = I_c \cdot T_c = C_2 \cdot V_{2f} + (C_0 + \frac{k_v}{2} \cdot V_{2f}) \cdot V_{2f} \quad (21)$$

gde je I_c konstantna struja punjenja pre prekida struje; T_c vreme punjenja; C_0 i k_v opisuju promenljivu kapacitivnost $C(V)$. R_2 se zatim računa kao $R_2 = \tau_2 / C_2$.

8. REZULTATI SIMULACIJE

Simulacija je trajala 2.000 s i napravljena tako da je prvo korišćena struja punjenja od 40 A u vremenu od 220 s, nakon toga struja pada na nulu i ostaje nula do 1.100 s, gde se događa struja pražnjenja od -40 A do 1.300 s i onda struja opet pada na nulu do kraja vremena simulacije. Nakon struje punjenja i pražnjenja od 40 A dobija izgled napona i struje superkondenzatora kao na slici 8. Na slici 6 se vidi da kada struja padne na 0, prvo dolazi do pada napona zbog unutrašnje omske otpornosti, a zatim i do eksponencijalnog pada usled vremenskog kašnjenja na kapacitivnostima. I nakon toga dolazi do struje pražnjenja i tada je pad napona veoma izražen. Bitno je primetiti da se napon superkondenzatora menja izuzetno brzo, i da je zbog toga njegova velika prednost kada se ima potreba za brzim čuvanjem ili isporukom energije.

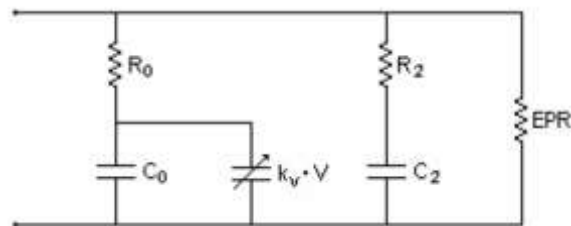
9. ZAKLJUČAK

Rezultati dobijeni eksperimentalno se nešto razlikuju od rezultata dobijenih simulacijom zbog lošijih karakteristika baterije, koje su posledica velikog broja ciklusa punjenja i pražnjenja. Takođe, i izgled dobijenog napona nije isti. Dodavanjem još RC paralelnih grana mogu se dobiti bolji rezultati, ali to povećava složenost i otežava izračunavanje elemenata ekvivalentnog kola potrebnih za simulaciju rada baterije.

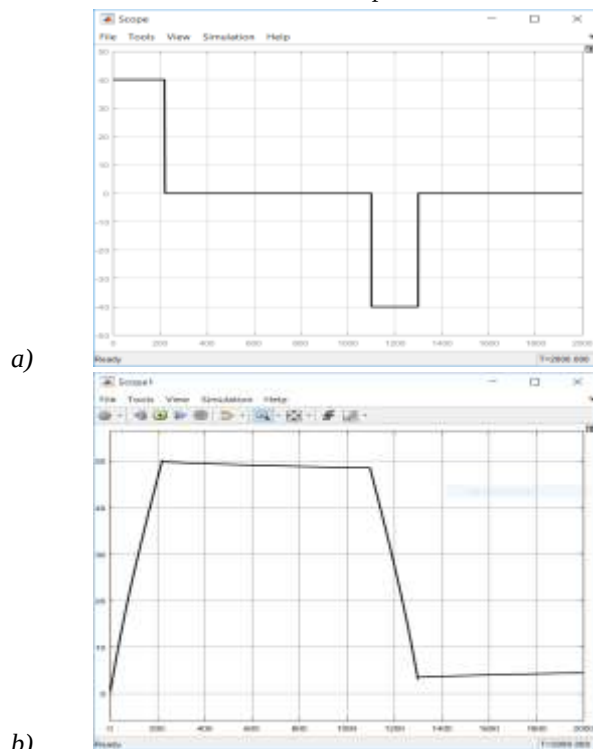
10. LITERATURA

- [1] <http://zebu.uoregon.edu/disted/ph162/l8.html>
- [2] R.A. Jackey, "A Simple, Effective Lead-Acid Battery Modeling Process for Electrical System Component Selection", The MathWorks, 2007.
- [3] M. Ceraolo, "New Dynamical Models of Lead-Acid Batteries", IEEE Trans. on Power Systems, Vol. 15, No. 4, Nov. 2000, pp.1184-1190.
- [4] L.W. Yao, J.A. Aziz, P.Y. Kong, N.R.N. Idris, "Modeling of Lithium-Ion Battery Using MATLAB /Simulink", IECON 2013, Nov.2013, Vienna.
- [5] Y. Diab, F. Auger, E. Schaeffer, M. Wahbeh, "Estimating Lithium-Ion Battery State of Charge and Parameters Using a Continuous-Discrete Extended Kalman Filter", Energies, 2017, 10, 1075; doi:10.3390/en10081075
- [6] L.H. Seim, "Modeling, Control and Experimental Testing of a Supercapacitor/Battery Hybrid System - Passive and

Semi-Active Topologies", Norwegian Institute of Life Sciences, M.Sc. Thesis, 2011.



Slika 7. Ekvivalentno kolo superkondenzatora [6]



Slika 8. a) Izgled struje prilikom simulacije b) Izgled napona prilikom punjenja i pražnjenja strujom od 40 A.

Kratka biografija:



Mirko Macanović rođen je u Novom Sadu 1994. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Energetska elektronika i električne mašine odbranio je 2018.god. kontakt: mirkomacanovic@yahoo.com



Octavian Cornea završio je osnovne, master i doktorske studije na Politehničkom Univerzitetu u Temišvaru 1996., 1997. i 2007. god., respektivno. Bavi se obnovljivim izvorima električne energije i mikro mrežama.



Vladimir A. Katić, red.prof. rođen je 1954. god. u Novom Sadu. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu 1978. god., a magistrirao i doktorirao na Elektrotehničkom fakultetu Univerziteta u Beogradu 1981. i 1991. god., respektivno. Oblasni interesovanja su energetska elektronika, obnovljivi izvori i kvalitet električne energije, i električna vozila.

**ИСПИТИВАЊЕ ПАРАМЕТАРА КВАЛИТЕТА ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ НА
НАФТНОМ ПОЉУ КОМПАНИЈЕ „НИС а.д. Нови сад“****POWER QUALITY PARAMETERS INDICES ON OIL FIELD OF "NIS a.d Novi Sad"
COMPANY**Марко Роган, Владимир Катић, *Факултет техничких наука, Нови Сад***Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

Кратак садржај – У овом раду је описано мерење параметара квалитета електричне енергије на нафтном пољу компаније „НИС а.д. Нови Сад“. Дефинисани су основни параметри квалитета електричне енергије, затим је описана коришћена опрема и анализирани су резултати мерења.

Кључне речи: *Квалитет електричне енергије, нафтно поље, дубинска клипна пумпа*

Abstract – *This paper describes the measurement of the power quality parameters on oil field of "NIS a.d. Novi Sad" company. The basic parameters of the power quality are defined, then it is described used equipment and measurement results are analyzed.*

Keywords: *Power quality, oil field, pumpjack*

1. УВОД

Проблем техничког квалитета електричне енергије је било која појава која се огледа у нарушавању основних параметара напона у устаљеним или прелазним стањима, нарушавању таласних облика напона и струја, њиховом фазном померају као и девијацији учестаности. У идеалном случају напон електричне мреже би требао да буде синусног таласног облика, номиналне фреквенције, номиналне ефективне вредности, док је идеална струја такође синусоидална, без хармонијске дисторзије и у фази са напонам. У реалности је међутим квалитет испоручене електричне енергије нарушен разним неправилностима, као што су: пренапони, поднапони, краткотрајни нестанци напона, пикови напона, фликери, виши хармоници и др. Доминантни утицај на физички квалитет напона имају нелинеарни потрошачи (уређаји енергетске електронике, електричне машине, рачунарска опрема, електролучне пећи и остали), и транзијентне појаве услед комутација у систему код прекидача, аутоматско поновно укључење, укључење/искључење великих потрошача, итд. [1].

Циљ овог рада је да прикаже стање квалитета на електромоторном погону нафтних пумпи и анализира резултате мерења на једном нафтном пољу.

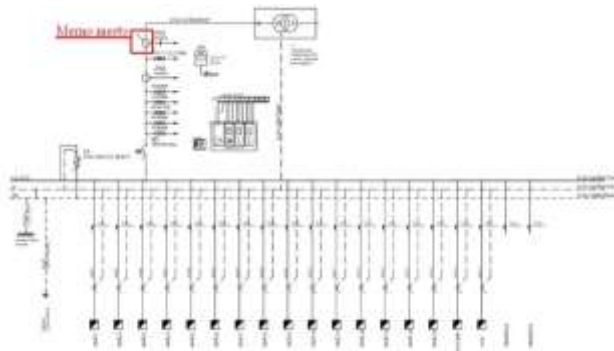
НАПОМЕНА:

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

**2. ОПИС ПОСМАТРАНОГ ПОГОНА И
ДУБИНСКЕ КЛИПНЕ ПУМПЕ НА НАФТНОМ
ПОЉУ КОМПАНИЈЕ „НИС а.д. Нови Сад“****2.1. Опис погона**

Посматрани погон, гледано са технолошке стране представља систем за прикупљање нафте са нафтних поља, затим припрему и даљи транспорт нафте.

Погон чини велики број различитих врста машина, са присутношћу управљачке електронике. Погон ради непрестано, али постоји периодика рада технолошких процеса, односно периодика рада машина, што смањује укупну потрошњу погона. На слици 1. је приказана једнополна шема НН постројења трансформаторске станице посматраног погона.



Слика 1. Једнополна шема НН постројења трафо станице посматраног погона

Потрошачи се напајају са трансформатора преносног односа 20/0,4 kV/kV и снаге 1000 kVA, трансформатор је такође у власништву НИС-а. Инсталисана снага свих потрошача у погону 960,34 kW. Неки асинхрони мотори садрже фреквентне регулаторе за контролу брзине, који су због свог нелинеарног рада представљају велики извори струјних хармоника.

2.2. Опис дубинске клипне пумпе

Основни принцип деловања дубинске пумпе темељи се на преносу погонске енергије са површине до нивоа урањања дубинске пумпе механичким начином, тј. клипним шипкама [2].

На слици 2 приказана дубинска клипна пумпа на којој су испитивани параметри квалитета електричне енергије.



Слика 2. Дубинска клипна пумпа (качаљка) на којој су мерени параметри КЕЕ

Неки основни подаци дубинске клипне пумпе су:

- Тип качаљке: UP 90, фабрички број:89290
- Редуктор: R35, фабрички број:28474
- Асинхрони мотор: 11 kW, фабрички број:093219
- Заптивна глава: Тип 2 9/15 5000
- Тубинг: Туб-16330-000-034
- Клипна шипка: Клип-16351-000-034

У саставу овог система дубинског пумпања налази се и индуктивни грејач снаге 6 kW, који има задатак да загреје тзв. „тешку“ нафту која се експлоатише. На пољима где се експлоатише „лака“ нафта, нису потребни грејачи.

3. ИСПИТИВАЊЕ КВАЛИТЕТА ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ ПОГОНА И ДУБИНСКЕ КЛИПНЕ ПУМПЕ

3.1. Избор мерног инструмента

У случају погона, циљ мониторинга је био окарактерисање рада система, а у случају дубинске клипне пумпе циљ мониторинга је био део надзора

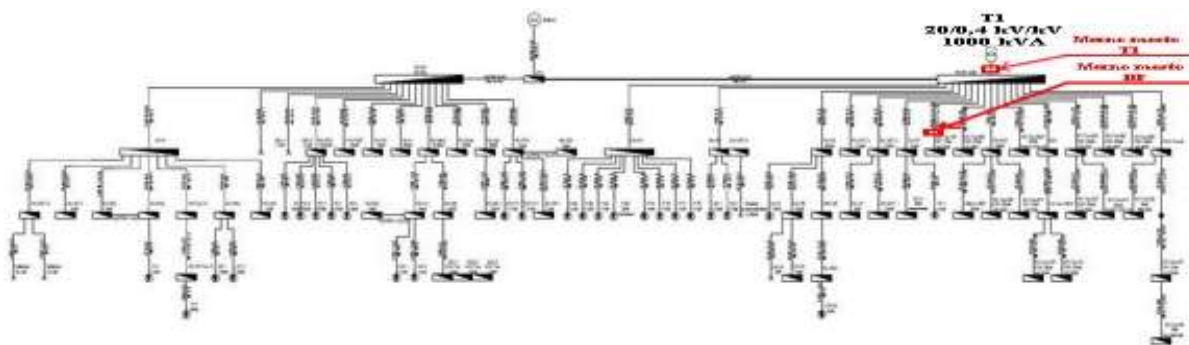
постројења на лицу места. Мерења су вршена инструментом за анализу квалитета електричне енергије типа C.A 8335 француског произвођача CHAUVIN ARNOUX, приказаног на слици 3.



Слика 3. Мерни инструмент C.A. 8335 CHAUVIN ARNOUX са опремом [3]

3.2. Постављање инструмента и поступак мерења

На слици 4 приказана је блок шема напојног трансформатора посматраног погона. Трансформатор је номиналног преносног односа 20 kV/0,4 kV, номиналне снаге 1000 kVA. Спрега трансформатора је DYn5 и напаја сабирнице 3x100x10mm² Cu са којих се напајају остали потрошачи. Након трансформаторске станице, са главних сабирница се „скида“ напон за мерења за потребе погона, а мерење параметара дубинске клипне пумпе врши се на кабловском изводу PP00 4x150mm² Al, у НН постројењу.



Слика 4. Блок шема напајања погона

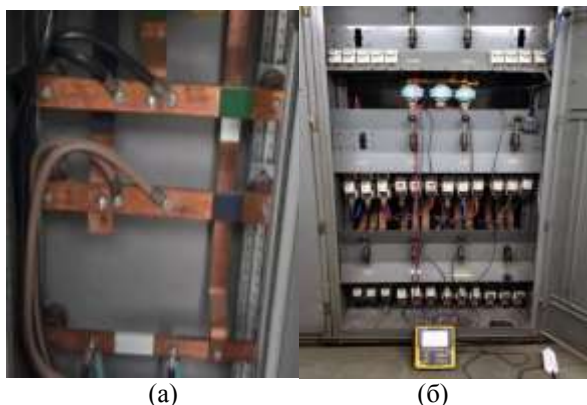
Изабрано је трофазно мерење без нултог вода, а за мерење струје изабране су AmpFlex струјне сонде (слика 5) које имају мерни опсег до 6500A, и погодне су за постављање на било који тип проводника (каблови, сабирнице и др.).



Слика 5. Одабране AmpFlex струјне сонде (до 6500A) [3]

На мерном месту за мерење параметара погона (слика 6), напонске сонде су приказане у НН постројењу на сабирнице одмах поред струјних мерних трансформатора преносног односа 1600A/5A, а струјним сондама су се обувале саме сабирнице. Мерни уређај је прикључен на монофазно АС напајање преко оближње утичнице.

На мерном месту за мерење параметара дубинске клипне пумпе (слика 7), напонске сонде су приказане такође у НН постројењу и то испод дришера (кућишта осигурача), тачније на спој кабловских папучица, а струјним сондама су обухваћени каблови.



Слика 6. Сабирнице (а) и прикључење сонди на сабирнице (б)



Слика 7. Прикључење сонди на кабловски извод за дубинску клипну пумпу

4. ПОРЕЂЕЊЕ РЕЗУЛТАТА МЕРЕЊА СА СТАНДАРДИМА И ПРЕПОРУКАМА ЗА КВАЛИТЕТ ЕЛЕКТРИЧНЕ ЕНЕРГИЈЕ

4.1. Поређење резултата мерења посматраног погона са стандардима и препорукама за КЕЕ

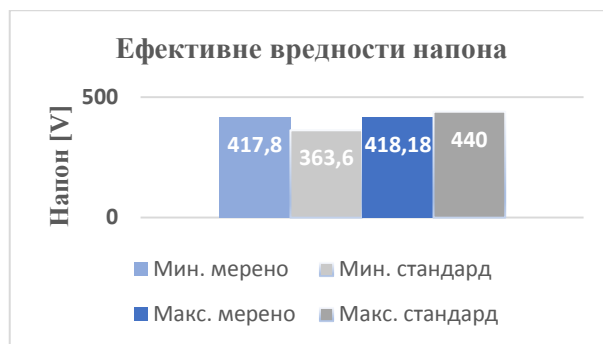
Мрежна фреквенција током периода мерења има веома мала одступања од номиналне вредности 50 Hz и задовољава опсег 49,5 Hz до 50,5 Hz, дефинисан стандардом EN 50160 (слика 8).



Слика 8. Граничне вредности мрежне фреквенције

Са аспекта квалитета напона, ефективне вредности међуфазног напона задовољавају граничне вредности $\pm 10\% U_n$, дефинисане стандардом EN 50160 (слика 9).

Укупно хармонијско изобличење напона (THDU) износи 10,82% и премашује дозвољени опсег $< 8\%$, дефинисан стандардом EN 50160 (слика 10).



Слика 9. Граничне вредности напона



Слика 10. Граничне вредности укупног хармонијског изобличења напона THDU

Крест фактор напона износи 1.49 г.ј. и излази из оквира дозвољене вредности за синусоидалне таласе од 1,414 (слика 11), што значи да у систему постоје пикови напона. Ова појава има негативне ефекте по изолацију, односно смањује век трајања диелектрика.



Слика 11. Граничне вредности крест фактора напона

Фликер (треперење) напона износи 0,27% и неће негативно утицати на систем, занемарљив је у односу на дозвољену граничну вредност < 1 дефинисану стандардом EN 50160 (слика 12).



Слика 12. Граничне вредности фликера напона

Са аспекта квалитета струје на главној сабирници трансформатора, постоји много проблема и сви индикатори квалитета електричне енергије указују на то да квалитет струје није задовољавајући, вероватно из разлога јер су у погону доминантни нелинеарни потрошачи. Укупно хармонијско изобличење струје (THDI) је изузетно велико 30,08% и ни у једном тренутку не задовољава граничну вредност < 8% дефинисану стандардом EN 50160 (слика 13).



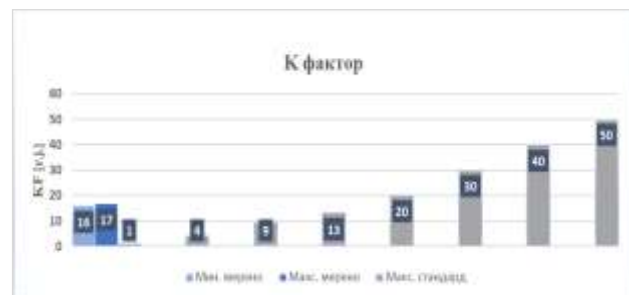
Слика 13. Граничне вредности укупног хармонијског изобличења THDI

Крест фактор струје износи 1,52% и излази из оквира дозвољене вредности за синусоидалне таласе од 1,414 (слика 14), што може представљати потенцијални проблем.



Слика 14. Граничне вредности крест фактора струје

На слици 15 приказане су вредности К фактора и његове граничне вредности за различите типове трансформатора. Трансформатор би требао да је са К фактором 20, с обзиром на то да је К фактор приближно 17. Ако трансформатор није добро димензионисан (ако је изабран са мањим К фактором од 20), потребно је смањити оптерећење трансформатора да би се избегла прегревања, а касније и евентуални кварови.



Слика 15. Граничне вредности К фактора

5. ЗАКЉУЧАК

Већина мерених вредности посматраног погона задовољавају критеријуме квалитета електричне енергије, осим укупног хармонијског изобличења напона и струје (THDU, THDI) и крест фактора напона и струје.

Такође, већина мерених вредности за дубинску клипну пумпу задовољавају критеријуме КЕЕ, осим укупног хармонијског изобличења струје (THDI), појединачних хармоника напона (5. хармоник) и струје (3., 5., 7.) и крест фактора напона и струје.

Како је хармонијско изобличење струје поприлично велико и неупада у оквиру предвиђенестандардима, требало би се извршити детаљна анализа потрошача и утврдити који то потрошачи негативно утичу на квалитет електричне енергије. Када се то утврди, требало би се тај проблем решити употребом филтара како пасивних, тако и активних или заменом потрошача енергетски ефикаснијим са мањим утицајем на квалитет, наравно ако постоји финансијска могућност и потреба за тим. Крест фактор напона и струје излази из дозвољених граничних вредности, али ситуација није заборављајућа.

6. ЛИТЕРАТУРА

- [1] В. Синић, Ж.В. Деспотовић, "Утицај нелинеарних потрошача мале снаге на квалитет електричне енергије", XXIII Симпозијум Енергетика 2017, Златибор.
- [2] ***, „Експлоатација дубинским пумпама Механички систем“ – www.rgf.rs
- [3] www.chauvin-arnoux.com/

Кратка биографија:



Марко Роган рођен је у Требињу 1991. год. Мастер радна Факултету техничких наука из области Електротехнике и рачунарства – Енергетска електроника и електричне машине одбранио је 2018.год.



др Владимир А. Катић, ред.проф. рођен је 1954. године у Новом Саду. Дипломирао је на Факултету техничких наука у Новом Саду 1978. год., а магистрирао и докторирао на Електротехничком факултету Универзитета у Београду 1981. и 1991. године, респективно. Од 2002. године је редовни професор Факултету техничких наука Универзитета у Новом Саду. Области интересовања су енергетска електроника, обновљиви извори електричне енергије, квалитет електричне енергије и електрична возила.

РЕКОНСТРУКЦИЈА ТРАНСФОРМАТОРСКЕ СТАНИЦЕ „МЕТАЉКА“**RECONSTRUCTION OF THE SUBSTATION „METALJKA“**

Лазар Симић, Зоран Стојановић, Факултет техничких Наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА

Кратак садржај – У овом раду биће приказана једна типична трансформаторска станица (ТС) 35/10 kV, старије производње, која је до скоро била у погону. Такође у овом раду ће бити описана реконструкција ТС, која подразумева замену комплетне опреме новијом, модернијом и поузданијом опремом. Прорачуни су урађени у складу са методологијом из Стандарда SRPS IEC 60909.

Кључне речи: Трансформаторска станица, ћелија, опрема, прорачун, струја квара, заштита, потрошачи.

Abstract - In this paper, one typical substation of 35/10 kV, older production will be shown, which was in operation until recently. Also in this paper will be described the reconstruction of substation, which implies the replacement of complete equipment with newer, more modern and reliable equipment. The calculations were made in accordance with the methodology of SRPS IEC 60909.

Keywords: Substation, switchgear, equipment, calculation, fault current, protection, consumers.

1. УВОД

Трансформаторска станица је лоцирана 2 km од места Митровац, на локацији Крња Јела, која се налази на планини Тари, недалеко од пута Митровац-Брана Лазићи, на надморској висини од 1100 m. Код одређивања макролокације, најповољније место било је у близини раскршћа путева Митровац – Лазићи – Крушчица.

Трансформаторска станица 35/10 kV „Метаљка“ (у даљем тексту ТС) служи за снабдевање потрошача на спољним објектима РХЕ Бајина Башта, који су лоцирани на планини Тара и то:

- Пумпног постројења Ђурићи 35 kV;
- ТС Брана Лазићи 10 kV;
- Фабрике пијаће воде и ТС Крушчица 10 kV;
- Потрошача Горње улазно – излазне грађевине ГУИГ;

као и за напајање дистрибутивних потрошача на планини Тари:

- Дечије одмаралиште на Митровцу 10 kV;
- Митровац 10 kV;
- Далековод за село Растешиште 10 kV;
- Далековод Крушчица за село Вежања 10 kV.

2. СТАРО ПОСТРОЈЕЊЕ ТС „МЕТАЉКА“

Систем напајања ТС чине постројење 35 kV, два уљна трансформатора 35/10 kV, од којих је један у погону, док је други у хладној резерви, постројења 35 и 10 kV и система сопствене потрошње 0,4 kV, који се напаја преко трансформатора 10/0,4 kV.

Ћелије 35 kV имају следећу намену:

- Ћелија бр. 1 – трансформатор I;
- Ћелија бр. 2 – ДВ РХЕ (део са растављачем);
- Ћелија бр. 3 – ДВ РХЕ (део са кабловском главом);
- Ћелија бр. 4 – трансформатор II;
- Ћелија бр. 5 – ДВ Ђурићи (део са прекидачем);
- Ћелија бр. 6 – ДВ Ђурићи (део са кабловском главом).

Целокупно постројење 35 kV је ваздушно металом оклопљено изведено у згради са апаратима и сигурносним размацима који одговарају прописима за унутрашњу монтажу. Опрема је смештена у отвореним префабрикованим ћелијама у специјалном извођењу фирме „Минел“. Свака ћелија је од суседне потпуно одвојена, а прелаз сабирница из ћелије у ћелију се врши кроз проводне изолаторе. Пошто је трансформаторска станица „Метаљка“ без посаде (непоседнута трансформаторска станица), прекидачи имају, поред ручног и електромоторни погон. Прекидачи су малоуљни, $I_n=630$ А. Погон растављача је ручни, јер се њима само изузетно манипулише, што се по налогу уклопничара из команде РХЕ препушта да обави чувар или друго овлашћено лице. Ћелије 35 kV су зидне, са предње стране преграђене жичаним вратима и са орманима за смештај опреме [1].

ТС се напаја из поља Е9 са сабирница I (прве) секције сопствене потрошње РХЕ Б. Башта 35 kV. Из поља Е9, подземни кабал NPZO 3x70 mm², дужине L=9180 m полази успонском трасом (NPZO-23A, L=2400 m на првом стрмљем делу трасе, затим NPZOP-13A, L=7350 m). Кабл у ТС долази у ваздушну (отворену-стари назив) доводну ћелију поља бр. 3 једноструког система сабирница 35 kV. Ћелија 35 kV бр.3 садржи уљну кабловску главу доводног кабла. Из доводне ћелије бр.3 сабирнице 35 kV воде у доводну ћелију бр. 2.

НАПОМЕНА:

Овај рад је проистекао из мастер рада чији ментор је био др. Зоран Стојановић, ванр. проф.

Трансформатори су смештени напољу уз сами зид трафостанице. Трансформатор бр.1 је номиналне снаге 1600 kVA, а трансформатор бр. 2 је номиналне снаге 2500 kVA. Трансформатори су трофазни, уљни са природним хлађењем (ONAN), за спољну монтажу, преносног односа 35/10 kV, спреге Yy0. Трансформатори су постављени у својим зиданим кадама на темељним стопама са Че-шинама. Сабирна уљна јама је заједничка. На 35 kV страни трансформатори су голим бакарним сабирницама спојени са одговарајућом 35 kV хелијом (пролаз Си – сабирнице кроз отвор у зиду објекта кроз шупљи изолатор, улаз у хелију одозго), а на 10 kV страни каблом. Голе бакарне сабирнице су на слободном простору заштићене мрежом. Трансформатори могу напајати оба система сабирница 10 kV што је омогућено системом растављача у 10 kV хелијама.

Намена хелија 10 kV:

- Хелија бр. 1 – ДВ Митровац;
- Хелија бр. 2 – ДВ Дечије одмаралиште;
- Хелија бр. 3 – ДВ Лазићи;
- Хелија бр. 4 – спојна хелија;
- Хелија бр. 5 – трансформатор I;
- Хелија бр. 6 – трансформатор II;
- Хелија бр. 7 – ДВ Улазна грађевина;
- Хелија бр. 8 – ДВ Раститште;
- Хелија бр. 9 – мерење;
- Хелија бр. 10 – ДВ Крушчица филтерско постројење;
- Хелија бр. 11 – ДВ Крушчица II;
- Хелија бр. 12 – сопствена потрошња.

Ово постројење је монтирано у истој просторији у којој је разводно постројење 35 kV. Целокупно постројење 10 kV је састављено из оклопљених слободностојећих хелија са двоструким сабирницама. Хелије су изоловане једна од друге, а унутар хелија су постављене сигурносне преграде. Хелије 10 kV су стандардни производ фабрике „Минел“.

Систем напајања 10 kV се састоји од два система сабирница 10 kV са могућношћу њиховог преспајања (растављачко – спојна хелија бр.4). Доводна поља 10 kV су хелија бр. 5 – довод трансформатор бр.2 и хелија бр. 6 - довод трансформатор бр. 1.

Изводна поља 10 kV су подељена на изводе за напајање спољних објеката РХЕ „Б.Башта“, изводе за напајање дистрибутивних потрошача и извод за напајање трансформатора 10/0,4 kV сопствене потрошње. Поље бр. 9 је мерно поље. Сваки довод и извод 10 kV има могућност прикључења и на I систем и на II систем сабирница 10 kV. Сви прекидачи 10 kV су постављени на колицима, која се могу по потреби извући изван хелије. Прекидачи су малоуљни, $I_n=630$ А.

Постројење 10 kV је ваздушно (отворено – стари назив) металом оклопљено.

Сопствена потрошња:

Као извор наизменичног напона постоји трансформација са 10 kV на 0,4 kV (хелија бр.12). Трансформатор је снаге 50 kVA, спреге Yzn5.

Главни потрошачи на 0,4 kV, 50 Hz су:

- Осветљење;
- Грејање;
- Исправљач за пуњење акумулаторских батерија 110 V jcc;
- Претварачи напона 110 V jcc у 48 V jcc (DC/DC конвертори).

Максимално једноремено оптерећење укључујући и грејање износи 46,79 kW.

Исправљач за пуњење батерија 110 V jcc има снагу од око 5,5 kW.

Систем заштите од превисоког напона додиром је TN-C-S.

Развод једносмерног напона 110 V jcc:

Извор једносмерног напона 110 V jcc је исправљач снаге 5.5 kW у паралелном споју са оловном стационарном акумулаторском батеријом. Напон 110 V jcc се користи за напајање постројења 35 и 10 kV и остале опреме као што је:

- командно – сигнална опрема;
- напајање нужног светла;
- напајање моторних погона прекидача;
- напајање уређаја телемеханике;
- општу сигнализацију.

Електричне заштите:

Електричне заштите су електромеханичке које се напајају са секундарних струјних и напонских мерних трансформатора за заштиту и мерење. Основни скуп електричних заштита у хелијама 35 kV и 10 kV су: прекострујна и земљоспојна док су трансформатори опремљени Бухолц релеом и контактним термометром (термичка заштита). Сопствена потрошња нема посебну релејну заштиту осим својих прекидача.

3. РАЗЛОЗИ РЕКОНСТРУКЦИЈЕ ТС

ТС је пуштена у рад 1974. године, што значи да је опрема у великој мери потрошила свој животни век. Поред тога, иако је то било предвиђено пројектом, алтернативни извор напајања никада није реализован, тако да је једини извор напајања кабл из РП 35 kV РХЕ. Присутан је и проблем обезбеђења резервних делова, а дошло је и до повећање потрошње дистрибутивних потрошача [1].

Сви горе наведени разлози намећу потребу реконструкције ТС. Основни технички захтеви реконструкције ТС су:

- Замена постојећих енергетских трансформатора новим, снаге 4000 kVA и провера напојног кабла из РХЕ;
- Обезбеђење додатног напајања ТС 35/10 kV из још једног извора напајања, независног од РХЕ „Бајина Башта“;
- Реконструкција РП 35 kV, која подразумева комплетну замену опреме у постојећим хелијама 35 kV и доградњу још једне доводне хелије за алтернативни извор напајања;
- Реконструкција РП 10 kV, која подразумева уградњу потпуно новог РП 10 kV са једноструким сабирницама;

- Реконструкција развода сопствене потрошње ТС која подразумева замену постојећег пираленског трансформатора са новим сувог типа;
- Замена постојећег исправљача и акубаторија, са новом опремом;
- Ново РП 35 kV и 10 kV, нови развод сопствене потрошње ТС и нови исправљачи морају бити опремљени системом микропроцесорске заштите и управљања;
- Пројектом предвидети замену свих каблова 10 kV од ниженапонске стране трансформатора 35/10 kV. Нови каблови треба да су типа ХНЕ – 49А, 10 kV, пресека који треба одредити прорачуном.

4. ПРОРАЧУН СТРУЈЕ КРАТКОГ СПОЈА

У циљу провере опреме на механичко и термичко напрезање у случају кратког споја извршени су одговарајући прорачуни струја кратког споја меродавних за димензионисање електро опреме у ТС која се напаја из РХЕ „Бајина Башта“. Прорачуни су урађени у складу са методологијом из Стандарда SRPS IEC 60909. Као карактеристична места квара, меродавна за избор нове опреме у ТС изабрана су најкритичнија места квара на 35 kV и 10 kV напонском нивоу [2].

Прорачуни струја кратког споја су извршени на следећи начин:

-На основу параметара постојеће опреме (стање пре ревитализације ХЕ „Бајина Башта“) израчуната је компонента струје квара на сабирницама 220 kV у РП Зауглине која потиче из ХЕ/РХЕ „Бајина Башта“, те је на основу ове вредности и података ЕМС-а о вредности укупне струје квара на сабирницама 220 kV у РП Зауглине одређена и компонента струје квара из мреже:

$$I_m = 16,34 \text{ kA} \quad Z_m = (0,85 + j8,51) \Omega$$

-Уз усвојену претпоставку да компонента струје квара из мреже остаје иста и после ревитализације ХЕ, те на основу усвојених параметара опреме ревитализоване ХЕ, одређене су и вредности компонената струја квара из ХЕ/РХЕ „Бајина Башта“, односно добијени су потребни подаци за прорачун струја квара на следећим местима:

-35 kV сабирнице у ХЕ „Бајина Башта“:

- Субтранзијентна струја $I''_{F1} = 4,53 \text{ kA}$;
- Ударна струја $I_{ud} = 12,01 \text{ kA}$;
- Термичка, једносекундна струја $I_{th} = 4,75 \text{ kA}$;

-35 kV сабирнице у ТС „Метаљка“:

- Субтранзијентна струја $I''_{F2} = 3,32 \text{ kA}$;
- Ударна струја $I_{ud} = 6,06 \text{ kA}$;
- Термичка, једносекундна струја $I_{th} = 3,35 \text{ kA}$;

-10 kV сабирнице у ТС „Метаљка“:

- Субтранзијентна струја $I''_{F3} = 3,12 \text{ kA}$;
- Ударна струја $I_{ud} = 6,79 \text{ kA}$;
- Термичка, једносекундна струја $I_{th} = 3,18 \text{ kA}$.

5. НОВО ПОСТРОЈЕЊЕ ТС „МЕТАЉКА“

За развод електричне енергије на напону 35 kV предвиђено је типско металом оклопљено разводно постројење за унутрашњу монтажу са једним системом сабирница подужно секционисаним, састављено од осам ћелија (Слика 1). Сабирнице су од изолованог електролитичког бакра. Разводно постројење 35 kV смештено је заједно са разводним постројењем 10 kV у просторију разводних постројења која се налази у склопу трафостанице [3].

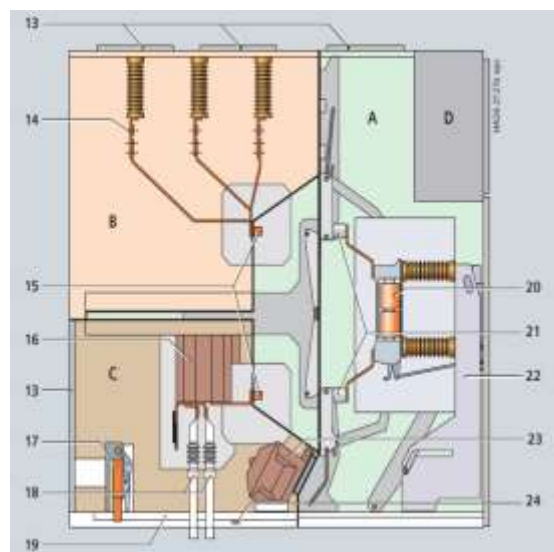
Називни напон сабирница је 36 kV, номинална струја 1250 А, димензије сабирница 80x10. Прекидачи су вакумски, номиналне струје 1250 А [4].

Постројење се састоји од следећих ћелија:

- Н01 - доводна ћелија из разводног постројења 35 kV РХЕ „Бајина Башта“;
- Н02 - ћелија трансформатора - ТМ1 35/10,5 kV, 4 MVA;
- Н03 - изводна ћелија за трансформаторску станицу 35/10 kV „Ђурићи“;
- Н04 - резервна ћелија;
- Н05 - спојна ћелија;
- Н06 - додатак спојне;
- Н07 - ћелија трансформатора – ТМ2 35/10,5 kV, 4 MVA;
- Н08- доводна ћелија из дистрибутивне мреже 35 kV „Бајина Башта“.

Предвиђена су два трофазна двонамотајна уљна трансформатора преносног односа $(35 \pm 2 \times 2,5\%) / 10 \text{ kV}$, снаге 4000 kVA сваки. Трансформатори су предвиђени за спољашњу монтажу и смештени су у близини зграде ТС.

На напонском нивоу 35 kV, веза трансформатора са постројењем 35 kV преко 35 kV кабла, типа ХНЕ 49 3x(1x70 mm²). На нижем напону веза трансформатора са разводним постројењем 10 kV остварена је кабловима ХНЕ 49 3x(1x70 mm²).

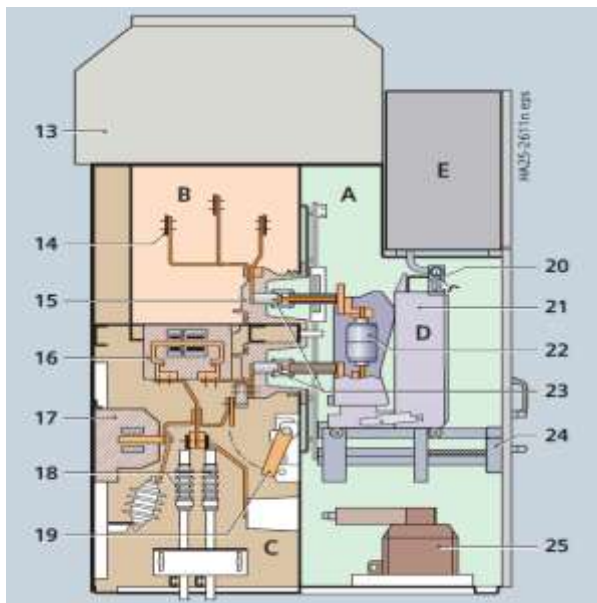


Слика 1. Попречни пресек ћелије 35 kV [4]

За развод електричне енергије на напону 10 kV предвиђено је типско металом оклопљено разводно постројење за унутрашњу монтажу са једним системом сабирница подужно секционисаним, састављено од седамнаест ћелија (Слика 2).

Врши се измена једнополне шеме разводног постројења 10 kV у смислу да се раздвоје потрошачи РХЕ „Бајина Башта“ и дистрибутивне мреже, да би се напајали свако са својег трансформатора, односно секције, са могућношћу спајања преко спојног поља.

Називни напон сабирница је 12 kV, номинална струја 1250 А, димензије сабирница 80x10. Прекидачи су вакумски, номиналне струје 800 А [5].



Слика 2. Попречни пресек ћелије 10 kV [5]

За напајање електричном енергијом нисконапонских потрошача предвиђен је један двонамотајни трофазни суви трансформатор преносног односа $(10 \pm 2 \times 2,5 \%) / 0,4 \text{ kV}$, снаге 100 kVA. Трансформатор је предвиђен за унутрашњу монтажу у ћелији =K01 извода за трансформатор ТМЗ 10/0,4 kV сопствене потрошње ТС.

Хлађење трансформатора остварује се природном циркулацијом ваздуха.

Веза трансформатора са постројењем 10 kV остварена је једножилним каблом пресека 50 mm^2 , а са постројењем 0,4 kV остварена је петожилним каблом РР00 1 kV $5 \times 4 \text{ mm}^2$.

6. ЗАКЉУЧАК

Како је ТС пуштена у рад 1974. године, њена опрема је у великој мери потрошила свој животни век. Поред тог проблема, присутан је и проблем обезбеђења резервних делова, као што је дошло и до повећање потрошње дистрибутивних потрошача на Митровцу. Комплетна замена опреме је урађена у складу са прорачунима из поменутог стандарда. Детаљан прорачун је приказан у мастер раду.

7. ЛИТЕРАТУРА

- [1] Техничка документација *ХЕББ*.
- [2]<https://support.industry.siemens.com/cs/product/3ah5322-2.....-vacuum-cb-36kv-pcd-275mm-16ka-1250a?pid=78604&mlfb=3AH5322-2...&mfn=ps&lc=en-DK> (приступљено у септембру 2018.)
- [3]<https://support.industry.siemens.com/cs/products/3ae5554-1.....-sion-3ae5-12kv-25ka-800a?pid=511733&mlfb=3AE5554-1....&mfn=ps&lc=en-SV> (приступљено у септембру 2018.)
- [4]https://w3.siemens.com/powerdistribution/global/SiteCollectionDocuments/en/mv/switchgear/air-insulated/8bt2/catalogue-8bt2_en.pdf (приступљено у септембру 2018.)
- [5]https://w3.siemens.com/powerdistribution/global/SiteCollectionDocuments/en/mv/switchgear/air-insulated/nxair/catalogue-nxair-family_en.pdf (приступљено у септембру 2018.)

Кратка биографија



Лазар Симић рођен је у Бајиној Башти 1990. год. Мастер рад на Факултету техниких наука из области Електротехнике и рачунарства – Електроенергетика – Енергетска електроника и електричне машине одбранио је 2018. год.

**PRIMENA AKTER PROGRAMSKOG MODELA NA MIKROSERVISNU
ARHITEKTURU SERVISA ZA SKLADIŠTENJE PODATAKA U MEMORIJI**
**APPLICATION OF ACTOR PROGRAMMING MODEL ON MICROSERVICE
ARCHITECTURE FOR IN-MEMORY DATA STORAGE SERVICE**

Marko Jelača, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu analizirano je skladištenje podataka u memoriji primenom akter programskog modela. Predložena je mikroservisna arhitektura servisa za skladištenje podataka u memoriji upotrebom Microsoft Azure Service Fabric platforme. Predloženo rešenje se sastoji od menadžer aktera koji upravljaju velikim brojem aktera koji skladište podatke u svojoj memoriji. Efikasna raspodela aktera po particijama je realizovana upotrebom algoritma za particionisanje brojeva. Dostupnost i pouzdanost su obezbeđeni replikacijom i trajnim čuvanjem podataka u bazi podataka. Performanse rešenja su merene na lokalnom klasteru promenom broja aktera.

Ključne reči: Skladištenje podataka u memoriji, akter programski model, mikroservisna arhitektura.

Abstract – This paper analyzes data storing in memory using actor programming model. Paper proposes a microservice architecture for in-memory data storage service using Microsoft Azure Service Fabric platform. The proposed solution consist of manager actors who coordinates with large number of actors for data storing. Evenly distribution of actors across partition is done using number partitioning algorithm. Availability and reliability are provided with replication and permanent data storage in the database. Performance of the proposed solution was measured on local cluster, changing the number of actors.

Keywords: Storing in memory, akter programming model, microservice architecture

1. UVOD

Skladištenje podataka u memoriji je jedan od osnovnih konceptata u računarstvu. Osnovni princip jeste da skladištenje podataka u memoriji ubrzava operacije. Kada želimo da dobavimo neki podatak, čije dobavljanje je relativno skupa operacija po pitanju vremena i resursa, mi taj podatak skladištimo u memoriji kako bi sledeće dobavljanje tog podatka bilo jeftinije. Neki od glavnih razloga zašto je skladištenje podataka u velikim sistemima veoma važno jeste to što je skladištenje podataka najvažniji deo softvera koji utiče na to koliko brzo aplikacija reaguje na zahteve, kao i činjenica da gubitak podataka nije prihvatljiv u poslovnim operacijama.

NAPOMENA:

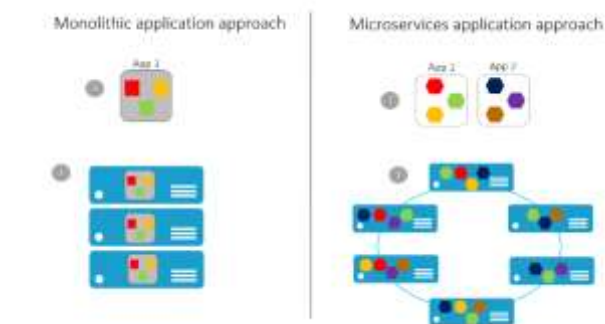
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Srđan Vukmirović, vanr.prof.

Računarstvo u oblaku (eng. *cloud computing*) predstavlja model koji omogućava pristup na zahtev deljenim računarskim resursima (kao što su računarska mreža, serveri, skladište podataka i servisi) [1]. Neke od karakteristika ovog modela jesu korišćenje neograničenih resursa, smanjenje infrastrukturnih rizika kupovinom novih resursa, skalabilnost i mogućnost da platimo samo one resurse koje zaista trošimo.

2. MIKROSERVISNA ARHITEKTURA

Mikroservisni arhitektonski stil u razvoju softvera podrazumeva razvoj softvera kao skup malih servisa, pri čemu se svaki servis izvršava na svom procesu i komuniciraju koristeći protokole kao što su *HTTP/HTTPS*, *Web sockets* ili *Advanced Message Queuing Protocol* (skr. *AMQP*) [2]. Mikroservisi [3] se definišu kao mali autonomni servisi koji rade zajedno. Ovaj arhitektonski stil veoma brzo preovlađuje u industriji kao održiva alternativa monolitnim aplikacijama i servisno-orijentisanoj arhitekturi (eng. *service-oriented architecture*, skr. *SOA*). Prednost ovakvih servisa je što su usko povezani sa poslovnom funkcijom koju treba da izvršavaju [4].

Razlike između monolitne i mikroservisne arhitekture su prikazane na slici 1.



Slika 1. Razlike monolitne i mikroservisne arhitekture [11]

Monolitna aplikacija predstavlja funkcionalnosti specifične za domen i podeljene su uglavnom po nivoima kao što su *web*, poslovna logika, podaci (1). Skaliranje takve aplikacije radimo tako što je kloniramo na više servera / virtualnih mašina (2). Suprotno tome kod mikroservisne arhitekture svaka nezavisna poslovna logika predstavlja jedan servis (3). Skaliranje aplikacije prema mikroservis arhitektonskom stilu podrazumeva isporuku svakog servisa posebno kreiranjem instanci servisa na više servera / virtualnih mašina (4) [11].

3. AKTER PROGRAMSKI MODEL

Akter (eng. *Actor*) programski model predstavlja matematički model konkurentnog izračunavanja koji predstavlja aktere kao osnovne jedinice konkurentnog izračunavanja. Pojam je prvi put definisan 1973 u radu [5] gde je akter predstavljen kao aktivan agent, koji se ponaša u skladu sa dobijenim naznakama. Kao odgovor na poruku koju dobije, akter može da: donese logičku odluku, kreira nove aktere, pošalje poruku nekom drugom akteru ili da odluči šta će se desiti sa narednom porukom koju dobije. Akteri mogu da menjaju svoje privatno stanje, dok stanje drugog aktera mogu da menjaju samo razmenom poruka. Aplikacije srednjeg nivoa, odnosno nivoa poslovne logike, su najčešće objektno-orijentisane, jer uključuju interakciju sa velikim brojem objekata iz realnog svega. Primer su višekorisničke igre, društvene mreže, telemetrija. Kako su objektni iz realnog sveta nezavisni, tako i softverski objekti koji treba da ih modeluju ne treba da dele stanje i potrebno je da komuniciraju asinhronom razmenom poruka [6]. Takvi objekti se nazivaju akteri.

4. PRIMENA AKTER PROGRAMSKOG MODELA NA PROBLEM SKLADIŠTENJA PODATAKA

U [6] i [7] se predstavlja ideja da sloj za upravljanje podacima treba da bude apstrakcija akter modela kako bi se postigla visoka mogućnost skaliranja, konkurentnost i modularnost.

Enkapsulacijom stanja, podržavanjem *single-threaded* semantike za manipulaciju stanjem i asinhronom razmenom poruka, akter model predstavlja generalan, modularan i konkurentan model za izračunavanje. Pored toga izdržljivost i globalnu konzistentnost između aktera i upravljanje otkazima sistema nije obezbeđena od strane samog modela, već je ostavljena korisniku da je sam obezbedi. Pored ovih nedostataka, mnogi radni okviri kao što su Akka, Erlang i Orleans su često korišćeni za kreiranje sistema za skladištenje podataka u memoriji, dizajnirani tako da se skaliraju na milione aktera distribuiranih na stotine servisa korišćenjem *Cloud* infrastrukture [7]. U [6] i [7] su predstavljena načela koja opisuju upravljanje podacima primenom akter programskog modela. Pomenuta su samo neka od predstavljenih načela.

Za modularnost akter orijentisani sistem za skladištenje podataka u memoriji treba da podržava apstrakciju konkurentnog i distribuiranog logičkog aktera. Logički akter predstavlja entitet koji komunicira razmenom poruka. Logički akter se definiše kao entitet koji obrađuje zahtev prosleđen od strane klijenta ili drugog aktera. Akter model upravljanje životnim ciklusom aktera pruža samoj aplikaciji. Kod sistema kao što je sistem za skladištenje podataka u memoriji, potrebno je omogućiti različite mehanizme za upravljanje životnim ciklusom aktera, kao što su dinamičko i statičko kreiranje aktera [7].

Kako bi se obezbedio brži odziv na zahteve, asinhrona razmena poruka postaje obavezna. *Single-threaded* semantika akter modela omogućava prilagođavanje klasičnih mehanizama baza podataka kao što su npr. transakcije [7]. Transakcija obuhvata više aktera koji se nalaze uglavnom na različitim serverima i nezavisno upisuju svoje stanje u skladište podataka. Zbog toga

atomičnost i izdržljivost zahtevaju korišćenje transakcionih protokola [6].

Pozivanje metode aktera zahteva da je njegov identitet poznat. Poželjno je da je identitet aktera definisan njegovom lokacijom. Identitet može biti trajan, kao npr. primarni ključ u tabeli baze podataka ili može biti prolazan, odnosno da se prilikom aktivacije aktera kreira novi [6].

Pored toga što se akteru pristupa preko njegovog identiteta, korisno je da postoji mogućnost pristupa i preko drugih atributa [6], [7], [8]. U slučaju skladištenja istorijskih podataka u akter orijentisanu bazu u memoriji, dodatno ubrzanje prilikom pretrage može se postići indeksiranjem aktera prema datumu.

Ako bi se akteri tretirali kao objekti, objektno-orijentisan upitni jezik može biti primenjen na akter-orijentisane sisteme. Ako bi se akter-orijentisana baza predstavila kao graf, gde su čvorovi akteri, a grane reference između aktera, tada upiti mogu biti usmereni (eng. *path-oriented*), kao npr. SPARQL regularni izrazi [9].

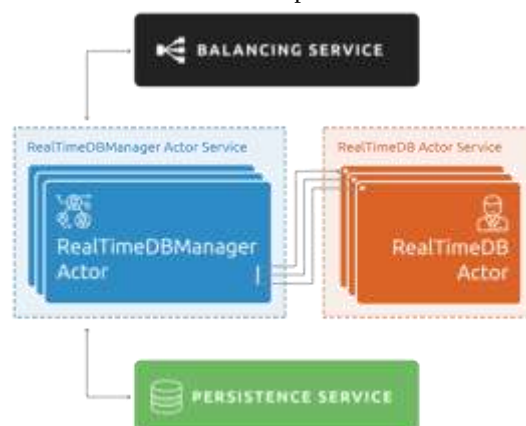
5. PREDLOG IMPLEMENTACIJE REŠENJA

5.1. Arhitektura predloženog rešenja

Osnovna ideja predloženog rešenja jeste podela podataka na određeni broj aktera. U opštem slučaju postoje menadžer (koordinator) akteri koji mogu predstavljati table u bazi podataka ili particionisanu tabelu u bazi podataka. Pored menadžer aktera, postoje i akteri koji su odgovorni za skladištenje podataka odgovarajućeg menadžer aktera.

Kao rešenje problema skladištenja istorijskih podataka, kreiran je distribuirani servis za pametno skladištenje podataka. Servis je implementiran uz oslonac na akter programski model upotrebom *Microsoft Azure Service Fabric* platforme. Sastoji se od *RealTimeDBManager* akter servisa, *RealTimeDB* akter servisa, *Balancing* servisa i *Persistence* servisa, prikazani na slici 2.

RealTimeDBManager akter servis dobija podatke odgovarajućeg tipa podataka i služi kao koordinator za skladištenje istorijskih podataka tog tipa. Akteri se identifikuju preko odgovarajućeg identifikatora. Svaki od *RealTimeDBManager* aktera ima reference na određeni broj *RealTimeDB* aktera koji skladište podatke. Pored skladištenja podataka u *RealTimeDB* akterima, *RealTimeDBManager* akter je odgovoran i za dobavljanje podataka od *RealTimeDB* aktera u određenom vremenskom periodu upotrebom asinhronne razmene poruka.



Slika 2. Arhitektura predloženog rešenja

5.2. Partitionisanje podataka

Partitionisanje predstavlja jedan od osnovnih principa prilikom kreiranja skalabilnih servisa. Partitionisanje se može posmatrati kao podela stanja i proračuna na manje jedinice kako bi poboljšali skalabilnost i performanse. Jedan od oblika partitionisanja jeste partitionisanje podataka (eng. *sharding*).

Kako je svaki akter servis, ustvari *stateful* servis, svaka particija akter servisa će sadržati određeni skup aktera. Particija *stateful* servisa se može posmatrati kao skalabilna jedinica koja je dostupna preko svojih replika koje su distribuirane i balansiranje na nodovima (virtualnim mašinama). *Stateful* servisi mogu biti kreirani sa različitim šemama za partitionisanje podataka.

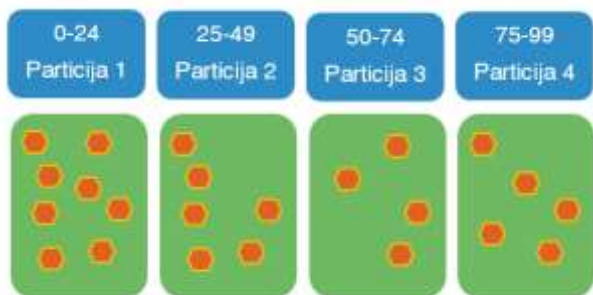
5.3. Primena algoritma za partitionisanje brojeva na problem partitionisanja aktera

Akter servisi koriste isključivo *Int64* partitionisanje sa upotrebom celog *Int64* opsega za vrednosti ključeva prilikom partitionisanja.

Kada se kreira akter, potrebno mu je dodeliti odgovarajući jedinstveni identifikator reprezentovan *ActorId* klasom. Moguće je koristiti korisnički definisane identifikatore, kao što su GUID, UUID, stringovi i *Int64* vrednosti. Kada se koristi GUID, UUID ili string, identifikator se konvertuje u *Int64* opseg. Kada se koristi eksplicitno *Int64* vrednost kao identifikator aktera, tada će se taj akter automatski mapirati na odgovarajuću particiju bez prethodnog konvertovanja. Na ovaj način se može uticati na kojoj particiji će se akter kreirati [11].

Za potrebe ravnomerne raspodele aktera na particije korišćen je algoritam za rešavanje problema partitionisanja brojeva (eng. *Number partitioning problem*, skr. NPP). Problem partitionisanja brojeva jeste da dati skup brojeva podeli na međusobno isključive podskupove tako da je suma brojeva u svakom podskupu jednaka ili približno jednaka [10]. Predlog implementiranog rešenja koristi pohlepnu heuristiku za proračun smeštanja aktera na particije.

Rad algoritma će biti objašnjen na primeru koji ilustruje slika 3. Kako bi se jednostavnije objasnio algoritam, koristiće se *Int64* šema za partitionisanje sa minimalnim ključem 0, a maksimalnim 99. Postoje 4 particije, vrednosti ključeva iz opsega su ravnomerno raspodeljene na particije 0-24, 25-49, 50-74 i 75-99.



Slika 3. Partitionisanje aktera

Kada *RealTimeDBManager* akter treba da kreira novi *RealTimeDB* akter, šalje poruku *Balancing* servisu, koji

odredi koja je particija najmanja i prosledi *RealTimeDBManager* akteru sledeću vrednost iz opsega ključeva za tu particiju. U situaciji prikazanoj na prethodnoj slici, particija 3 ima najmanji broj kreiranih aktera (id aktera na ovoj particiji su 50, 51, 52, 53), tako da će *Balancing* servis vratiti prvi naredni ključ iz opsega ključeva koji odgovara toj particiji, a to je ključ 54.

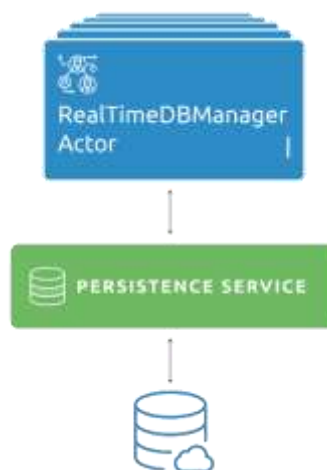
5.4. Dostupnost i pouzdanost podataka

Kritičan deo u obezbeđivanju visoke dostupnosti jeste obezbediti otpornost servisa na otkaze. Kako bi se zaštitili od mogućeg gubitka podataka potrebno je periodično kopiju podataka skladištenih u memoriji, skladištiti na nekom trajnom skladištu podataka, na primer relacionoj bazi podataka koja podatke skladišti na disk. Do gubitka podataka u memoriji može doći iz više razloga.

Otkaz pojedinih nodova na *Service Fabric* klasteru može dovesti do gubitka servisa na tom nodu, kao i gubika stanja tih servisa. U takvom slučaju poželjno je posedovati više replika servisa. Implementirano rešenje podrazumeva upotrebu *Service Fabric Reliable Services* replikacionog mehanizma [11]. Svaki od kreiranih aktera predstavlja primarnu repliku tog aktera. Pored primarne replike postoje i dve sekundarne replike od koje jedna postaje primarna u slučaju otkaza primarne replike.

Samo u slučaju inicijalizacije servisa za skladištenje podataka u memoriji potrebno je vršiti čitanje podataka iz relacione baze podataka, u suprotnom RAM memorija treba da bude primarno skladište podataka u koju se podaci unose i iz koje se čitaju.

Slika 4 prikazuje perzistenciju podataka aktera na *Azure SQL* bazu podataka. Kada dobije podatak, *RealTimeDBManager* akter šalje taj podatak na *Persistence* servis koji ima ulogu skladištenja podataka u relacionoj bazi. Pored toga što vrši skladištenje podataka u bazu podataka, *Persistence* servis vrši i inicijalizaciju podataka u memoriji prilikom startovanja servisa. Učitava podatke iz relacione baze i šalje ih na *RealTimeDBManager* akter servis, koji potom vrši raspodelu podataka na *RealTimeDB* aktere. Da bi *Persistence* servis mogao da čita podatke iz baze, mora da zna koje tipove podataka treba da učitava iz baze podataka. Skup tipova podataka se inicijalno čita iz baze podataka, a ažurira se prilikom čuvanja novog tipa podataka u bazu.



Slika 4. Perzistencija podataka

6. PERFORMANSE REŠENJA

Vršena su merenja performansi sistema. Za testni skup podataka korišćena je relacionala baza koja sadrži oko 9000 istorijskih merenja snaga po generatoru. Merenja performansi sistema su izvršena na lokalnom klasteru, računaru konfiguracije: CPU i5 2.20GHz, RAM 12GB. Klaster sadrži 5 nodova. *RealTimeDB* akter servis je particionisan na 3 particije, dok je *RealTimeDBManager* akter servis particionisan na 2 particije.

Performanse su merene za dva slučaja u kojima je broj kreiranih aktera različit. Tabela 1 i tabela 2 sadrže podatke o broju entiteta koji su dobavljeni prilikom upita, vreme trajanja izvršavanja upita na implementiranom rešenju (M) i vreme izvršavanja upita na relacionoj bazi podataka (DB). Tabela 1. predstavlja rezultate vremena dobavljanja svih podataka po generatoru u situaciji kada je maksimalan broj entiteta koje akter može da skladišti 100.

ID	1	2	3	4	5	6	Avg
Ent	9075	9300	9259	8509	8527	9300	8995
M [ms]	398	458	434	405	436	395	421
DB [ms]	1791	664	1196	1263	1325	2311	1425

Tabela 1. Vremena dobavljanja svih merenja po generatoru – 100 entiteta po akteru

Tabela 2. predstavlja rezultate vremena dobavljanja svih podataka po generatoru u situaciji kada je maksimalan broj entiteta koje akter može da skladišti 50. U ovom slučaju je ukupan broj aktera u sistemu duplo veći nego u prethodnom slučaju. Što znači da prilikom pretrage merenja po određenom kriterijumu (vremenskom opsegu) sistem treba da pretraži duplo veći broj aktera i zbog toga je vreme potrebno za izvršavanje upita veće u odnosu na prethodni slučaj.

Tabela 2. Vremena dobavljanja svih merenja po generatoru – 50 entiteta po akteru

ID	1	2	3	4	5	6	Avg
Ent	9075	9300	9259	8509	8527	9300	8979
M [ms]	851	816	986	882	976	865	905
DB [ms]	2194	2152	2032	1985	1926	2216	2062.2

Na lokalnom klasteru, rad servisa se izvršava konkurentno. Na *Cloud* klasteru izvršavanje servisa na različitim nodovima je paralelno i zbog toga bi veliki broj particija na kojima se upiti izvršavaju paralelno omogućio brže performanse. Performanse zavise od fizičkih karakteristika samih nodova, kao i od broja nodova na klasteru.

7. ZAKLJUČAK

U radu je predstavljeno rešenje skladištenja podataka u memoriji upotrebom akter programskog modela i mikroservisnog arhitektonskog šablona. Analizirani su radovi koji se bave pomenutom tematikom i neki predlozi

rešenja predstavljeni su u ovom radu. Na osnovu nekih od tih predloga implementirano je osnovno softverko rešenje koje omogućava particionisanje podataka na veliki broj aktera koji u svojoj memoriji skladište podatke. Problem ravnomernog iskorišćenja resursa rešen je upotrebom algoritma za particionisanje brojeva.

Kako bi se obezbedila pouzdanost podataka, realizovano je repliciranje podataka između aktera, gde postoji jedna primarna instanca svakog aktera i dve sekundarne replike, i perzistencija podataka na *Azure SQL* bazu podataka.

8. LITERATURA

- [1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology," *Nist Spec. Publ.*, vol. 145, p. 7, 2011.
- [2] C. de la Torre, B. Wagner, and M. Rousos, *.NET Microservices: Architecture for Containerized .NET Applications*, V1.0. Redmond, Washington: Microsoft Corporation.
- [3] S. Newman, *Building Microservices*, 1st ed. Sebastopol, CA: O'Reilly media, Inc., 2015.
- [4] M. Fowler and J. Lewis, "Microservices." [Online]. Dostupno: <https://martinfowler.com/articles/microservices.html>. [Pristupljeno: 12-sep-2018].
- [5] C. Hewitt, P. Bishop, and R. Steiger, "A Universal Modular ACTOR Formalism for Artificial Intelligence," *Proceeding IJCAI'73 Proceedings of the 3rd international joint conference on Artificial intelligence*, 1973, pp. 235–245.
- [6] P. Bernstein, "Actor-Oriented Database Systems," in *Proceedings of the 2018 IEEE 34th International Conference on Data Engineering*, 2018, pp. 13–14.
- [7] V. Shah and M. A. V. Salles, "Actor Database Systems: A Manifesto," no. 1, 2017.
- [8] P. A. Bernstein, T. Kiefer, and D. Maier, "Indexing in an Actor-Oriented Database," *Conf. Innov. Database Res.*, 2017.
- [9] P. Hitzler, M. Krötzsch., and S. Rudolph, *Foundations of Semantic Web Technologies*. 6000 Broken Sound Parkway NW, Suite 300: Chapman & Hall/CRC textbooks in computing, 2010.
- [10] R. E. Korf, "Multiway Number Partitioning," *Proc. Twenty-First Int. Jt. Conf. Artif. Intell.*, pp. 538–543, 2009.
- [11] "Introduction to microservices on Azure | Microsoft Docs." [Online]. Dostupno: <https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview-microservices>. [Pristupljeno: 23-sep-2018].

Kratka biografija:



Marko Jelača rođen je 12.09.1994. godine u Sremskoj Mitrovici. Završio je Mitrovačku gimnaziju u Sremskoj Mitrovici 2013. godine. Fakultet tehničkih nauka u Novom Sadu je upisao 2013. godine, a diplomirao na smeru Elektroenergetski softverski inženjering 2017. Iste godine upisuje master studije na smeru Primenjeno softversko inženjerstvo.

PRISTUP NADGLEDANJU INDIKATORA RADA IOT ČVORIŠTA U REALNOM VREMENU**ACCESS TO MONITORING INDICATOR OF WORKING OF IOT HUB IN REAL TIME**

Dejan Živković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – Upravljanje sistemima

Kratak sadržaj – U ovom radu opisano je proširenje programske podrške postojećeg sistema kućne automatizacije. Proširenje podrazumijeva dodavanje kontrolera, u vidu novog modula, koji će prikupljati trenutnu statistiku rada sistema. To se odnosi na prikupljanje informacija o broju promjena stanja uređaja i broju aktivacija scena u sistemu. Takođe, kontroler implementira i opciju za podešavanje vremenskog intervala u toku kojeg će statistika biti prikupljana. Dat je i kratak opis IoT tehnologije, koja se koristi u opisanom programskom rješenju za spregu kućnih uređaja sa „oblak“ računarskim sistemom, radi bolje i kvalitetnije analize podataka i informacija o radu sistema.

Ključne reči: *Sistemi kućne automatizacije, IoT, „oblak“ računarstvo*

Abstract – This paper describes the expansion of software support for the existing home automation system. Extension implies the addition of a controller in the form of a new module that will collect current system operating statistics. This refers to the collection of information about the number of changes in the condition of the device and the number of activations of the scenes in the system. Also, the controller implements the option to set the time interval during which the statistics will be collected. Also, there is short description of IoT technology used in the described software solution for interconnecting home appliances with a "cloud" computer system, for better and quality data analysis and information on system operation.

Keywords: *Home automation systems, IoT, "cloud" computing*

1. UVOD

Sistem kućne automatizacije je tehnološko rješenje koje obuhvata automatizaciju većine elektronike, električnih i tehnoloških funkcija u kući. Upotrebljava kombinaciju hardverskih i softverskih rješenja koje omogućavaju kontrolu i upravljanje aparatima i uređajima u kući. Kuća sa sistemom automatizacije se naziva „pametna kuća“. Uređaji unutar sistema kućne automatizacije se povezuju i komuniciraju međusobno preko lokalne (LAN) ili bežične mreže.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Velimir Čongradac, vanred. prof.

Čitav sistem kućne automatizacije zahtjeva programsku podršku za upravljanje sistemom, instalaciju kontrolera za uređaje i aparate, senzore temperature i pokreta, kao i druge komponente [1].

Internet stvari (eng. Internet of Things) ili skraćeno IoT, je mreža fizičkih objekata ili „stvari“, ukomponovanih sa softverom, elektronikom, senzorima i mrežnom spregom, koja omogućava ovim objektima da sakupljaju i razmjenjuju podatke.

Pametna kuća je primjena internet tehnologija u domaćinstvu. Kombinuje prikupljanje, komunikaciju, skladištenje i analizu podataka, softverski i električni dizajn, grupu tehnologija, kao što su čuvanje energije i zaštita okoline. IoT omogućava da se vrijednosti objekata očitaju sa senzora i da se kontrolišu udaljeno preko postojeće mrežne infrastrukture.

To otvara mogućnost za neposredniju integraciju između stvarnog svijeta i računarskih sistema i rezultuje povećanom efikasnošću, preciznošću i ekonomskom dobiti. Svaka „stvar“ je jedinstveno identifikovana preko svog ugrađenog računarskog sistema, ali može i da razmjenjuje informacije sa drugim komponentama postojeće internet infrastrukture [2].

2. INTERNET STVARI (IoT)

IoT može da se definiše kao mreža povezanih uređaja sa jedinstvenim identifikatorima u obliku IP adrese, koji imaju ugrađene tehnologije ili su opremljeni tehnologijama koje im omogućavaju da prikupljaju informacije sa senzora i da komuniciraju međusobno i/ili sa okolinom u kojoj se nalaze.

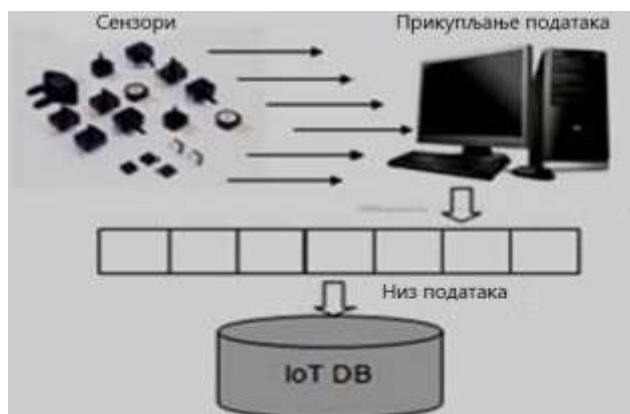
Potencijal i realnost IoT-a se ne ogleda u povezanosti IoT objekata niti u ugrađenoj tehnologiji i elektronici, kao što su senzori, aktuatori i mogućnosti povezivanja. Ogleda se u načinu na koji se IoT koristi da bi unaprijedio uvid u podatke, da automatizuje, digitalizuje, optimizuje i u zrelijim fazama transformiše procese, poslovne modele pa čak i industrijskih procesa u pogledu digitalne transformacije.

Nisu svi mrežno povezani uređaji IoT povezani, ali su svi IoT uređaji mrežno povezani. Šta više, u IoT se koristi IP protokol, preciznije IPv6. Stoga, o IoT se može govoriti samo ako su „stvari“ jedinstveno adresibilne, preko IP adrese ili URI-ja (Uniform Resource Identifier). IoT ne čine samo povezani uređaji, nego i hardver, softver, mrežni i komunikacioni protokoli, kao i mnoga toga drugog potrebnog za kreiranje IoT rješenja [3].

2.1. Prikupljanje podataka u sistemu kućne automatizacije

Pametna kuća je virtuelno-fizički sistem izgrađen na IoT, računarima i pametnim kućnim aparatima, sa ljudskom interakcijom preko kućnih komunikacionih mreža i interneta. Kao sakupljač podataka i pristupnik internetu, mrežni prolaz (eng. gateway) pametne kuće nadgleda uređaje pametne kuće koji kontrolišu kućno okruženje i služe korisnicima. Tradicionalno, mrežni prolaz, npr. modem ili ruter, povezuje internet konekciju sa kućnom lokalnom mrežom. Takođe, upravlja mrežom za većinu kućnih mrežnih uređaja, kao što su pametni telefoni, pametni električni aparati i dr. [4]. Na slici 1 je prikazan jednostavan dijagram prikupljanja podataka u sistemu kućne automatizacije koji se oslanja na IoT. Prikazan Kućni IoT uređaji, kao što su pametni ormarići, video

nadzor, daljinski kontroleri i slično, su povezani preko različitih komunikacionih tehnologija da bi obezbjedili različite tipove pametnih aplikacija, uključujući nadzor okoline, bezbjednost, kućnu automatizaciju...[5]. Cjelokupno mrežno upravljanje ovim kućnim aplikacijama se odvija u pametnom mrežnom prolazu, tzv. gejtveju. On takođe sarađuje sa spoljnim sistemima, poput „oblak“ (eng. cloud) i internet servisa. Da bi se obezbjedili mrežni servisi sa dobrim kvalitetom ličnog iskustva korisnika (eng. quality-of-experience), ili skraćeno QoE, pametni gejtvej mora da prikupi ogromnu količinu podataka za analizu, npr. u „oblaku“. Analiza podataka će se usredsrediti na podatke o mjerenjima mrežnog kvaliteta servisa (eng. quality-of-service), ili skraćeno QoS, bezbjednosti i ponašanju korisnika pametne kuće [6].



Slika 1. Dijagram prikupljanja podataka sistema kućne automatizacije baziranog na IoT [4]

3. KONCEPT RJEŠENJA

Postojeće rješenje se sastoji od 3 komponente i to:

Softverske komponente koja obezbjeđuje funkcionalnosti kućne automatizacije, nazvane „kućni upravljač“;

„Oblak“ servis koji povezuje krajnje korisnike sa njihovim domom bilo gdje da se nalaze;

Klijentske aplikacije koje omogućavaju krajnjim korisnicima da kontrolišu svoje domove sa Android, iOS ili uređajima baziranim na mrežnom pretraživaču.

Kućni upravljač radi ili na gejtvej uređaju kućne automatizacije ili se integriše unutar programskog okruženja uređaja „domaćina“ (npr. WiFi/3G/4G ruter, zvučna karta, personalni računar i dr.). Podržava najmodernije bežične komunikacione protokole danas, kao što su Zigbee, Z-Wave, IP. Kreira univerzum u kojem su sva čvorišta apstraktna u odnosu na svoje usluge i svojstva. Podržava jednostavnu integraciju novih protokola kućne automatizacije, koristi javni aplikativni programski interfejs (API), podržava konfiguraciju preko mrežnih interfejsa, upravlja čvorištima pomoću lokacije ili tipa, podržava napredne mehanizme scena, povezuje se na „oblak“ servise preko MQTT protokola i omogućava pristup nekim funkcijama „oblak“ servisa, poput daljinskog upravljanja, dijagnostike, ažuriranja firmvera i slično.

„Oblak“ servis omogućava kontrolu domaćinstva izvan lokalne mreže i dozvoljava udaljenu dijagnostiku i konfiguraciju raspoređenih gejtveja kućne automatizacije. Može da se koristi kao kompletno „oblak“ rješenje ili da

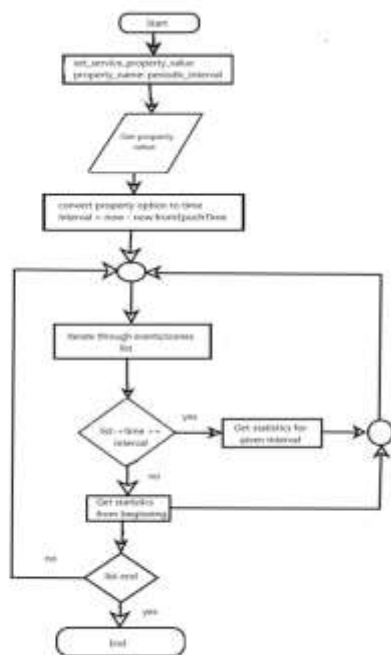
se integriše sa postojećim „oblak“ servisom. Zasnovan je na novoj mrežnoj tehnologiji, poznatoj kao MQTT, obezbjeđuje najveću bezbjednost i robusnost sa najmanjim kašnjenjem u komunikacionom lancu kućne automatizacije. Ključne komponente rješenja su aplikativni server, baza podataka i pozadinski servisi. Cjelokupna komunikacija je bazirana na sigurnim protokolima. Dodatna mjera bezbjednosti je ta što se gejtveji uvijek povezuju na predefinisano adresu na „oblaku“ koja se ne može mijenjati.

Klijentske aplikacije obezbjeđuju finu kontrolu doma pomoću aplikacija lakih za upotrebu. Ključna ideja je da se sakrije složenost tehnologije od korisnika iza grafičkog korisničkog interfejsa (GUI). Aplikacije u svakom trenutku imaju najažurnije informacije o gejtvejima i čvorištima.

U cilju proširenja postojećeg rješenja potrebno je realizovati kontroler u okviru „kućnog upravljača“ koji će pratiti rad internih modula i na osnovu prikupljenih informacija davati statistiku o radu sistema. Potrebno je da kontroler prati broj ažuriranja uređaja u sistemu i broj izvršavanja scena. To podrazumijeva ukupan broj ažuriranja uređaja, odnosno izvršavanja scena, od početka rada sistema i prosječan broj ažuriranja uređaja, odnosno izvršavanja scena, u jedinici vremena. Podešavanje kontrolera treba da omogući opciju za podešavanje jedinice za prosječan broj izvršavanja na svakih 15 minuta, 30 minuta i na svakih sat vremena.

Pored opcije za podešavanje, kontroler treba da omogući skup funkcija za dobavljanje statistike: za sve uređaje, za

statistika. Na primjer, ukoliko se iz korisničke aplikacije zada opcija 0, vršiće se prikupljanje statistike u vremenskom prozoru od 15 minuta. Ukoliko je ta vrijednost zadana u prvih 15 minuta rada sistema, vršiće se dobavljanje statistike od početka rada sistema. Ukoliko se vrijednost intervala zada nakon 15 minuta rada sistema, trenutak u kome se zada vrijednost, smatra se početnim trenutkom dobavljanja statistike, i od tog trenutka pa u narednih 15 minuta će se vršiti dobavljanje statistike. Na slici 4 prikazan je algoritam dobavljanja statistike u zadatom vremenskom intervalu.



Slika 4. Algoritam dobavljanja statistike u jedinici vremena

4. REZULTATI RADA

Verifikacija rada se vrši pomoću skripte napisane u interpreteru skripti koja je dodata u postojeće programsko rješenje. U cilju verifikacije rada pokrenuta je klijentska aplikacija u kojoj je kreirana scena koja će se aktivirati kad se otvori ili zatvori senzor vrata. Korištena je i pametna utičnica, čije je stanje mijenjanje ručno.

U tabeli 1 prikazan je ciklus ispitivanja i verifikacije rada sistema. Tabela prikazuje opis izvršenih testova i njihove rezultate.

Tabela 1. Prikaz izvršenih testova i njihovih rezultata

Test	Rezultat
Dobavljanje inicijalnog broja promjena uređaja i scena u sistemu	PROŠAO
Ispis trenutnog broja promjena stanja uređaja i scena u zadatom vremenskom intervalu od 15 minuta	PROŠAO

Ispis statistike u periodu mirovanja sistema sa zadatim intervalom od 15 minuta	PROŠAO
Ispis trenutnog broja promjena stanja uređaja i scena u zadatom vremenskom intervalu od 30 minuta	PROŠAO
Ispis statistike u periodu mirovanja sistema sa zadatim intervalom od 30 minuta	PROŠAO
Ispis statistike od početka rada sistema	PROŠAO

5. LITERATURA

- [1] Home Automation System, URL <https://www.techopedia.com/definition/29999/home-automation-system>, September 2018.
- [2] A Data Acquisition and Control System in Smart Home Based on the Internet of Things, Zhubing Hu, Chengde Petroleum College, 2016.
- [3] S. Suresh and P. V. Sruthi, "A review on smart home technology," in 2015 Online International Conference on Green Engineering and Technologies (IC-GET), Nov 2015, pp. 1–3.
- [4] A Smart Home Gateway Platform for Data Collection and Awareness, Pan Wang, Feng Ye, Member, IEEE, Xuejiao Chen
- [5] S. Guoqiang, C. Yanming, Z. Chao, and Z. Yanxu, "Design and implementation of a smart iot gateway," in 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Aug 2013, pp. 720–723.
- [6] P. Wang, S. Y. Zhang, and X. J. Chen, "Sfarima: A new network traffic prediction algorithm", in 2009, First International Conference on Information Science and Engineering, Dec 2009, pp. 1859–1863

Kratka biografija:



Dejan Živković rođen je u Derventi 1990. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2018.god.
kontakt: zivkovicdejan230@gmail.com

РЕГУЛАЦИЈА ТЕМПЕРАТУРЕ ТЕРМАЛ УЉА У ПРОЦЕСУ ПРЕДЖЕЛИРАЊА ПАСТЕ У ПОГОНУ ЗА ПРОИЗВОДЊУ PVC ПОДА**TEMPERATURE CONTROL OF THERMAL OIL IN PASTE JELLIFICATION PROCESS IN THE PVC FLOOR PRODUCTION PLANT**

Александра Максимовић, Факултет техничких наука, Нови Сад

Област – Електротехника и рачунарство

Кратак садржај – Предмет овог рада је анализа тренутног стања регулације температуре термал уља у фабрици за производњу хетерогених PVC подова. Најпре биће детаљно приказан опис система и технолошког процеса који се у њему дешава. Посебан акценат биће на томе како је реализована регулација температуре термал уља и које су кључне компоненте регулационог система. Затим ће бити појашњене све функције и параметри који се могу подешавати у коришћеном регулатору, као и које од њих се користе у конкретном случају. На крају, биће приказани одзии температуре у различитим случајевима, након подешених параметара регулатора.

Кључне речи: температура, регулација, параметри

Abstract – The subject of this study is an analysis of the current state of temperature control of thermal oil in paste jellification process at the plant for heterogeneous PVC floors production. First of all, a detailed description of the system and technological process will be presented. A special emphasis will be on how temperature of thermal oil is currently regulated and what are the key components of the regulation system. Then all the functions and parameters that can be set in the chosen regulator will be explained, with special accent on which one are used in the particular case. Finally, after the adjusted parameters of the regulator, temperature charts will be shown, for different situation in explained process.

Keywords: temperature, regulation, parameters

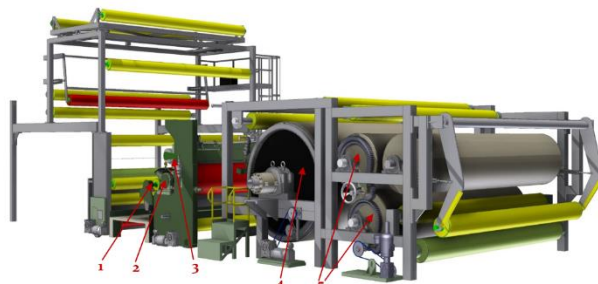
1. УВОД

Процес производње хетерогених PVC подова је такав да се на основу од стакленог флиса наноси више слојева PVC пасте. Паста представља смешу PVC праха, омекшивача и стабилизатора. Процес производње је континуалан, што значи да нема паузе између наношења појединачних слојева.

Изглед опреме потребне за процес наношења једног слоја пасте приказан је на слици 1. Робна трака преко уводног ваљка долази до премазног ваљка где се испушта паста.

Дебљина наноса пасте се регулише подешавањем зазора између премазног ваљка и премазног ножа,

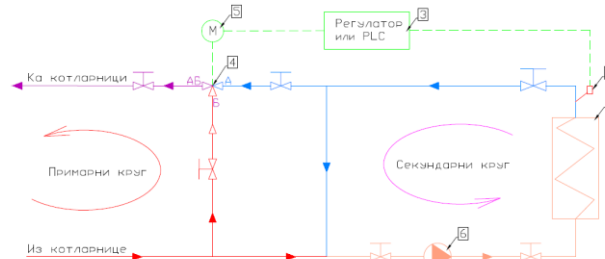
тако што се нож подиже или спушта у зависности од захтева производње. Даље, робна трака прелази преко цилиндра за преджелирање, где се одвија процес згушњавања пасте на високој температури и испаравања дела течности. Бубањ је изграђен од дуплог плашта, који чине унутрашња и спољашња цев. Између њих се налазе преграде спиралног облика кроз које протиче термално уље, загрејано на температуру задате вредности. Веома је битно да температура не одступа од задате вредности, како не би било утицаја на квалитет финалног производа. Након процеса преджелирања, робна трака се преноси до ваљака хлађења који теже да смање температуру робе и припреме је за наношење следећег слоја.



Слика 1. Изглед опреме: 1. Уводни ваљак, 2. Премазни ваљак, 3. Премазни нож, 4. Цилиндар за преджелирање, 5. Ваљци за хлађење

2. ПРИНЦИП НАМЕШАВАЊА ТЕРМАЛ УЉА У ЦИЛИНДРУ ЗА ПРЕДЖЕЛИРАЊЕ

Температура термалног уља потребна за процес преджелирања пасте добија се намешавањем врелог уља које стиже из котларнице (280°C) и уља које излази из бубња. Уље из котларнице (означено црвеном бојом на слици 2) се преко трокраког вентила додаје уљу које излази из цилиндра за преджелирање (означено црвеном бојом на слици 2).



Слика 2 – Принципска шема намешавања термалног уља: 1. Цилиндар за преджелирање, 2. Мерач температуре, 3. Управљачки склоп, 4. Трокраки мешни вентил, 5. Актуатор, 6. Пумпа секундарног круга

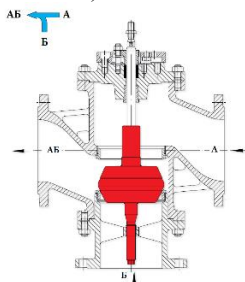
НАПОМЕНА:

Овај рад је проистекао из мастер рада чији ментор је био проф. др Дарко Марчетић.

На овај начин трокраки вентил директно намешава уље које се враћа ка котларници, а индиректно уље које одлази ка цилиндру. Трокраким вентилом управља серво мотор који прецизно остварује потребан положај вентила, додајући таман онолико врућег уља колико је потребно да се одржи жељена температура.

2.1. Трокраки (мешни) вентил

У посматраном систему користи је трокраки вентил *Ari STEVI 423*, у функцији мешног вентила. Састоји се од три крака, два улазна (А и Б на слици 3) и једног излазног (АБ на слици 3).



Слика 3 – Трокраки мешни вентил

Вруће термал уље из котларнице долази преко преко улаза Б у мешни вентил. На улаз А долази уље које излази из бубња. Уље са улаза А и Б се меша и преко излаза АБ враћа у котларницу где се догрева на 280°C.

Ради лакшег схватања принципа рада, биће објашњени екстремни случајеви, који у посматраном систему у трајном раду нису реални. Када је клип вентила подигнут у крајњи горњи положај, са улаза А се не преноси уље на излаз, већ само са улаза Б. Уз котларнице се враћа у котларницу и тај ток представља примарни круг на слици 2. У том случају уље које излази из бубња, поново се враћа у њега (секундарни круг на слици 2). Кружење уља у секундарном кругу омогућава пумпа (на слици 2 означена бројем 6). Када је клип вентила спуштен у крајњи доњи положај укупна количина уља које долази из котларнице иде ка бубњу, а уље које излази из бубња враћа се назад ка котларници.

У реалном случају, док постоји одношење топлоте приликом проласка робне траке преко бубња, клип вентила не може бити константно у крајњем горњем положају. Када би имали такву ситуацију, уље би при сваком проласку имало све нижу температуру и не би били остварени услови потребни за процес који се одвија. Такође, клип вентила у редовном раду не може бити ни у крајњем доњем положају. Температура долазног уља у бубањ би била 280°C, што је превелика вредност за потребе процеса. Клип мешног вентила може бити у неком од крајњих положаја одређени временски период у специфичним случајевима, а то су првобитно загревање бубња на старту производње и хлађење бубња по завршетку производње.

Постоје три карактеристична режима рада мешног вентила, а то су:

1.Старт производње- Мешни вентил је у крајњем горњем положају. Раздвојени су примарни и

секундарни круг. Термал уље које циркулише кроз бубањ је смањене температуре, око 30°C. Пре наилазак робне траке на бубањ потребно је загрејати уље на жељену температуру 150°C. При томе се део енергије губи, одлазећи у атмосферу. Нема губљења енергије услед проласка робне траке преко бубња.

2. Редован рад- У овом режиму постоје два поремећаја. У оба случаја мешни вентил се налази у средњем положају и намешава уље из котларнице и повратно уље из бубња како би остварио потребну температуру у бубњу. Први поремећај се дешава када робна трака долази први пут до бубња. Тада долази до наглог одношења веће количине топлоте, те вентил мора пропустити више врућег уља из котларнице. Други поремећај се дешава периодично, када долази до промене ширине робне траке. Приликом тога долази до промене количине одношене енергије. Уколико је промена са веће ширине на мању, количина одношене енергије се смањује, док се код промене са мање ширине на већу, та количина повећава. Мешни вентил треба да испрати промене и да променом положаја клипа додаје више или мање врућег уља. На тај начин он одржава жељену температуру и при сваком поремећају.

3. Завршетак производње- Температуру термал уља је потребно смањити на што мању вредност, да не би дошло до даљих губитака енергије; то се може постићи једноставно, подизањем клипа вентила у крајњи горњи положај.

2.2 Линеарни актуатор

Линеарни актуатор *ARI – Premio 2,2.5kN* користи се за отварање/затварање или регулисање положаја вентила. Он претвара ротационо кретање синхроног мотора у праволинијско кретање клипа. Синхрони мотор покреће навојно вретено у једну или другу страну (увлачи га или извлачи). Навојно вретено је повезано са клипом вентила, те се клип креће горе или доле у зависности од смера ротације мотора.

Синхрони, мотор и остале електричне компоненте овог актуатора се налазе испод његовог поклопца и на тај начин су заштићене од спољних утицаја. На штампаној плочи постоје 2 трополна конектора, један на који се доводе сигнали са регулатора и други преко ког се шаљу сигнали ка синхронном мотору. Први трополни конектор има три прикључка на улазном делу, са ознакама 1/N, 11 и 14. Излазни прикључци су одведени на штампану плочу. На штампаној плочи се налазе гранични прекидачи S1 и S2, на које се доводи сигнал са прикључака 11 и 14. Са контаката граничних прекидача се сигнал води на улазне прикључке другог тропинског конектора. Сигнал са контаката прекидача S1 води се на прикључак M_A, а са контаката прекидача S2 на прикључак M_C. Ради лакшег схватања, на слици 10 приказана је електрична шема повезивања.

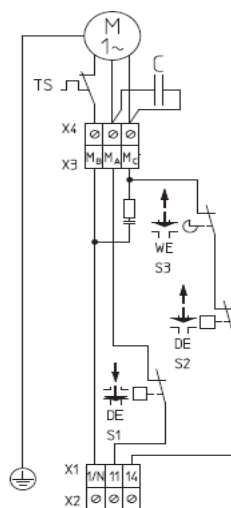
Контакти граничних прекидача су нормално затворени (мирни), што значи да мотор има услов за рад све док полуга не додирне један од граничних прекидача и на тај начин отвори његов контакт. Са прикључка 11, преко прекидача S1, доводи се на

прикључак M_A сигнал мотору за кретање у једну страну. Докле год постоји напон на M_A навојно вретено се извлачи, клип вентила се креће ка доле, односно отвара улаз вентила А ка излазу АБ. У случају да клип дође до крајњег доњег положаја, полуга ће додирнути гранични прекидач S1.

Његови контакти ће се раздвојити и на тај начин ће нестати напон на прукључку M_A , те ће се мотор зауставити. Са прикључка 14, преко прекидача S2, на прикључак M_B доводи се сигнал мотору за кретање у супротном смеру. Докле год постоји напон на M_B навојно вретено се увлачи, клип вентила се креће ка горе.

Дешава се супротан процес од претходно објашњеног. Затвара се пут течности од улаза А, а отвара се пут од улаза Б ка излазу АБ.

У случају да клип дође до крајњег горњег положаја, полуга ће додирнути гранични прекидач S2 и његови контакти ће се раздвојити. На тај начин ће се прекинути струјни круг на прукључку M_B , па ће се мотор зауставити.



Слика 4. Шема повезивања електричних компоненти актуатора

2.3 Регулатор температуре

Посматрајући објашњени систем, јасно је да мешни вентил мора имати такво управљање да у сваком моменту остварује положај у ком ће намешано уље постићи жељену температуру. Обзиром да постоје губици енергије услед преласка робне траке преко цилиндра за преджелирање, потребно је имати повратну информацију о тренутној температури уља у цилиндру. Та информација се шаље регулатору OMRON E5EK.

На основу разлике између задате вредности и повратне информације о стварној вредности температуре, регулатор има задатак да одреди да ли је потребно померити клип мешног вентила горе или доле, односно додати више или мање топлог уља.

У овом случају коришћена је PT100 температурна сонда. Постављена је на повратном цевоводу. Сигнал са сонде се повезује трожишно на улазне прикључке 11, 12 и 13 на регулатору температуре, као што је приказано на слици 5.

Контролни сигнал се шаље регулатору преко два релејна излаза (контролни излаз 1 и 2 на слици 5). Контакти релејних излаза су нормално отворени. Један треба да проводи у случају отварања вентила, а други у случају затварања, што се подешава при параметрирању регулатора у његовом менију. Сигнал са контролног излаза 1 доводи се на тропински конектор актуатора, на прикључак 11 (са слике 4), док се сигнал са контролног излаза 2 доводи на прикључак 14 (са слике 4). Уколико је остварена температура мања од задате, клип вентила би требало да се креће ка доле, и активиран је контролни излаз 1. Ако је остварена температура већа од задате, клип вентила би требало да се креће ка горе и активира се контролни излаз 2.



Слика 5. Повезивање улазних и излазних сигнала регулатора

3. ПАРАМЕТРИ РЕГУЛАТОРА ТЕМПЕРАТУРЕ ТЕРМАЛ УЉА У ПРОЦЕСУ ПРЕДЖЕЛИРАЊА

Параметри регулатора OMRON E5EK су подељени у следећих 9 нивоа (мода): PROTECT MODE, MANUAL MODE, Level 0, Level 1, Level 2, Setup mode, Expansion mode, Option mode, Calibration mode.

Пре свега се подешавају параметри у Setup режиму. Тип улазног сигнала је подешен на Pt100, -199.9 – 650.0 °C. Контролни излаз 1 се подешава на функцију *heat* (грејање), контролни излаз 2 на функцију *cool* (хлађење). То значи, да уколико је остварена температура мања од задате, активираће се контролни излаз 1. Актуатор добија услов да покрене клип вентила ка доле и отвори пут врућем уљу из котларнице. У супротном случају активира се контролни излаз 2. Затим се изврши калибрација улазног сигнала у Calibration режиму подешавања.

Auto-tuning, тј. аутоматско тражење параметара PID регулатора није могуће јер га је потребно покренути у току редовне производње, што би довело до велике количине шкарт производа и није исплатљиво. Стога, параметри су искуствено подешавани. Вредности параметара су: P=20, I=233 и D=45.

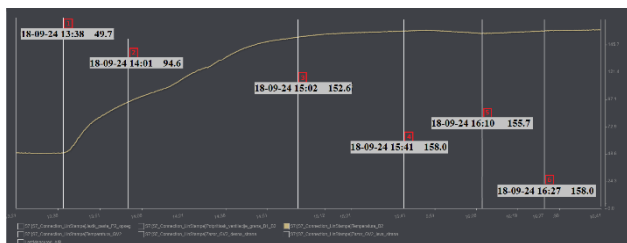
4. ЗАКЉУЧАК

Параметри регулатора су усклађени према најкритичнијем случају за регулацију, а то је редован рад производње. То је најкритичнији случај у смислу, да је тада најбитније да температура термал уља не одступа од задате вредности ни приликом промене типа или ширине производа, ни приликом производње истог производа.

По старту загревања, потребно је да термал уље достигне жељену температуру од 158°C, без

оптерећења, односно без одношења топлоте са површине цилиндра за преджелирање. Приликом почетка загревања уља, нагло се отвори пут врелом уљу из котларнице што се манифестује наглим порастом температура до неке вредности. Када температура достигне око 90°C, обзиром да је разлика између задате и остварене вредности мања него у старту, регулатор шаље актуатору информацију да прави мање помераје. Последица тога је одзив температуре са блажим одзивом (на слици 6 део карактеристике између 2 и 4). У тачки 4, температура је постигла стационарну вредност. Може се приметити да је одзив аperiодичан и да загревање уља траје око 60min што је сасвим прихватљиво време.

Први поремећај се дешава када робна трака први пут дође до цилиндра. На слици 6, бројем 6 је означен пад температуре приликом наиласка робне траке ширине 2,5m. Због наглог одношења топлоте, температура опадне за 3°C. Након врло кратког времена, благим повећањем протока врелог уља, поново се постиже жељена температура од 158°C. Одзив је брз и аperiодичан и оваква осцилација не утиче на квалитет производа.



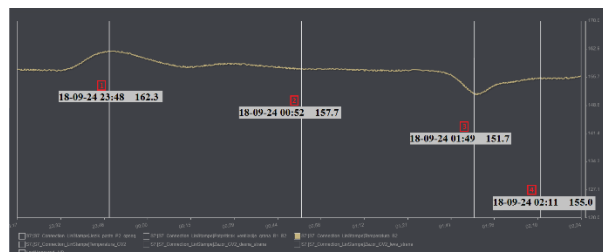
Слика 6. Одзив температуре: 1-4. Приликом првобитног загревања, 5-6. Приликом првог наиласка робне траке по старту производње

Други поремећај се дешава приликом промене ширине робне траке са веће на мању или обрнуто, што је и приказано на слици 7. Најпре се дешава промена ширине са 3,5m на 2,5m. Смањује се количина енергије која се односи са површине бубња, те температура на повратном воду постаје већа. Регулатор шаље актуатору информацију да смањи доток врућег уља из котларнице.

Највећа вредност температуре које уље има приликом овог поремећаја износи 162,3 степена, што се може видети са слике 7, означено бројем 1. Температура је већа од задате свега пар минута и има мањег једне периоде осцилације. Друга полупериода има знатно мању амплитуду од прве, што показује добру стабилност система регулације. На слици 7, означено бројем 2, види се да је температура поново постигла задату температуру (157,7°C). Овакав одзив при промени ширине не представља претњу квалитету производа.

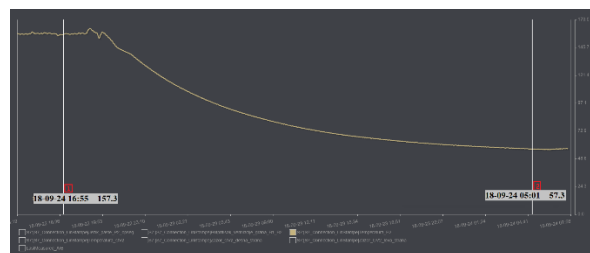
У другом делу графикона може се видети промена температуре приликом наиласка производа веће ширине, при чему је промењена и задата температура са 158°C на 155°C.

Робна трака при томе повећава количину енергије коју односи са собом, па сонда на повратном воду детектује мању температуру од 151,7°C. Регулатор шаље сигнал актуатору да повећа доток врућег термал уља. Неколико минута касније температура достиже 155°C што се може видети на слици 7, означено бројем 4. Достицање жељене вредности траје кратко и одзив је готово аperiодичан чему се и тежи. Оваквим одзивом температуре не угрожава се квалитет готовог производа.



Слика 7. Одзив температуре приликом промене ширине производа

На слици 8 се може видети одзив температуре по завршетку производње, када се зада температура од 0°C. Тада вентил затвара доток врелог уља из котларнице, односно потпуно раздваја примарни и секундарни круг циркулације. Температура уља које кружи кроз бубањ полако опада. Тај процес је као што се може видети доста спорији, али то није од великог значаја.



Слика 8. Одзив температуре приликом завршетка производње

5. ЛИТЕРАТУРА

- [1] Др Филип Кулић дипл.инж.ел., Драган Матић дипл.инж.ел., др Веран Васић дипл.инж.ел., мр Ђура Орос, *Подешавање параметара PI регулатора температуре апсорпционог уља у погону екстракције у процесу производње биљних уља*, Факултет техничких наука, 21000 Нови Сад, Трг Доситеја Обрадовића 6.
- [2] „Operating and Installation Instructions Thrust actuator ARI-PREMIO“, 2008, доступно на: http://www.ari-armaturen.com/_appl/files_tb/files/0040501000-2.pdf
- [3] „E5EK Digital Controller User's Manual“, 2018, доступно на: https://www.fa.omron.com.cn/data_pdf/closed/mnu/h085-e1-03_e5ek_discon.pdf?id=1436

**UPRAVLJANJE HVDC PRENOSOM ZASNOVANIM NA PRETVARAČIMA KAO
NAPONSKIM IZVORIMA****CONTROL OF VOLTAGE SOURCE CONVERTER BASED HVDC TRANSMISSION**Aleksandar Kokotović, Marko Vekić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu su razmatrane najvažnije osobine sistema koji se koriste za jednosmjerni prenos električne energije. Naglašene su prednosti, kao i nedostaci u odnosu na naizmjenične sisteme. Predstavljene su osnovne komponente sistema. Opisano je upravljanje sistemom koristeći strujnu regulaciju. Na kraju su dati rezultati simulacije rada sistema u ustaljenom stanju kako bi se potvrdilo njegovo očekivano ponašanje u tom režimu.

Ključne riječi: HVDC prenos, pretvarači sa naponskim izvorom, strujno upravljanje sistemom

Abstract – The most important characteristics of DC system used for power transmission are discussed in this paper. Advantages, as well as disadvantages comparing to AC systems are emphasized. The main HVDC system components are presented. Current control is detailed as a method to regulate power exchange between two independent AC grids. In order to verify expected behaviour, simulation results of a system in steady state operation are published and explained.

Keywords: HVDC Transmission System, Voltage Source Converters, Current Control Method

1. UVOD

Prva javno upotrebljena mreža za prenos električne energije, izgrađena krajem 19. vijeka. U to doba odvijao se „Rat struja“ između Edisona sa jedne i Tesle i Vestinghausa sa druge strane koji su na kraju dobili Tesla i Vestinghaus otvorivši na taj način put korišćenju naizmjenične struje (AC) za proizvodnju, prenos i potrošnju električne energije. Pošto je tehnologija zasnovana na naizmjeničnoj struji bila naprednija u odnosu na onu zasnovanu na jednosmjernoj struji u pogledu proizvodnje, pouzdanosti, transformisanja kao i naponskih nivoa u prenosu, ubrzo je, postavši standardizovana, postala i stub razvoja tadašnje elektroenergetike [1].

S druge strane, razvoj HVDC sistema je započet sredinom 20. vijeka i zasnivao se na upotrebi snažnih tiristorskih prekidača. Ubrzani napredak i primjena počeli su devedesetih godina 20. vijeka, zahvaljujući razvoju brzih prekidačkih uređaja – snažnih tranzistorskih prekidača.

NAPOMENA:

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

Najnovijom HVDC tehnologijom moguće je postići sva poboljšanja koja se postižu primjenom FACTS uređaja u naizmjeničnom prenosu, a pored toga, povezivanje naizmjeničnih sistema različitih učestanosti u praksi se ostvaruje jedino jednosmjernom (DC) vezom. Stoga se, potpuno opravdano, može reći da moderna HVDC veza predstavlja najprilagođeniji prenosni sistem [2]. Motiv ovog rada je bio da se predstavi upravljanje jednim savremenim HVDC sklopom zasnovanim na IGBT prekidačima. Upotrebljen je tzv. dvostrani (eng. Back-to-Back) pretvarač koji se sastoji od dva trofazna invertora sa tri nivoa raspregnuta preko zajedničkog DC kola. Cilj je bio prikazati upravljanje tokom snage između dvije nezavisne mreže naizmjeničnog napona povezane preko HVDC sistema.

2. PREGLED HVDC TEHNOLOGIJE

Kao rezultat uvođenja HVDC tehnologije u praktičnu upotrebu nastale su dvije velike, uzajamno povezane dobite vezane za sistem prenosa električne energije. To su povećanje prenosnih kapaciteta i obezbjeđenje dodatne prilagodljivosti sistema, upravljanjem tokom snaga. HVDC prenosni sistemi koriste pretvarače energetske elektronike za povezivanje naizmjeničnih i jednosmjernih mreža. Prednost HVDC prenosa u odnosu na naizmjenični prenos su [3]:

- mogućnost povezivanja naizmjeničnih trofaznih sistema koji nisu sinhronizovani (npr. sistema čije su učestanosti 50 i 60 Hz);
- manji gubici zbog prenosa reaktivne snage, jer je uticaj parazitnih kapacitivnosti i induktivnosti gotovo zanedarljiv;
- najveći dio prenosnog kapaciteta koristi se za prenos aktivne snage;
- povećanje stabilnosti naizmjeničnih sistema međusobnim povezivanjem;
- koristi se manje provodnika za prenos iste količine energije, a moguće je koristiti samo jedan provodnik za prenos (u tom slučaju povratni put je zemlja, pri tome se koriste elektrode ogromne površine ukopane duboko u zemlju);
- provodnici su mnogo tanji (jer je skin-efekat zanemarljiv) i jeftiniji od provodnika koji se koriste u visokonaponskim naizmjeničnim mrežama;
- mogućnost ograničenja struje kratkog spoja u slučaju kvara, što ovu tehnologiju čini manje opasnom po ljude i životnu okolinu;

- smanjenje cijene izgradnje trase prenosnog voda (manje dimenzije zbog smanjenog broja provodnika);

Svi HVDC sistemi koriste pretvarače energetske elektronike, koji se mogu podijeliti u dvije grupe:

- mrežom vođeni pretvarači, koji se zasnivaju na polu-upravljivim poluprovodničkim prekidačima - tiristorima;
- samokomutujući pretvarači, koji se zasnivaju na puno-upravljivim poluprovodničkim prekidačima - tranzistorima (u pitanju su tranzistori sa izolovanim gejtom).

Klasični HVDC sistemi, koji su zasnovani na tiristorskim prekidačima, imaju prednost jedino kada je u pitanju prenos velikih snaga, kao i efikasnost pojedinačnih prekidača.

Sa druge strane, prednosti modernih HVDC sistema, koji su zasnovani na tranzistorskim prekidačima, su odsustvo potrebe za ugrađivanjem kompenzatora reaktivne snage zahvaljujući mogućnosti jednostavnog upravljanja tokom reaktivne snage, smanjene dimenzije neophodnih filtara usljed pojave harmonika višeg reda u talasnom obliku napona na izlazu pretvarača, kao i manji gubici na nivou cjelokupnog pretvaračkog sklopa.

Prema vrsti izvora koji koriste na svojim priključcima na jednosmjernoj (DC) strani, pretvarači mogu biti sa [4]:

- strujnim izvorom (Current Source Converter);
- naponskim izvorom (Voltage Source Converter).

2.1. Topologije pretvarača sa naponskim izvorom u sklopu HVDC sistema

Monofazni polumosni (Half-Bridge) DC/AC pretvarač, predstavlja osnovni gradivni element višefaznog (trofaznog) pretvarača, koji nalazi primjenu u HVDC sistemima u vidu raznih topologija.

Kada radi u invertorskom režimu, ovaj pretvarač na svom izlazu daje pravougaoni talasni oblik napona sa dva nivoa. Takav talasni oblik napona sadrži harmonike neparnog reda koje je potrebno odstraniti kako bi ostala prisutna jedino osnovna komponenta napona. Način da se otklone neželjene sinusne komponente napona je da se na izlaz invertora postavi visokopropusni filter.

Filtiri za niže harmonike (treći, peti, sedmi, deveti) bi bili izuzetno glomazni i skupi, pa je zato prvo potrebno „pomjeriti“ harmonike nižeg reda na veće učestanosti, a onda ih i ukloniti odgovarajućim filtrom. „Pomjeranje“ harmonika nižeg reda na veće učestanosti se ostvaruje upotrebom tehnike impulsno-širinske modulacije.

Postoji veliki broj različitih topologija pretvarača koji na svom izlazu mogu da daju talasni oblik napona sa više nivoa (u invertorskom režimu rada). One se mogu podijeliti na klasične i moderne [5].

3. KOMPONENTE HVDC PRENOSNOG SISTEMA

Na slici 1 prikazana je pojednostavljena pofazna šema HVDC sistema zasnovanog na pretvaračima sa naponskim izvorom. Osnovni elementi sistema su: pretvarači, energetski transformatori, fazne prigušnice, filtri viših harmonika, kondenzatori, vodovi, prekidači i uzemljivači.

4. UPRAVLJANJE HVDC PRENOSNIM SISTEMOM

Tokovima aktivne i reaktivne snage u ustaljenom stanju u HVDC sistemu se upravlja nezavisno, prema sljedećim jednačinama:

$$P = \frac{V \cdot V_m \cdot \sin \delta}{X}, \quad (1)$$

$$Q = \frac{V \cdot V_m \cdot \cos \delta - (V_m)^2}{X}, \quad (2)$$

gdje je:

V – moduo vektora napona na izlazu pretvarača, sa naizmjenične strane;

V_m – moduo vektora napona na sabirnici na koju se priključuje visokopropusni filter;

δ – ugao snage (razlika uglova vektora napona V i V_m);

X – reaktansa fazne prigušnice.

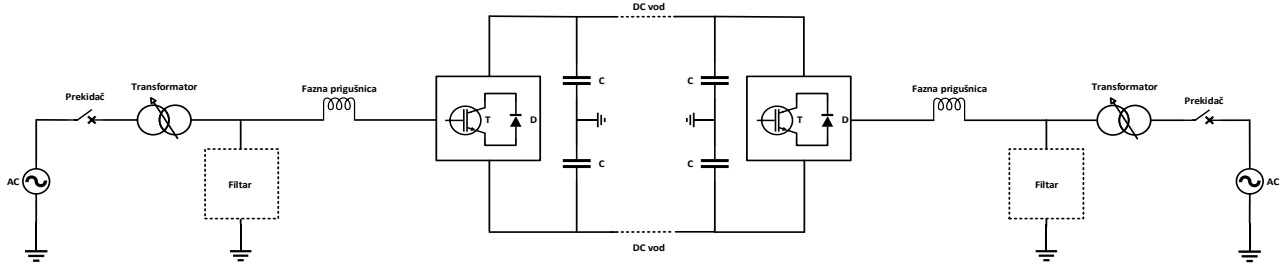
Dakle, HVDC sistem zasnovan na pretvaračima sa naponskim izvorom, u izvedbi prikazanoj na slici 1, odlikuje se mogućnošću jednostavnog upravljanja tokovima aktivne i reaktivne snage, koje se razmjenjuju između naizmjeničnih sistema. Postupak upravljanja je zasnovan na zadavanju referentnih vrijednosti za aktivnu i reaktivnu snagu sklopu koji upravlja jednim pretvaračem (neka je to, zbog daljeg razmatranja, lijevi pretvarač). Sklopu koji upravlja drugim pretvaračem (desni pretvarač) se, pored reference za reaktivnu snagu, zadaje i referentna vrijednost napona jednosmjernog međukola. Referentna vrijednost za reaktivne snage se zadaje nezavisno kod oba pretvarača.

Upravljanje radom pretvarača se vrši regulatorima koji se nalaze u okviru upravljačkih sklopova pretvarača. Regulator aktivne i reaktivne snage nezavisno upravlja razmjennom snage između jednosmjerne strane lijevog pretvarača i naizmjeničnog sistema sa iste strane. Regulator napona jednosmjernog međukola treba da obezbijedi stabilan napon, kako bi regulator snage vršio upravljanje prema izrazima (1) i (2). Proračuni upravljačkih promjenljivih se vrše u dq koordinatnom sistemu. Da bi to bilo moguće potrebno je stalno imati podatak o stvarnom položaju rotirajućeg koordinatnog sistema. Određivanje položaja koordinatnog sistema se vrši u fazno zatvorenoj petlji (Phase Locked Loop - PLL).

4.1. PLL u okviru upravljačkog sklopa HVDC sistema

Da bi upravljanje pretvaračima sa naponskim izvorom koji pripadaju HVDC sistemu bilo izvedeno jednostavno i brzo, potrebno je da se veličine koje zahtjevaju regulatori pretvore iz faznog u dq sistem. Pogodnost dq sistema je što se bilo koja trofazna veličina iz faznog sistema u dq sistemu može predstaviti sa dvije međusobno upravne komponente. One se obrću zajedno sa koordinatnim sistemom, što znači da su u ustaljenom stanju jednosmjerne. Pretvaranje veličina iz abc u dq koordinatni sistem se u oba upravljačka sklopa koji pripadaju pretvaračima vrši na osnovu određenog položaja dq koordinatnog sistema. Oba upravljačka sklopa sadrže po jedan PLL, koji pretvara fazne napone mreže V_{mabc} u napone u dq koordinatnom sistemu V_{mdq} i prilagođava brzinu obrtanja dq koordinatnog sistema sve dok q komponenta napona mreže ne postane jednaka nuli,

$V_{mq} = 0$. Kada se to dogodi, položaj dq sistema je određen. Jedna od mogućnosti za praktično ostvarenje PLL-a jeste upotreba integralno-proporcionalnog (IP) regulatora kako bi se izbjegla nula u prenosnoj funkciji, a



Slika 1. Pofazna šema HVDC sistema koji je zasnovan na pretvaračima sa naponskim izvorom

4.2. Strujno upravljanje u okviru HVDC sistema

Kombinacijom matematičkog modela HVDC sistema u ustaljenom stanju i matematičkog modela pretvarača u okviru HVDC sistema u dq koordinatnom sistemu, dobijaju se uopštene jednačine za zadavanje potrebnih komponenti modulišućeg signala u dq sistemu [6]:

$$m_d = \frac{2}{V_{DC}} \cdot \left[\left(L \frac{di_d}{dt} + Ri_d \right) + V_{md} - \omega_0 Li_q \right], \quad (3)$$

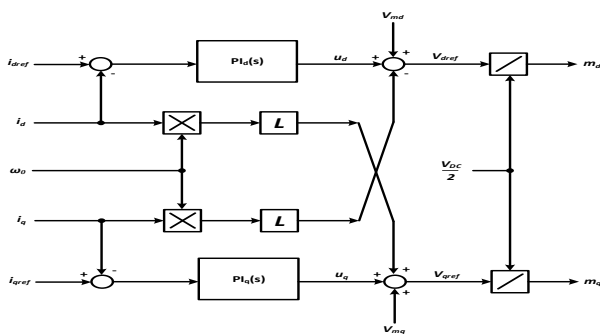
$$m_q = \frac{2}{V_{DC}} \cdot \left[\left(L \frac{di_q}{dt} + Ri_q \right) + V_{mq} + \omega_0 Li_d \right]. \quad (4)$$

Prema izrazima (3) i (4), očigledno je da su za upravljanje d i q komponentama modulišućeg signala neophodna dva proporcionalno-integralna regulatora, po jedan za obje ose. Jednačine koje opisuju rad PI regulatora predstavljaju dva međusobno raspregnuta linearna sistema prvog reda:

$$L \frac{di_d}{dt} = Ri_d + u_d, \quad (5)$$

$$L \frac{di_q}{dt} = Ri_q + u_q, \quad (6)$$

gdje su sa u_d i u_q označene upravljačke promjenljive kojima se neposredno upravlja d (3) i q (4) komponentama indeksa modulacije. Strujna blok šema upravljanja je data na slici 2.



Slika 2. Blok šema upravljanja d i q komponentama struje HVDC sistema

Upravljanje se vrši na osnovu signala greške koji predstavlja razliku između vrijednosti referentnih i stvarnih komponenti struja i_d i i_q , redom.

Zatim se nove vrijednosti komponenti modulišućeg signala koriste za upravljanje d i q komponentama napona na izlazu pretvarača, koje, u stvari, predstavljaju upravljačke promjenljive kojima se zadaju nove vrijednosti d i q komponenti struje.

time i preskok u odzivu koji bi nastao kod uobičajenog PI regulatora.

4.3. Upravljanje tokom snaga u okviru HVDC sistema

Ako je napon u tački priključenja između pretvarača i sistema stabilan, tada je u ustaljenom stanju $V_{mq} = 0$, dok je napon V_{md} jednak vršnoj vrijednosti faznog napona V_m . Ako se pretpostavi da je napon V_m stabilan, onda su izrazi za aktivnu i reaktivnu snagu u dq sistemu:

$$P = \frac{3}{2} V_m i_d, \quad (7)$$

$$Q = -\frac{3}{2} V_m i_q. \quad (8)$$

Dakle, upravljanje tokovima snaga se sprovodi tako što se na osnovu referentnih vrijednosti za razmjenu aktivne i reaktivne snage, P_{ref} i Q_{ref} , zadaju referentne vrijednosti za d i q komponente struje lijevog pretvarača, i_{dref} i i_{qref} :

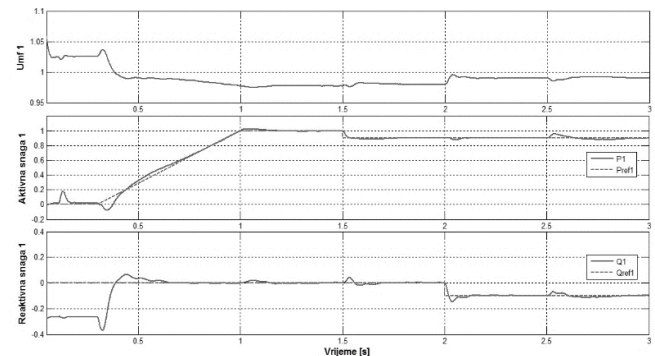
$$i_{dref} = \frac{2}{3V_m} P_{ref}, \quad (9)$$

$$i_{qref} = -\frac{2}{3V_m} Q_{ref}, \quad (10)$$

na osnovu kojih se upravlja d i q komponentama struje.

5. REZULTATI SIMULACIJE RADA SISTEMA

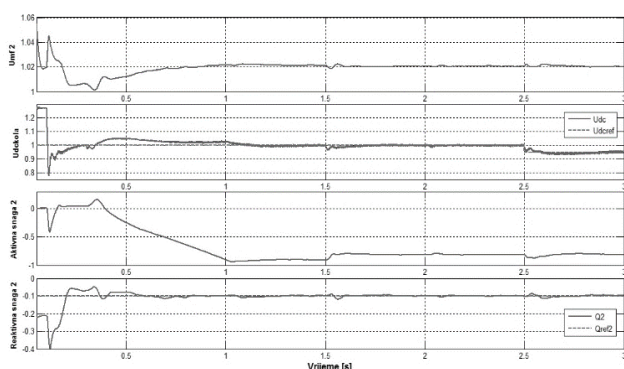
Simulacija rada HVDC sistema izvršena je u *Simulink* programskom paketu, koji je sastavni dio programskog okruženja *MATLAB*. Dužina trajanja simulacije je iznosila 3 s. Na grafiku sa slike 3 prikazani su dobijeni rezultati za „lijevu polovinu“ sistema sa slike 1. Oni su označeni brojem „1“. Na grafiku sa slike 4 prikazani su dobijeni rezultati za „desnu polovinu“ sistema sa slike 1. Oni su označeni brojem „2“. Naizmjenični sistem broj 1 preko HVDC sistema isporučuje aktivnu snagu naizmjeničnom sistemu broj 2.



Slika 3. Uporedni pregled dobijenih rezultata simulacije za „lijevu polovinu“ sistema sa slike 1

U trenutku $t = 0,3 \text{ s}$ se, dovodenjem naponskih impulsa na gejtove IGBT-ova u sklopu pretvarača broj 1, uključuje ovaj pretvarač. Kao što je već objašnjeno, on služi za upravljanje tokom aktivne snage. On radi (tokom simulacije) u ispravljačkom režimu. Smjer toka aktivne i reaktivne snage ima pozitivan predznak ako se snage isporučuju od naizmjeničnog sistema (mreže) ka pretvaraču. Odmah nakon uključivanja pretvarača broj 1 njemu se, kao referenca za vrijednost aktivne snage (isprekidana kriva), zadaje nazivna vrijednost aktivne snage (1 r.j.). Ona se dostiže u trenutku $t = 1 \text{ s}$. U trenutku $t = 1,5 \text{ s}$ se zadaje smanjenje aktivne snage za 0,1 r.j.

Referentna vrijednost za reaktivnu snagu, koja se zadaje pretvaračem broj 1, ima nultu vrijednost sve do trenutka $t = 2 \text{ s}$, kada se smanjuje za 0,1 r.j. Ono što se još može uočiti jeste zavisnost povećanja vršne vrijednosti faznih napona (prvi grafik) u tački priključenja filtera od povećanja reaktivne snage koja se isporučuje mreži.



Slika 4. Uporedni pregled dobijenih rezultata simulacije za „desnu polovinu“ sistema sa slike 1

Pretvarač broj 2 se uključuje u trenutku $t = 0,1 \text{ s}$. On tokom simulacije radi u invertorskom režimu i ima ulogu upravljanja naponom jednosmjernog međukola. Drugi grafik sa slike 4 prikazuje promjenu ovog napona (puna kriva) tokom simulacije. Po uključanju pretvarača broj 2 njemu se, kao referenca za vrijednost napona jedno-smjernog međukola, zadaje nazivna vrijednost tog napona (1 r.j.). Ova vrijednost se u trenutku $t = 2 \text{ s}$ smanjuje na 0,95 r.j.

Vrijednost aktivne snage koja se preko pretvarača broj 2 isporučuje naizmjeničnom sistemu (mreži) broj 2 ima suprotan znak od vrijednosti aktivne snage koju naizmjenični sistem (mreža) broj 1 isporučuje pretvaraču broj 1, ali je po apsolutnoj vrijednosti manja, što je posljedica gubitaka u pretvaračima i u ostalim komponentama sistema.

Referentna vrijednost za reaktivnu snagu se održava na $-0,1 \text{ r.j.}$ tokom trajanja simulacije. Upravljanje tokom reaktivne snage se odvija nezavisno između naizmjeničnih sistema i pretvarača u sklopu HVDC sistema.

6. ZAKLJUČAK

Ovaj rad daje osvrt na visokonaponske jednosmjerne prenosne sisteme sa strane sa koje se oni nalaze u prednosti u odnosu na klasične, naizmjenične prenosne sisteme. Kao najvažnije prednosti HVDC sistema se mogu navesti jednostavno upravljanje tokovima aktivne i reaktivne snage, pružanje ekonomski isplativih i prostorno prihvatljivih rješenja problema vezanih za reaktivnu snagu, održavanje naponske stabilnosti i

povezivanje nesinhro-nizovanih sistema. Zahvaljujući navedenim prednostima, potpuno je shvatljiv nagli razvoj i porast broja izgrađenih HVDC prenosnih sistema u 21. vijeku. Istorijski gledano, tri decenije stariji, pretvarači zasnovani na tiristorima su jedino po kapacitetu za prenos snaga ostali nenadmašeni u odnosu na pretvarače zasnovane na brzim IGBT-ovima. Važan zaključak je bio da se upravljanje takvim sistemom sprovodi tako što jedan pretvarač upravlja tokom aktivne snage, a drugi upravlja naponom jednosmjernog međukola. Upravljanje tokom reaktivne snage se postiže nezavisno kod oba pretvarača. Postupak upravljanja se vrši u dq sistemu, pošto su u njemu veličine u ustaljenom režimu jednosmjerne. Podatak o trenutnom položaju rotirajućeg dq sistema daje PLL. Za upravljanje tokom aktivne i reaktivne snage korišćena je strujna petlja.

Rezultati simulacije su pokazali očekivane vrijednosti.

7. LITERATURA

- [1] Kim, C. K., Sood, V. K., Jang, G. S., Lim, S. J., Lee, S. J.: „HVDC Transmission: Power Conversion Applications in Power Systems“, John Wiley & Sons (Asia), Clementi Loop, Singapore, 2009.
- [2] Arrillaga, J., Liuand, Y. H., Watson, N. R.: „Flexible Power Transmission: The HVDC Options“, John Wiley & Sons, Chichester, West Sussex, England, 2007.
- [3] Ostojić, R.: „Visokonaponski jednosmjerni prenos“, INFOTEH - Jahorina, Vol. 8, Ref. F-14, p. 889-893, Jahorina, BiH, 2009.
- [4] Acha, E., Agelidis, V., Anaya-Lara, O.: „Power Electronic Control in Electrical Systems“, Newnes Books, Oxford, England, 2002.
- [5] Gupta, K. K., Bhatnagar, P.: „Multilevel Inverters: Conventional and Emerging Topologies and Their Control“, Academic Press, Cambridge, Massachusetts, USA, 2017.
- [6] Yazdani, A., Iravani, R.: „Voltage-Sourced Converters in Power Systems: Modeling, Control, and Applications“, John Wiley & Sons, Hoboken, New Jersey, USA, 2010.

Kratka biografija:



Aleksandar Kokotović rođen je u Trebinju 1994. godine. Osnovnu i srednju školu završio je u Bileći. Osnovne akademske studije je započeo 2013. godine na Fakultetu tehničkih nauka, na studijskom programu Energetika, elektronika i telekomunikacije. Diplomski rad je odbranio u septembru 2017. godine. Iste godine je upisao Master akademske studije, smjer Energetska elektronika i električne mašine. Master rad je odbranio u oktobru 2018. godine.



Marko Vekić je docent na Fakultetu tehničkih nauka u Novom Sadu, na Katedri za Energetsku elektroniku i pretvarače. Oblast interesovanja su energetska elektronika u prenosnim i distributivnim mrežama i kvalitet električne energije.

**ANALIZA PERFORMANSI ALGORITAMA UČENJA USLOVLJAVANJEM U OKVIRU
STARCRAFT 2 OKRUŽENJA****ANALYZING THE PERFORMANCES OF DEEP REINFORCEMENT LEARNING
ALGORITHMS IN THE STARCRAFT 2 ENVIRONMENT**

Saša Lalić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj –Ovaj rad se bavi analizom performansi algoritama učenja uslovljavanjem pri rešavanju problema iz kompjuterske igre Starcraft 2. Algoritmi koji su implementirani i poređeni su A3C i Deep-Q Learning. Za svaki algoritam ispitan je uticaj različitih parametara obuke, kao što su broj preskočenih koraka agenta, i stopa učenja neuronske mreže. Pokazalo se da oba algoritma reaguju isto na promene parametara, i da u problemima koji ne zahtevaju česte akcije da bi dostiglo optimalno rešenje, preskakanje većeg broja akcija ubrzava obuku algoritma, i dovodi do boljeg rešenja u istom vremenskom periodu za obuku. Dok smanjenje stope učenja dovodi do lošijeg rešenja u svim slučajevima. Oba algoritma su postigli rezultate u problemima upravljanja jedinicama, ali nisu postigli značajne rezultate u izgradnji baze.

Abstract – This paper studies the performance of deep reinforcement learning algorithms in solving a subset of problems in the Starcraft 2 environment. Algorithms that were studied were: A3C and Deep-Q Learning. Each algorithm was tested with a different set of training parameters, such as the number of skipped agent steps, and the neural network learning rate. Both algorithms perform similarly in accordance to parameter changes, when dealing with the problems that do not require a great number of actions to achieve an optimal solution. Consequently parameters values that skip greater number of actions lead to better results given the same training time. Reducing the learning rate leads to a decrease of performances for both algorithms in all of the problems. Both algorithms have achieved satisfiable results in problems that mostly involve management of units. However the results were quite low tasks that included the construction of a base.

Ključne reči: Starcraft 2, učenje uslovljavanjem, duboko učenje, A3C, Deep-Q learning

1. UVOD

Oblast mašinskog učenja se bavi razvijanjem algoritama i tehnika rešavanja problema koji nisu primereni klasičnim tehnikama, i koji rešavaju kompleksne probleme koji su nekad mogli rešiti samo ljudi. Jedna oblast koja se dobro pokazala je učenje uslovljavanjem (*reinforcement learning*).

NAPOMENA:

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

Reinforcement learning se zasniva na interakciji sa pravim ili simuliranim okruženjem i bavi se algoritmima koji direktno interaguju sa okruženjima.

Reinforcement learning agent interaguje sa svojim okruženjem u diskretnim vremenskim koracima. U svakom trenutku t agent dobija opažanje (opis stanja okruženja) nakon čega on bira akciju iz skupa trenutno dostupnih akcija koju prosleđuje okruženju. Okruženje onda prelazi u sledeće stanje. Cilj agenta je da skupi što više nagrade može. Nagrada predstavlja meru uspeha algoritma. Moguće je uvesti kaznu, koja je mera neuspeha agenta. Glavne razlika u odnosu na nadgledano učenje su to što *reinforcement learning* ne zahteva postojanje tačnih parova ulaznih i izlaznih vrednosti, i nije potrebna eksplicitna ispravka neoptimalnih akcija. Umesto toga je fokus na performansama, što uključuje nalaženje ravnoteže između istraživanja (skupljanja novih podataka) i iskorištavanja postojećih podataka. *Reinforcement learning* pruža rešenja u poljima robotike kao i u oblasti igranja video igara.

Ovaj rad predstavlja i poredi rezultate obuke algoritama *reinforcement learning*-a u rešavanju nekoliko problema iz igre Starcraft 2 [2]. Starcraft 2 je vojna naučno fantastična strategija u stvarnom vremenu koju razvija i objavljuje Blizzard Entertainment. Korišćeni algoritmi su Q-učenje (*Q-learning*) i Asinhroni prednost akter-kritičar (*Asynchronous Advantage Actor-Critic*, A3C). Oni predstavljaju najpoznatije algoritme u upotrebi za rešavanje problema iz domena kompjuterskih igara. Problemi nad kojima se algoritma obučavanju predstavljaju skupljanje resursa (*CollectMineralShards*), traženje neprijatelja (*FindAndDefeatZerglings*), borba sa neprijateljem (*DefeatRoaches*) i izgradnja baze (*BuildMarines*). Algoritmi su trenirani sa nekoliko različitih skupova parametara da bi se utvrdila podešavanja pod kojima algoritmi postižu najbolje rezultate za dati problem. Najbolji rezultati su postignuti od strane A3C algoritma, dok je najbolji skup parametara za oba algoritma bio povećanje `step_mul` (vrednost koja određuje koliko koraka agent preskače pri interakciji sa okruženjem) vrednosti što je dovelo do brže obuke u slučaju skupljanja resursa i traženja neprijatelja, a lošije u slučaju borbe. Nijedan algoritam nije uspeo da postigne značajne rezultate u izgradnji baze u vremenu za obuku koje smo mi imali.

Preostali deo rada organizovan je na sledeći način. U narednoj sekciji naveden je uporedni prikaz prethodnih rešenja i vladajućih stavova na temu sentiment analize. Treća sekcija posvećena je definiciji problema i metodologiji za njegovo rešavanje. Četvrta sekcija se bavi

evaluacijom dobijenih rezultata, a peta sumarizacijom rada i pravcima daljeg razvoja.

2. PRETHODNA REŠENJA

Reinforcement learning primenjen na rešavanje problema u video igrama je tema mnogih radova. Rešavanjem problema *Starcraft 2* se bavilo dosta radova, pokrivajući različite probleme, algoritme, tehnike obuke. U [3] se uvodi SC2LE okruženje, okruženje za *reinforcement learning* bazirano na igri *Starcraft 2*. Pored specifikacije *Starcraft 2* domena, pružaju *Python* bazirani interfejs za komunikaciju sa okruženjem igre, niz mini igara koji se fokusiraju na različite elemente *Starcraft 2 gameplay*. *Replay*¹ podatke pravih ekspertnih igrača kao i bazne rezultate za neuronske mreže trenirane na osnovu ovih podataka da predvide rezultate igara i akcija igrača. Na kraju ovaj rad predstavlja bazne rezultate *reinforcement learning* agenata nad *Starcraft 2* domenom. Rad [3] predstavlja osnovu od koje naš rad počinje. U njemu su zadati i kreirani radni okviri i biblioteke koje omogućavaju i olakšavaju *reinforcement learning* nad problemima *Starcraft 2*. U [1] se opisuju *pysc2* okruženje za učenje i procenjuju delotvornost A3C algoritma u rešavanju nekoliko mini mapa. Fokus rada je poređenje različitih arhitektura neuronskih mreža A3C algoritma. Pored kraće diskusije individualnih problema koji su rešavani, ovaj rad obrađuje i rezultate transfer learning-a (uzimanje težina neuronske mreže nakon obuke nad jednim problemom, i korišćenjem istih za početak obuke mreže nad drugim). Rezultati ovog rada su mnogo pogodniji poređenju sa našim od [3] zato što su za treniranje algoritama autori imali značajno manji broj iteracija.

U [4] je izvršena procena primerenosti *reinforcement learning* algoritama za rešavanje zadatka mikro menadžmenta borbenih jedinica (pozicioniranje i upotreba posebnih sposobnosti jedinica za svaku ponaosob umesto grupno) u komercijalnoj *real time strategy* (RTS) igri *Starcraft: Broodwar* (preteča *Starcraft 2*). Primenjene tehnike su varijacije *Q-learning* (detaljnije u 3.5.1) i Sarsa (λ) algoritama, jednostavne jednokoračne verzije kao i sofisticiranijih verzija koje koriste *eligibility trace*² da nadoknade problem odgođenih nagrada³. Cilj je dizajn agenta koji može da uči nenadgledan u kompleksnom okruženju, i koji bi eventualno preuzeo zadatke koje je dotad rešavala neadaptivna deterministička igrina umetna inteligencija (*Artificial intelligence*, AI). U [6] se obrađuju moderni i klasični pristupi *reinforcement learning*. Od većeg značaja nama je njihova implementacija A2C (preteča A3C algoritma koja koristi samo jednog agenta za obuku), opis i diskusija

¹ Replay je interaktivan snimak igre, koji sadrži podatke o stanju jedinica i drugih elemenata igre, a ne samo sliku i zvuk.

² Privremeni zapis uslova koje dovode do nekog događaja, *eligibility trace* obeležava parametre koji su asocirani sa događajem kao podložne promenama tokom učenja. Na ovaj način *eligibility trace* premošćava jaz između događaja i informacija za učenje.

³ nagrada koje se dešavaju dalje u budućnosti, problem je dati akcijama koje dovode do njih odgovarajuću težinu, da bi one bile zanemarene i ispraćene ako dovode do bolje nagrade od drugih akcija

arhitektura mreže i korišćenog algoritma koja upotpunjuje i potvrđuje rezultate iz [3].

3. METODOLOGIJA I ALATI

U ovoj sekciji se nalazi pregled svih alata, okruženja, biblioteka i algoritama korištenih u okviru ovog rada.

3.1. Starcraft 2

Starcraft 2 je naučno fantastična, vojna, *real-time* strateška video igra koju razvija i izdaje *Blizzard Entertainment*. Izašla je u Julu 2010 za *Windows* i *Mac OS X* operativne sisteme.. Verzija korišćena za ovaj rad je 4.2.4.

3.2. PySC2 – StarCraft 2 Learning Environment

PySC2 [5] je *Python* komponenta za *Starcraft 2 Learning Environment* koju je napravio *DeepMind*. Otkriva *Blizzard Entertainment Starcraft 2 Machine Learning API* kao *Python Reinforcement Learning* okruženje. *PySC2* pruža interfejs *reinforcement learning* agentima za interakciju sa *Starcraft 2*, za dobijanje posmatranja (podataka o stanju igre, položaj jedinica, dostupne akcije, itd.) i slanje akcija.

3.3. CUDA

NVIDIA CUDA biblioteka [8] za duboke neuronske mreže je GPU ubrzana biblioteka primitiva za duboke neuronske mreže. Pruža veoma optimizovane implementacije standardnih rutina kao što su konvolucija unapred i unazad, *pooling*, normalizacija i aktivacione funkcije. *CUDA* ubrzava često korištene radne okvire za duboko učenje uključujući *Caffe2*, *MATLAB*, *Microsoft Cognitive Toolkit*, *Tensorflow* i *Theano*.

3.4. Tensorflow

Tensorflow [7] je *open source* simbolička matematička biblioteka korištena za programiranje toka podataka i aplikacija mašinskog učenja kao što su neuronske mreže. Pruža API-je za *python*, *C++*, *Haskell*, *Java*, *Go* i *Rust*. Takođe je moguće koristiti biblioteke kao što su *Keras* koje bi pojednostavile kreiranje algoritama ili dodale dodatni sloj apstrakcije radi lakše promene biblioteka.

3.5. Reinforcement learning

Reinforcement learning je oblast mašinskog učenja inspirasana psihologijom ponašanja, koja se bavi načinom na koji softverski agenti treba da preduzimaju akcije u okruženju da bi maksimizirali neki pojam kumulativne nagrade. *Reinforcement learning* se razlikuje od standardnog nadgledanog učenja u tome da nije potrebno da postoje ispravni ulaz/izlaz parovi, i neoptimalne akcije ne moraju biti eksplicitno ispravljene. Fokus je umesto toga na performansama, što zahteva nalaženje ravnoteže između istraživanja (neistraženog prostora) i iskorištavanja (trenutnog znanja) [9]. Agent uči interagujući sa svojim okruženjem diskretnim vremenskim koracima. U svakom trenutku t , agent prima opažanje, koje tipično uključuje nagradu. Onda agent bira akciju iz skupa dostupnih akcija, koju prosleđuje okruženju. Okruženje onda prelazi u sledeće stanje i određuje nagradu asociranu sa prelazom između ovih stanja. Cilj agenta u *reinforcement learning* je da skupi što veću nagradu može. Značajni elementi *reinforcement learning* su upotreba uzoraka zarad optimizacije performansi, i upotreba funkcija aproksimacije da bi se izborili sa velikim okruženjima. Zahvaljujući njima se

reinforcement learning može koristiti u situacijama gde model okruženja nije poznat ali postoji analitičko rešenje ili ako postoji samo simulacija modela okruženja, i interakcija je jedini izvor podataka o okruženju.

3.5.1. Q-Learning

Q-learning je tehnika *reinforcement learning* korištena u mašinskom učenju. Cilj je naučiti politiku koja govori agentu koju akciju da preduzme pod kojim okolnostima. Ne zahteva model okruženja i može rešiti probleme sa stohastičnim prelazima i nagradama bez izmena. *Q-learning* skladišti svoje podatke u tabele, pristup koji loše funkcioniše sa većim brojem stanja ili akcija. *Q-learning* može biti kombinovan sa aproksimacijom funkcija, što omogućava primenu algoritma na većim problemima čak i ako je prostor stanja kontinualan. Jedno rešenje je koristiti veštačku neuronsku mrežu kao aproksimator funkcije. Algoritam je funkcija koja računa kvalitet kombinacije stanja i akcije. Pre nego što učenje počne *Q* se inicijalizuje na arbitrarnu fiksnu vrednost. Onda u svakom vremenu *t* agent bira akciju, posmatra nagradu, ulazi u sledeće stanje i koriguje *Q*. Jezgro algoritma je iteracija vrednosti, koja koristi težinski prosek (*weighted average*) stare vrednosti i nove informacije. Nagrada date akcije može biti pomnožena faktorom popusta, koji određuje značaj budućih nagrada.

Deep Q-learning je varijanta *Q-learning* koji koristi duboku konvolucionu neuronsku mrežu sa slojevima konvolucionih filtera da oponaša efekte receptivnih polja. *Reinforcement learning* je nestabilan kada je nelinearni aproksimator funkcije kao neuronska mreža korišćen da predstavlja *Q*. Ova nestabilnost dolazi iz korelacije koja se nalazi u sekvenci opažanja, činjenica da male izmene *Q* mogu značajno promeniti politiku i distribuciju podataka, i korelacije između *Q* i ciljnih vrednosti. Implementacija korištena u ovom radu je *Deep Q-learning*, koji u svakom koraku za učenje uzima nasumično sortirani podskup (*batch*) svih prelaza stanja/akcija iz prošle iteracije problema.

3.5.2. Asynchronous advantage actor-critic (A3C)

A3C algoritam [10] je izdao Google DeepMind. Na velikom broju *deep reinforcement learning* zadataka A3C je brži, jednostavniji i bolji nego DQN. Takođe radi na kontinualnim kao i na diskretnim prostorima akcija. Zbog ovoga je postao dominantan algoritam u rešavanju problema sa kompleksnim stanjem i akcionim prostorom.

Actor-critic je kombinacija beneficija iterativnih metoda baziranim na politici i vrednosnih iterativnih metoda kao što je *Q-learning*. A3C će proceniti funkciju vrednosti *V* (koliko je dobro neko stanje) i politiku (skup izlaza verovatnoće akcija). Ove procene vrše potpuno povezani slojevi na vrhu neuronske mreže. Bitno je da agent koristi procenu vrednosti (kritičar) da koriguje politiku (akter) inteligentnije nego tradicionalne metode gradijenta politike⁴.

Advantage je način računanja pravila korigovanja politike, uzimajući nagradu nakon uračunatih faktora popusta kao procenu *Q* vrednosti, i oduzimajući od nje rezultat vrednosne funkcije *V*.

Asynchronous se odnosi na mogućnost paralelizacije agenta. A3C algoritam poseduje jednu globalnu neuronsku mrežu, i skup proizvoljne veličine agenata i njihovih okruženja. Svaki agent počinje sa istim parametrima kao i globalna neuronska mreža. On onda interaguje sa svojom kopijom okruženja i skuplja iskustvo. Kada skupi dovoljno iskustva, npr. dođe do kraja okruženja, njegovo iskustvo se koristi za računanje gubitaka vrednosti i politike. Iz njih se izvlače gradijenti za korigovanje parametara mreže i njima se koriguje globalna mreža. Na kraju svake ovakve iteracije agent postavlja vrednosti svoje mreže na vrednosti globalne mreže i nastavlja dalje. Implementacija korištena u ovom radu je ovde opisani A3C algoritam.

4. EKSPERIMENTALNA POSTAVKA

U ovoj sekciji predstavljamo neke od problema u Starcraft 2 i evaluiramo algoritme nad njima. Predstavljamo eksperimentalnu postavku za evaluaciju algoritama. Eksperimentalna postavka se sastoji od problema i primenjenih algoritama. Pod problemima smatraju se nekoliko konkretnih zadataka u igri Starcraft 2. Primenjeni algoritmi su *Q-learning* i A3C, reinforcement learning algoritmi koji su obučavani nad ovim problemima.

4.1. Problemi

Problemi su predstavljeni kao nekoliko malih mapa u Starcraft 2. Izabrali smo sledeće probleme da bi smo pokrili specifične probleme iz domena Starcraft 2.

CollectMineralShards – agent počinje sa dva marinca (vrsta jedinice u igri) i mora ih selektovati i pomerati po mapi da bi skupio minerale. Svaki mineral vredi jedan poen, raspoređeni su nasumično po mapi, i ako agent uspe da skupi sve minerale pre nego što mu vreme istekne mapa se ponovo puni novim mineralima. Agent ima 2 minuta za jednu iteraciju ove mape.

FindAndDefeatZerglings – agent počinje sa 3 marinca i mora istražiti mapu da nađe i pobedi individualne zerglinge. Ovo zahteva pomeranje kamere i efikasno istraživanje. Svaki zergling ubijen vredi 1 poen, svaki marinac izgubljen vredi -1 poen. Agent ima 2.5 minuta za jednu iteraciju ove mape.

DefeatRoaches – agent počinje sa 9 marinaca i mora pobediti 4 roach-a. Svaki put kad pobedi sve roach-eve dobije još 5 marinaca pojačanja i još 4 roach-a kao protivnike. Svaki roach vredi 10 poena a svaki izgubljeni marinac -1. Agent ima 3 minuta za jednu iteraciju ove mape.

BuildMarines – agent počinje sa ograničenom bazom, i cilj mu je građenje marinaca. Svaki marinac kojeg napravi vredi jedan poen. Dok je prostor akcija njemu ograničen isključivo na akcije potrebne da bi on napravio marince u pitanju je još uvek kompleksan problem. Agent mora skupljati resurse, izgraditi *supply depo*, baraku i onda istrenirati marince tim redosledom. Agent ima 15 minuta za jednu iteraciju ove mape.

4.2 Rezultati

Na osnovu rezultata možemo izvući nekoliko zaključaka. U svim problemima A3C algoritam je značajno bolji od *Q-learning* algoritma. U svim problemima sem u slučaju *DefeatRoaches* problema je treći skup parametara

⁴ Gradijent politike je tip tehnike učenja uslovljavanjem koja optimizuje politike koristeći *gradient descent*.

najbolji. Veći broj iteracija dovodi do boljih rezultata algoritma, ako problem ne zahteva brze reakcije da bi bio optimalno rešen. U uslovima gde su resursi za obuku algoritama ograničeni, stoga davanje algoritmu ređe priliku da izdaje akcije ubrzava obuku jer stiže da obradi više iteracija. Drugi skup parametara se pokazao loše u svim slučajevima, iz čega se vidi da su ovi problemi isuviše kompleksni da bi postigli veliki napretke u učenju u početnim iteracijama obuke, na koje drugi skup parametara stavlja fokus. *Q-learning* se pokazao kao lošiji algoritam zbog manje paralelizacije i samim tim manje brzine obuke, ali i zbog toga što *Q-learning* ne računa gubitak politike kao A3C već se samo oslanja na *advantage* pri računanju gubitka funkcije. Naši rezultati su značajno gori od rezultata sličnih istraživanja zbog ograničenja hardvera i vremena, što je smanjilo broj iteracija obuke algoritama, kao količinu različitih skupova parametara obuke koju možemo zadati.

Ljudski rezultati su značajno bolji ne samo od naše implementacije algoritama, već i od implementacija u svim drugim dosadašnjim radovima. Uzrok ovoga je velika kompleksnost *Starcraft 2*, koja još uvek predstavlja izazov sadašnjim *reinforcement learning* algoritmima.

Tabela 1. Usporedni rezultati evaluacije svih algoritama

Tip problema/Algoritam	A3C	Deep Q-learning
CollectMinerals	34.63	0.63
CollectMinerals (manji learning rate)	28.68	0
CollectMinerals (manja učestalost akcija)	36.91	2.98
FindAndDefeatZerglings	7.45	0
FindAndDefeatZerglings (manji lr)	2.58	-1.72
FindAndDefeatZerglings (manja učestalost akcija)	8.44	6.1
DefeatZerglings	37.37	12.23
DefeatZerglings (manji lr)	9.99	0
DefeatZerglings (manja učestalost akcija)	6.99	4.09
BuildMarines	0	0
BuildMarines (manji lr)	0	0
BuildMarines (manja učestalost akcija)	0	0

5. ZAKLJUČAK

Problem koji je u ovom radu rešavan je implementacija i analiza performansi A3C i *Q-learning* algoritama u rešavanju problema iz domena *Starcraft 2*.

Ovi algoritmi su implementirani uz oslonac na *Tensorflow* i *pysc2* biblioteke, napisane u programskom jeziku *python*. Oba algoritma su implementirana sa sličnom arhitekturom neuronske mreže, sa dva konvolucionarna sloja i potpuno povezanim slojem za izlaze neprostorne akcije i vrednosti stanja, i *softmax* slojem za izlaz prostorne akcije. *Q-learning* algoritam računanje *advantage* vrši u izlaznim slojevima neuronske mreže, zbog čega njegovi izlazni slojevi imaju podelu na dva odvojena sloja, dok A3C algoritam *advantage* računa u koraku ažuriranja algoritma, pa je njegova arhitektura jednostavnija. A3C algoritam podržava obuku koristeći više agenata istovremeno, tako što vrši ažuriranje neuronske mreže posle jedne iteracije nad problemom svakog agenta, dok *Q-*

learning radi samo sa jednim agentom. *Q-learning* sadrži dve identične neuronske mreže, od kojih jednu koristi samo za generisanje ciljnih vrednosti u obuci, i koju izjednačava sa glavnom neuronskom mrežom svakih 50 koraka obuke da bi postigao stabilniju i konzistentniju obuku. Svaki algoritam je treniran osam sati u realnom vremenu nad 4 problema: *CollectMinerals*, *DefeatRoaches* i *BuildMarines*, *FindAndDefeatZergling*. Svaki algoritam je takođe treniran sa 3 skupa različitih parametara za obuku za svaki problem. Rezultati za svaki algoritam su dobijeni uzimajući prosečan rezultat 100 iteracija nad problemom po završetku obuke.

Pokazali smo da su svi izabrani problemi sa izuzetkom *BuildMarines* problemi koje i *Q-learning* i A3C algoritam mogu savladati, sa dovoljno vremena obuke. Naša arhitektura algoritama se loše pokazala na *BuildMarines*, ali zbog dužine i kompleksnosti mape, teško je utvrditi da li je problem manjak vremena obuke, sa obzirom da su u *pysc2* radu tek posle nekoliko miliona koraka obuke uspeali postići neke rezultate, nešto što prevazilazi resurse nama na raspolaganju. Dalji pravci istraživanja uključuju modifikaciju algoritama specifično za svaki problem, isprobavanje većeg broja parametara obuke, i rešavanje drugih problema u okviru *Starcraft 2*. Mogući pravac takođe predstavlja primena ovih algoritama na druge kompleksne igre, kao što je *Dota 2*.

6. LITERATURA

- [1] A. Basel, P. G. Keerthana "Asynchronous Advantage Actor-Critic Agent for Starcraft II", 22.7.2018,
- [2] Starcraft 2 Windows PC version, Blizzard Entertainment, 2010.
- [3] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, "StarCraft II: A New Challenge for Reinforcement Learning", 16.08.2017.
- [4] S. Wender, I. Watson, "Applying Reinforcement Learning to Small Scale Combat in the Real-Time Strategy Game StarCraft:Broodwar", 2012
- [5] <https://github.com/deepmind/pysc2>
- [6] R. Ring, "Replicating DeepMind StarCraft II Reinforcement Learning Benchmark with Actor-Critic Methods", 2018
- [7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, "TensorFlow: Large-scale machine learning on heterogeneous systems", 2015, Software available from tensorflow.org.
- [8] NVIDIA cuDNN, <https://developer.nvidia.com/cudnn>
- [9] L. Kaelbling, M. Littman, A. Moore, "Reinforcement Learning: A Survey", Journal of Artificial Intelligence Research. 4: 237–285, Archived from the original on 20.11.2001. (1996).
- [10] A. Juliani, "Asynchronous Actor-Critic Agents (A3C)", 17.12.2016.

KRATKA BIOGRAFIJA



Saša Lalić je rođen 12.08.1994. godine u Subotici. Osnovnu školu „Hunjadi janoš“ u Čantaviru završio je 2009. godine. Gimnaziju „Svetozar Marković“ u Subotici završio je 2013. god. Iste godine upisao se na Fakultet tehničkih nauka, odsek Računarstvo i automatika. Osnovne studije je završio 2017. godine, nakon čega upisuje master akademske studije na Fakultetu tehničkih nauka, smer Računarstvo i automatika. Položio je sve ispite propisane planom i programom.

KOMPRESIJA MULTISPEKTRALNIH SATELITSKIH SNIMAKA KORIŠĆENJEM HEVC STANDARDA**COMPRESSION OF MULTISPECTRAL SATELLITE IMAGES USING HEVC STANDARD**Marko Adamović, *Fakultet tehničkih nauka, Novi Sad***OBLAST - ELEKTROTEHNIKA I RAČUNARSTVO****2. MULTISPEKTRALNE SLIKE**

Kratak sadržaj – Sateliti za multispektralno snimanje Zemlje, kao što su američki Landsat 8 i evropski Sentinel-2, svakodnevno prave veliku količinu fotografija visoke rezolucije. Njihovo skladištenje zahteva veliki digitalni prostor, što uvećava cenu opreme. Landsat snima slike u nekomprimovanom formatu, dok Sentinel koristi JPEG2000 kompresiju. Cilj ovog rada je da uporedi High Efficiency Video Coding (HEVC) i JPEG2000 standarde za kompresiju ovih fotografija i utvrdi da li bi HEVC bio bolje rešenje.

Ključne reči: HEVC, JPEG2000, kompresija, Landsat, Sentinel, multispektralne fotografije

Abstract – Satellites for multispectral imaging of Earth, such as American Landsat 8 and European Sentinel-2, daily make huge amounts of high resolution images. Their storage requires lots of digital space, which increases equipment costs. Landsat saves images in uncompressed format, while Sentinel uses JPEG2000 compression. Goal of this paper is to compare High Efficiency Video Coding (HEVC) and JPEG2000 standards for compression of these images, and determine if HEVC would be a better solution.

Keywords: HEVC, JPEG2000, compression, Landsat, Sentinel, multispectral imaging

1. UVOD

Skladištenje satelitskih multispektralnih fotografija visoke rezolucije može biti veoma skupo sa stanovišta potrebnog digitalnog prostora i cene opreme. Američki Landsat 8 dnevno snimi do 740 fotografija, od kojih se svaka sastoji od 11 kanala veličine oko 100 MB. Dva satelita evropskog Sentinel-2 programa dnevno objave preko 4 TB fotografija i snime celu površinu Zemlje svakih pet dana. Landsat fotografije se čuvaju u nekomprimovanom GeoTIFF formatu, a Sentinel koristi JPEG2000 kompresiju. U oba slučaja se radi o 16-bitnim slikama.

High Efficiency Video Coding (HEVC), poznat i kao H.265, je moderan standard za video kompresiju i u ovom radu ćemo se baviti ispitivanjem validnosti korišćenja ovog standarda u svrhe kompresije satelitskih multispektralnih snimaka.

NAPOMENA:

Ovaj rad je proistekao iz master rada čiji mentori su bili: dr Branko Brkljač, doc. i mast. inž. Miloš Radošavljević, istr. sar.

Multispektralne fotografije predstavljaju skup fotografija koje su uslikane u različitim opsezima talasnih dužina u elektromagnetnom spektru. Obuhvataju opsege svetlosti u vidljivom i one van vidljivog spektra (različite opsege infracrvene i ultraljubičaste svetlosti).

To omogućava prikupljanje podataka nedostupnih korišćenjem konvencionalnih metoda fotografisanja. Zamenom jednog ili više kanala iz vidljivog spektra (crveni, zeleni i plavi) sa drugim kanalima dobijaju se lažno obojene (false-color) slike, koje, zbog boljeg kontrasta omogućuju lakše razaznavanje između npr. zelenila i vode ili kamena i betona. Ovakve fotografije se koriste u vojne svrhe za detekciju nagaznih mina i balističkih raketa, pri snimanju dokumenata i umetničkih dela, u medicini i biologiji za ispitivanje tkiva i opekotina. Koriste se i za snimanje Zemlje iz svemira pomoću satelita, što je poljoprivredi i biologiji omogućilo praćenje vegetacije i useva, kao i promene plodnosti zemljišta.

Geolozi mogu lakše da proučavaju mineralni sastav zemlje na velikom području. Posmatranje zagađenosti i vlažnosti vazduha i prisustvo aerosoli je od velikog značaja za ekologiju. Opservacija okeana, glečera i vulkana je takođe postala lakša, a službe za brzo reagovanje tokom katastrofa takođe imaju benefit od ove tehnologije [1].

3. SATELITI SENTINEL-2 I LANDSAT 8

Sentinel-2 je misija Evropske svemirske agencije (ESA) za snimanje multispektralnih slika širokih regiona, visoke rezolucije i sastoji se od dva satelita. Nalaze se u polarnoj orbiti sa fazom od 180°, što znači da su pozicionirani jedan naspram drugog u odnosu na Zemlju. Nadleću iste predele u okolini ekvatora svakih 5 dana, a oblete Zemlju 14.3 puta dnevno. Poseduju optičke instrumente za snimanje 13 spektralnih kanala sa rezolucijama od 10, 20 i 60 m i širinom područja snimanja od 290 km. Program Sentinel-2 je prvenstveno pružanje informacija vezanih za poljoprivredu i šumarstvo, a služi i za praćenje glečera, zagađenosti jezera i priobalnih voda, promena u pokrivenosti zemljom, snegom, itd. Takođe, fotografije poplava, erupcija vulkana, požara i drugih nepogoda pomažu pri koordinaciji spasilačkih timova i prve pomoći u pogođenim regionima.

Landsat 8 je nastao kao kolaboracija između agencija NASA i United States Geological Survey (USGS), a lansiran je 2013. godine sa ciljem da nastavi 40 godina

dugu misiju opservacije Zemlje is svemira i daje rezultate uporedive sa onima dobijenim od prethodnih Landsat sistema. Posедуje dva optička instrumenta: Operational Light Imager (OLI), koji ima 9 kratkotalasnih kanala i Thermal InfraRed Sensor (TIRS), sa dva dugotalasna kanala. Senzori mogu da rade neovisno jedan od drugog iako scene snimaju istovremeno. Deset kanala ima prostornu rezoluciju od 30 m, a jedan od 15 m, dok je širina snimanja oko 185 km. Satelit se nalazi u blisku polarnoj, heliosinhronoj orbiti, što znači da iste regione preleti uvek tokom istog lokalnog vremena. Na ovaj način fotografije istih regiona uvek imaju identično osvetljenje i mogu se lako porediti. Landsat fotografije se koriste u iste svrhe kao i one dobijene od Sentinel programa.

4. JPEG2000 i HEVC

JPEG2000 je nastao 2000. godine kao naslednik JPEG standarda iz potrebe za efikasnim, fleksibilnim i interaktivnim prikazivanjem slika. Jedan od glavnih zahteva standarda je bio da, bez obzira na implementaciju, od najjednostavnije do najsloženije, uvek može da ima efikasnu interakciju sa istom, efikasno kompresovanom slikom, bez obzira na njenu rezoluciju, bitsku dubinu, bit rate, ili broj komponenti [2].

Kodni tokovi u ovom standardu su regioni od interesa koji omogućavaju pristupanje različitim delovima slike sa različitim stepenima kvaliteta. Ovakva skalabilnost daje mogućnost da se slika jednom kompresuje, a da može da se dekompresuje na mnogo načina, sa stepenima kvaliteta koji trebaju korisniku. Primer bi bio upravo pri pregledanju satelitskih snimaka koji su mnogo veće rezolucije od klasičnih monitora. Puna slika smanjenog kvaliteta može da se očitava relativno brzo, a dok korisnik zumira sliku da pregleda region koji ga interesuje, dekodir taj region očitava sa većim kvalitetom. Ovo se zove skalabilnost odnosa signal-šum ili skalabilnost distorzije [2].

Slika se prvo transformiše iz RGB palete boja u nepovratnu YCbCr, ili povratnu modifikovanu YUV paletu. Nakon toga se slika deli na blokove koji se transformišu i koduju nezavisno jedni od drugih. Mogu biti bilo koje veličine, ali svi moraju biti istih dimenzija, osim eventualno onih na dnu i na desnom obodu slike. Nad ovim blokovima se dalje vrši *diskretna wavelet transformacija* (DWT), a nakon nje skalarna kvantizacija nad koeficijentima, smanjujući broj bita potrebnih za njihovo predstavljanje na uštrb kvaliteta.

Na izlazu se dobija set celih brojeva koji se koduju bit po bit. Korak kvantizacije se može menjati u zavisnosti od željenog kvaliteta i stepena kompresije.

Istraživanja su pokazala [3] da kod kompresije sa gubicima, pri malim bit rate-ovima, u poređenju sa HEVC i H.264 kodekom podešenim da koriste samo I-frejmove, JPEG2000 daje bolje rezultate. Ova prednost se gubi rastom bite rate-a. Kod kompresije bez gubitaka, HEVC korišćenjem I-frejмова daje uporedive rezultate kao i JPEG2000.

HEVC - High Efficiency Video Coding standard je nastao kao zajednički projekat ITU-T Video Coding Experts Group (VCEG) i ISO/IEC Moving Picture Experts Group (MPEG) organizacija za standarde.

Razvijen je od 2010. godine, a, prva verzija standarda je objavljena 2013. godine. Rastom zahteva za video sadržajem veće rezolucije rasla je i potreba da se zameni njegov prethodnik, H.264/MPEG-4 AVC. HEVC je dizajniran sa ciljem da podrži sve aplikacije svog prethodnika, sa dodatnim fokusom na veću rezoluciju i korišćenje arhitektura za paralelno izračunavanje [4]. Pored toga, rađeno je i na efikasnosti kodovanja, otpornosti na greške i integraciji sistema za prenos.

Nakon primanja nizova bita, slika deli na blokove, a koder odlučuje kako će ti oni biti raspoređeni. Prva slika u video sekvenci, kao i svaka prva slika u parametrima određenim delovima sekvence, biće kodovana koristeći intra predikciju (tzv. I-frejм).

Kod intra predikcije, koder koristi samo informacije iz trenutnog frejма. Za ostale frejmove može da koristi i inter predikciju, kod koje se informacije dobijaju iz prethodno kodovanih frejмова. To se postiže pomoću podataka o pokretu koji se sastoje od izabrane referentne slike i vektora pokreta, koji se primenjuju na prediktovane semplove za svaki blok. Uz pomoć ovih predikcija moguće je odbaciti redundantne podatke iz slike. Nakon što se dobije signal razlike originalne slike i predikcije, nad njime se izvrši linearna prostorna transformacija, uglavnom dvodimenzionalna *diskretna kosinusna transformacija* (DCT). Transformisani koeficijenti se skaliraju i kvantizuju, čime se uklanjaju manje bitni podaci. Konačno, kvantizovani koeficijenti se entropijski koduju koristeći *kontekstno adaptivni, binarni aritmetički koder* (CABAC).

Dekoder će iz dobijenog niza bita generisati identičnu predikciju kao u koderu. Kvantizovani transformisani koeficijenti se rekonstruišu inverznim skaliranjem i inverznom transformacijom, kako bi se dobio rezidualni signal približno identičan originalnom. On se sabira sa predikcijom i rezultat može da se propusti kroz jedan ili dva filtera za korekciju izobličenja nastalih pri kvantizaciji i obradi slike baziranoj na blokovima. Rekonstruisani podaci se smeštaju u bafer, kako bi služili za predikciju ostalih frejмова, ako se koristi inter mod.

5. POSTUPAK TESTIRANJA

Kako bismo uporedili performanse HEVC i JPEG2000 kodeka, bilo je potrebno koristiti nekomprimovane slike u RAW formatu. Slike Landsat-a 8 već dolaze u nekomprimovanom GeoTIFF formatu i pogodne su za potrebe ovog rada. Preuzete su sa internet sajta *EarthExplorer* [5], a u paketu se nalazi 12 16-bitnih slika jednog regiona, po jedna za svaki od 11 kanala i jedna za kontrolu kvaliteta, kao i dva fajla sa meta informacijama koji nisu od značaja za potrebe ovog rada. Paketi su definisani datumom i vremenom kada je snimak napravljen. Test set koji smo izabrali predstavlja region Vojvodine u okolini Zrenjanina i ima oznaku: *LC08_L1TP_186028_20170414_20170501_01_T1*. Slika za kontrolu kvaliteta (BQA), je od velikog značaja za istraživače i uvek bi trebala da se kompresuje bez gubitka podataka (lossless). Zbog toga smo se opredelili da nju ne koristimo u ovom testu. Takođe, slika 8. kanala ima duplo veću rezoluciju (i veću veličinu fajla), pa smo odlučili da ni nju ne uvrstimo u test zbog pojednostavljenja postupka,

što neće značajno uticati na rezultate poređenja performansi kompresije. Treba dodatno obratiti pažnju da različiti paketi mogu imati slike različitih rezolucija, a test set na kojem smo mi radili ispitivanja ima rezoluciju od 7801x7911 piksela. Pošto slike dolaze u GeoTIFF formatu, bilo je potrebno iz njih ukloniti geografske podatke iz zaglavlja, jer ih HEVC ne podržava. Ovo smo postigli pravljenjem skripte u Python programskom jeziku i korišćenjem GDAL biblioteke koja sadrži alate za rad sa geografskim podacima. Skripta iz slika uklanja geografske podatke, a pošto HEVC na ulazu traži video fajl, sve slike smo takođe kombinovali u jedan binarni stream i snimali ih u yuv formatu. Test program za JPEG2000 otvara GeoTIFF uz upozorenje da fajlovi sadrže tagove koje ne može da očita, te da će oni biti ignorisani. Zbog toga za JPEG2000 nije bilo potrebno vršiti nikakvu predobradu podataka. Ipak smo, za potrebe poređenja rekonstruisanih slika sa originalnim, morali da napravimo i set originalnih slika konvertovanih u RAW format.

Kao test program za JPEG2000 kompresiju smo koristili *OpenJPEG* [6] referentni softver verzije 2.3.0. Postupak je bio da se svaka od 10 slika prvo kompresuje u JPEG2000 format sa određenim nivoom kvaliteta koji zadajemo jednim od ulaznih parametara i izračunamo veličinu dobijenog fajla. Sumiranjem veličina nekomprimovanih i deljenjem sumom veličina komprimovanih fajlova dobijamo formulu za odnos kompresije (1):

$$\text{odnos kompresije} = \frac{\sum \text{nekompresovani fajlovi}}{\sum \text{kompresovani fajlovi}} \quad (1)$$

Parametar za podešavanje željenog kvaliteta kompresije kod *OpenJPEG*-a predstavlja vršni odnos signal-šum (PSNR) u dB, a program će pokušati da podesi parametre tako da se na izlazu dobije slika čija je PSNR vrednost što približnija zadatoj. U našem testu smo koristili vrednosti u opsegu od 34 do 134 sa korakom 2, što je dalo rezultate uporedive sa onima dobijenim HEVC kompresijom. Ipak, pošto se ne dobije vrednost identična zadatoj, potrebno je uporediti kvalitet rekonstruisane slike sa originalnom. To smo postizali dekomprimovanjem slika u RAW format, koje smo poredili sa prethodno pripremljenim RAW verzijama originalnih slika, korišćenjem dodatnog programa koji smo napisali u jeziku C, čiji je izlaz PSNR razlika poređenih slika. Menjanje svih zadatih kvaliteta, poređenje pomenutim programom i izdavanje rezultata smo automatizovali dodatnim programom pisanim u programskom jeziku C++.

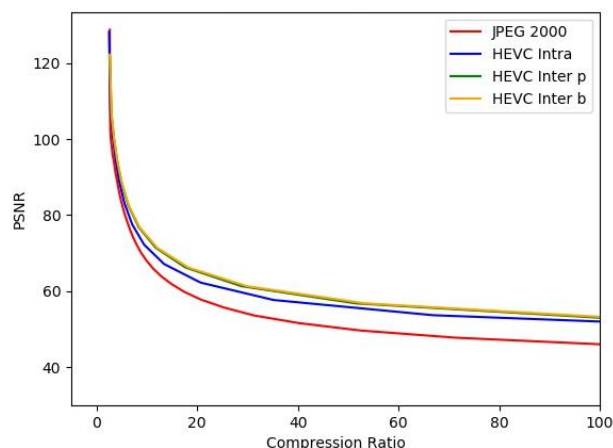
Referentni test softver za HEVC kodek koji smo koristili je *HM* (HEVC Test Model) [7], verzije *HM-15.0+RExt-8.1*. Za njega smo napravili tri konfiguraciona fajla koji sadrže opcije za korišćenje mod predikcije: jedan za intra (I) i dva za inter predikciju koja koristi P ili B frejmove. U njima se takođe nalaze i podaci o veličini blokova, kvantizacionim parametrima i ostalim opcijama koje će biti aktivirane pri kompresiji. HM takođe traži i dodatni konfiguracioni fajl u kojem se nalaze informacije o ulaznoj sekvenci (putanja, rezolucija, bitska dubina, format boja, itd.). I frejmovi ne zavise od prethodno dekodovanih frejmova, ali zato imaju najmanju kompresiju. Koriste se u intra predikciji, ali i kao prvi frejmovi u sekvenci kod inter predikcije. P frejmovi zavise od

prethodno dekodovanih, referentnih frejmova, ali samo onih koji su u snimku vremenski bili pre njih. Imaju bolju kompresiju od I frejmova i koriste se kod inter predikcije. Za razliku od njih, B frejmovi mogu da koriste kao referentne frejmove one koji su vremenski u snimku dolaziti nakon njih. Bitno je samo da su dekodovani pre trenutno posmatranog frejma, što HEVC podržava.

Kvalitet kompresije se u HEVC-u podešava zadavanjem kvantizacionog parametra QP, koji za 16-bitne slike može biti između -48 i 51. Mi smo ga menjali sa korakom 6. Nakon kompresije se dobija izveštaj iz kojeg smo uzimali vrednosti PSNR, veličinu dobijenog fajla, kao i vreme trajanja procesa. Pošto je HEVC napravljen kao video kodek, on i na izlazu daje jedan fajl koji sadrži svih 10 slika kao frejmove video snimka. Proces menjanja konfiguracionih fajlova, parametra QP i snimanja dobijenih rezultata smo takođe automatizovali pisanjem programa u jeziku C++. Ovako dobijene rezultate smo mogli dalje lako da poredimo i prikažemo u eksternim programima za iscrtavanje grafika, kao što su Matlab ili Python.

6. REZULTATI

Na sledećoj slici su prikazani uporedni rezultati kodovanja HEVC-om korišćenjem sva tri različita konfiguraciona fajla: jedan za intra predikciju, i po jedan za inter predikciju sa P i B frejmovima, kao i rezultati JPEG2000 kompresije. X osa predstavlja odnos kompresije koji smo dodatno ograničili na 100 radi boljeg pregleda krive (prave vrednosti prelaze 6000), dok Y osa predstavlja PSNR vrednost.



Slika 1. Poređenje rezultata kompresije

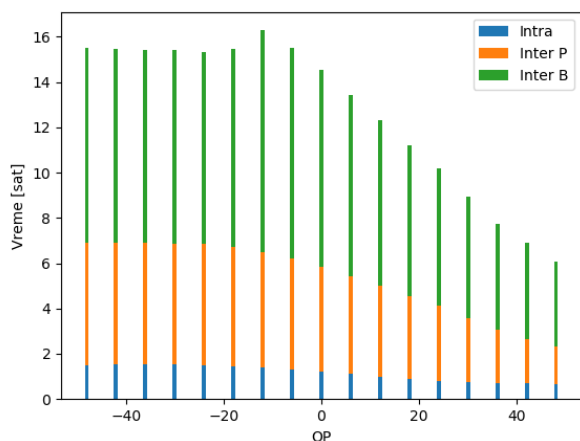
Sa slike 1. se vidi da HEVC daje primetno bolje rezultate u poređenju sa JPEG2000. Intra predikcija je nešto slabija u odnosu na inter, dok korišćenje P i B frejmova daje uporedive rezultate.

Tabela 1. Rezultati kompresije za PSNR od 70 dB i odnos kompresije 10

	PSNR u dB pri odnosu kompresije 10	Odnos kompresije pri PSNR 70 dB
JPEG2000	68	8.9
HEVC Intra	71.6	11.15
HEVC Inter P	74.2	13.34
HEVC Inter B	74.5	13.91

U tabeli 1. nalazi se poređenje vrednosti PSNR za zadati odnos kompresije 10, kao i poređenje odnosa kompresije za zadati PSNR 70.

HEVC ima mnogo veću kompleksnost, pa se i očekivalo da će dati bolje rezultate. Cena je, međutim, mnogo duže vreme kompresije. JPEG2000 kodeku je trebalo nekoliko sati da izvrši testove nad svih 10 kanala, za sve različite vrednosti zadatog kvaliteta (50 u našem testu). HEVC-u je, sa druge strane, za set od 10 slika i samo jedan zadati QP, u proseku trebalo oko 1.5 sat pri inter predikciji, između 3 i 7 sati za intra P, a između 7 i 16 sati za intra B predikciju, što se vidi na slici 2.



Slika 2. Vreme komprimovanja

Hardverskom implementacijom i odabirom jednog određenog skupa parametara bi proces kompresije HEVC-om trajao svega nekoliko minuta. Takođe se postavlja i pitanje kompromisa između performansi i kompleksnosti.

7. ZAKLJUČAK

Jedan od ciljeva ovog projekta je bio uspostavljanje modela testiranja, kao i paket funkcija i skripti koji će omogućiti dalja istraživanja u oblasti lossy kompresije nad satelitskim slikama. Rezultati pokazuju da HEVC može da bude bolja opcija od JPEG2000 za kompresiju ovakvih fotografija, jer postiže bolji kvalitet pri istim nivoima kompresije. Potrebno je vršiti dalja istraživanja kako bi se utvrdilo koji parametri daju zaista najbolje rezultate. Nakon toga bi se mogla napraviti hardverska implementacija kodeka koja bi mnogo ubrzala postupak kompresije i dekompresije. Postojeći softver za pregledanje i obradu satelitskih fotografija bi trebao da dobije odgovarajuća proširenja za podršku za rad sa ovako komprimovanim slikama.

8. LITERATURA

- [1] Optics & Photonics news, "Multispectral imaging moves into the mainstream", https://www.osa-opn.org/home/articles/volume_23/issue_4/features/multispectral_imaging_moves_into_the_mainstream/ (pristupljeno u septembru 2018.)
- [2] D.S. Taubman, M.W. Marcellin, "JPEG2000: Standard for interactive imaging", *Proceedings of the IEEE*, Vol. 90, pp. 1336-1357, August 2002.
- [3] Q. Cai, L. Song, G. Li, N. Ling, "Lossy and lossless intra coding performance evaluation: HEVC, H.264/AVC, JPEG 2000 and JPEG LS", *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, December 2012.
- [4] G. J. Sullivan, J.R. Ohm, W.J. Han, T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, no. 12, pp. 1649 - 1668.
- [5] <https://earthexplorer.usgs.gov/> (pristupljeno u septembru 2018.)
- [6] <http://www.openjpeg.org/> (pristupljeno u septembru 2018.)
- [7] <https://hevc.hhi.fraunhofer.de/> (pristupljeno u septembru 2018.)

Kratka biografija:



Marko Adamović rođen je u Novom Sadu 1986. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Telekomunikacije i obrada signala odbranio je 2018. god.
kontakt: adam.thor@gmail.com

**RAZVOJ APLIKACIJE ZA TRENUTNU RAZMENU PORUKA NA ANDROID
SISTEMIMA U PROGRAMSKOM JEZIKU KOTLIN****FRAMEWORK INSTANT CHAT MESSAGES APPLICATION IMPLEMENTATION FOR
ANDROID PLATFORM USING KOTLIN PROGRAMMING LANGUAGE**

Nikola Škrbić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U radu je predstavljena implementacija aplikacije za trenutnu razmenu poruka u realnom vremenu. Takođe rad opisuje i korišćenje Kotlin programskog jezika na Android platformi kao i praktičnu primenu same aplikacije. Sama implementacija i način funkcionisanja su takođe detaljnije opisani kroz poglavlja kako bi se dočarala kompletna slika sistema.

Ključne reči: Android, chat, aplikacija, softver, Kotlin, Firebase.

Abstract – *This paper represents a study about chat application for instant message communication in real time on Android platform. It also describes a Kotlin programming language and its usage in implementation on Android platform as well as practical usage of given application. Concrete implementation and functionality of the application is also included in this work and is further elaborated in coming chapters in purpose of supplementing it for better complete image of whole system.*

Keywords: *Android, chat, application, software, kotlin, Firebase.*

1. UVOD

Mobilna aplikacija je program koji je namenjen prenosnim uređajima kao što su pametni telefoni, tableti i pametni satovi. Prvobitno su služile za obavljanje osnovnih funkcija, ali kako je tehnologija sve više napredovala, dobile su sasvim novi oblik. Danas, mobilne aplikacije su postale svakodnevica i koriste se u različite svrhe. Kalkulator, video igra, kompas, online prodavnica su neke od mobilnih aplikacija. Izrada aplikacija postala je kompleksnija pa se dosta pažnje počelo posvećivati njihovim arhitekturama. Kako bi se arhitektura uklopila u samu izradu, potrebno je poštovati komponente operativnog sistema na kom se ona primenjuje, samim tim i životni vek aplikacije koja se izvršava na mobilnom uređaju. Treće poglavlje će se baviti Kotlin programskom jeziku [2]. prednosti korišćenja, takođe će biti više reči i o specifikaciji samog programskog jezika. Kroz četvrto poglavlje govoriće se o novim tehnologijama, Google Firebase, Ubrizgavanje zavisnosti i Gradle sistem automatizacije. Peto poglavlje govoriće o samoj implementaciji aplikacije. Biće detaljnije objašnjen model podataka koji se koristi u aplikaciji, način prijave na

sistem, arhitektura sistema, komunikacija sa udaljenim servisima, kao i parcijalno učitavanje podataka i načinu testiranja istog. Poslednje poglavlje biće posvećeno zaključku ovog istraživanja i projekta.

2. ANDROID SISTEM

Android [1] je mobilni operativni sistem kompanije Google zasnovan na Linuks operativnom sistemu i njegovom jezgri, prvenstveno dizajniran za mobilne uređaje sa ekranom osetljivim na dodir, kao što su pametni telefoni i tablet uređaji. Korisnički interfejs Androida je zasnovan na direktnoj manipulaciji objektima na ekranu, korišćenjem ulaza u vidu dodira koji odgovaraju pokretima u realnom svetu kao što su prevlačenje, pritiskanje i unos teksta pomoću virtuelne tastature. Varijante Android operativnog sistema se koriste i na igračkim konzolama, digitalnim kamerama, personalnim računarima i drugim elektronskim uređajima. U odnosu na klasično programiranje, programerima je na početku problem da shvate da nema klasičnih konstruktora i inicijalizacija nad androidovim klasama, nego da treba da se prati životni ciklus sistema. Kako aplikacija ima životni ciklus tako i svaka aktivnost ima svoj životni ciklus, od pokretanja aplikacije pa sve do zaustavljanja. Aktivnost predstavlja jednu stranicu koju korisnik vidi i ona spada u jednu od četiri komponente androida. Ostale komponente Androida su servisi, dobavljači sadržaja i prijemnici poruka. Za svaku fazu životnog ciklusa definisana je odgovarajuća metoda koje su redom nazvane onCreate(), onStart(), onResume(), onPause(), onStop(), onDestroy(), onRestart() i pozivaju se u različitim životnim fazama aplikacije. Kao što je već pomenuto, ovakav način programiranja ne liči na do sada viđene i samim tim predstavlja prepreku koja se treba prevazići prilikom inicijalnog razvoja aplikacije. Ulazak u samu materiju i način njegovog funkcionisanja predstavlja poduhvat sam za sebe, pa su programeri mobilnih aplikacija dosta traženi, jer su pored poznavanja samog sistema, za sam razvoj potrebni i uređaji koji koštaju, te nisu svi u mogućnosti da programiraju na njima.

3. KOTLIN

Kotlin [2] je objektno orijentisani programski jezik nastao kao želja za unapređenjem programskog jezika Java [3]. Konkretno Android se na mobilnim telefonima pokreće pomoću Java virtualne mašine, tako što se kreira virtuelna mašina na telefonu, na koju će biti spuštena Android aplikacija. Kotlin programski jezik je razvijen od strane JetBrains kompanije koja stoji iza mnogih alata za pisanje programskog koda i zasniva se na otvorenom kodu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinać.

Sintaksa koda nije kompatibilna sa Javom, ali Kotlin ima mogućnost komunikacije sa Javom, odnosno mogu se koristiti u istom Java paketu. Android studio od verzije 3.0 zvanično podržava Kotlin kao jezik za izradu Android aplikacija što je uveliko doprinelo popularnosti isog. Kotlin stavlja naglasak na interoperabilnost, sigurnost, jasnoću, lakšu kontrolu pozadinskih procesa i kao takav predstavlja “nadogradnju” Jave. Sledi detaljniji opis neke od najbitnijih funkcionalnosti Kotlina. Funkcija proširenja nam omogućuje da dodatno proširimo već postojeću metodu (ili klasu) bez nasleđivanja. Funkcija visokog nivoa je ona funkcija koja prima funkciju kao parametar ili vraća funkciju. Isto tako, postoje i funkcije prvog reda koje primaju i vraćaju sve vrste parametara osim funkcije. U Kotlinu se null vrednost može lako izbeći upotrebom ? operatora koji prvo proverava da li postoji vrednost. Ako vrednost postoji, program nastavlja da se izvršava, ako ne postoji kod se ne izvršava i ne dolazi do pucanja. Kotlin dopušta imenovanje argumenata koje nam omogućava menjanje redosleda argumenata u funkciji. U Kotlinu postoje dve vrste literala: stringovi koji mogu da izbegnu neke karaktere i stringovi koji mogu da sadrže znak za novu liniju. Ovo je u suštini isto kao u Javi, a druga vrsta literala je kada su stringovi sa tri znaka navodnika: """ . Unutar para ovih znakova string može da sadrži bilo koji karakter i bezbroj linija. Ovo je prednost u odnosu na javu, koja nema ovu mogućnost. Ponekad nam treba objekat za male izmene neke klase, bez eksplicitne deklaracije nove klase, odnosno singleton. Java ovaj problem rešava pomoću anonimnih unutrašnjih klasa, dok Kotlin realizuje ovaj problem uz pomoću objektnih izraza i deklaracija objekata. Ključna reč za kreiranje singleton klase je object. Opseg se formira uz pomoć rangeTo funkcije koja sadrži operator .. koji se koristi u kombinaciji sa operatorom in. Opseg je definisan za bilo koji uporediv tip. Za razliku od Jave u Kotlinu klase nemaju statičke metode. U većini slučajeva jednostavno se koriste funkcije na nivou paketa. Ako je potrebno napisati funkciju koja se poziva bez instanciranja određene klase, i potreban je pristup neki poljima unutar te klase, može da se napiše kao član unutar Companion object-a. Još malo približnije, ako se deklarise Companion object unutar jedne klase, mogu da se pozivaju sve metode unutar ovog objekta s istom sintaksom pozivanja statičkih metoda u Javi i C#, koristeći samo ime klase kao kvalifikator. Često se kreiraju klase čija je glavna namena držanje podataka. U takvoj klasi neke standardne funkcionalnosti i korisne funkcije često se mehanički mogu izvesti iz podataka. U Kotlinu ove klase nazivaju se Data class i označavaju se kao podaci. Asinhrono programiranje je sve više potrebno i može se reći da je novo. Bez obzira da li kreiramo server, desktop ili mobilnu aplikaciju važno je da se pruži iskustvo koje nije samo težnost programa već i skalabilnost kada je potrebno. Postoji mnogo pristupa ovom problemu, a Kotlin ima veoma fleksibilnu podršku za to - korutine. Korutine takođe pružaju mogućnost konkurentnog programiranja. Nema povratnih funkcija, osluškivača ili nešto drugo. Korutina se može zamisliti kao lagana nit. Kao i niti, korutine mogu da se izvršavaju paralelno, mogu čekati jedna drugu i komunicirati. Najveća razlika je u tome što su korutine jeftine, gotovo besplatne, u pogledu performansi u odnosu na prave niti.

Može da se stvori na hiljade korutina i da ne bude toliko zahtevno koliko hiljadu niti može biti ozbiljan izazov za savremenu mašinu.

4. KOMPONENTE SISTEMA

Prilikom razvoja aplikacije korišćene su različite komponente sistema kako bi se poboljšale neke od funkcionalnosti, kao i omogućio brži i lakši razvoj.

4.1. Firebase

Firebase [11] je mobilna i Web platforma koja snabdeva programere sa mnoštvom alata i servisa kako bi im pomogla u razvijanju kvalitetne aplikacije. Nudi nekoliko funkcionalnosti koje su sem analitike koja je odmah dostupna po konektovanju sa projektom, podeljene na tri celine: razvoj, rast i dobit. U ta tri dela su redom smeštene funkcionalnosti prvo koje su nam potrebne uglavnom u fazi razvoja, zatim koje su nam potrebne u fazi rasta aplikacije i funkcionalnosti vezane za marketing. U projektu je korišćena Firebase baza podataka, (*Firebase Realtime Database je Cloud*) podatke čuva u JSON obliku i sinhronizuju se u realnom vremenu sa svim prijavljenim klijentima (korisnicima). Čak i ako napravimo aplikaciju koja može da radi na više platformi (iOS, Android i Web) svi klijenti dobijaju ažurirane podatke [4]. Njene ključne mogućnosti su da u realnom vremenu ne koriste se standardni upiti za podatke, Realtime Database je dizajnirana tako da stalno sinhronizuje podatke sa korisnicima. Svaki put kada se neki podatak u bazi izmeni, svaki konektovan uređaj dobija podatak unutar milisekunde vremena. Zbog toga je Firebase i pogodan za aplikacije koje simuliraju dopisivanje u realnom vremenu i oslanjaju se na žive podatke.

4.2. Ubrizgavanje zavisnosti

Ubrizgavanje zavisnosti spada u jedan od dizajn šablon. Osnova šablona kaže da ukoliko je klasi A potrebna klasa B da završi neki zadatak, može da se kaže kao da je to zavisnost B od A, odnosno A zavisi od B. Čitava poenta sistema jeste da se načini što raslojenijim kako bi bio robusniji i time olakšale buduće izmene istog. Obzirom da se kod mnogo češće čita i prepravljeno nego što se piše ili kreće iz početka, potrebno ga je izdeliti u module ili celine koje su maksimalno nezavisne jedne od drugih. Tako se i u ovom radu odlučilo da bude implementiran sličan princip, time je napravljena raslojenost sistema o kojoj će biti više reči u odeljku slojevi arhitekture.

4.3. Gradle sistem automatizacije

Android Studio koristi Gradle [5] sistem za izgradnju, objavljivanje i testiranje aplikacije koji ima mogućnost upravljanja zavisnostima. Gradle je jedan od najpoznatijih sistema za automatizaciju te ga koriste mnogi softverski projekti. Android studio koristi dodatak koji se koristi za komunikaciju sa Gradle sistemom ali nije potpuno zavistan sa Android Studio-m, što znači da se može izgraditi Android aplikacija pomoću komandne linije i bez Android Studia. Isto tako, Gradle je baziran na Java virtualnoj mašini (Java Virtual Machine, JVM [12]), čime je omogućeno pisanje vlastitih skripti u Kotlin programskom jeziku. Android aplikacija sastoji se od dve Gradle datoteke (build.gradle) od kojih se jedna odnosi na aplikacijski modul, a druga na celokupni projekt koja ima zajednička svojstva za sve module. U Gradle datoteci koja se odnosi na modul navedene su neke početne postavke za projekt (aplikacijska oznaka, minimalna verzija SDK [13]

Android paketa, verzija koda itd.), postavke za potpisivanje Android aplikacije kojom možemo ubrzati celi proces te postavke za produkcijsku verziju aplikacije i verziju aplikacije dok je aplikacija u izradi.

5. RAZVOJ APLIKACIJE ZA TRENUTNU RAZMENU PORUKA

U ovom odeljku će biti više reči o funkcionalnosti same aplikacije, detaljnijim opisima samog sistema. Posvetiće se pažnja organizaciji projekta, glavnim svojstvima i karakteristikama aplikacije i arhitekturi koja predstavlja osnovu čitave aplikacije, kao i načinu testiranja istog.

5.1. Pregled i organizacija projekta

Android projekat je generalno organizovan po paketima koji su smešteni unutar vrhovnog paketa koji po običaju ima naziv kao i sama aplikacija. Jedine stvari koje se ne nalaze u njemu, jesu upravo fajlovi i skripte koje služe za pokretanje aplikacije, kao i uvezivanje eksternih zavisnosti poput biblioteka i drugih modula. Taj odvojeni deo se sveobuhvatno nalazi u Gradle Scripts vrhovnom fajlu, gde se nalaze svi Gradle potrebni skripti.

5.2. Arhitektura

Jedan od bitnih faktora prilikom kreiranja projekta jeste arhitektura. Sve više pažnje i značaja daje se arhitekturi. Ona razdvaja komponente i čini projekat modularnijim, lakšim za testiranje i kod lakšim za održavanje. Model-View-ViewModel (MVVM) [6] arhitektura, odnosno šablon je nastala od dobro poznate kompanije Microsoft. MVVM se sve više koristi u izradi Android aplikacija i primarno služi da bi razdvojila prezentacioni sloj od same logike. Ukoliko ne implementiramo okvavku ili sličnu arhitekturu u projekat, sva logika se nalazi u pogledu (View). Na ovaj način kod postaje neprikladan za pisanje testova i nije sklon lakim promenama. Idealna MVVM arhitektura je postignuta kada se može menjati korisnički interfejs, a da logika ostane nepromenjena. Model se odnosi na model domena koji predstavlja stanje sadržaja ili na sloj za pristup podacima koji predstavlja sadržaj. View predstavlja ono što korisnik vidi na ekranu. Prikazuje korisniku podatke koje dobavlja iz modela i stupa u interakciju sa korisnikom pomoću raznih akcija. ViewModel predstavlja apstrakciju modela koji sadrži javna svojstva i komande. Vezivanje podataka omogućuje automatizaciju komunikacije između View-a i ove komponente. Ovde je opisano stanje podataka koji stižu iz Model komponente preko UseCase komponente. UseCase predstavlja sponu između View-a [14] i Model-a i unutar ovog dela arhitekture smešta se biznis logika aplikacije. Slojevi donose razdvajanje korisničkog interfejsa od same logike aplikacije, što omogućava aplikaciji modularnost, testabilnost i održivost koda. Unutar arhitekture razlikujemo tri sloja: 1) Prezentacioni sloj, 2) Domenski sloj, 3) Sloj podataka. Prezentacionom sloju pripadaju delovi pogleda (Fragment-s [16]) i aktivnosti (Activity [15]). Može slobodno da se kaže da su oni "glupi". Izvršuju naredbe i javljaju o događajima koji su se desili od strane korisnika. Prezentacioni sloj vezan je za domenski sloj, koji u sebi sadrži biznis logiku slučajeva korišćenja (UseCase). Skladište podataka (Repository) je deo arhitekture korišćen za obradu podataka. Podaci mogu da se dobiju iz više različitih izvora. Podaci o konverzacijama i korisnicima se čuvaju na Cloud Firebase-u, korisnička podešavanja čuvana su u deljene resurse (SharedPreferences [17]). Domain sloj koji rukuje

diskretnim delovima poslovne logike aplikacije. Ovo nam omogućuje da centralizujemo biznis logiku i da promena logike bude na jednom mestu. Jedan izvor podataka je implementiran tako da pormene nad bazom podataka obaveste sve one koji je osluškuju da se desila promena. Na ovaj način baza podataka nam služi kao jedini izvor podataka, a drugi delovi joj pristupaju preko skladišta podatka. Bez obzira da li koristimo drugu aplikaciju za dovođenje podataka, Firebase Database ili neki drugi servis, baza podataka bi trebala da bude jedini izvor.

5.3. Model podataka

Model podataka koji je korišćen dovoljan je za osnovnu funkcionalnost čet aplikacije. Ukoliko bi se aplikacija dalje razvijala sa novim funkcionalnostima neophodno je i proširivanje modela podataka. Model kontakta predstavlja jedinstveno identifikovanog korisnika aplikacije koji se uspešno registrovao na sitem i biće prikazan u listi svih mogućih kontakata u samoj aplikaciji. Konverzacija predstavlja skladište svih poruka između jednog ili više kontakata, odnosno korisnika sistema, kao i neke od opisnih informacija poput da li korisnik trenutno kuca na konverzaciju, ima i nepročitanih poruka i koliki im je broj i tome sličnih stvari. Sam model čet poruke predstavlja konkretan sadržaj jedne poruke sa svojim tipom, datumom kada je poslata, sadržajem, statusom (primljena, pročitana, neuspelo slanje), pripada tačno jednoj konverzaciji kao i jednom korisniku.

5.4. Opis i funkcionalnosti aplikacije

Aplikacija omogućava časkanje sa drugim korisnicima ove aplikacije. Na početnoj strani nalazi se lista konverzacija koje je korisnik ili drugi učesnik započeo i lista korisnika koji koriste aplikaciju. Svaka konverzacija ima ime. Unutar svake konverzacije nalazi se lista poruka koje koje korisnik može da pregleda. U svakom trenutku korisnik može da pošalje novu poruku, a sama poruka može biti različitog tipa, kao na primer običan tekst, slika, lokacija. Svaka poruka ima datum kada je kreirana i status da li je poslata, dostavljena ili pregledana. Korisnik može da započne konverzaciju sa drugim korisnicima aplikacije, pored toga ima mogućnost brze pretrage konverzacija i korisnika. Prijava na sistem je neophodna za aplikaciju zbog identifikacije korisnika. Za prijavu na sistem korišten je Google Sign-In [7]. Ovo nudi bezbedan sistem za potvrđivanje identiteta i smanjuje teret prilikom prijavljivanja korisnika. Omogućuje da se korisnik prijavi sa svojim Google nalogom, isti nalog koji koristi za poštu i druge usluge. Ako korisnik koristi postojeći nalog na Android uređaju, sistem će da ponudi prijavu sa već postojećim nalogom ili pak omogućiti prijavu pomoću novog. Svaki korisnik mora da poseduje Google nalog da bi imao mogućnost prijave na aplikaciju. Za komunikaciju sa udaljenim servisom korišćena je Retrofit biblioteka. Retrofit je REST (Representational state transfer) [19] klijent gradjen nad OkHttp [18] klijentom dizajniran za Android. Koristi anotacije kako bi opisao različite HTTP [20] zahteve, pruža mogućnosti kao što su dinamički parametri unutar URL-a, parametre upita, prilagođena zaglavlja, skidanje i učitavanje datoteka, pravljenje lažnih odgovora (Mocking response). Za izvršavanje Retrofit zahteva potreban je model klasa koja koristi serijalizaciju odgovora, interfejs u kojem je definisan tip HTTP zahteva i Retrofit instanca koja se koristi za izvršavanje HTTP zahteva definisanog u interfejsu. Svaki

interfejs predstavlja kolekciju različitih Application programming interface (API) poziva, svaka metoda unutar interfejsa koja želi da se koristi kao HTTP zahtev mora da sadrži anotaciju zahteva koji obavlja. Parcijalno učitavanje podataka je odrađeno uz pomoć Paging biblioteke. Ona nudi mogućnost integracije sa RecyclerView-om [8], komponentom androida, koja se i obično koristi za velike liste u kombinaciji sa živim podacima (LiveData). Parcijalno učitavanje može da se implementira na sledeće načine: 1) Samo lokalna baza, 2) lokalna baza i internet, 3) samo server.

5.5. Testiranje

Radi potrebe aplikacije, odrađena su nasumična testiranja koja su ili bila ručno rađena od strane programera ili uz pomoć Monkey Test [21] programa koji nakon podešavanja određenih parametara nasumično prolazi kroz aplikaciju i klikće po ekranu. Pored toga, rađeni su unit testovi, kako bi se proverila konzistentnost prethodnih verzija aplikacije prilikom verzionisanja, unapređivanja i menjanja. Poslednji testovi koji su bili rađeni jesu User Interface (UI) testovi, kojima se utvrđivali da su određeni pogledi, liste ili pak animacije bile odrađene i prikazane

prilikom različitih stanja aplikacije. Oni su odrađeni uz pomoć biblioteke Espresso-a [9], koja ima mogućnost snimanja akcija korisnika, tako da omogućava reprodukciju određenih slučajeva korišćenja a samim tim i proveru konzistentnosti iste.

6. ZAKLJUČAK

Svrha rada bila je razvijanje aplikacije za časkanje uz pomoć najmodernijih tehnologija i osigurati kvalitet razvijene programske podrške. Primećeno je da je bitno obratiti pažnju na arhitekturu programskog rešenja, kako bi se u daljim koracima na brži i lakši način aplikacija mogla graditi i menjati. Korišćenjem najnovijih alata i biblioteka pomoglo je oko implementacije sistema što je svakako preporuka za svakog programera koji bude imao sličnu problematiku. Jedna od mana koja se protezala tokom postavljanja aplikacije jeste dodatno vreme koje je potrebno utrošiti prilikom izrade osnove arhitekture, pogotovo ako programeri nisu upoznati sa tehnikama koje se koriste. Ali kao što je već objašnjeno u radu kompletno vreme koje je dodatno uloženo tokom postavljanja projekta i pisanja koda po nekim pravilima uglavnom se isplati već prvom prilikom kada nešto treba menjati u projektu. Upravo iz razloga što će takav kod biti neopisivo lakše održavati nego kod koji je pisan bez jasno definisanih pravila. Kako Google još od kada je kupio Android nastoji da što više olakša posao programerima na ovoj platformi, tako kontinuirano unapređuje biblioteke koje su u njegovom vlasništvu, a koje se koriste za razvoj Android aplikacija. Firebase platforma još od svog nastanka 2011. godine se pokazala kao platforma u koju vredi investirati i koja ima budućnost. Firebase pre svega ubrzava razvoj aplikacije, a i da je to postao Google-ov glavni cilj u održavanju, svedoči činjenica da se sve više aplikacija i softvera opredeljuje za korišćenje njihovih usluga. Postoji mnoštvo servisa iz Firebase-a koji posao od možda jedan ili dva dana kodiranja (prolaske kroz faze razvoja), mogu da obave u pet minuta. Aplikacije za časkanje su jedan od prvih primera korišćenja ove platforme, još pod imenom Envolv [10]. Što se tiče

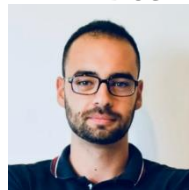
praktičnih utisaka, tokom pisanja aplikacije naišlo se na dosta dobru dokumentaciju, čime je dodatno olakšan razvoj. Na osnovu sprovedenog istraživanja uz oslonac na novo naučene tehnologije i servise, može se zaključiti da bi poznavanje ovih servisa i tehnologija bilo korisno svakom Android programeru i znatno mu olakšalo posao.

Kroz rad se moglo primetiti da su korišćene najnovije tehnologije, ali i ne da su bile jedine prilikom implementacije softverskog rešenja. Šabloni i principi koji su korišćeni su predstavljeni kako bi programeru olakšali proces implementacije softverskog rešenja, ali takođe i ovaj deo može biti zamenjen sa odgovarajućom alternativom. Tokom izrade softverskog rešenja arhitekture, korišćeni alati su se u tom trenutku smatrali za odgovarajućim. Da li je ovo najbolje rešenje? Odgovor na ovo pitanje je teško definisati, tako da se ostavlja čitaocu da sam nađe najrazumnije rešenje za svoj problem uz primer i teorijsku osnovu datu kroz ovaj rad.

7. LITERATURA

- [1] Android, official website, <https://www.android.com/>
- [2] Kotlin, Official documentation, <https://kotlinlang.org/docs/reference/android-overview.html>
- [3] Java, The Java Language, <https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf>
- [4] Firebase Realtime Database, official documentation, <https://firebase.google.com/docs/database/>
- [5] Gradle build system, official website, <https://gradle.org/>
- [7] Google Sign-in, android developers, <https://developers.google.com/identity/sign-in/android/start-integrating>
- [8] J. Sonmez, Software Testing Basics, Usersnap, <https://usersnap.com/blog/software-testing-basics/>
- [9] R. Ajesh, Espresso Basics, Medium, <https://medium.com/mindorks/android-testing-part-1-espresso-basics-7219b86c862b>
- [10] Envolv <https://www.envolve.com/>
- [11] Firebase library, Official documentation, <https://firebase.google.com/docs/android/setup>
- [12] Java virtual machine, JavaWorld, <https://www.javaworld.com/article/3272244/core-java/what-is-the-jvm-introducing-the-java-virtual-machine.html>
- [13] Software development kit, SDK, Wikipedia https://en.wikipedia.org/wiki/Software_development_kit
- [14] View, Android Developers <https://developer.android.com/reference/android/view/View>
- [15] Activity, Android Developers <https://developer.android.com/reference/android/app/Activity>
- [16] Fragment, Android Developers <https://developer.android.com/guide/components/fragments>
- [17] SharedPreferences, Android Developers <https://developer.android.com/reference/android/content/SharedPreferences>
- [18] OkHttp, official github repository <https://github.com/square/okhttp>
- [19] Representational State Transfer, REST, codecademy <https://www.codecademy.com/articles/what-is-rest>
- [20] Hypertext Transfer Protocol, HTTP, Wikipedia https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- [21] Monkey Testing, Wikipedia https://en.wikipedia.org/wiki/Monkey_testing

KRATKA BIOGRAFIJA:



Nikola Škrbić rođen je 1994. godine u Novom Sadu. Završio je srednju školu Gimnazija "20. oktobar" u Bačkoj Palanci 2013. godine. Fakultet tehničkih nauka u Novom Sadu je upisao 2013. godine. Ispunio je sve obaveze i položio je sve ispite predviđene studijskim programom.

RAZVOJ INFORMACIONOG SISTEMA ZA AUTO-ŠKOLE DRIVING SCHOOL INFORMATION SYSTEM DEVELOPMENT

Nina Babić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu predstavljena je važnost odabira odgovarajućih tehnologija pri razvoju nekog informacionog sistema, kao i analiza odabranih tehnologija. Osim toga, detaljno je analiziran sam razvijani sistem. Sistem je predstavljen i analiziran kroz prikaz modela sistema, implementiranih klasa i delova koda koji su preuzeti iz implementiranih klasa kao i kroz demonstraciju izgleda, odnosno korisničkog interfejsa razvijane aplikacije.

Ključne reči: Informacioni sistem, Spring Boot, JHipster, Maven, Hibernate, Angular

Abstract – This thesis represents the importance of the choice of technologies used for implementation of certain system, along with analysis of chosen technologies. Furthermore, detailed analysis of developed system is given. System is represented and analyzed through review of system database model, implemented classes and code parts taken from implemented classes, as well as through demonstration of user interface of developed application.

Keywords: Information system, Spring Boot, JHipster, Maven, Hibernate, Angular

1. UVOD

Uporno se stremi ka tome da se svaki segmet poslovanja ili bilo kog delovanja neke institucije digitalizuje, te je sasvim izvesno da se digitalizuju i segmenti obrazovnog sistema.

Kako i pohađanje nastave (kako teorijske, tako i praktične) za obuku za vožnju, predstavlja neki vid obrazovanja, sasvim je logično da će talas digitalizacije pogoditi i ovu vrstu obrazovnih ustanova.

U ovom radu će biti posvećena pažnja tome koje tehnologije su korišćene za razvoj ovog sistema, jer je autor akcenat stavio na to kako odabrati tehnologije koje uslovno rečeno rade za nas. Drugim rečima, kako odabrati tehnologije u zavisnosti od toga kakav sistem razvijamo, jer su tehnologije upravo tu da nam pomognu, a ne da nam odmognu.

Osim toga, osvrnućemo se na to kakav je model baze podataka za razvijani sistem. Dalje će biti dat detaljniji uvid u implementaciju sistema. Konačno, biće napravljen i pregled implementirane aplikacije, odnosno korisničkog interfejsa.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinac.

2. KORIŠĆENE TEHNOLOGIJE

Pre početka razvoja same aplikacije, najbitnija stvar je odabrati odgovarajući set tehnologija pomoću kojih ćemo razvijati istu, jer su upravo tehnologije, te koje rade za nas. One treba da nam olakšaju posao i da nam pomognu da postignemo uspeh, dok bi s druge strane, izbor loših tehnologija vrlo lako mogao biti glavni preduslov za propadanje projekta. U nastavku će biti navedene neke od korišćenih tehnologija pri razvoju informacionog sistema za auto-škole.

2.1. JHipster

JHipster je open source alat koji je vrlo zgodan za korišćenje. Poznato je da je nekad prava muka povezati sve željene tehnologije. Tu na scenu stupa JHipster koji to radi za vas. On predstavlja generator aplikacija koji kreira SpringBoot i Angular aplikaciju. Dakle, dobijate aplikaciju sa Java back-end-om i Angular 6 front-end-om. Drugim rečima, korišćenjem JHipster-a dobićemo takozvanu ljušturu projekta, čime nam je uštedeno vreme na povezivanju tehnologija, koje sada možemo potrošiti na sam razvoj aplikacije.

2.2. Maven

Maven predstavlja alat za automatizovano bildovanje backend-a, kao i za dependency menadžment, između ostalog. Maven je taj koji nam pomaže da definišemo kako će se neka aplikacija build-ovati (build life cycle), odnosno od čega aplikacija zavisi kako bi joj bilo obezbeđeno to od čega zavisi (dependency management).

2.3. Spring Boot

Spring Boot predstavlja alat koji automatski obavlja sve ono što nam je zadavalo muke. Dakle, vrši automatizovanu konfiguraciju Spring-a kao i konfiguraciju Spring Bean-ova, ali konfiguraciju koju je moguće u svakom trenutku pregaziti sa malim izmenama koje su nama potrebne.

2.4. Hibernate

S obzirom na to da vrlo lako može da dođe do neslaganja između objektnog i relacionog modela sistema u smislu predstave podataka u jednim i drugim, potreban nam je alat koji će nam skratiti vreme razvoja samog sistema, time što će olakšati objektno-relaciono mapiranje. Upravo iz tog razloga nam je potreban Hibernate alat koji, između ostalog služi za mapiranje objektnog u relacioni model i primenu objektno-orijentisanih koncepata u relacionim bazama podataka.

2.5. Angular

Angular predstavlja TypeScript baziranu platformu koja je tu da nam olakša razvoj front-end dela naše web aplikacije. Karakteristično za Angular jeste to što predstavlja MVC framework, što samim developerima omogućava da primene MVC strukturu na klijentsku stranu aplikacije koju razvijaju.

3. MODEL BAZE PODATAKA

Baza podataka je neizostavna komponenta svakog informacionog sistema, bez obzira na to da li je u pitanju sistem sa kompleksnom biznis logikom, ili se radi relativno jednostavnom sistemu.

S obzirom na to da je reč o za sada nekomplikovanom sistemu, model baze za auto-školu se sastoji od četiri, uslovno rečeno, obične tabele, kao i od tri vezne tabele. Sama baza podataka je kreirana u MySQL-u.

Svaka od tabela ima svoj primarni ključ, dok neke tabele sadrže i strani ključ drugih tabela, jer su na neki način povezane.

Osim toga, kao što je već napomenuto postoje i tri vezne tabele, koje predstavljaju vezu više-prema-više, odnosno Many-to-many relaciju.

Model baze podataka za sistema auto-škola se sastoji iz sledećih tabela:

1. instruktor
2. učenik
3. vozilo
4. raspored
5. raspored_učenik
6. raspored_vozilo
7. raspored_instruktor

4. IMPLEMENTACIJA

U nastavku će biti navedene neke od klasa implementiranog sistema, sa prikazom delova koda nekih od njih.

4.1. Bean

U aplikaciji imamo nekoliko entiteta, to su učenik, instruktor, vozilo i raspored. Svaki od tih entiteta ima klasu, odnosno bean koji ga predstavlja. Bitno je napomenuti da je svaki od bean-ova anotiran anotacijom `@Entity`, koja ističe da je dati objekat, odnosno datu klasu moguće mapirati na tabelu u bazi podataka.

4.2. Controller

Kontroleri, spadaju u prezentacioni sloj aplikacije. Kada korisnik pošalje neki zahtev, odnosno odluči da na primer ažurira neki entitet, taj zahtev prihvata upravo kontroler. Drugim rečima, kontroleri rukuju `Http request`-ovima.

U kontroleru se nalaze osnovne CRUD (Create, Read, Update, Delete) operacije, međutim sama logika nije u njima. Kontroleri nam između ostalog služe da obave proveru da li su zahtevi klijenta validni.

Ukoliko određeni zahtev nije validan, kontroler će poslati odgovor u vidu greške. Ako zahtev klijenta zaista jeste validan, kontroler gađa servisnu klasu u kojoj se u stvari nalazi sva logika za operaciju za koju je klijent poslao zahtev. Svaki od kontrolera je anotiran anotacijom `@RestController`, kako bi taj kontroler postao vidljiv Spring kontejneru.

4.3. Service

Sva biznis logika aplikacije smeštena je upravo u servisne klase. Servisne klase su korišćene kako bi se jasno odvojilo ono što ima veze sa web delom aplikacije, od onoga što predstavlja neku uopštenu biznis logiku.

Servisne klase označene su `@Service` anotacijom, iz istog razloga iz kog su kontroleri bili označeni `@Controller` anotacijom.

4.4. Module

Dok su prethodno navedene klase pripadale back-end delu aplikacije, moduli, kao i klase koje će biti navedene u nastavku pripadaju front-end delu aplikacije.

Module (modul) predstavlja kontejner koji sadrži različite komponente razvijanog sistema (servise, direktive i sl.). Svaki od entiteta (u kontekstu Angular-a svaki entitet predstavlja komponentu), ima svoj modul u kom su navedene komponente od kojih ta komponenta (npr. komponenta učenik) zavisi. Osim toga svaki modul sadrži deklarisanje komponenti koje mu pripadaju.

4.5. Model

Kao i na back-end-u, tako je i na front-end-u potrebno da imamo klasu koja će predstavljati određeni entitet. Na front-end-u je to tzv. Model. Model predstavlja reprezentaciju DTO-a (Data Transfer Object) kog dobavljamo sa back-end-a.

4.6. Service

Kako bi se izbeglo da komponente u Angular-u budu te koje manipulišu podacima, jer to definitivno nije dobra praksa, uvedeni su servisi. Uvođenjem servisa, omogućeno je da komponente budu čiste i efikasne.

Ono što je bitno napomenuti, jeste upotreba `HttpClient`-a u svakom od implementiranih servisa. `HttpClient` predstavlja klijentski HTTP API za Angular aplikacije. Upravo njega koristimo kako bismo poslali određeni http zahtev, bio to GET, POST, PUT, DELETE ili sl. Na slici 1 prikazana je metoda `update`, koja šalje http zahtev PUT, kako bi se ažurirali podaci određenog učenika.

```
update(učenik: IUčenik): Observable<EntityResponseType> {  
  const copy = this.convertDateFromClient(učenik);  
  return this.http  
    .put<IUčenik>(this.resourceUrl, copy, { observe: 'response' })  
    .pipe(map((res: EntityResponseType) => this.convertDateFromServer(res)));  
}
```

Slika 1. Prikaz metode za ažuriranje podataka učenika

4.7. Component

Kada pričamo o komponentama, možemo slobodno reći da je sve u Angularu neki vid komponente. Na komponente možemo gledati kao osnovne jedinice građe nekog modula.

Svaka od komponenti, označena je `@Component` dekoratorom. Funkcija ovog dekoratora jeste da obavesti Angular da se radi o komponenti, osim toga za ovaj dekorator su vezani i određeni metapodaci, koji Angularu mogu da kažu mnogo toga, kao što je npr. putanja do template-a ove komponente.

4.8. Template

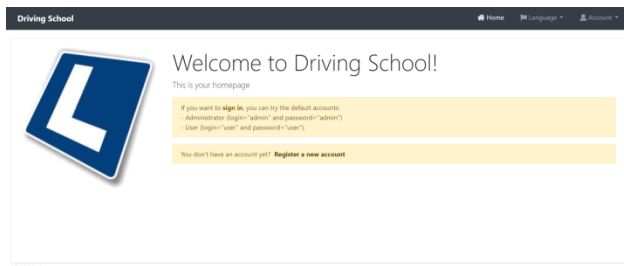
Dok smo implementacijom komponente definisali ono šta korisnik može da radi, implementacijom template-a defi-

nišemo ono šta je korisniku vidljivo. Jezik koji koristimo za implementaciju template-a u Angular-u jeste HTML.

U okvirima template-a, skoro sva HTML sintaksa je validna. Osim toga, rečnik HTML-a template-a možemo obogatiti našim direktivama i komponentama, koje se javljaju kao novi elementi ili atributi.

5. PRIKAZ APLIKACIJE

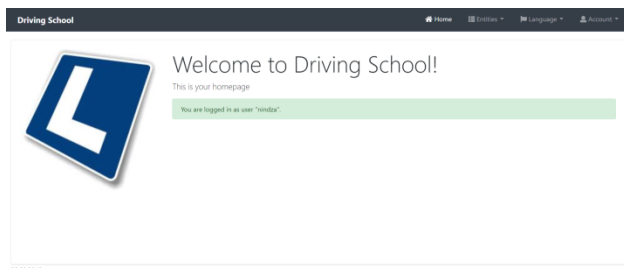
Nakon što je korisnik pristupio aplikaciji, ono što mu se inicijalno prikazuje jeste home stranica. Na home stranici korisnik može da odabere željeni jezik, pre logovanja u aplikaciju. Na slici 2 je moguće videti kako izgleda home stranica razvijane aplikacije.



Slika 2. Početna stranica pre prijave korisnika

Kako bi se korisnik ulogovao u aplikaciju, potrebno je da unese svoje kredencijale, odnosno korisničko ime i lozinku.

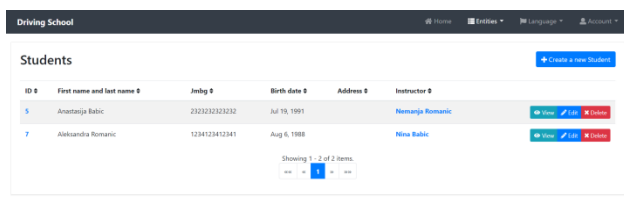
Po uspešnom prijavljivanju na sistem, korisniku se prikazuje početna stranica, koju možete videti na slici 3.



Slika 3. Početna stranica nakon uspešne prijave korisnika

U meniju je korisniku ponuđen Home, koji vodi na početnu stranicu. Ono što korisnik još može da vidi u meniju, između ostalog jesu entiteti u aplikaciji. Osim toga korisnik u svakom trenutku može da promeni jezik sistema. Mogući jezici su engleski i srpski jezik. Takođe, ulogovani korisnik može da upravlja svojim nalogom, moguće funkcionalnosti se nalaze u stavci menija pod nazivom Account.

Kao što je ranije navedeno, entiteti u sistemu su instruktori, učenici, vozila i rasporedi. Na slici 4 moguće je videti kako izgleda stranica koja predstavlja entitet učenik.



Slika 4. Entitet učenik

Svaki od entiteta je moguće izmeniti (Edit), videti detalje istog (View), kao i obrisati ga (Delete). Osim toga,

korisnik u svakom trenutku može da doda novog instruktora, učenika, vozilo i raspored (Create new ...). Na slici 5 je moguće videti kako izgleda forma za unos novog instruktora.

Slika 5. Forma za kreiranje novog instruktora

Kao što se vidi na slici 5, neka polja imaju indikator crvene boje, dok druga nemaju. Ti indikatori predstavljaju vid validacije na front-end-u. U ovom slučaju je to validacija za obaveznost polja. To nam govori da sve dok korisnik ne unese podatke za sva obavezna polja, on neće biti u mogućnosti da submit-uje formu, odnosno neće moći da kreira novog Instruktora.

Kao što je prethodno napomenuto, moguće je videti detalje svakog entiteta. Forma za detalje služi isključivo za čitanje, odnosno u njoj nije moguće menjati vrednosti koje su unete za datog Instruktora. Međutim, klikom na dugme Edit u formi za detalje, moguće je preusmeriti se na formu za ažuriranje podataka o Instrukturu. Kako izgleda forma za detalje, moguće je videti na slici 6.

Slika 6. Forma za pregled detalja instruktora

Formu za izmenu podataka o instrukturu možete videti na slici 7. Ova forma izgleda veoma slično formi za unos novog instruktora. U implementaciji one predstavljaju istu formu, sa određenim uslovima vezanim za to šta u kojoj formi treba da bude vidljivo korisniku, te ih upravo iz tog razloga korisnik ne vidi u istom formatu.

Slika 7. Forma za ažuriranje podataka instruktora

Kao što je već navedeno, korisnik može da obriše određeni entitet. Kada korisnik pozove akciju brisanja klikom na dugme Delete koje je moguće videti na slici 4, otvara se pop-up dijalog (slika 8).

Slika 8. Potvrda za brisanje Instruktora

Dijalog koji je prikazan na slici 8, namenjen je za to da korisnik potvrdi da zaista želi da obriše određeni entitet. Ukoliko korisnik zaista želi da obriše odabrani entitet, potrebno je da klikne na dugme Delete. Međutim, ukoliko želi da odustane, potrebno je da klikne na dugme Cancel.

6. ZAKLJUČAK

Ovim radom je obuhvaćen proces od odluke koje tehnologije koristiti do toga kako je sam sistem implementiran. Sama razvijana aplikacija predstavlja samo začetak ideje o digitalizaciji sistema neke auto-škole. Za sada je namenjena samo instruktorima i administrativnim radnicima zaposlenim u datoj auto-školi. Međutim, autor ima razvijenu ideju kako taj sistem unaprediti, kao i šta je potrebno uraditi kako bi sama aplikacija mogla da donese koristi svim potencijalnim članovima jedne auto-škole, od učenika, preko instruktora, administrativnih radnika, do menadžmenta iste.

Kada se govori o učenicima, ideja je da se aplikacija proširi posebno dizajniranim rasporedom samo za njih, koji će obuhvatati detaljni nedeljni raspored prvo teorijske, a zatim i praktične nastave. Osim toga, postoji i ideja da se sistem proširi i testovima koji će olakšati učenicima da se sprema za polaganje teorijskog ispita. Sam entitet učenik bi mogao da se proširi još jednim poljem koje bi označavalo broj izlazaka na teorijske ispite, kao i poljem koje bi označavalo i broj izlazaka na teren kako bi se polagao praktičan ispit vožnje. Dakle, konačna ideja je da učenici imaju poseban korisnički nalog, putem kog bi videli elemente aplikacije koje su njima od interesa.

Kada je reč o instruktorima, takođe postoji ideja da se napravi optimizovaniji raspored nastave, u odnosu na implementirani, kako bi im bilo omogućeno još jednostavnije praćenje obaveza koje imaju.

Osim toga, moguće je sistem proširiti i izveštajima, koji bi u mnogome olakšali posao administrativnim radnicima zaposlenim u auto-školi, te bi ih tako jedan klik delio od toga da dobiju uvid u npr. prošlomesečno zauzeće nekog instruktora i sl.

Korišćenjem ovog sistema bi se drastično smanjila mogućnost oko pogrešnog ugovaranja termina nastave, ili prezauzetosti određenog instruktora, dok drugi instruktor nema u nadležnosti nijednog učenika. Osim toga mogao bi da se prati status svakog učenika uvidom u to koliko puta je izašao na teorijski odnosno praktičan ispit. Zatim bi se olakšano pratila potreba za npr. proširivanjem voznog parka, ili nastavničkog osoblja. Te bi to predstavljalo odličan primer korišćenja tehnologija tako da nam olakšaju svakodnevicu.

7. LITERATURA

- [1] *About JHipster*,
<https://www.jhipster.tech/> (pristupljeno u septembru 2018.)
- [2] *What is Maven*,
<https://maven.apache.org/what-is-maven.html>
(pristupljeno u septembru 2018.)
- [3] *Spring Boot*,
<https://spring.io/projects/spring-boot>
(pristupljeno u septembru 2018.)
- [4] *What is Hibernate*
<https://www.linkedin.com/learning/java-database-access-with-hibernate/what-is-hibernate> (pristupljeno u septembru 2018.)
- [5] *What is Angular*
<https://angular.io/docs> (pristupljeno u sept. 2018.)

Kratka biografija:



Nina Babić rođena je u Bačkoj Topoli 23.10.1991. Školske 2010/2011 je upisala Fakultet tehničkih nauka u Novom Sadu. Zvanje diplomirani inženjer elektrotehnike i računarstva stekla je 2015. godine. Položila je sve predmete predviđene planom i programom na master akademskim studijama na odseku: Primenjeno softversko inženjerstvo.

ФУНКЦИОНАЛНА ПАРАДИГМА У ПРОГРАМСКОМ ЈЕЗИКУ C#**FUNCTIONAL PARADIGM IN C# PROGRAMMING LANGUAGE**Марко Иветић, *Факултет техничких наука, Нови Сад***Област – Рачунарство и аутоматика****Кратак садржај** – *Тема рада је представљање концепата функционалног програмирања. У оквиру рада, поред описа самих концепата, приказани су и примери њихове употребе у програмском језику C#.***Кључне речи:** *Функционално програмирање, Објектно-оријентисано програмирање, Ламбда изрази, Чисте функције, Непроменљивост***Abstract** – *The subject of this paper is presenting concepts of functional programming. Beside the description of concepts, it also contains examples of their usage in C# programming language.***Keywords:** *Functional programming, Object-oriented progr., Lambda expression, Pure functions, Immutability***1. УВОД**

Програмирање је термин под којим се најчешће подразумева креирање рачунарских програма, што укључује детаљну теоријску разраду проблема, проналажења концептуалног решења и имплементацију коришћењем неког од програмских језика. Програмски језици [1] су формални језици који се могу користити за контролу понашања машина, нарочито рачунара. Програмски језици се користе да олакшају комуникацију са рачунаром приликом организовања и манипулације информација, али и прецизно изразе алгоритме.

Поред програмских језика развијено је и неколико програмских техника (парадигми) за писање програма.

Програмска парадигма [2] представља основни стил програмирања, служећи као начин прављења структуре и елемената рачунарских програма. Различити програмски језици подржавају различите парадигме програмирања. Већина језика подржава само једну парадигму, али има и језика који могу подржати и више парадигми.

Програмске парадигме се могу поделити у две групе, императивну и декларативну. Код императивне, у коју спадају процедурално и објектно-оријентисано програмирање, акценат је стављен на начин на који ће се задатак обавити. Код декларативне групе, у коју спадају функционално и логичко програмирање, акценат је на самом задатку и томе шта је потребно обавити, а не и како.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Александар Купусинац.

2. УВОД У ФУНКЦИОНАЛНО ПРОГРАМИРАЊЕ

У рачунарству, функционално програмирање [3] је програмска парадигма која третира програм као израчунавање математичких функција и избегава стања и променљиве податке. Посматра појединачне операције као израчунавање математичких функција. Супротно од стила императивног програмирања, код кога је акценат на промени стања, код функционалног програмирања акценат је стављен на примену функција. Спада у декларативне програмске парадигме јер акценат ставља на то шта посматрани код, односно апликација ради, а не како ради. Развојем математичког стила програмирања долази до удаљавања од конкретне рачунарске архитектуре која извршава програм. То за последицу има слабије перформансе функционалних језика у односу на императивне, те су они мање заступљени у развоју комерцијалних софтверских решења.

Предност функционалних језика је што су они ближи људском размишљању него рачунарима. У пракси, разлика између математичке функције и појма функција који се користи у императивном програмирању је у томе што императивне функције могу да имају бочне ефекте, јер могу да промене вредности већ извршених израчунавања. Због тога, њима недостаје референцијална транспарентност, другим речима, исти израз може да доведе до различитих резултата у различитим тренуцима, у зависности од стања програма који се извршава.

Са друге стране, у функционалном коду, излазна вредност функције зависи само од аргумената који се проследе функцији, па ће функције f два пута, са истом вредношћу аргумента x произвести исти резултат. Елиминисањем бочних ефеката, разумевање и предвиђање понашања програма може да постане много лакше, и ово је једна од кључних мотивација за развој функционалног програмирања. Елиминација бочних ефеката подразумева елиминацију измене вредности претходно израчунатих резултата (*mutable data*) и стања апликације.

Корени функционалног програмирања леже у ламбда рачуну [4], формалном систему развијеном током 1930-их (развио их је Алонзо Черч) ради проучавања дефиниције и примене функција и рекурзије. Ламбда рачун пружа теоријски оквир за израчунавање и описивање функција.

Иако се ради о математичкој апстракцији а не о програмском језику, он представља основу скоро свих модерних функционалних програмских језика.

Ламбда рачун се користи у рачунарству због своје корисности у приказивању функционалног размишљања и итеративне редукције. Ламбда рачун садржи два упрошћавања која чине семантику једноставном. Поједностављује израчунавање уводећи неименоване функције и *curring*. Прво поједностављење - ламбда рачун третира функцију „анонимно“, без давања експлицитних имена. Друго поједностављење - ламбда рачун користи само функције једног улаза. Обична функција која захтева два улаза, може се написати као једна еквивалентна функција која прима један улаз, а да излаз враћа другу функцију, која заузврат прима један улаз.

У функционалном програмирању [2] се све заснива на идеји композиције једноставних функција, како би се добила комплексна структура програма, док се код објектног програмирања све заснива на композицији објеката (граф објеката).

LINQ [5] је базиран на композицији функција. *LINQ* (*Language Integrated Query*) је језик структурираних упита за претраживање, како локалних колекција објеката, тако и удаљених извора података, на такав начин да не нарушава безбедност података.

Он омогућава скраћено писање кода у раду са било каквим колекцијама и динамичким састављањем упита. Омогућава рад са различитим изворима података коришћењем истог начина кодирања.

2.1 Разлике функционалног у односу на објектно-оријентисано програмирање

Оно што је главна разлика ова два начина програмирања је сама чињеница да они представљају две различите парадигме у програмирању.

Функционално програмирање спада у декларативну (фокус на томе шта се ради) а објектно-оријентисано у императивну (фокус на то како се ради) програмску парадигму. Затим, код ООП-а редослед операција је изузетно битан. Када се измени стање неког објекта од стране различитих делова апликације, извршавање одређених функционалности може директно да зависи од стране њиховог редоследа.

Такође, са становишта једног дела апликације, неки други део апликације може објекат оставити у невалидном стању, што доводи до грешке, односно негативног *side* ефекта.

3. ОСНОВНИ КОНЦЕПТИ ФУНКЦИОНАЛНОГ ПРОГРАМИРАЊА

Функционално програмирање уводи неколико фундаменталних принципа, базирајући се на математичким основама ламбда калкулуса. У овом поглављу су представљени ти принципи, односно основни концепти функционалне програмске парадигме.

3.1 Функције вишег реда (Higher Order function)

У функционалним језицима, функција је основни градивни блок кода (за функцију се каже да је *first-class citizen*). Као што се у објектно-оријентисаним језицима креирају и користе објекти, тако се на исти

начин у функционалним језицима креирају и користе функције. Једноставно речено, функција се може креирати, доделити некој променљивој, проследити другој функцији као аргумент или се вратити из друге функције као повратна вредност.

Функција која може да прихвати неку другу функцију као свој аргумент је функција вишег реда [6]. То је важан концепт који омогућава једноставну и природну имплементацију инверзије контроле, који је један од најважнијих принципа у развоју софтвера. Према концепту инверзије контроле, објекти не треба да креирају друге објекте од којих зависе, већ те објекте добијају екстерно.

3.2 Непроменљиви подаци (Immutability)

У императивном програмирању, варијабла може имати вредност, другачију у односу на прослеђену. Такође *for*, *while* и *do while* петље захтевају *mutable* варијабле чије промене доводе до услова за прекид и излазак из петље. У функционалном програмирању стање објекта се не мења. Уколико постоји потреба да се промени стање неког објекта, прави се копија оригиналног објекта и затим се та копија мења уместо промене самог оригиналног прослеђеног објекта.

3.3 Чисте функције (Pure functions)

Чиста функција [7] је функција која нема бочне ефекте, која може да се позове велики број пута и да сваки пут врати исти резултат, који само и једино зависи од аргумената те функције, а не од глобалног стања. Користи вредности својих улазних параметара и не приступа нити мења вредности других променљивих. Самим тим се избегава нежељени *side* ефекти који могу проузроковати *bug*. Чињеница да функције у функционалном програмирању конзистентно враћају резултат за исте улазне вредности, оставља се слобода кориснику таквих функција да их позива као „црну кутију“.

3.4 Лењо израчунавање (Lazy evaluation)

Лењо израчунавање [8] је стратегија израчунавања која одлаже само израчунавање израза све док његова вредност није неопходна програму, и која такође избегава поновно израчунавање. Лењо израчунавање је врло често комбиновано са мемоизацијом. Након што је вредност функције израчуната за вредност једног или више улазних параметара, резултат се смешта у упоредну табелу (колекцију). Следећи пут када се та функција позове, претражује се колекција како би се видело да ли је резултат за те улазне параметре већ израчунат. Ако јесте, резултат из колекције се просто врати позиваоцу функције.

3.5 Pattern matching

Усклађивање структуре објекта са унапред одређеним обрасцима, тако да се може наставити са израчунавањем само ако је објекат у складу са одређеном структуром и вредностима.

3.6 Референцијална транспарентност (*Referential transparency*)

Постоји један важан аспект чистих функција који произилази из чињенице да оне зависе само од њихових аргумената, а то је референцијална транспарентност [9]. Референцијална транспарентност омогућава да се безбедно замени читав израз са својом одговарајућом вредношћу у свим каснијим референцама на њега, без промене понашања програма.

3.7 Рекурзија

Понављање делова програмског кода у смислу петљи не постоји у функционалним језицима, него се такви процеси решавају рекурзијом. Рекурзија може попримити многе облике у функционалним језицима и сматра се да је ефикаснија од механизма петљи у императивним језицима. То је и један од разлога што и већина императивних језика подржава рекурзију.

4. МЕТОДЕ И ПРИМЕРИ КОРИШЋЕЊА КОНЦЕПАТА ИЗ ФУНКЦИОНАЛНОГ ПРОГРАМИРАЊА У ООП-У НА ПРИМЕРУ C#

Сви претходно наведени концепти функционалног програмирања имају своје репрезенте у C#-у, што значи да се могу користити без оклевања. Међутим, C# је објектно-оријентисан језик и без обзира колико су концепти функционалног програмирања привлачни они нису природни за C# и морају се користити на адекватан начин.

У функционалном програмирању контрола тока извршавања нема *if-then-else*, нема петљи, него постоји *pattern matching*. Функција добије аргументе и ти аргументи се упоређују са очекиваним шаблонима. За сваки шаблон се зна шта треба да се уради. Направи се унија тих шаблона за које се зна, ако је она једнака домену функције, онда је та функција тотална, дефинисана за сваки аргумент. Било шта да позивалац функције проследи, функција ће моћи да произведе резултат. Слика 4.1 приказује функцију за израчунавање Фибоначијевих бројева, која је базирана на рекурзивним позивима.

```
1 using System;
2
3 namespace Demo
4 {
5     class Program
6     {
7         static long Fibonacci(int n) =>
8         {
9             n < 2
10             ? n
11             : Fibonacci(n - 1) + Fibonacci(n - 2);
12         }
13         static void Main(string[] args)
14         {
15             for (int i = 0; i < 40; i++)
16             {
17                 Console.WriteLine($"{i} \t {Fibonacci(i)}");
18             }
19             Console.ReadLine();
20         }
21     }
22 }
23
```

Слика 4.1 Једноставна функција за израчунавање Фибоначијевих бројева

Код са Сlike 4.1 даје исправне резултате, међутим време извршавања овог кода, који се може сматрати једноставним, траје веома дуго. Због рекурзивних позива, ова функција дуго ради, иако је сама по себи доста тривијална. И управо овај пример показује како се понашају перформансе функција у пракси. Ова функција ће рекурзивно, за пример *Fibonacci* од 10

($f(10)$), позвати $f(9) + f(8)$, затим да би израчунала $f(9)$ мора да позове $f(8)$ поново и $f(7)$, што намеће то да ће се онај крајњи $f(1)$ позвати велики број пута. Једини битан елемент, везан за овај пример, је тај да се примети да ће тај $f(1)$ који се позива велики број пута, сваки пут вратити исти резултат. У функционалном програмирању се таква функција зове „Чиста функција“.

У чистим функционалним језицима, који гарантују да су све функције чисте (*Pure*), постоји један принцип који се зове мемоизација. Мемоизација [10], у рачунарству, је техника оптимизације која се користи да би се убрзало извршавање рачунарских програма тако што се складиште резултати „скупих“ позива функција и враћају кеширани резултат када се поново јаве исти улази.

Ефекти коришћења мемоизације су најчешће позитивни и утрошак меморије је у прихватљивом обиму. На слици 4.2 приказан је пример функције за израчунавање Фибоначијевих бројева која користи кеширање тј. принцип мемоизације.

```
16 static Dictionary<int, long> dynamicCache = new Dictionary<int, long>();
17
18 static long DynamicFibonacci(int n)
19 {
20     long value;
21     if (dynamicCache.TryGetValue(n, out value))
22     {
23         return value;
24     }
25     value = n < 2
26         ? n
27         : DynamicFibonacci(n - 1) + DynamicFibonacci(n - 2);
28     dynamicCache[n] = value;
29     return value;
30 }
31
```

4.2 Full memorization

Динамичко програмирање [11] је метод за решавање сложеног проблема, тако што се он „разбије“ у колекцију једноставних и сваки од подпроблема се решава само једном и притом се решења чувају. У функцији *DynamicFibonacci*, свака појединачна ставка листе је израчуната на захтев, и не више од једном. Подаци се кеширају као парови кључ-вредност где је кључ примљени аргумент функције а вредност, повратна вредност те функције. Када тражени кључ у кешу не постоји, позива се целокупно тело функције. На крају функције, повратна вредност се сачува у кешу. На тај начин кеш је универзалан, независан од функције.

Следећи пример показује како стање објекта може утицати на повратну вредност једне функције. На слици 4.3 је приказан код који има *side ефекат*.

```
1 namespace Demo
2 {
3     class Program
4     {
5         class BankCard
6         {
7             public DateTime ValidBefore { get; set; }
8             public decimal Balance { get; set; }
9
10            public decimal GetAvailableAmount(decimal desired)
11            {
12                if (DateTime.Now.CompareTo(ValidBefore) >= 0)
13                    return 0;
14                return Math.Min(Balance, desired);
15            }
16        }
17        static void Main(string[] args)
18        {
19            BankCard card = new BankCard()
20        {
21            ValidBefore = DateTime.Now,
22            Balance = 1000
23        };
24        Console.WriteLine(card.GetAvailableAmount(1000));
25        Console.WriteLine(card.GetAvailableAmount(1000));
26        Console.ReadLine();
27    }
28 }
29
```

Слика 4.3 BankCard класа

Следећа слика (4.4) приказује како се функција *GetAvailableAmount* може позивати са истим аргументом али сваки пут да врати другачији резултат.

```

21         return Math.Min(100, balance, desired);
22     }
23 }
24 namespace
25 {
26     static void Main(string[] args)
27     {
28         BankCard card = new BankCard()
29         {
30             ValidBefore = DateTime.Now.AddSeconds(10),
31             Balance = 100;
32         };
33         decimal available1 = card.GetAvailableAmount(20);
34         card.Balance = 15;
35         decimal available2 = card.GetAvailableAmount(20);
36         Thread.Sleep(1000);
37         decimal available3 = card.GetAvailableAmount(20);
38         Console.WriteLine("available1 : " + available1.ToString());
39         Console.WriteLine("available2 : " + available2.ToString());
40         Console.WriteLine("available3 : " + available3.ToString());
41         Console.ReadLine();
42     }
43 }

```

Слика 4.4. Позивање методе *GetAvailableAmount*

На овом примеру се може видети како стање једног објекта може утицати на повратну вредност функције. Постоје три позива функције *GetAvailableAmount* и сва три позива враћају различити резултат. Први пут ће функција вратити 20, други пут 15, док ће трећи пут вратити 0, и то је типична ситуација у објектно-оријентисаном програмирању, која се избегава у функционалном. Гледано са стране функционалног програмирања, функција је чиста ако зависи само од улазних вредности и нема видљивог нежељеног ефекта. Функција *GetAvailableAmount* није чиста, зависи од променљивог стања објекта, који се може променити између позива.

Решење овог проблема би било да се уклони зависност од променљивих стања, или да се стања учине непроменљивим. Оба *property*-ја могу постати *read-only* и константни за цео „животни век“ објекта, а време које се проверава у методи, може бити прослеђено као аргумент.

На тај начин се метода чини чистом. Остатак кода, који инстанцира објекат класе *BankCard* и позива њену функцију се неће компајлирати више. *Immutable state* (непроменљиво стање) је неопходно уколико је циљ да функција врати увек исти резултат за исте вредности аргумената.

Структура која се користи у *Immutable* колекцијама је *ImmutableList*. *ImmutableList* је балансирано AVL стабло које гарантује да је дубина до $\log_{\text{aritama}}(n)$. Балансирање се врши динамички. Када постане „нагнуто“ на једну страну, онда се оно динамички ребалансира.

5. ЗАКЉУЧАК

Већини програмера објектно-оријентисана парадигма је једино позната. Разлог томе је мала примена функционалне парадигме у развоју модерних софтверских решења, пре свега због слабијих перформанси функционалних језика. Међутим, данас, уз јефтине рачунарске компоненте и већу процесорску моћ рачунара, проблем перформанси функционалних језика је лако премостив.

Елементи функционалног програмирања су корисни, али исто тако не треба уклонити оно што је добро у објектно-оријентисаном програмирању. Нити је функционално програмирање „лек за све болести“, нити је то објектно-оријентисано.

Обе парадигме имају своје предности и мане, па примена концепата из функционалног програмирања на местима где објектно заостаје, може представљати крајњи циљ који ће умногоме олакшати и улепшати код.

Коришћењем концепата из функционалног програмирања омогућава код са мање *bug*-ова, где се лакше и безбедније праве измене због *mutability* концепта и елиминације *side* ефеката. У оквиру рада, представљени су концепти функционалног програмирања и примери њихове употребе у објектно-оријентисаном језику као што је *C#*.

6. ЛИТЕРАТУРА

- [1] https://en.wikipedia.org/wiki/Programming_language - Programming language
- [2] https://en.wikipedia.org/wiki/Programming_paradigm - Programming paradigm
- [3] https://en.wikipedia.org/wiki/Functional_programming - Functional programming
- [4] https://en.wikipedia.org/wiki/Lambda_calculus - Lambda calculus
- [5] <https://msdn.microsoft.com/en-us/library/bb308959.aspx> - LINQ
- [6] https://en.wikipedia.org/wiki/Higher-order_function - Higher-order function
- [7] https://en.wikipedia.org/wiki/Pure_function - Pure function
- [8] https://en.wikipedia.org/wiki/Lazy_evaluation - Lazy evaluation
- [9] https://en.wikipedia.org/wiki/Referential_transparency - Referential transparency
- [10] <https://en.wikipedia.org/wiki/Memoization> - Memoization
- [11] https://en.wikipedia.org/wiki/Dynamic_programming - Dynamic programming

Кратка биографија:



Марко Иветић, рођен је 23. септембра 1993. године у Новом Саду, Србија. Завршио је средњу економску школу „Светозар Милетић“ у Новом Саду, смер Финансијски техничар. Школске 2012/2013. године уписао је Факултет техничких наука у Новом Саду, одсек Рачунарство и аутоматика, усмерење Примењене рачунарске науке и информатика. Звање дипломирани инжењер електротехнике и рачунарства стекао је 2016. године. Исте године уписао је мастер академске студије на одсеку Рачунарство и аутоматика, усмерење Софтверско инжењерство. Положио је све испите прописане планом и програмом.

МЕТОДОЛОГИЈЕ УПРАВЉАЊА ПРОЈЕКТОМ СА АКЦЕНТОМ НА SCRUM**PROJECT MANAGEMENT METHODOLOGIES WITH ACCENT ON SCRUM**Милица Говедарица, *Факултет техничких наука, Нови Сад***Област – РАЧУНАРСТВО И АУТОМАТИКА**

Кратак садржај - У раду су описане популарне методологије за управљање пројектом. Направљено је угрубо поређење двије главне групе методологија. Акценат рада је стављен на *scrum*, који је до детаља објашњен.

Кључне речи: *Методологије управљања пројектом, agile, scrum, sprint, backlog, product*

Abstract - The paper gives an overview of the project management methodologies. The shallow comparison between the two main groups of methodologies is done. Accent of paper is on scrum, and scrum is explained to details.

Keywords: *Project management methodologies, agile, scrum, sprint, backlog, product*

1. УВОД

Развијање модерних апликација ријетко је дјело појединца. Дизајнирање, кодирање, тестирање и паковање апликација захтјева *cross functional* тим (тим који се састоји од људи из различитих области) који чине дизајнери, програмери, тестери и графички дизајнери који заједно раде на задацима. Ово постаје још захтјевније узимајући у обзир чињеницу да су захтјеви чешће недефинисани и непотпуни него обрнуто. Комбиновање свега наведеног представља процес развијања апликације у циљу испуњавања очекивања корисника. Апликација треба да буде стабилна, да је достављена у временски остављеном року и унутар предвиђаног буџета.

Почетком 1970-тих креће развој разних методологија намјењених превазилажењу ових потешкоћа. На почетку најпопуларнија је била *waterfall* (водопад) методологија. Међутим, током 1980-тих, величина и сложеност апликација се повећава и расте потреба за флексибилношћу захтјева.

С једне стране ово значи да апликација неће у потпуности испуњавати захтјеве корисника, а са друге стране повећава се квалитет апликације и њена стабилност. Било како било, резултат ове методологије неријетко представљају апликације које не садрже карактеристике и функционалности неопходне да се постигне зацртани циљ, као и апликације које имају пуно мана због проблема с квалитетом.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Александар Купусинац, ванр. проф.

2. ОСНОВНЕ МЕТОДОЛОГИЈЕ

Постоји много приступа управљању пројектом, али се посебно истичу два начина организације на које се касније ослањају остали. То су:

1. Традиционални приступ управљању пројектом и
2. Агилни

Традиционални приступ је линиеаран, све фазе се изводе у секвенци. Пројекат је, унапријед, у потпуности испланиран без икаквог простора остављеног потенцијалним промјенама. Због ових разлога, традиционални приступ се сусреће са проблемима буџета и времена.

Агилни ставља акценат на тимски рад, сарадњу са клијентом и флексибилност. Представља итеративан процес, и труди се да у свакој итерацији спреми низ функционалности који може да буде показан клијенту.

Традиционални је најпогоднији за мале и једноставне пројекте, који прате линеарни приступ. Изненадне промјене или било какве компликације могу да блокирају читав процес и приморају тим да се врати уназад како би кренуо из почетка.

Агилни је најпогоднији за комплексне пројекте. Комплексни пројекти могу да имају више повезаних фаза и свака фаза може да буде зависна од много других за разлику од линеарног гдје једна фаза зависи тачно од једне друге фазе. Разлог због којег је агилни приступ бољи за комплексније пројекте јесте управо могућност адаптације.

На слици [1] је приказана табела поређења карактеристика традиционалног и агилног приступа.

Characteristics	Traditional Project Management	Agile Project Management
Organizational structure	Linear	Iterative
Project scale	Large-scale	Small and medium scale
Development model	Life cycle model	Evolutionary delivery model
User requirements	Clearly defined before coding or implementation	Interactive input
Client involvement	Low	High
Restart Cost	High	Low
Development process	Life cycle model	Evolutionary delivery model
Development model	Fixed	Easily changeable
Testing	Once coding is done	Every iteration
Architecture	Creates current and predictable requirements	Creates current requirements
Requirements	Standard and known in advance	Emergent with rapid changes

Слика 1. Традиционални наспрам агилног приступа

3. SCRUM

Scrum представља подкуп Агилне методологије, помоћу њега може да се имплементира агилно окружење. Пошто агилни процес нема кораке које треба испратити, **scrum** пружа начин да се унесе агилни процес у пројекат. Може да се користи као улазна тачка за друга агилна окружења. Није базиран само на софтверске пројекте, већ има доста ширу примјену.

Scrum је:

- Лаган
- Једноставан за разумјети
- Тежак за савладати

Scrum се користи при развоју софтвера, хардвера, уградбених система, интерактивних мрежа, аутономних возила, школа, влада, маретинга, управљања операцијама организација и скоро свему што користимо у свакодневном животу. Стручност **scrum**-а је у раду с комплексним проблемима. Посебно је ефективан при итеративном и инкременталном преносу знања. Суштину **scrum**-а чини мали тим људи.

Појединачан тим је веома флексибилан и лако се адаптира. Они настављају да раде сами, у склопу другог или других тимова који развијају, пуштају у промет, користе и одржавају производ који користе хиљаде људи. Они сарађују и заједно раде користећи софистициране развојне архитектуре и циљана окружења.

Оно што раздваја **scrum** од других агилних методологија јесу специфични концепти и праксе које су подјелени у три категорије:

- Улоге
- Артефакти
- Временске кутије

Scrum значајно повећава продуктивност и смањује вријеме развијања наспрам класичног **waterfall** процеса. Омогућава организацији да се прилагоди лако брзо мјењајућим захтјевима, и да произведе производ који испуњава еволуирајуће пословне захтјеве. Предности **scrum**-а:

- Повећање квалитета достављаних производа
- Боље подношење промјена(и очекивање промјена)
- Пружање бољих естимација, а при том коришћење мање времена да се постигну
- Боља контрола над производом као и његовим стањем

Три стуба којих свака имплементација емпиријске контроле процеса треба да се ослања су:

- Транспарентност
- Инспекција
- Адаптација

Scrum тим се састоји од:

- **Product owner**
- **Development team**
- **Scrum master**

Product owner је одговоран за повећање вриједности производа који је резултат рада **development** тима. Начин на који се ово ради варира од организације до организације, **scrum** тима до **scrum** тима и индивидуе до индивидуе.

Development тим се састоји од професионалаца који раде на достављању потенцијалних верзија инкремента **off-Done** које могу бити пуштене у коришћење на крају сваког **sprint**-а. Инкремент **off-Done** захтјева **sprint review**. Само чланови тима могу да креирају инкремент **off-Done**.

Scrum master је одговоран за промовисање и подстицање **scrum**-а као што је дефинисано у **Scrum Guide**. **Scrum master** ово постиже тако што помаже свима да разумију **scrum** теорију, правила и вриједности.

Срце **scrum**-а јесте **sprint**, спакован у временску кутију од мјесец или мање, током кога инкремент **off-Done**, искористив и спреман. **Sprint** има константно трајање у току развојног труда. Нови **sprint** почиње одмах након закључка претходног **sprint**-а.

Циљ **sprint**-а представља оно што може да буде имплементирано из **product backlog**-а. Пружа смјернице тиму поводом тога зашто граде тај инкремент. Креиран је у току планирања **sprint**-а. Циљ **sprint**-а пружа мало флексибилности **development** тиму што се тиче функционалности које ће бити имплементирани. Селектоване ставке **product backlog**-а достављају једну кохерентну функционалност, која може да буде циљ **sprint**-а. Циљ може да буде било која кохерентност која захтјева да тим заједно ради на повезаним функционалностима наспрам оних засебних. Како тим ради, имају у уму циљ. Како би га постигли, имплементирају функционалности. Уколико се испостави да посао није оно што је очекивано, тим сарађује с **product owner**-ом о потенцијалним промјенама домена **product backlog**-а за текући **sprint**-а.

Scrum прописује четири формална догађаја за инспекцију и адаптацију:

- **Sprint Planning** – планирање спринта
- **Daily Scrum** – дневни састанци
- **Sprint Review** – ревизија спринта
- **Sprint Retrospective** – ретроспектива спринта

Планирање **sprint**-а даје одговоре на следећа питања:

- Шта ће бити одрађено у инкременту који представља резултат планираног **sprint**-а?
- Како да се постигне тај инкремент?

Daily Scrum је догађај ограничен на петнаест минута дневно у коме учествује **development** тим. Одвија се сваки дан у току **sprint**-а. На њему, тим планира посао који намјерава да одради у наредна двадесет-четири сата. Овако оптимизује сарадњу тима и перформансе прелазећи кроз посао урадјен у претходних двадесет-четири сата, тј. од последњег **daily scrum**-а, као и прогнозирање надолazeћег посла. **Daily scrum** се одвија сваки дан на истом мјесту у исто вријеме, како би се смањила комплексност.

Структуру састанка поставља **development** тим, и може да буде извршена на различите начине уколико је фокус идаље на напретку ка циљу. Неки тимови ће да користе питања, други ће бити више окренути дискусији.

Примјери структуре:

- Шта сам јуче урадио како бих помогао тиму да постигне циљ?
- Шта ћу данас да урадим како бих помогао тиму да постигне циљ?
- Да ли видим неке препреке које спречавају мени или тим у достизању циља?

Чланови тима се често нађу након састанка како би детаљније продискутовали, или прилагодили, поново испланирали остатак посла.

Sprint review се одржава на крају сваког **sprint**-а како би се размотрио инкремент и адаптирао **product backlog** уколико је то потребно. У току овог догађаја, тим и дионичари разматрају оно што је постигнуто. Базирано на том и на свим измјенама, учесници разматрају следеће ставке које би могле да се одраде како би оптимизовале вриједност.

Ово је састанак информативног карактера, није састанак информисања о стању пројекта. Он такође служи за презентацију инкремента како би добили реакцију и потенцијалне измјене.

Ретроспектива представља могућност разматрања перформанси и прављење плана шта треба да се поправи за наредни **sprint**.

Product backlog представља сортирана листу која се састоји од свега што је потребно производу. Она је јединствени извор захтјева свих промјена које треба да се одраде на производу. **Product owner** је одговоран за **product backlog**, укључујући његов садржај, доступност и сортирање. Он није никад комплетан. Најранији развој креира преглед иницијалних и најбоље разјашњених захтјева. Напредује као што и производ и окружење у којем ће да се производ користи напредује. Динамички је, константно се мјења у циљу идентификације онога што је потребно производу како би био пристојан, компетитиван и користан. Уколико производ постоји, постоји и **product backlog**.

Sprint backlog је низ ставки из **backlog**-а селектоване за **sprint**, плус план како доставити инкремент производа и остварити циљ **sprint**-а. Представља прогнозу развојног тима о томе које ће функционалности да буду наредни инкремент и који посао је потребан да би се оне доставиле. Чини видљивим сав посао који развојни тим идентификује као неопходан за достизање **sprint** циља. Како би се осигурао константни напредак он укључује бар једну ставку унапређења високог приоритета, која је идентификована на претходној ретроспективи. Представља план са довољно детаља који се мјења са радом који је објашњен на дневном **scrum**-у. Развојни тим модификује спринт беклог у току спринта. Промјене се јављају како тим ради на плану и сазнаје више о послу који је потребан да се постигне циљ.

Инкремент представља збир свих ставки из **product backlog**-а које су завршене у току **sprint**-а и вриједност инкремента свих претходних **sprint**-ова. На крају **sprint**-а, нови инкремент мора бити **Done**, што значи да мора да буде у стању у ком може да се користи и да испуњава **scrum**-ову дефиницију **off Done**. Инкремент је тијело које се састоји из готовог посла који подржава емпиризам. Корак је ка циљу или визији. Он мора бити оспособљен за употребу без обзира да ли **product owner** одлучи да га достави.

Scrum се ослања на транспарентност. Одлуке да се оптимизује вриједност и контрола ризика су базиране на запаженом стању артефакта. Уколико је транспарентност комплетна, ове одлуке имају добру подлогу. У случају да није, онда ове одлуке могу да имају мане, вриједност може да опадне и ризици могу да се повећају.

Кад се ставка **product backlog**-а или инкремент прогласе за **Done**, сви треба да разумију шта то тачно значи. Дефиниција зависи од тима, али је кључно да сви разумију шта значи да је посао завршен, како би се осигурала транспарентност. Ово је дефиниција **off Done** за **scrum** тим, и користи се како би се процјенило који је посао готов.

Термини који се најчешће користе приликом рада унутар **scrum** окружења:

- **Daily Scrum** – кратки састанак на којем се описује шта се ради тај дан
- **Definition of Done** – кад је пројекат завршен и сусреће заједничка очекивања
- **Development Team** – одговорни за управљање послом који је повезан за сваки **sprint**
- **Forecast** – ствари коришћене за имплементацију **sprint**-а
- **Increment** – мали кораци који воде ка завршетку пројекта
- **Product Backlog** – посао који треба да се одради тачно задатим редом
- **Product Backlog Refinement** – када **product owner** и тим додају детаље **product backlog**-у
- **Product Owner** – менаџер који је одговоран за производ и тим
- **Ready** – обострано разумјевање **product backlog** када се планира **sprint**
- **Scrum** – окружење за сарадњу тима приликом рада на комплексним пројектима
- **Scrum Board** – визуелизација рада који је у току у виду табле
- **Scrum Master** – улога која је слична тренеру који помаже тиму коришћењем свог експертског знања
- **Scrum Team** – **product owner**, **development** тим и **scrum master**
- **Self-Organization** – аутономија тима унутар оквира пројектиних циљева
- **Sprint** – кратки задаци, секвенцијални, по завршетку једног одмах креће други
- **Sprint Backlog** – оно што тим треба да уради у **sprint**-у

- **Sprint Goal** – циљ *sprint*-а, сочно речено
- **Sprint Planning** – кратко планирање наредног *sprint*-а
- **Sprint Retrospective** – картка ретроспектива завршеног *sprint*-а
- **Sprint Review** – кратка оцјена *sprint*-а како би се унаприједио следећи
- **Stakeholder** – иницијатор пројекта који није члан тима
- **Velocity** – просјечна количина *product backlog* претворена у инкремент пројекта у току *sprint*-а

4. ЗАКЉУЧАК

Циљ овог раде јесте да помогне при одабиру методологије којом ће да се управља пројектом. На само питање како одабрати праву методологију има краatak одговор: зависи од пројекта. Углавном све се своди на повјерење, да ли клијенат вјерује агенцији да ће успјешно да достави пројекат, и да ли је спреамн да плаћа неуспјех на путу до успјеха.

У крајности није битно за коју се методологију одлучите, докле год акценат остаје на оном што је заиста битно, а то је сам пројекат. Методологије служе само томе да олакшају процес достављања пројекта.

5. ЛИТЕРАТУРА

- [1] Jeff Sutherland and Ken Schwaber, Scrum Guide, 2010, <https://www.scrumguides.org/>
- [2] Jason Westland, Scrum Methodology, 2018, <https://www.projectmanager.com/blog/scrum-methodology>
- [3] Jeff Sutherland, Ken Schwaber and others, Agile manifesto, 2001, <http://agilemanifesto.org/>
- [4] Kira Carr, Agile Project Management Vs. Traditional Project Management, 2017, <https://www.knowledgehut.com/blog/agile/agile-project-management-vs-traditional-project-management>

Кратка биографија:

Милица Говедарица рођена је 1994. године у Требињу, Босна и Херцеговина. Завршила је гимназију „Перо Слијепчевић“ у Гацку 2013. године. Факултет техничких наука у Новом Саду уписала је 2013. године. Испунила је све обавезе и положила све испите предвиђене студијским програмом.

MICROSOFT AZURE-OBA ARHITEKTURA BEZ SERVERA**MICROSOFT AZURE SERVERLESS ARCHITECTURE**Десанка Тодић, *Факултет техничких наука, Нови Сад***Област – РАЧУНАРСТВО И АУТОМАТИКА**

Кратак садржај – У раду су описане архитектуре система информационих технологија од првих система грађених само традиционалном монолитном архитектуром, затим преко система грађених помоћу микросервиса до систем без сервера и коришћења готове инфраструктуре за градњу жељеног система. Акцент рада је стављен на архитектуру без сервера **Microsoft Azure** и његове **Azure** функције које су детаљно објашњене.

Кључне ријечи: традиционална монолитна архитектура, микросервисна архитектура, архитектура без сервера, cloud computing, **Azure** функције, платформа као сервис, **Azure** портал, **API**

Abstract – The paper describes the architecture of the information technology system from the first systems built only by traditional monolithic architecture, then through the systems built using microservice to the serverless architecture and using the ready infrastructure to build the desired system. The focus of work is on the Microsoft Azure serverless architecture and its Azure Functions that are explained in detail.

Keywords: traditional monolithic architecture, microservice architecture, serverless architecture, cloud computing, Azure functions, platform as a service, Azure portal, API

1. УВОД

Од алата преко машина до рачунара, људи траже начине за аутоматизовање поновљеног рада и стандардизацију задатака којима се том приликом баве тако да могу да се усредсреде на специјализоване доприносе завршавању задатака и рјешавању проблема. Паралелно с тим, индустрија информационих технологија је еволуирала, и људи су се трудили да повећају ефикасност на свим нивоима система, од централне процесорске јединице до фарми сервера, у нади да ће се остварити највећи излазни учинак система.

Архитектура без сервера је тачка у коју ова два тока конвергирају. То је тачка у којој је појединачни напор најсјајније фокусиран на специфичан задатак, а неупотребљивост система је сведена на минимум.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био др Александар Купусинац, ванр. проф.

У свету без сервера, програмери стварају рјешења умјесто инфраструктуре и надгледају извршење. Не закупају се најчешће беспослена виртуелне машина него се плаћају само временски интервали у којима она ради. У **Microsoft Azure**-у постоји много потрошних сервиса које се могу повезати заједно како би се формирало комплетно рјешење. Кључна компонента архитектуре без сервера **Microsoft Azure** јесу **Azure** функције.

2. ТРАДИЦИОНАЛНА АРХИТЕКТУРА АПЛИКАЦИЈА

Већина традиционалних апликација изграђена је коришћењем трослојног шаблона архитектуре апликација: слој презентације (кориснички интерфејс), средњег слоја (функционална логика процеса) и слоја података (рачунарска подаци и приступ подацима). Сваки ниво ради на намјенском серверу и статички је конфигуриран са именом хоста (енг. **hostname**) и **IP** (енг. **Internet Protocol**) адресама сервера осталих нивоа од којих зависи. Ове апликације су распоређене на статичкој инфраструктури која се не мијења. Традиционалне апликације имају веома мало знања о инфраструктури на којој су покренуте и претпостављају да се инфраструктура неће промјенити или пропасти. Трослојна архитектура састоји се из следећих слојева:

- **Презентациони слој.** Ово је највиши ниво апликације. Ниво презентације приказује информације везане за такве услуге као што су прегледање робе, куповина и садржај корпе. Комуницира са другим нивоима помоћу којих се резултати преносе на слој претраживача/клијента и свих осталих нивоа у мрежи. Једноставно, то је слој који корисници могу директно приступати (као што је веб страница или графички кориснички интерфејс оперативног система).
- **Слој апликације (пословна логика, логички ниво или средњи ниво).** Логички ниво се извлачи из нивоа презентације и, као свој слој, контролише функционалност апликације примјеном детаљне обраде.
- **Слој података.** Ниво података укључује механизме перзистености података (сервери базе података, дијелење датотека итд.) и слој за приступ подацима који енкапсулира механизме перзистености и излаже податке. Ниво за приступ подацима треба да обезбједи **API** (енг. **Application programming interface**) нивоу

апликације да излаже методе за управљања складиштеним подацима без излагања или стварања зависности од механизма за складиштење података. Избјегавање зависности од механизма за складиштење дозвољава ажурирања или промјене без да то утиче на клијенате апликационог слоја или да они буду упознати са промјеном. Као и код раздвајања било којег нивоа, постоје трошкови за имплементацију и често побољшана скалабилност и одрживост буде на рачуна опадања преформанси.

3. МИКРОСЕРВИСНА АРХИТЕКТУРА

„Микросервис је техника развоја софтвера – варијанта архитектонског стила сервисно-оријентисане архитектуре (SOA¹-а) који структурира апликацију као скуп услуга који се лако повезују“ [1]. У архитектури микросервиса, сервиси су заокружене цјелине намјењене обављању неке функционалности и протоколи којим се комуницира са овим сервисима имају само неколико правила и пракси, понекад и само једно које је лако пратити. Комуникација између микросервиса се обавља програмским интерфејсима API-јима који су независни од програмског језика нпр. **Representational State Transfer (REST)** – архитектонски стил који дефинише скуп ограничења који се користе за креирање веб сервиса.

Најважнија предност разлагања апликације у различите мање сервисе је у томе што побољшава модуларност и чини апликацију лакшом за разумјевање, развој, тестирање и отпорност на разрушавање архитектуре. Такође, паралелним развојем омогућено је да мали аутономни тимови могу самостално да развијају своје сервисе. Због тога архитектуре базиране на микросервисима омогућавају непрекидну испоруку и примјену у систем.

Додавање нових функционалности у вишеслојној архитектури обично повећа сложеност бекенд (енг. **backend**) слоја (пословна логика) и значајно компликује средњи слој. Код је ипак организован по модулима, па повећање броја функција доводи до повећања броја модула у средњем слоју апликације. Да би се одговорило на повећане захтјеве, скалирање се врши тако што се покреће више инстанци бекенд апликација и сентичним модулима.

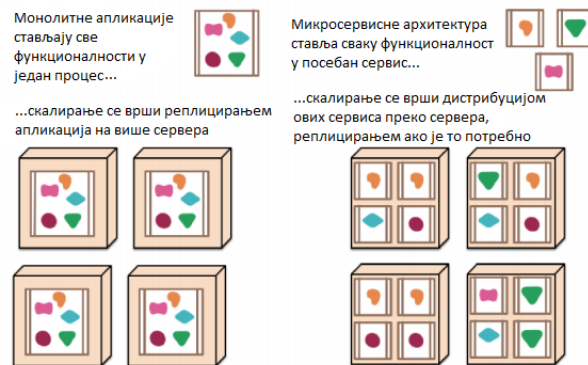
Битне разлике:

- Нема глобализационих модула са “шпагети” кодом.
- Компоненте се могу развијати потпуно независно једна од друге.
- Различити програмски језици, извршна окружења, оперативни системи, хардвер, база података.
- Компоненте се могу релативно једноставно мијењати.

¹ енгл. **Service-oriented architecture** - архитектонски стил развоја софтвера гдје се сервиси који комуницирају међусобно пружају другим софтверским компонентама на услугу захваљујући комуникационом протоколу преко мреже.

- Може се замијенити и комплетна технологија модула.
- Могуће различито скалирање различитих компоненти.

Разлике у скалирању између традиционалне (монолитне) и микросервисне архитектуре представљене и објашњене су на слици 1.



Слика 1. Разлике у скалирању

4. АРХИТЕКТУРА БЕЗ СЕРВЕРА

У основи, архитектура без сервера представља следећу еволуцију сервиса рачунарства у облацима, изграђених на врху **PaaS**-а (платформа као сервис), апстрахујући виртуалне машине, окружење апликације и екстерне зависности путем везивања тако да се програмери могу једноставно фокусирати на код за имплементацију пословне логике.

„Рачунарство у облацима је парадигма информационих технологија која омогућава свеобухватан приступ заједничким базама конфигурираних системских ресурса и сервиса вишег нивоа који се брзо могу обезбиједити уз минималне напоре управљања, често преко интернета“ [3]. Три главне категорије сервиса рачунарства у облацима су: инфраструктура као сервис (**IaaS**), платформа као сервис (**PaaS**) и софтвер као сервис (**SaaS**). „Инфраструктура као сервис (енг. **Infrastructure as a service**) је облик рачунарства у облацима који пружа виртуелизоване рачунарске ресурсе преко интернета“ [4].

„Платформа као сервис (енг. **Platform as a service**) је комплетно окружење за развој и успостављање окружења у облацима, уз ресурсе који омогућавају да пруже све од једноставних апликација заснованих на облацима до софистицираних апликација које користе облаке“ [5].

Поред горе наведених основних облика рачунарства у облацима, важно је споменути и **FaaS** – функција као сервис (енг. **Function as a service**) који представља еволуцију PaaS-а и на његовим принципима изграђена је архитектура без сервера са **Azure** функцијама.

„Софтвер као сервис (енг. **Software as a service**) је модел дистрибуције софтвера у ком независни добављач (енг. **third-party provider**) хостује апликације и чини их доступним корисницима преко интернета“ [6].

4.1. ПРЕДНОСТИ АРХИТЕКТУРЕ БЕЗ СЕРВЕРА

Из пословне перспективе:

- Трошак који произлази из апликације без сервера заснован је на броју извршења функција, мјерено у милисекундама умјесто сатима.
- Омогућава агилност процеса тако што мање јединице које се могу распоређивати резултирају бржом испоруком функција на тржиште, повећавајући способност прилагођавања промјенама.
- Опадају трошкови запошљавања бекенд инжењера.
- Смањени оперативни трошкови пошто су сервери на клауду.

Из перспективе програмера:

- Смањена обавеза пошто нема бекенд инфраструктуре за коју би се могло бити одговорно.
- Лакше оперативно управљање.
- Брже постављање окружења.
- Скалабилно, нема потребе да се брине о броју истовремених захтјева.
- Подстиче иновације пошто је релативно нова архитектура.

Из перспективе корисника апликације:

- Ако компаније користе као конкурентску предност да брже испоручују нове функционалности апликације, онда корисници добијају нове функционалности брже него раније.
- Корисници могу лакше да обезбиједје сопствени бекенд за складиштење (нпр. **Dropbox**, **Google Drive**).
- Апликације ове врсте могу да понуде кеширање на клијентској страни и да тако омогуће боље корисничко искуство када је ван мреже (енг. **offline**).

4.2. МАНЕ АРХИТЕКТУРЕ БЕЗ СЕРВЕРА

Из пословне перспективе:

- Смањена укупна контрола пошто бекенд слој контролише продавац сервиса рачунарства у облацима.
- Може доћи до гашења сервиса који архитектура нуди.
- Сигурносни ризик пошто приступ свему може да има и клауд продавац.
- Сви ови недостаци могу бити ублажени алтернативама отвореног кода (енг. **open-source**), али то повећава трошкове апликације.

Из перспективе програмера:

- Незрела технологија доводи до фрагментације компоненти и нејасних најбољих пракси.
- Архитектонска сложеност за функције које су превише сложене.
- Технички могуће да сусједне функције могу да преносе системске ресурсе испод хаубе, па да се појаве невалидни подаци.
- Тестирање локално је незгодно. Понекад је потребно изложити своју машину мрежи

пребацавањем приватне у јавну адресу да би клауд сервис могао да комуницира са програмеровом машином.

- Трајање извршења је ограничено. Обично 300 секунди сервер има времена да изврши функцију пре него што саопшти да није успио у извршавању. Тако да функције не смију бити превише сложене.
- Недостатак оперативних алата, поготово за дибаговање.

Из перспективе корисника апликације:

- Ако се исправно не имплементира архитектура, апликација би могла пружити лоше корисничко искуство као резултат повећане учесталости захтјева.

5. АРХИТЕКТУРА БЕЗ СЕРВЕРА НА AZURE ПЛАТФОРМИ

Архитектура без сервера на **Azure** платформи укључује **Azure** функције, **Logic Apps** (апликације за логику) и **Event Grid** (мрежа догађаја). Ови сервиси раде заједно и повезују се са бројним другима ресурсима. Серверлес платформа функционише са свиме из базе и складишта до аналитике и машинског учења тј. вјештачке интелигенције.

Поред ових сервиса, могуће је користити и **Application Insights** (увид у апликацију), серверлес платформу за снимање дијагностичких трагова и телеметрије (технологија која омогућава даљинско мјерење и пренос информација од удаљених локација до оператора). **Application Insights** су доступни за апликације свих врста (десктоп, мобилне или веб) као и имплементације серверлеса.

6. АРХИТЕКТУРА БЕЗ СЕРВЕРА СА AZURE ФУНКЦИЈАМА

Архитектура без сервера омогућава чисту раздвојеност између кода и окружења у коме је хост. Имплементира се код у функцији која је позвана преко окидача. Након што се функција изврши, сви ресурси који су којој били потребни могу бити ослобођени. Окидач може да буде ручни, да се окида послјег одређеног времена, **HTTP** захтјева, или отпремање датотеке. Резултат окидача је извршење кода. Иако сервери без сервера варирају, већина пружа приступ унапред дефинисаним **API**-има и везама за поједностављивање задатака као што је додавање у базу података или резултате чекања.

Архитектура без сервера је архитектура која се у великој мјери ослања на апстракцији удаљеног окружења хоста да би се фокусирала на код. Може се сматрати као архитектура са мање сервера.

Рјешења заснована на контејнерима омогућава програмерима да користе постојеће скрипте за билд (енг. **build** – направити, подизати) да би обавили објаву кода до спремне слике серверлеса. Остале имплементације користе постојећа **PaaS** рјешења за пружање скалабилне архитектуре.

Апстракција значи да **DevOps** тим не мора да објезбеђује нити управља серверима нити специфичним контејнерима.

Постоје двије верзије **Azure Functions** фрејмворка. Стара верзија подржава комплетан **.NET** фрејмворк, а нова омогућава **.NET Core** и развој између платформи. Језици који су подржани поред **C#** су **JavaScript**, **F#** и **Java**, а ту је и експериментална подршка за **Python**, **PHP**, **TypeScript**, **Batch**, **Bash** и **PowerShell**. Функције креиране на порталу објезбеђују синтаксу из скрипти. Функције креиране као самостални пројекти могу бити отпремљење на сервер уз потпуну подршку и могућности које пружа платформа.

7. ЗАКЉУЧАК

Овом раду описане су архитектуре помоћу којих се могу градити системи информационих технологија. Прва архитектура која је објашњена јесте традиционална монолитна архитектура. Затим слиједи описивање микросервисне архитектуре које представља напредак у односу на традиционалну. Даље је објашњена архитектура без сервера са акцентом на архитектуру без сервера помоћу **Azure** функција. Ниједна од ових архитектура није савршена и има и предности и мане, стога у зависности од проблема који се ријешава, нека је боља, а нека лошија у односу на другу. За апликације које треба да ријешавају сложене проблеме, коначна архитектура обично обухвата више различитих архитектура и тиме се ствара хибридна апликација.

8. ЛИТЕРАТУРА

[1] Chen, Lianping (2018). Microservices: Architecting for Continuous Delivery and DevOps, https://www.researchgate.net/publication/323944215_Microservices_Architecting_for_Continuous_Delivery_and_DevOps

[2] Gautambiztalkblog, Microservices approach vs traditional approach, 2016,

<https://gautambiztalkblog.com/tag/microservices-approach-vs-traditional-approach/>

[3] Wikipedia, Cloud computing, 2018,

https://en.wikipedia.org/wiki/Cloud_computing

[4] Margaret Rouse, Infrastructure as a Service (IaaS), 2018,

<https://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-IaaS>

[5] Microsoft Azure, What is PAAS, 2018,

<https://azure.microsoft.com/en-us/overview/what-is-paas/>

[6] Margaret Rouse, Software as a Service (SaaS), 2016,

<https://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service>

[7] The agile admin, What is DevOps, 2017,

<https://theagileadmin.com/what-is-devops/>

[8] Margaret Rouse, Internet of Things (IoT), 2018,

<https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>

[9] Vangie Beal, Big Data, 2018,

https://www.webopedia.com/TERM/B/big_data.html

[10] Robin Weston, Serverless architectures and continuous delivery, 2017,

<https://www.gocd.org/2017/06/26/serverless-architecture-continuous-delivery/>

Кратка биографија:

Десанка Тодић рођена је 1994. године у Брчком, Босна и Херцеговина. Завршила је гимназију „Васо Пелагић“ у Брчком 2013. године. Факултет техничких наука у Новом Саду уписала је 2013. године. Испунила је све обавезе и положила све испите предвиђене студијским програмом.

ADAPTACIJA I ANALIZA SELENIUM TESTOVA U C# PROGRAMSKOM JEZIKU**ADAPTATION AND ANALYSIS OF SELENIUM TESTS IN C# PROGRAMMING LANGUAGE**

Boško Šogura, *Fakultet tehničkih nauka, Novi Sad*

Oblast – SOFTVERSKO INŽENJERSTVO

Kratak sadržaj – U okviru ovog rada opisan je Selenium radni okvir za implementaciju veb automatskih testova. Manipulisanje i izvršavanje akcija nad elementima veb stranice pomoću WebDriver-a. Upotreba Page Object obrasca kao najbolje prakse prilikom kreiranja test scenarija. Analiza test scenarija u najpopularnijim pretraživačima.

Ključne reči: Selenium, WebDriver, Automatski testovi, Page Object obrazac

Abstract – The subject of this paper was description of the Selenium framework for implementation Web automated tests. Manipulate and execute actions over Web page elements using the WebDriver. Using the Page Object Model as best practice in creating a test scenario. Analysis of the test scenario in the most popular browsers.

Keywords: Selenium, WebDriver, Test automation, Page Object Model

1. UVOD

Testiranje predstavlja samo jedan dio procesa kontrole kvaliteta softvera. Različite metode se koriste za testiranje različitih slojeva softvera, a ono što je svima zajedničko jeste da se mogu izvoditi manuelno ili automatski. I jedan i drugi način imaju svoje prednosti i mane. U poslednje vrijeme sve se više teži automatizaciji jer se automatizacijom testiranja skraćuje i samo vrijeme testiranja.

2. AUTOMATSKO TESTIRANJE

Softveri sa kritičnom misijom prolaze kroz rigorozne funkcionalne testove, koji su često podržani radnim okvirima (eng. Framework) za automatsko testiranje. Automatizacija ovakvih platformi kao i održavanje kvaliteta softvera kroz verzije su od ključnog značaja za poslovanje. Preduzeća se često suočavaju sa dilemom između balansiranih troškova i upravljanja resursima kako bi se osiguralo da platforme za testiranje pokrivaju sve poslovne scenarije i da isporučeni softver nema grešaka [1]. Implementacijom odgovarajuće platforme za automatsko testiranje, preduzeća mogu značajno povećati brzinu i tačnost procesa testiranja.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinać, van. prof.

2.1. Zašto Framework?

Framework definiše način preduzeća da organizuje stvari pomoću “Jedinstvenog Standarda”. Prateći ovaj standard, projektni tim postigao bi:

- prikaz automatskih testova bez skripta,
- testove vođene podacima,
- koncizno izveštavanje,
- standardizovano skriptovanje i konzistentnost i
- implementaciju i ponovnu upotrebljivost.

3. SELENIUM

Selenium je prenosni radni okvir za testiranje veb aplikacija. On obezbeđuje alat za pisanje testova bez potrebe da se uči ili zna test skripting jezik (eng. Selenium IDE). Takođe, pruža test jezik specifičan za domen testiranja (Selenese) u kom se mogu pisati testovi na najpopularnijim programskim jezicima kao što su: C#, Groovy, Java, Perl, PHP, Python, Ruby i Scala. Testovi se mogu pokretati u većini modernih veb pretraživača. Selenium može da se koristi na Windows, Linux i macOS platformama. On je open-source softver, objavljen pod Apache 2.0 licencom, što znači da ga veb programeri mogu preuzeti i koristiti bez naknade [2].

3.1. Selenium korisnički API

Kao alternativa pisanju testova u Selenese-u, testovi se mogu pisati i u različitim programskim jezicima. Ovi testovi potom komuniciraju sa Selenium-om pozivom metoda njegovog korisničkog API-ja. Selenium trenutno nudi korisničke API-je za Javu, C#, Ruby, JavaScript i Python. Sa novom verzijom Selenium-a 2, uveden je novi korisnički API, koji ima WebDriver kao centralnu komponentu. Međutim, stari API, koji koristi Selenium klase, još uvijek je podržan i održava se.

3.2. Selenium WebDriver

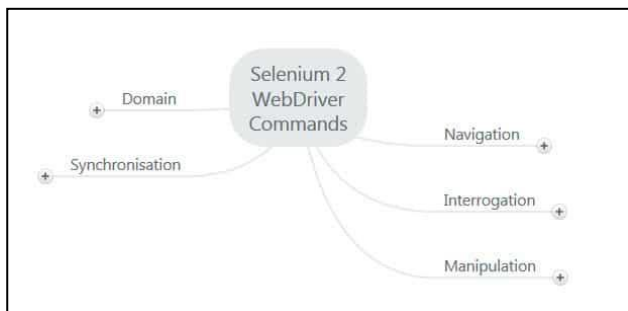
Selenium WebDriver je naslednik Selenium RC-a. On prihvata komande iz Selenese-a ili korisničkog API-ja i šalje ih pretraživaču. Ovo se implementira pomoću posebnog driver-a za pretraživače, koji šalje komande samom pretraživaču i vraća rezultat, odnosno odgovor pretraživača. Većina driver-a za pretraživače zaista pokreće i pristupa aplikaciji pretraživača (kao što su Firefox, Chrome, Internet Explorer, Safari, Edge).

Tu je i HtmlUnit driver, koji omogućava simuliranje pretraživača (nema grafičku reprezentaciju pretraživača). Za razliku od Selenium-a 1, gdje je Selenium server bio neophodan za pokretanje testova, Selenium WebDriver-u nije potreban poseban server za izvršavanje testova. Umjesto toga, WebDriver direktno pokreće instancu pretraživača i kontroliše je.

Međutim, Selenium Grid se može koristiti u kombinaciji sa WebDriver-om za izvršavanje testova na udaljenim sistemima.

Kada je moguće, WebDriver koristi funkcije operativnog sistema a ne JavaScript komande koje su specifične za pretraživač, radi pokretanja istog. Ovaj pristup zaobilazi probleme sa razlikama između komandi operativnog sistema i JavaScript-a, uključujući bezbjednosna ograničenja [3].

Ovo je savršena mapa uma za testere koji koriste WebDriver na projektima automatskog testiranja. Na slici 1. prikazano je osnovnih pet kategorija Selenium WebDriver komandi [4].



Slika 1. Osnovne kategorije WebDriver komandi

3.3. Page Object Model obrazac

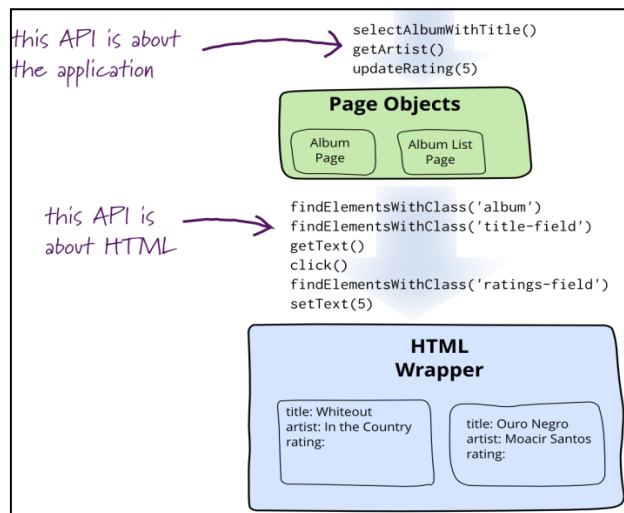
POM je najčešće korišćen dizajn obrazac u kom se svaka veb stranica (ili bar značajne) posmatra kao klasa. Na svakoj od ovih klasa stranica (otuda i Page Object) mogu se definisati njihovi elementi i metode koje su specifične za tu stranicu. Svaka klasa treba da predstavlja stranicu veb aplikacije. Ovaj obrazac predstavlja sloj između test skripti i korisničkog interfejsa i enkapsulira funkcije veb stranice [5].

Page Factory je dodatak POM obrascu koji se koristi za inicijalizaciju veb elemenata koji su definisani na stranici, u klasi koja "preslikava" tu veb stranicu. Elementi se mogu definisati koristeći anotacije pomoću Page Factory obrasca [6].

Kada se pišu testovi koji testiraju određenu veb stranicu, potrebno je da se koriste elementi i funkcionalnosti te veb stranice.

Međutim, ukoliko takvi testovi direktno manipulišu HTML elementima, oni će biti krhki na promjene u korisničkom interfejsu. POM obrazac omogućava manipulaciju veb elementima u test scenarijima bez kopiranja po HTML-u u samim test skriptama.

Slika 2. prikazuje kakve metode bi trebalo da se koriste u test skriptama. Osnovno pravilo POM obrasca jeste da omogući softverskom klijentu da vidi i uradi sve ono što može i čovjek na stvarnim veb stranicama.



Slika 2. Primjer Page Object Model

4. OPIS PROJEKTOG RJEŠENJA

4.1. ICodeFactory sajt

ICodeFactory je kompanija koja izvršava proces razvoja softvera od ideje do proizvodnje [7]. Na njihovom oficijalnom sajtu www.ICodeFactory.com mogu se pronaći osnovne informacije o firmi, usluge koje pružaju, tehnologije i projekti na kojima rade, najnovije vijesti vezane za poslovanje firme, konkurse za otvorene pozicije u firmi kao i mnoge druge informacije.

Sajt takođe pruža nekoliko administratorskih funkcionalnosti, ali je prvo potrebno prijaviti se na stranici za prijavljivanje. Funkcionalnosti koje prijavljeni korisnik može da izvrši (zavisno od njegovih prava) su:

- pregled, unos, izmjene i brisanje aktuelnih vijesti koje se prikazuju u sekciji "News",
- pregled, unos, izmjene i brisanje trenutno otvorenih pozicija koje se prikazuju u sekciji "Career",
- pregled, unos, izmjene i brisanje drugih aktivnih korisnika,
- pregled, kreiranje, izmjene i brisanje aktivacionih kodova za razne servise,
- pregled log fajlova i
- pregled, čuvanje i skidanje dokumenata.

4.2. Test scenario

Test scenario predstavlja niz akcija koje zajedno testiraju određenu funkcionalnost. U daljem nastavku ovog poglavlja biće analiziran scenario za dodavanje nove vijesti na ICodeFactory sajtu. Koraci ovog scenarija su:

1. Otvaranje početne stranice sajta i klik na poruku o kolačićima.
2. Klik na dugme za prijavljivanje i prelazak na stranicu za prijavljivanje.
3. Unos korisničkog imena i lozinke u odgovarajuća polja i klik na dugme za prijavljivanje.
4. Navigacija na administratorsku stranicu za pregled vijesti kao prijavljen korisnik.

5. Klik na dugme za dodavanje nove vijesti i prelazak na stranicu za dodavanje vijesti.
6. Unos potrebnih informacija za kreiranje vijesti i klik na dugme za dodavanje.
7. Provjera očekivanih i dobijenih rezultata nakon dodavanja vijesti.

4.3. Inicijalizacija testa

Ovaj dio testa je zajednički za sve test scenarije. U ovom koraku se radi inicijalizacija baze, kreiranje WebDriver objekta, pronalazak i klik na poruku o kolačićima.

4.4. Prelazak na stranicu za prijavljivanje

NewsRelatedScenarios klasa sadrži test scenarije koji testiraju funkcionalnosti vezane za vijesti sajta. Slika 3. prikazuje metodu za inicijalizaciju spomenute klase. Nakon inicijalizacije testa, vrši se kreiranje objekta početne stranice (eng. HomePage).

Sledeći korak testa jeste prelazak na stranicu za prijavljivanje.

To se izvršava pozivom NavigateToLoginPage metode nad futer objektom početne stranice. Metoda vraća objekat stranice za prijavljivanje čime se završava ovaj korak testa.

```
[TestFixture]
-- references | Bosko Sogura, 2 days ago | 1 author, 1 change
class NewsRelatedScenarios : BaseScenario
{
    private ApplicationStructure.AdminPages.NewsPage newsPage;

    [SetUp]
    -- references | Bosko Sogura, 2 days ago | 1 author, 1 change
    public override void Initialize()
    {
        base.Initialize();

        // Login for all these tests as content editor and navigate to admin news page:
        var homePage = new HomePage(Driver);
        var loginPage = homePage.Footer.NavigateToLoginPage();
        homePage = loginPage.LoginAsContentEditor();
        newsPage = homePage.MainMenu.NavigateToAdminNewsPage();
    }
}
```

Slika 3. Prikaz Initialize metode klase koja sadrži scenarije o vijestima

NavigateToLoginPage metoda locira dugme za prelazak na stranicu za prijavljivanje, izvršava klik nad dugmetom i zatim vraća kreiranu instancu stranice za prijavljivanje.

4.5. Prijavljivanje korisnika

U ovom koraku se unose korisničke informacije u odgovarajuća polja na stranici za prijavljivanje. Na slici 3. u metodi za inicijalizaciju, može se vidjeti metoda LoginAsContentEditor koja se poziva nad objektom stranice za prijavljivanje. Na slici 4. je prikazana metoda za prijavljivanje korisnika sa pravima menadžera sadržaja.

Metoda prihvata opcione parametre koji su korisničko ime i lozinka.

Ukoliko se ne navedu parametri, korisničko ime i lozinka se uzimaju iz app.config fajla gdje su definisani različiti korisnici, sa različitim pravima, odnosno grupama kojim pripadaju.

Zatim se poziva metoda LoginUser kojoj se kao parametri proslijeđuju korisničko ime i lozinka.

```
5 references | Bosko Sogura, 3 days ago | 1 author, 1 change
public HomePage LoginAsContentEditor(string username = null, string password = null)
{
    if (username == null || password == null)
    {
        username = UserStore.Editor.Username;
        password = UserStore.Editor.Password;
    }

    return LoginUser(username, password);
}
```

Slika 4. Prikaz metode za prijavljivanje korisnika kao menadžera sadržaja

4.6. Navigacija na administratorsku stranicu za pregled vijesti

Poslednja komanda u litalize metodi, koja je prikazana na slici 3. jeste navigacija na administratorsku stranicu za pregled vijesti (eng. NewsPage). U predhodnom koraku testa, nakon prijave korisnika, vraćena je instanca početne stranice sajta. U ovom koraku, pozivamo metodu NavigateToAdminNewsPage nad objektom menija koji se nalazi u klasi početne stranice.

4.7. Navigacija na administratorsku stranicu za dodavanje vijesti

Nakon izvršavanja koda Initialize metode, prelazi se na glavnu metodu testa. Na slici 5. je prikazana Home_LoginAndAddNews_CheckNewsTitleInTable metoda, koja predstavlja glavni dio test scenarija koji se opisuje. Na početku ove metode se deklariraju osnovne informacije kao što su naslov vijesti i broj postojećih vijesti do sada. Ove informacije se koriste i na kraju za samu verifikaciju, odnosno provjeru da li je test izvršio očekivane promjene. Nakon definisanja informacija za verifikaciju, poziva se metoda za prelazak na administratorsku stranicu za dodavanje vijesti (eng. AddNewsPage).

```
[Test]
0 references | Bosko Sogura, 3 days ago | 1 author, 1 change
public void Home_LoginAndAddNews_CheckNewsTitleInTable()
{
    // arrange:
    var testTitle = "test title";

    var oldNumberOfNews = newsPage.NewsCount;
    var addNewsPage = newsPage.NavigateToAddNews();

    // act:
    newsPage = addNewsPage.InsertTitle(testTitle)
        .InsertDescription("Some description")
        .InsertContent("insert text")
        .TogglePublish()
        .SelectPublishDate(DateTime.Now.AddDays(-5))
        .SaveNews();

    // assert:
    Assert.AreEqual(oldNumberOfNews + 1, newsPage.NewsCount);
    Assert.AreEqual(testTitle, newsPage.GetNewsTitle(3));
}
```

Slika 5. Prikaz glavnog dijela testa za dodavanje nove vijesti

NavigateToAddNews metoda vrši lociranje odgovarajuće stavke menija pomoću identifikatora, pa zatim izvršava akciju klika nad njim. Nakon klika očekuje se prelazak na stranicu za dodavanje vijesti. Ovaj korak se završava kreiranjem objekta administratorske stranice za dodavanje vijesti.

4.8. Unos potrebnih informacija i kreiranje vijesti

Ovo je najvažniji korak test scenarija. Takođe, korak u kom najčešće padaju testovi. U ovom koraku se unose potrebne informacije za kreiranje nove vijesti u odgovarajuća polja i zatim se pokušava snimanje unijete vijesti. Na slici 5. se vidi nekoliko poziva različitih metoda nad objektom stranice za dodavanje vijesti, koje vrše simulaciju unošenja određenih informacija u polja za unos. InsertTitle metoda vrši lociranje polja za unos u koji se upisuje naslov vijesti. InsertDescription metoda se koristi za dodavanje opisa vijesti, itd.

4.9. Provjera dobijenih i očekivanih rezultata pri završetku testa

Nakon unosa vijesti, vrši se provjera da li je ona stvarno unijeta. Prvo se provjerava da li se broj vijesti na stranici za pregled uvećao za jedan u odnosu na stari broj prije dodavanja. Druga provjera je da li maloprije unijeti naslov vijesti odgovara naslovu u tabeli gdje se izvršilo dodavanje. Ukoliko obje provjere vrte pozitivne rezultate, smatra se da je test uspješno izvršen.

4.10. Usporedna analiza

Tabela 1. Usporedna analiza testova u Chrome-u, Firefox-u i Internet Explorer-u

Pretraživač	Prosječno vrijeme testa [s]	Postotak uspješnosti [%]
Chrome	21,86	95
Firefox	31,64	80
Internet Explorer	41,93	100

Analiza je izvršena tako što se test scenario, koji je opisan u poglavlju 4.2. pokrenuo 20 puta u svakom od navedenih pretraživača. Zatim se izračunalo prosječno vrijeme testa i postotak uspješnosti izvršavanja u svakom od pretraživača.

Iz tabele 1. može da se zaključi da Chrome pretraživač ima najkraće prosječno vrijeme izvršavanja testova. Takođe, uspješnost ovih testova je veoma visoka (95%), što ga stavlja na prvo mjesto ove analize. Mozilla Firefox ima duže prosječno vrijeme izvršavanja testova od Chrome-a za čitavih 10 sekundi. Sa uspješnošću testova od 80%, što znači da će svako peto izvršavanje testa da bude neuspješno. Firefox predstavlja najslabiji pretraživač od svih analiziranih kada se uzme u obzir trajanje testova i uspješnost istih. Internet Explorer ima prosječno vrijeme izvršavanja testova za 20 sekundi duže od Chrome-a, i 10 sekundi duže od Firefox-a. Ono što ipak stavlja IE na drugo mjesto ove analize, jeste činjenica da se testovi izvršavaju sa uspješnošću od 100%.

Potrebno je spomenuti i da za inicijalizaciju baze treba u prosjeku 9 sekundi po testu. Što znači da se prosječna vremena testova mogu dodatno skratiti ukoliko se primjene druge metode za inicijalizaciju baze. Prosječno vrijeme testa se može smanjiti još više upotrebom kolačića da bi se izbjegao dio vezan za prijavljivanja korisnika u svakom test scenariju.

5. ZAKLJUČAK

U ovom radu opisan je problem osiguravanja kvaliteta veb aplikacije i njegovo rješenje uvođenjem automatskog testiranja. Automatski testovi su poželjni u svakoj aplikaciji jer testiraju funkcionalnosti sa istim podacima u istom okruženju svaki put. Tako da se, nakon svake izmjene na aplikaciji koja se nalazi u produkciji, mogu pokrenuti testovi koji testiraju sve funkcionalnosti aplikacije. Tako se može lako utvrditi bilo kakva greška nastala sa poslednjim izmjenama.

Možda i najvažnija osobina opisanog projekta za testiranje jeste lako prebacivanje između instanci pretraživača u kojima se test scenariji izvode. To se radi izmjenom odgovarajuće promjenljive app.config fajla. Na taj način se pokriva veliki dio različitih produkcionih okruženja (pretraživača) u kojima korisnici mogu da koriste aplikaciju.

Moguća poboljšanja ovog projekta za testiranje bi bila upotreba kolačića kao mehanizma za prijavljivanje korisnika. Pomoću kolačića, mogu se direktno u pretraživač unijeti podaci o korisniku, tako da bi veb aplikacija smatrala da je korisnik već prijavljen i dozvolila mu pristup određenim funkcionalnostima. Takođe, upotreba snimaka stanja baze prije i posle testa i zatim vraćanje na stanje prije testa. Trenutno se koristi inicijalizacija baze iznova prije svakog testa, što znači da se čitava baza obriše i opet popuni odgovarajućim podacima. Upotreba ovih mehanizama bi dodatno smanjila vrijeme izvršavanja test scenarija.

6. LITERATURA

- [1] <http://toolsqa.com/selenium-webdriver/automation-framework-introduction/> (pristupljeno u septembru 2018.)
- [2] <https://www.seleniumhq.org/> (pristupljeno u septembru 2018.)
- [3] "The Architecture of Open Source Applications: Selenium WebDriver", www.aosabook.org. (pristupljeno u septembru 2018.)
- [4] <https://www.swtestacademy.com/selenium-webdriver-api/> (pristupljeno u oktobru 2018.)
- [5] <http://www.assertselenium.com/automation-design-practices/page-object-pattern/> (pristupljeno u oktobru 2018.)
- [6] <https://www.guru99.com/page-object-model-pom-page-factory-in-selenium-ultimate-guide.html> (pristupljeno u oktobru 2018.)
- [7] <https://www.icodetfactory.com/about> (pristupljeno u oktobru 2018.)

Kratka biografija:



Boško Šogura rođen je u Bijeljini 1993. god. Master rad na Fakultetu tehničkih nauka iz oblasti Softverskog inženjerstva – Testiranje softvera odbranio je 2018.god. kontakt: bolesogura93@gmail.com

OSNOVNI PRINCIPI, ŠABLONI I PRAVILA ZA PISANJE ČISTOG KODA

BASIC PRINCIPLES, PATTERNS AND RULES FOR WRITING CLEAN CODE

Saša Momčilović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U okviru rada opisane su najbolje prakse koje dovode do čistog dizajna prilikom rešavanja određenog problema. Pored dizajna akcenat je stavljen i na implementaciju, tj. na konkretne preporuke prilikom pisanja koda kako bi bio na visokom stepenu održivosti.

Ključne reči: Objektno orijentisano programiranje, SOLID principi, dizajn šablona, C#, čist kod.

Abstract – This document describes the best practices that lead to a clean design when solving a particular problem. In addition to design, emphasis is placed on the implementation, i.e. on concrete recommendations when writing code to be of a high degree of maintenance.

Keywords: Object oriented programming, SOLID principles, design patterns, C#, clean code.

1. UVOD

U radu biće više reči o tome kako napisati kod sa što većim stepenom održivosti, vodeći računa prvenstveno na kvalitet dizajna, a nakon toga i na kvalitet samog koda. Takođe, biće reči i o tradicionalnom načinu programiranja, kao i o problemima koje je imao proceduralni pristup programiranju, a koji su bili motiv za uvođenje velikih promena u ovoj oblasti.

Kao osnovni korak za uprošćavanje kompleksnog koda koji je prouzrokovao proceduralnim pristupom programiranju, biće predstavljeni aspekti objektno orijentisanog programiranja koji utiču na poboljšanje čitkosti koda i rešavanje problema proceduralnog programiranja. Nakon toga, akcenat će biti na SOLID principima. Biće opisano kako pridržavanje ovih principa pomaže da se dizajn i kod učine što kvalitetnijim i čitljivijim. Nakon SOLID principa prelazi se na dizajn šablone koje strukturno dosta pomažu poboljšanju dizajna, a samim tim i koda. Nakon aspekata koji se u većoj meri odnose na dizajn, potrebno je opisati kako da dobar dizajn ne bude narušen pisanjem nečitkog koda.

2. MOTIVACIJA

Rešenja problema u oblasti informacionih tehnologija su podložna promenama zahteva od strane klijenata. S tim u vidu, potrebno je bilo ustanoviti određena pravila koja će prouzrokovati pisanje koda takvog da je uz najmanje napore moguće dodati implementacije novih zahteva, a da se pritom ne naruši dotadašnja implementacija.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinać, van. prof.

U ranim fazama sveta informacionih tehnologija nailazilo se na probleme koji su znatno povećavali vreme potrebno da se neki posao obavi. Kod koji je bio pisan je bio neodrživ, nastao je veliki nedostatak programera.

3. OBJEKTNO ORIJENTISANO PROGRAMIRANJE

Kao što je već napominjano, programiranje je evoluiralo kroz više faza. Jedna od njih je bilo proceduralno programiranje za čije je probleme bilo potrebno naći adekvatno rešenje. Problemi proceduralnog programiranja u najvećoj meri rešeni su ustanovljavanjem objektno orijentisanog programiranja. Uvođenjem objektno orijentisanog programiranja, rešeni su problemi nepotrebnog čuvanja globalnih podataka, nemogućnost ponovne iskoristivosti koda, redundantnost koda, pravljanje složenih programa i tako dalje. Rešavanje pomenutih problema omogućeno je kroz četiri osnovna principa objektno orijentisanog programiranja, a to su: enkapsulacija, nasleđivanje, apstrakcija i polimorfizam.

Objektno orijentisano programiranje ujedinjuje funkcije i varijable u jednu celinu nazvanu klasom. Na taj način je omogućeno odvajanje logike u celine, tako da se svaka logika može dodeliti celini kojoj pripada.

3.1. Enkapsulacija

Enkapsulacija je jedan od četiri osnovna koncepta objektno orijentisanog programiranja. Omogućava zaštitu atributa i funkcija objedinjenih u celinu. Zaštita elemenata klase od klijentskih klasa, tj. klase koji je koriste, omogućena je kroz modifikatore pristupa elementima klase. Klasi koja koristi enkapsuliranu klasu se omogućava da vidi samo neophodne njene elemente, dok su interni elementi klase sakriveni, na taj način se klasa štiti spoljnih promena koje se ne bi trebale desiti, uprošćava se interfejs klase što daje određenu fleksibilnost prilikom dalje implementacije i testiranja klase.

3.2. Nasleđivanje

Nasleđivanje je koncept objektno orijentisanog programiranja koji omogućava izvođenje novih klasa iz postojeće, tako da nove klase dobijaju osobine bazne klase uz mogućnost dodavanja novih osobina. Osnovni oblici nasleđivanja su: jednostruko, višestruko, hijerarhijsko, višeslojno i kombinovano nasleđivanje. Sa stanovišta čistog koda nasleđivanje pruža: izbegavanje ponovnog pisanja koda, ponovna iskoristivost koda, jednostavnije proširivanje koda.

3.3. Apstrakcija

Apstrakcija je koncept koji omogućava definisanje ponašanja u baznoj klasi koje će biti implementirano u

klasi naslednici. Apstrakcija se postiže apstraktnim klasama koje mogu imati apstraktne članove koji se obeležavaju ključnom rečju **abstract** i koje nemaju telo. Prilikom kreiranja konkretnih klasa iz definisanih apstrakcija, apstraktnim članovima je potrebno implementirati ponašanje. Osim apstraktnih klasa apstrakcija se može postići i interfejsima.

3.4. Polimorfizam

Polimorfizam je koncept koji omogućava izvršavanje istih funkcija na različit način. Pomenute funkcije, tj. metode se nalaze u konkretnim klasama koje su podklase zajedničkoj baznoj klasi. Sa aspekta čitkog koda polimorfizam ima veliki udeo. Njime se može izbeći ne baš najčistije implementacije koje imaju veliki broj **if else** grananja. Polimorfizmom je omogućeno napraviti više objekata takvih da predstavljaju instance klasa koji su podklase iste bazne klase. Te klase će izvršavati istu metodu na različit način.

4. S.O.L.I.D principi

SOLID je kolekcija najboljih praksi objektno orijentisanih dizajn principa koji se mogu primeniti na novi dizajn, omogućavajući ostvarenje raznih ciljeva kao što su, veća održivost, intuitivna lokacija koda, slaba povezanost komponenti i tako dalje. [1]

SOLID je akronim koji predstavlja sledeće principe:

- SRP (**The Single Responsibility Principle**)
- OCP (**The Open Closed Principle**)
- LSP (**The Liskov Substitution Principle**)
- ISP (**The Interface Segregation Principle**)
- DIP (**The Dependency Inversion Principle**)

4.1. The Single Responsibility princip

SRP je princip koji nalaže da se svaka pojedinačna odgovornost nalazi u okviru tačno jedne klase. U situacijama kada se više odgovornosti nađe u okviru jedne klase može doći do problema prilikom dodavanja novog koda vezanog za jednu odgovornost. U tim situacijama može doći do narušavanja ostalih odgovornosti koji se nalaze u toj klasi. Moduli u koje je potrebno rasporediti odgovornosti trebaju imati što niži stepen međusobne povezanosti.

```
public class Student
{
    public void Save()...

    public double DoTheExam(Exam exam)...

    public Student FindById(long id)...
}
```

Slika 1. Klasa student koja narušava SRP princip

Na slici 1 nalazi se klasa **Student** koja ima metode **Save**, **DoTheExam** i **FindById**. Svaka od ovih metoda jeste usko povezana sa studentom, međutim ne predstavlja svaka metoda i odgovornost klase student. Klasa definisana kao na slici ima odgovornost rada sa kolekcijom studenata, što ne bi trebala biti odgovornost pojedinačnog objekta klase **Student**. Promene u implementaciji načina skladištenja studenata mogu uticati na logiku koja bi se zaista trebala nalaziti u klasi **Student**. Da bi se ovakav problem rešio potrebno je u jednoj klasi držati samo jednu odgovornost, tj. izdvojiti logiku vezanu

za skladište podataka u posebnu klasu. Ta klasa bi trebala preuzeti metode za čuvanje i za pronalaženje studenata po jedinstvenom identifikatoru.

4.2. The Open Closed princip

OCP princip nalaže da svaki modul treba biti: otvoren za proširenje i zatvoren za modifikaciju. Implementacije u kojima su klijentska i serverska klasa konkretne, je dobar osnov za primenu OCP principa. Da bi se primenio OCP princip potrebno je da serverska klasa bude implementacija određene apstrakcije. Takođe potrebno je i da klijentska klasa ne barata sa konkretnom implementacijom, već sa pomenutom apstrakcijom.

Počnimo od primera u kojem je potrebno implementirati rad sa zaposlenima u određenom sistemu. Konkretno, u zavisnosti od tipa zaposlenog potrebno je na određeni način izračunati platu koja zavisi od procenta na platu i dodatnog novca. Izvešćemo klasu **BonusCalculator** u kojoj će se u metodi **CalculateBonus** računati bonus za prosleđenog zaposlenog. U metodi je potrebno ispitati kog je tipa zaposleni, što će se čuvati u klasi **Employee**, i na osnovu tipa računati bonus. Prilikom dodavanja novog tipa potrebno je menjati metodu novim slučajem. Ovakav pristup krši OCP. Kao što je pomenuto, potrebno je uvesti apstrakciju nad klijentskom klasom, tj. u ovom slučaju nad klasom **Employee**. Pomenuta apstrakcija će imati onoliko implementacija koliko će biti vrsta zaposlenih, i u njima će biti podaci o procentu bonusa i dodatnom novcu za svakog zaposlenog. **BonusCalculator** klasa će raditi sa apstrakcijom i neće imati potrebu za menjanjem.

4.3. The Liskov Substitution princip

Formalna definicija LSP principa koju je iznela Barbara Liskov, po kojoj je ovaj princip i dobio naziv glasi: „Ako je S podtip tipa T, tada objekti tipa T mogu biti zamenjeni objektom tipa S, bez prekida rada programa.“ [2]

Funkcije koje koriste reference na baznu klasu moraju biti sposobne da koriste objekte klase naslednica bez znanja o tome. [3] Osim pomenutih definicija za LSP princip bitno je da klasa naslednica u svojoj implementaciji ne sme bacati nikakav novi izuzetak. Loš primer modelovanja koji nije u skladu sa ovim principom je u sledećem primeru. Ukoliko imamo klasu **Employee** koja ima referencu ka svom menadžeru. Svaki konkretni zaposleni može implementirati klasu **Employee**. Međutim, u hijerarhiji zaposlenih nemaju svi menadžera, te je potrebno tu logiku izmestiti u novu klasu koja predstavlja zaposlene koji mogu imati menadžera, a koja nasleđuje klasu **Employee**. Sve klase koje predstavljaju zaposlene koji nemaju menadžera će naslediti klasu **Employee**. Dok će klase koje predstavljaju zaposlene koje imaju menadžera naslediti novokreiranu klasu.

4.4. The Liskov Substitution princip

LSP se bavi nedostatkom koji prouzrokuju kompleksni interfejsi koji po svojoj definiciji primoravaju sve klase koje ga implementiraju da definišu implementaciju svih metoda interfejsa. Ukoliko su ti interfejsi preveliki, tj. sa velikim brojem članova, potrebno je te članove pregrupisati u više manjih interfejsa. Recimo da postoji interfejs **IFile** koji ima **property-e Title, Author, Duration, PageCount** i metode **GetVideoSnapshot, Play, Pause, Stop, GetTextContent**. Pomenuti članovi

interfejsa se mogu koristiti za implementaciju nekog fajla. Međutim ukoliko se radi o tekstualnom fajlu, metode kao što su **Play** nemaju svrhu. Potrebno je razdvojiti članove u više interfejsa koji će se moći, po potrebi i u kombinaciji, koristiti tako da ih klase koje predstavljaju bilo koji fajl mogu implementirati bez potrebe implementacije nepotrebnih metoda.

4.5. The Dependency Inversion princip

DIP princip se bazira na programiranju spram apstrakcije, tj. oslobađanja aplikacije zavisnosti od konkretnih implementacija određenih komponenti. To oslobađanje zavisnosti se postiže korišćenjem apstrakcija. Osnovna dva načela DIP principa su: 1. Moduli visokog nivoa ne bi trebali da zavise od modula niskog nivoa, oba trebaju zavisiti od apstrakcija. 2. Apstrakcija ne bi trebala zavisiti od detalja. Detalji trebaju zavisiti od apstrakcije. [4].

U situacijama kada se ne koristi apstrakcija iznad implementacija određenih komponenti onemogućeno je podmetanje druge implementacije prilikom promene zahteva od strane klijenata. Ukoliko dođe do novih zahteva, a ovaj princip nije ispoštovan, kod koji će se morati menjati može da prouzrokuje velike probleme i kršiće većinu SOLID principa. Pored pomenutog problema, veoma je otežano jedinično testiranje ukoliko se ovaj princip ne poštuje.

5. DIZAJN ŠABLONI

Dizajn šablona predstavljaju koncepte rešenja za neke ponavljajuće programerske probleme. Ovakvi šablони imaju veliki stepen ponovne iskoristivosti i veoma su lako prepoznatljivi u svetu programiranja, te će lako biti uočeni od programera koji za njih znaju i time im olakšati snalaženje u kodu koji je u skladu sa nekim šablonom. Postoje tri osnovne grupe dizajn šablona: kreacioni, strukturalni i šablони ponašanja.

5.1. Singleton

Singleton je strukturalni dizajn šablon koji omogućava pristup objektu određene klase sa jedne, svima dostupne, pristupne tačke. Takođe klasa koja implementira ovaj šablon treba obezbediti da postoji samo jedna instanca ove klase. **Singleton** se može kombinovati sa velikim brojem dizajn šablona.

5.2. Abstract factory

Abstract factory je dizajn šablon koji obezbeđuje kreiranje srodnih ili zavisnih objekata bez navođenja njihovih konkretnih klasa. [5] Predstavlja proširenje **Factory** i **Factory Method** dizajn šablona. **Abstract factory** omogućava da se korišćenjem osnovnih koncepata objektno orijentisanog programiranja ispoštuju ključni SOLID principi prilikom rešavanja određenih programerskih problema.

5.3. Composite

Composite dizajn šablon predstavlja strukturu rešenja za probleme u kojima je potrebno organizovati objekte u stablo. Mogu postojati dve vrste objekata, oni koji mogu imati još podobjekata i oni koji predstavljaju listove u stablu. Može se koristiti u kombinaciji sa dosta drugih dizajn šablona. Omogućava proširenja u kodu koje je u skladu sa SOLID principima.

5.4. Chain of responsibility

Chain of responsibility je dizajn šablon koji omogućava raspoređivanje objekata u lanac. Lanac se okida tako što klijentska klasa šalje zahtev prvom objektu u nizu, koji ili obradi zahtev ili ga prosledi sledećem objektu i tako dalje do poslednjeg objekta u lancu. Ovaj omogućava dizajniranje rešenja tako da ne narušava OCP i SRP princip.

5.5. Command

Ovaj dizajn šablon jedan je od najčešće korišćenih šablona. Enkapsulira komande u objekte kojima se može manipulirati. **Command** šablon razdvaja objekat koji zna kako da izvrši operaciju od objekta koji zna kako da je izvede. Ima veliku primenu prilikom kreiranja WPF aplikacija primenom MVVM šablona. Takođe, velika prednost ovog šablona jeste mogućnost implementacije **Undo/Redo** mehanizma.

5.6. Null object

Prilikom rešavanja programerskih problema često se nailazi na problem u kome bi jedno od rešenja bilo vraćanje **null** reference. U tim situacijama dobro je koristiti ovaj šablon. Kreira se objekat koji ima logiku jednaku logici koju bi klijentska klasa izvršila ukoliko bi u ranijoj implementaciji bio vraćen **null**. Često to bude prazna metoda. Ovaj dizajn šablon oslobađa klijentsku klasu od provere **null**-a. Ovaj šablon se može kombinovati sa **Singleton**, **Command**, **Strategy** i drugim dizajn šablonima.

5.7. Repository

Repository je šablon koji omogućava implementaciju rada sa skladištenjem podataka tako da razdvaja sloj poslovne logike aplikacije od samog skladišta gde će poslovna logika znati samo za apstrakciju. Tako da, kada dođe do promene zahteva o skladištu, može se menjati implementacija bez uticaja na poslovnu logiku.

5.8. Strategy

Strategy šablon predstavlja jedan od najjednostavnijih dizajn šablona sa veoma širokom primenom. Ovaj dizajn šablon služi za promenu ponašanja određenog objekta tako da on zna samo za apstrakciju, te mu je moguće po potrebi podmetnuti ponašanje koje odgovara jednoj od konkretnih implementacija te apstrakcije.

5.9. Facade

Facade dizajn šablon obezbeđuje interfejs za set operacija koje se dešavaju u određenom delu aplikacije. **Facade** oblikuje te operacije u jedan podsistem koji zna kojim komponentama da prosleđuje zahteve koji su došli od drugih podsistema. Ovaj dizajn šablon ima veliki broj varijacija.

6. OSNOVNA NAČELA PISANJA ČISTOG KODA

Osim bitnih koncepata i šablona vezanih za dizajniranje rešenja, potrebno je ispoštovati i smernice vezane za konkretnu implementaciju koje dovode do kompletno čitljivog i održivog rešenja.

6.1. Smisljeno imenovanje

Prilikom pisanja koda veoma bitnu ulogu ima imenovanje promenljivih, klasa, metoda i parametara. Potrebno je davati imena koja otkrivaju nameru člana koji se kreira.

Ukoliko ime zahteva komentar onda je potrebno razmisliti o boljem imenu. Postoje situacije u kojima je kod, koji je potrebno napisati, krajnje jednostavan, međutim i pored toga može dovesti do velike zabune ukoliko dođe do loših imenovanja. Imena klasa trebaju biti imenice, tako da se izbegavaju reči poput **Processor**, **Data**, **Manager** ili **Info**.

6.2. Metode

Prilikom pisanja metode treba obratiti pažnju na sve komponente metode: naziv, povratnu vrednost, broj parametara, tip parametara, naziv parametara, dužinu tela metode, strukturu tela metode i tako dalje.

Osnovna pravila vezana za pisanje metoda govore da telo metode treba biti što kraće i da metoda treba imati što manje parametara. Ukoliko je u metodi potrebno koristiti neke od naredba grananja, npr **if**, treba nastojati da u **if** bloku bude samo jedna linija koda, pa makar to bila jedna metoda koja obuhvata više radnji, nazvana tako da iz njenog naziva bude jasno šta će se obaviti u njoj. Kroz parametre metoda potrebno je prosledivati samo podatke koji su potrebni metodi. Prilikom pisanja metoda treba izbegavati opcione parametre, pogotovo ako je u pitanju virtualna metoda. Takođe, dobro je izbegavati **out** i **flag** parametre.

6.3. Komentari

U situacijama kada se kod pretvori u nečist i neodrživ, veliki deo programera se odluči da doda komentare kako bi olakšao čitanje tog koda. Međutim, najbolja reakcija na takav kod jeste refaktorisanje. Takođe, ukoliko postoji neki deo koda koji nije najčitljiviji ne bi se trebali ostavljati komentari iznad njega. Najbolja opcija u takvim situacijama je ubacivanje tog koda u metodu koja će svojim imenom reći šta taj kod radi. Ukoliko programer u određenim situacijama nije u mogućnosti da odradi neki deo posla odlučuje se da ostavi **TODO** komentar, koji služi kao podsetnik. Iako su takvi komentari korisni u današnjim razvojnim okruženjima, treba nastojati izbegavati potrebu za ostavljanje istih.

Dodatne neophodne podatke za određenu metodu ili klasu moguće je dopisati u **summary** blok koji se nalazi iznad klase ili njenih članova.

6.4. Korišćenje uslova

Prilikom grananja koda na više mogućih tokova koriste se **boolean** promenljive. Ukoliko je **boolean** promenljiva nazvana na adekvatan način ona se može koristiti u **if** grananju, bez poređenja sa **true** ili **false** vrednostima, jer bi se time dodatno uprostito kod, a bio bi i čitljiviji.

```
bool passwordLengthIsValid;  
  
if (_password.Length > 6)  
{  
    passwordLengthIsValid = true;  
}  
else  
{  
    passwordLengthIsValid = false;  
}
```

Slika 2. Kompleksniji način dodele vrednosti **boolean** promenljivama

Na slici 2 prikazan je komplikovaniji način dodele vrednosti **boolean** promenljivama koji se može znatno uprostiti načinom dodele sa slike 3.

```
var passwordLengthIsValid = _password.Length > 6;
```

Slika 3. Uprošćena dodela vrednosti **boolean** promenljivama

6.5. Rukovanje greškama

U slučaju kada dođe do određene predviđene greške u kodu potrebno je na adekvatan način obavestiti klijentsku klasu. Rukovanje vraćene poruke o eventualnoj grešci može biti zaboravljeno na klijentskoj klasi, te je potrebno baciti izuzetak. Osnovno pravilo za rukovanje izuzecima jeste da ih treba bacati što ranije, a hvatati što kasnije. Za neko specifično ponašanje treba dodati posebnu klasu koja nadležuje klasu **Exception**.

6.6. Ostala pravila o čistom kodu

Postoji još mnogo pravila koja dovode do čitko napisanog koda, kao što su: dopisivanje donje crte ispred privatnog polja, izbegavanje korišćenja regiona, brisanje neiskorišćenih **using**-a, formatiranje koda, korišćenje **var** ključne reči, korišćenje principa „**tell, dont ask**“, izbegavanje hardkodiranja, umereno korišćenje **Linq** biblioteke.

7. ZAKLJUČAK

U radu su predstavljena osnovna načela, principi i šabloni koji prouzrokuju pisanje boljeg dizajna i samog koda prilikom rešavanja određenih programerskih problema. Današnje aplikacije dostižu veoma visok nivo kompleksnosti, dolazi do velikog broja promene klijentskih zahteva, te je jako bitno voditi računa o pisanju koda koji ima visok stepen održivosti.

8. LITERATURA

- [1] Pablo's SOLID Software Development, LosTechies.com
- [2] Adaptive Code via C#: Agile coding with design patterns and SOLID principles, Gary McLean Hall
- [3] The Liskov Substitution Principle, Robert C. Martin
- [4] Agile Principles, Patterns and Practices in C#, Robert C. Martin
- [5] Design Patterns - Elements of Reusable Object Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides

Kratka biografija:



Saša Momčilović rođen je 09.02.1994. godine u Zrenjaninu. Srednju tehničku školu „Mileva Marić“ u Titelu, smer ekonomski tehničar, završio je 2013. godine. Iste godine upisao je Fakultet tehničkih nauka u Novom Sadu. Ispunio je sve obaveze i položio je sve ispite predviđene studiskim programom.

U realizaciji Zbornika radova Fakulteta tehničkih nauka u toku 2018. godine učestvovali su sledeći recenzenti:

Aco Antić	Dušan Uzelac	Matija Stipić	Slobodan Dudić
Aleksandar Erdeljan	Duško Bekut	Milan Rapajić	Slobodan Krnjetin
Aleksandar Ristić	Đorđe Ćosić	Milan Simeunović	Slobodan Morača
Bato Kamberović	Đorđe Lađinović	Milan Trifković	Sonja Ristić
Biljana Njegovan	Đorđe Obradović	Milan Trivunić	Srđan Kolaković
Bogdan	Đorđe Vukelić	Milan Vidaković	Srđan Popov
Kuzmanović	Đula Fabian	Milena Krklješ	Srđan Vukmirović
Bojan Batinić	Đura Oros	Milica Kostreš	Staniša Dautović
Bojan Lalić	Đurđica Stojanović	Milica Miličić	Stevan Gostojić
Bojan Tepavčević	Filip Kulić	Mijodrag Milošević	Stevan Milisavljević
Bojana Beronja	Goran Sladić	Milovan Lazarević	Stevan Stankovski
Branislav Atlagić	Goran Švenda	Miodrag	Strahil Gušavac
Branislav Nerandžić	Gordana	Hadžistević	Svetlana Nikoličić
Branislav Veselinov	Milosavljević	Miodrag Zuković	Tanja Kočetov
Branislava Kostić	Gordana Ostojić	Mirjana	Tatjana Lončar -
Branislava	Igor Budak	Damnjanović	Turukalo
Novaković	Igor Dejanović	Mirjana Malešev	Toša Ninkov
Branka Nakomčić	Igor Karlović	Mirjana Radeka	Uroš Nedeljčević
Branko Milosavljević	Ivan Beker	Mirko Borisov	Valentina Basarić
Branko Škorić	Ivana Katić	Miro Govedarica	Velimir Čongradec
Damir Đaković	Ivana Kovačić	Miroslav	Veran Vasić
Danijela Lalić	Ivana Miškeljin	Hajduković	Veselin Perović
Darko Čapko	Jasmina Dražić	Miroslav Popović	Vladimir Katić
Darko Marčetić	Jelena Atanacković	Mitar Jovanović	Vladimir Strezoski
Darko Reba	Jeličić	Mladen Kovačević	Vlado Delić
Dejan Ubavin	Jelena Borocki	Mladen Radišić	Vlastimir Radonjanin
Dejana Nedučin	Jelena Kiurski	Momčilo Kujačić	Vuk Bogdanović
Dragan Ivanović	Jelena Radonić	Nemanja	Zdravko Tešić
Dragan Ivetić	Jovan Petrović	Stanisavljević	Zoran Anišić
Dragan Jovanović	Jovanka Pantović	Nemanja Sremčev	Zoran Brujić
Dragan Kukolj	Ksenija Hiel	Nikola Đurić	Zoran Jeličić
Dragan Mrkšić	Laslo Nađ	Nikola	Zoran Mitrović
Dragan Pejić	Lazar Kovačević	Jorgovanović	Zoran Papić
Dragan Šešlija	Leposava Grubić	Nikola Radaković	Željen Trpovski
Dragana Bajić	Nešić	Ninoslav Zuber	Željko Jakšić
Dragana	Livija Cvetičanin	Ognjen Lužanin	
Konstantinović	Ljiljana Vukajlov	Pavel Kovač	
Dragana Šarac	Ljiljana Cvetković	Peđa Atanasković	
Dragana Štrbac	Ljubica Duđak	Petar Malešev	
Dragoljub Šević	Maja Turk Sekulić	Predrag Šiđanin	
Dubravka Bojanić	Marko Todorov	Radivoje Dinulović	
Dušan Dobromirov	Marko Vekić	Radovan Štulić	
Dušan Gvozdenac	Maša Bukurov	Slavica Mitrović	
Dušan Kovačević		Slavko Đurić	