

Interaktivno web-sučelje za analizu rada pretvarača

Mišljenović, Miloš

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:944861>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-23**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**INTERAKTIVNO WEB-SUČELJE ZA
ANALIZU RADA PRETVARAČA**

Diplomski rad

Miloš Mišljenović

Osijek, 2021.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit**

Osijek, 20.09.2021.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime studenta:	Miloš Mišljenović
Studij, smjer:	Diplomski sveučilišni studij Elektrotehnika
Mat. br. studenta, godina upisa:	D-1248, 06.10.2019.
OIB studenta:	91866537152
Mentor:	Prof. dr. sc. Denis Pelin
Sumentor:	Andrej Brandis
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Prof. dr. sc. Kruno Miličević
Član Povjerenstva 1:	Prof. dr. sc. Denis Pelin
Član Povjerenstva 2:	Dr. sc. Dragan Vulin
Naslov diplomskog rada:	Interaktivno web-sučelje za analizu rada pretvarača
Znanstvena grana rada:	Elektrostrojarstvo (zn. polje elektrotehnika)
Zadatak diplomskog rada:	Za odabrane elektroničke energetske pretvarače koji se koriste na laboratorijskim vježbama iz Osnova energetske elektronike napraviti interaktivno web- sučelje za prikazivanje karakterističnih vlnanih oblika struja i napona na osnovu unešenih parametara pretvarača. SUMENTOR s FERIT-a: Andrej Brandis Tema je rezervirana za studenta: Miloša Mišljenovića
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	20.09.2021.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 11.10.2021.

Ime i prezime studenta:

Miloš Mišljenović

Studij:

Diplomski sveučilišni studij Elektrotehnika

Mat. br. studenta, godina upisa:

D-1248, 06.10.2019.

Turnitin podudaranje [%]:

11%

Ovom izjavom izjavljujem da je rad pod nazivom: **Interaktivno web-sučelje za analizu rada pretvarača**

izrađen pod vodstvom mentora Prof. dr. sc. Denis Pelin

i sumentora Andrej Brandis

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD	1
2. PREGLED PODRUČJA TEME	2
3. ANALIZA RADA PRETVARAČA	3
3.1. Izravni istosmjerni pretvarači napona	3
3.1.1. Uzlazni istosmjerni pretvarači napona	4
3.1.2. Silazni istosmjerni pretvarači napona	8
4. BIBLIOTEKA REACT	11
4.1. Prezentacijski jezik HTML	11
4.2. CSS	12
4.3. Programski jezik JavaScript	13
4.4. Instalacija React-a	14
4.5. Korištenje dodatnih paketa u Reactu	14
4.6. Redux	15
4.7. Platforma Firebase	17
4.8. Biblioteka CanvasJS	18
5. IZRADA INTERAKTIVNOG SUČELJA ZA UZLAZNI ISTOSMJERNI PRETVARAČ NAPONA	19
5.1. Izrada komponenti koje su upotrebljive na više mjesta	19
5.2. Kreiranje skladišta, akcija i reduktora	20
5.3. Unos parametara i upotreba Redux-a	23
5.4. Izrada grafa karakterističnih valnih oblika	24
6. REZULTATI INTERAKTIVNOG WEB-SUČELJA	28
6.1. Korištenje interaktivnog web-sučelja	28
6.2. Usporedba rezultata simulacije i mjerenja	29
7. ZAKLJUČAK	38
LITERATURA	39
SAŽETAK	40

ABSTRACT	40
ŽIVOTOPIS	41
PRILOG	42

1. UVOD

Energetski pretvarači su danas sve češće u upotrebi. Mogu se naći i u najjednostavnijim kućanskim aparatima pa sve do električnog auta. Upravo rad takvih pretvarača te sama njihova analiza rada simulirana je kroz ovaj rad. Interaktivno web-sučelje omogućuje korisnicima lakši uvid u rezultate analize te kako određeni parametri utječu na karakteristične valne oblike napona i struje. Kolika je sklopna frekvencija te koliki je utjecaj reaktivnih komponenata na odziv, najbolje se uočava iz interaktivnih simulacija.

U praktičnom dijelu diplomskog rada bilo je potrebno napraviti simulaciju za svaki pretvarač koji se koristi na laboratorijskim vježbama iz kolegija Osnova energetske elektronike. Potrebno je za svaki pretvarač napraviti matematički model i uz pomoć React-a napraviti interaktivno web-sučelje koje uz pomoć CanvasJS-a prikazuje karakteristične valne oblike. Na osnovu dobivenih karakterističnih valnih oblika napona i struje korisnici dobivaju bolji uvid u način rada pretvarača.

Kroz treće poglavlje rada odrađena je analiza rada izravnih istosmjernih pretvarača. Četvrto poglavlje opisuje neke od osnovnih svojstava React-a, Firebase-a, Redux-a i programskih jezika. Za rad u React-u potrebno je poznavanje svega što je napisano u poglavlju četiri. Kroz peto poglavlje detaljno je objašnjeno kako je napravljena simulacija za jedan pretvarač. Šesto poglavlje predstavlja sam rad i testiranje na interaktivnom web-sučelju.

2. PREGLED PODRUČJA TEME

U priručniku [1], nalaze se analiza rada izravnih istosmjernih pretvarača, neizravnih istosmjernih pretvarača, neupravljivih ispravljača, upravljivih ispravljača te izmjenjivača. Iz literature preuzete su slike karakterističnih valnih oblika te na osnovu tih slika napravljena je analiza pretvarača. Na osnovu mjerenja i gotovih laboratorijskih vježbi napravljena je usporedba rezultata mjerenja i simulacije.

Iz knjige [2], preuzeta je analiza rada u diskontinuiranom načinu rada uzlaznog istosmjernog pretvarača napona. Preuzeta je upravljačka karakteristika, srednja vrijednost struje induktiviteta L_e te izraz faktora \tilde{R}_{uz} , iz kojeg je kasnije određen faktor vođenja. Također je preuzet i detaljan opis rada svih energetskih pretvarača, proces pretvorbe osnovnih pretvaračkih komponenti

U dokumentaciji [3], opisano je što je to React, od koga je razvijen te što je sve potrebno od osnovnih znanja za korištenje

Osnove HTML-a, način pisanja elemenata, popis elemenata, objašnjen je u dokumentaciji [4]

Osnove CSS-a, način pristupanja klasama te primjer i rezultat ispisa, objašnjen je u dokumentaciji [5]

Osnove JavaScript-a, definicija objašnjeno je u dokumentaciji [6]

Iz dokumentacije [7], preuzet je primjer koda za React-router paket te detaljno je objašnjeno koja linija što radi

Način rada Redux-a preuzet je iz dokumentacije [8]

Način rada akcija i reduktora, objašnjen je u dokumentacije [9]

Firebase je omogućio da aplikacija bude vidljiva svima, besplatan hosting. Popis mogućnosti nalazi se u dokumentaciji [10]

Iz literature [11], preuzeta je biblioteka CanvasJS i uvezena je u React, te uz pomoć CanvasJS nastali su svi grafovi u simulaciji

Popis elemenata koji mogu biti tip za unosno polje, nalazi se u dokumentaciji [12]

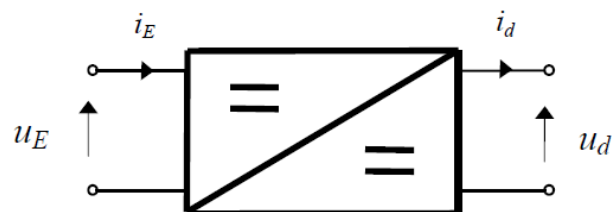
Način rada Omit-a preuzet je iz dokumentacije [13]

3. ANALIZA RADA PRETVARAČA

Rad interaktivnog web-sučelja zasniva se na poznavanju izraza za karakteristične valne oblike koji moraju unaprijed biti poznati. U analizi pretvarača napona proučavat će se dva načina rada pretvarača. Prvi način rada bit će kontinuirani način rada, a drugi će biti diskontinuirani način rada. Kontinuirani način rada se sastoji od dva intervala rada. Diskontinuirani način rada sastoji se od 3 intervala rada. Prvi interval rada predstavlja samo vođenje upravljivog ventila, dok drugi interval rada predstavlja samo vođenje neupravljivog ventila, a treći interval rada pojavljuje se u diskontinuiranom načinu rada i predstavlja interval u kojemu ne vodi niti jedan ventil.

3.1. Izravni istosmjerni pretvarači napona

Izravni istosmjerni pretvarači napona su uređaji koji povezuju istosmjerne mreže različitih napona. Izravni istosmjerni pretvarači napona povezuju naponski uvor i iduktivno trošilo ili strujni izvor i kapacitivno trošilo. Simbol izravnog istosmjernog pretvarača napona prikazan je na slici 3.1. [1].

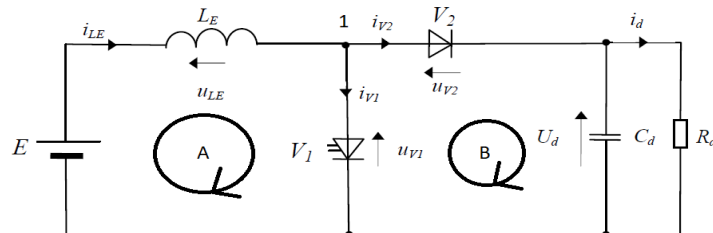


Slika 3.1. Simbol izravnog istosmjernog pretvarača napona[1]

Postoje dvije topologije istosmjernih pretvarača napona. Prva topologija istosmjernog pretvarača napona je silazni istosmjerni pretvarači napona (engl. *buck converter*, *step-down converter*). Druga topologija izravnih istosmjernih pretvarača napona je uzlazni istosmjerni pretvarači napona (engl. *boost converter*, *set-up-converter*).

3.1.1. Uzlazni istosmjerni pretvarači napona

Nadomjesna shema spoja uzlaznog pretvarača napona prikazana je na slici 3.2. U analizi pretvarača uzima se u obzir da su sve pretvaračke komponente idealne. Dodatno je pretpostavljena valovita struja izvora.



Slika 3.2. Shema uzlaznog pretvarača napona[1]

U kontinuiranom načinu rada, u prvom intervalu vodi ventil V1, a ne vodi ventil V2, a u drugom intervalu vodi ventil V2, a ne vodi ventil V1.

Nakon raspisivanja KZN-a dobivaju se jednadžbe:

$$A) \quad E = u_{LE} + u_{V1} \quad (3-1)$$

$$B) \quad u_{V1} = u_{V2} + u_d \quad (3-2)$$

Nakon raspisivanja KZS-a dobivaju se jednadžbe:

$$1) \quad i_{LE} = i_{V1} + i_{V2} \quad (3-3)$$

U intervalu 1, vodi ventil V1 pa je pad napona:

$$u_{V1} = 0 \quad (3-4)$$

Nakon ubacivanja jednadžbe (3-4) u (3-1) dobiva se:

$$E = u_{LE} \quad (3-5)$$

Jednadžba (3-5) može se zapisati kao:

$$E = L_E \frac{di_{LE}}{dt} \quad (3-6)$$

Nakon integriranja jednadžbe (3-6) dobije se izraz za struju kroz induktivitet:

$$i_{LE} = \frac{E}{L_E} t + i_{LE}(0) \quad (3-7)$$

gdje je $i_{LE}(0) = I_0$ početna vrijednost struje induktiviteta.

Interval 1 traje do trenutka t_1 kada ventil V1 dobije impuls za isklapanje.

U intervalu 2 vodi ventil V2, pa izraz (3-6) može se uvrstiti u (3-2), dobije se izraz:

$$E = L_E \frac{di_{LE}}{dt} + U_d \quad (3-8)$$

Nakon integriranja izraza (3-8) dobiva se izraz za struju izvora i_{LE} tijekom intervala 2:

$$i_{LE} = \frac{E - U_d}{L_E} (t - \alpha T_S) + i_{LE} + (\alpha T_S) \quad (3-9)$$

gdje je $i_{LE}(\alpha T_S) = I_1$ početna vrijednost struje induktiviteta u intervalu 2, a α je faktor vođenja:

$$\alpha = \frac{T_{V1}}{T_S} \quad (3-10)$$

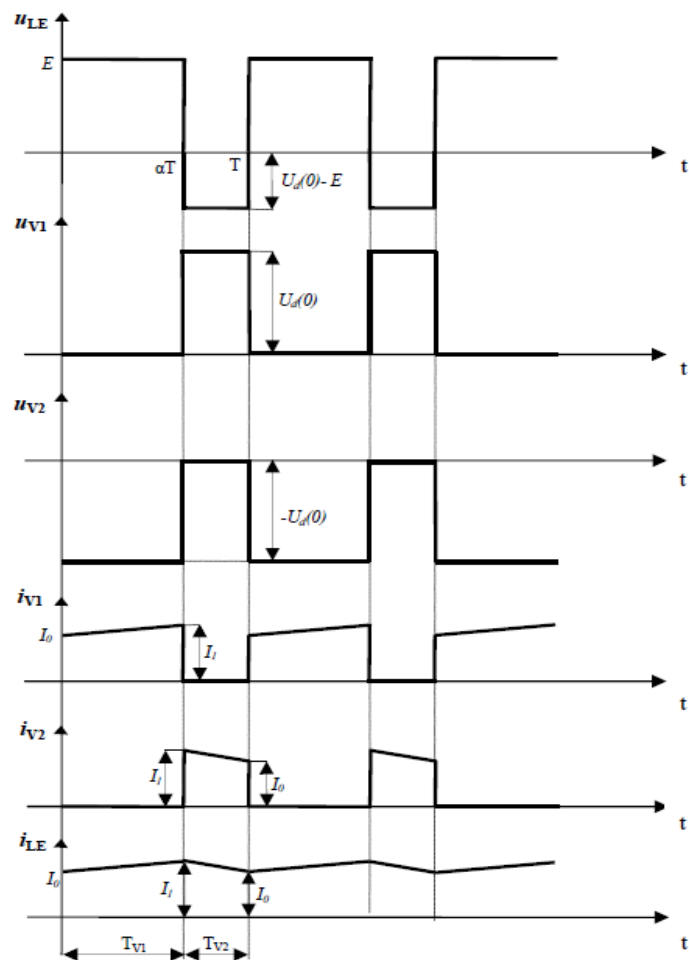
Kod periodičnog režima rada, napon na induktivitetu $U_{LE}(0) = 0$, za valni oblik napona na induktivitetu sa slike 3.3. vrijedi:

$$\alpha E T_S = (1 - \alpha)(U_d - E) T_S \quad (3-11)$$

Nakon uređivanja izraza (3-11) dobiva se izraz upravljačke karakteristike uzlaznog pretvarača:

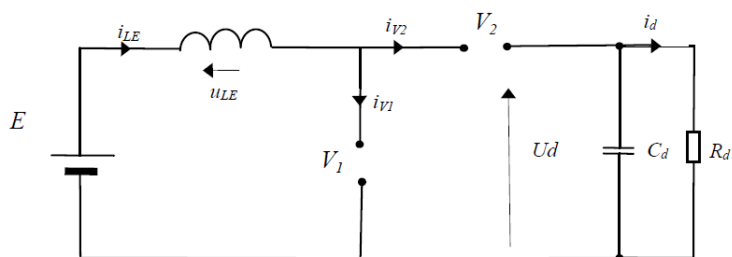
$$\frac{U_d}{E} = \frac{1}{1 - \alpha} \quad (3-12)$$

Slika 3.3. prikazuje karakteristične valne oblike uzlaznog pretvarača napona za kontinuirani režim rada.



Slika 3.3. Valni oblici uzlaznog pretvarača u kontinuiranom načinu rada [1]

Za diskontinuirani način rada pojavljuje se 3. interval (slika.3.4.). U 3. intervalu, dolazi do pada struje ventila V2 na nulu prije nego upravljivi ventil dobije ponovo impuls za uključenje. Slika 3.4. prikazuje nadomjesnu mrežu za 3. interval u analizi rada [2].



Slika 3.4. Nadomjesna shema spoja u 3. intervalu[1]

Prema [2] vrijedi:

srednja vrijednost struje induktiviteta L_e iznosi:

$$I_{LE}(0) = \frac{E}{\tilde{R}_{uz}} + I_d(0) = \frac{1}{2} \Delta i_e \frac{T_{V1} + T_{V2}}{T_s} \quad (3-13)$$

Faktor \tilde{R}_{uz} iznosi:

$$\tilde{R}_{uz} = \frac{2L_E}{\alpha^2 T_s} \quad (3-14)$$

Kako je zbog ravnoteže voltsekunda na induktivitetu L_E

$$ET_{V1} = (U_d - E)T_{V2} \quad (3-15)$$

Upravljačka karakteristika uzlaznog pretvarača napona dana je izrazom:

$$U_d = E \frac{1 + \sqrt{1 + \frac{4R_d}{\tilde{R}_{uz}}}}{2} \quad (3-16)$$

Ako iz relacije (3-16) izrazi \tilde{R}_{uz} iz toga slijedi da je izmjenični otpor jednak:

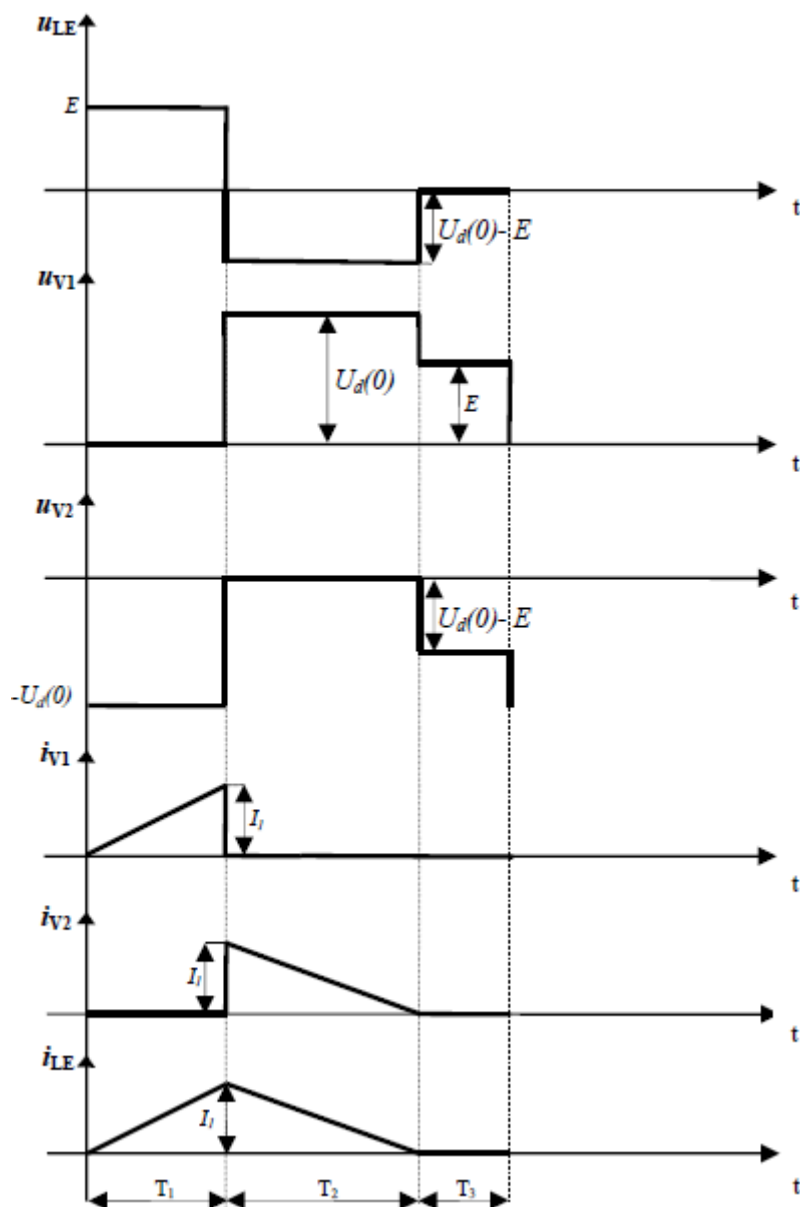
$$\tilde{R}_{uz} = \frac{R_d}{\left(\frac{U_d}{E}\right)^2 - \frac{U_d}{E}} \quad (3-17)$$

Ako iz relacije (3-14) izrazi α , dobije se da je faktor upravljanja za uzlazni pretvarač u diskontinuiranom načinu rada:

$$\alpha_1 = \sqrt{\frac{2L_E}{\tilde{R}_{uz} T_s}} \quad (3-18)$$

Vrijeme vođenja ventila V_2 dobije se iz izraza (3-15) i ono iznosi iznosi:

$$T_{V2} = \frac{E\alpha T_s}{U_d - E} \quad (3-19)$$

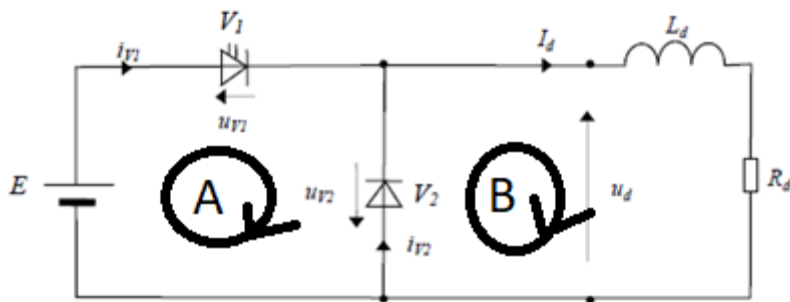


Slika 3.5. Valni oblici uzlaznog pretvarača u diskontinuiranom načinu rada [1]

3.1.2. Silazni istosmjerni pretvarači napona

Pri analizi matematičkog modela pretpostavljeno je da je $L_d/R_d \gg T_s$, te se podrazumjeva da je struja trošila konstanta, tj glatka; $i_d = I_d$.

Slika 2.6. predstavlja nadomjesnu shemu silaznog istosmjernog pretvarača napona



Slika 3.6. Nadomjesna shema spoja istosmjernog silaznog pretvarača napona[1]

Nakon raspisivanja KZN-a dobivaju se jednačbe:

$$A) \quad E = u_{V1} - u_{V2} \quad (3-20)$$

$$B) \quad u_{V2} = -u_d \quad (3-21)$$

Nakon raspisivanja KZS-a dobivaju se jednačbe:

$$1) \quad I_d = i_{V1} + i_{V2} \quad (3-22)$$

U intervalu 1 vodi upravljivi ventil V1 ($u_{V1} = 0$), a ne vodi ventil V2 ($i_{V2} = 0$). Ako se izraz (3-20) uvrsti u (3-21) dobiva se:

$$E = u_d \quad (3-23)$$

Neupravljivi ventil V2 ne vodi, iz izraza (3-22) dobiva se:

$$i_{V1} = I_d \quad (3-24)$$

Napon na neupravljivom ventilu V2 iznosi $-u_d$

U intervalu 2 ne vodi upravljivi ventil V1, ali vodi neupravljivi ventil V2 ($u_{V2} = 0$), iz izraza (3-21) dobiva se:

$$u_d = 0 \quad (3-25)$$

Iz izraza (3-20) dobiva se:

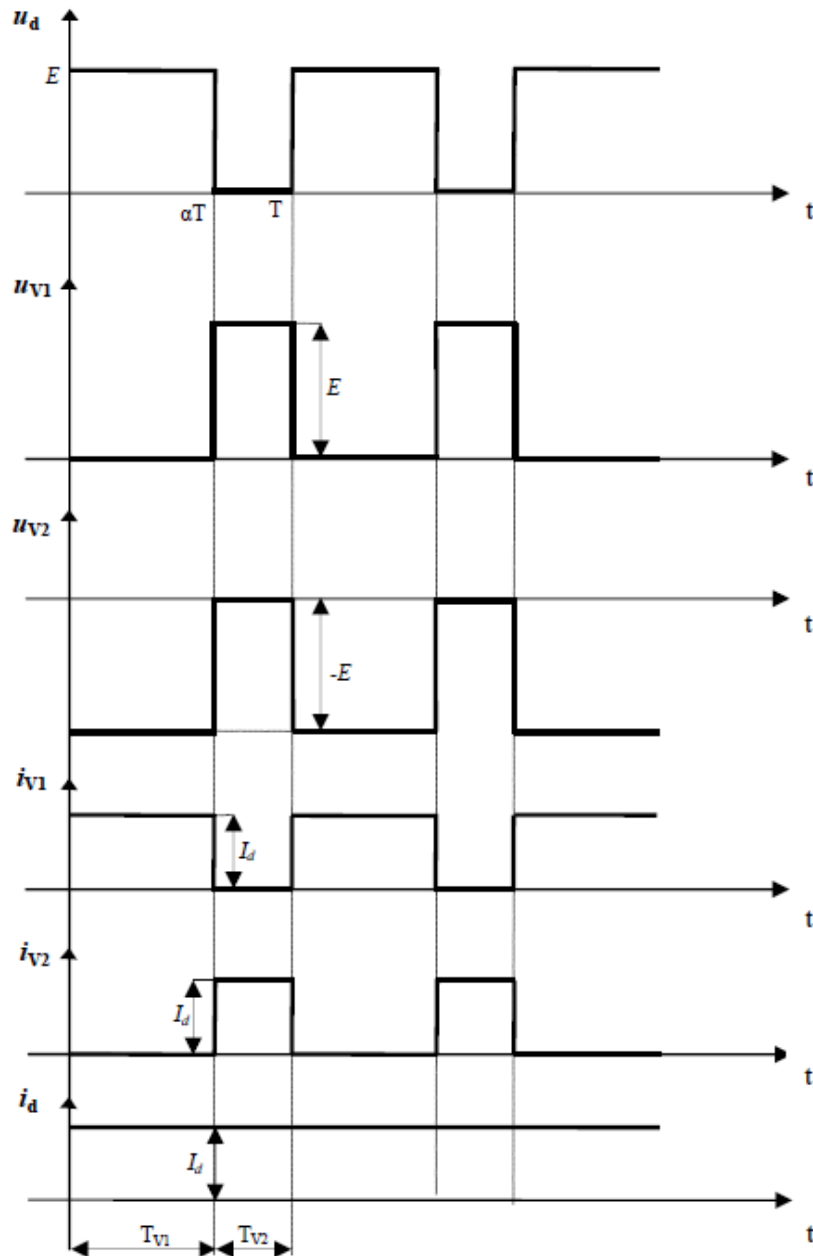
$$u_{V1} = E \quad (3-26)$$

Upravljivi ventil V1 ne vodi, iznos struje na upravljivom ventilu V1 iznosi ($i_{V1} = 0$).

Iznos struje na neupravljivom ventilu V2, dobiva se iz izraza (3-22), te je iznos struje ($i_{V1} = I_d$). Iz valnog oblika napona na trošilu sa slike 3.6., dobije se upravljačka karakteristika silaznog pretvarača:

$$\frac{U_d(0)}{E} = \alpha \quad (3-27)$$

Trošilo je modelirano strujnim izvorom, te je s toga moguć samo kontinuirani način rada. Slika 3.7. prikazuje karakteristične valne oblike silaznog pretvarača napona za kontinuirani režim rada.



Slika 3.7. Karakteristični valni oblici silaznog pretvarača u kontinuiranom načinu rada [1]

4. BIBLIOTEKA REACT

React je prema licenci Open Source JavaScript biblioteka za izgradnju korisnikovog sučelja, odnosno, dan je na korištenje svakome. Razvijen je od strane Facebooka i pojedinih programera i kompanija. React se može koristiti u razvoju jednostrukih stranica (engl. *Single-page*) ili mobilnih aplikacija. Koncentriran je samo na prikazivanje podataka u DOM-u (engl. *Document Object Model*). Za korištenje React-a je potrebno dobro poznavanje i korištenje JavaScripta, HTML-a (engl. *HyperText Markup Language*) i CSS-a (engl. *Cascading Style Sheets*) [3].

4.1 Prezentacijski jezik HTML

HTML je prezentacijski jezik za izradu web stranica. Hipertekst dokument stvara se pomoću HTML jezika. Hiperteks se odnosi na veze koje povezuju web stranice jednu s drugom, bilo unutar jedne web stranice ili između web stranica. HTML služi za dodavanje elemenata u web stranicu. Elementi koji se dodaju su: paragrafi, naslovi, slike, video, unosi podatak. Elementi koji su nabrojani kasnije se uređuju pomoću CSS-a, te tako tim elementima dodaje vizualni stil te animacije istih, a pomoću JavaScripta tim elementima se dodaje logika i sposobnost rada. HTML je vrlo jednostavan jezik i brzo se uči. Neki od primjera pisanja HTML koda te ispisa istih prikazan je dolje [4].

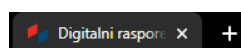
Primjer pisanja HTMLa:

```
<p>Primjer</p>  
<b>Primjer</b>  
<u>Primjer</u>  
<i>Primjer</i>  
<h1>Primjer</h1>  
<h6>Primjer</h6>
```

Primjer ispisa:

```
Primjer  
Primjer  
Primjer  
Primjer  
Primjer  
Primjer
```

Struktura HTML-a sastoji se od *head-a* i *body-a* unutar *html* taga. U *head* tag pišu se elementi poput *title*, *title* tag služi za prikaz naziva pojedine stranice, primjer je prikazan na slici 4.1. U *body tag* pišu se svi ostali elementi poput: paragrafa, slika, videa, unosa, naslova.



Slika 4.1. Prikaz title taga

Struktura pisanja:

```
<html>
  <head>
    <title>Digitalni raspored</title>
  </head>
  <body>
    <h1>Naslov</h1>
    <p>Paragraf</p>
    <i>Primjer</i>
  </body>
</html>
```

U *head* tag spadaju slijedeći elementi: *style*, *meta*, *link*, *script* i *base*.

4.2. CSS

CSS se koristi za izgled i raspored elemenata web stranice. Prvi korak za CSS je da postoji HTML element kojem će se dodavati izgled ili raspored elementa. Dva su načina pristupanja elementu HTML-a. Prvi način je način preko klase, a drugi način je preko id. U principu, isti je način, samo što se preko klase pristupa sa „ime__html__elementa“, a za id koristi se „#ime__html__elementa“ [5].

Pisanje HTML-a

```
<p
className="primjer__paragrafa">
Paragraf
</p>
```

```
<p id="primjer__paragrafa">
Paragraf
</p>
```

Pisanje CSS-a

```
.primjer__paragrafa {
font-family: Avenir;
font-style: normal;
font-weight: normal;
font-size: 16px;
color: red;
}
```

```
#primjer__paragrafa {
font-family: Avenir;
font-style: normal;
font-weight: normal;
font-size: 16px;
color: red;
}
```

Primjer ispisa

Paragraf

Paragraf

U gornjem dijelu primjera objašnjeno je definiranje klase i id te način pisanja CSS-a. Prilikom pisanja CSS-a korištene su osnovne naredbe za izgled samog fonta. Font je poprimio crvenu boju, tip fonta je *Avenir* te veličina samog fonta je 16px.

4.3. Programski jezik JavaScript

JavaScript bila je znana kao LiveScript, ali je Netscape promijenio ime u JavaScript jer je u to vrijeme bila popularna Java. Netscape je iskoristio popularnost Jave i promijenio naziv iz LiveScript u JavaScript kako bi se probili na tržištu, iako, Java i JavaScript nemaju nikakve zajedničke sličnosti te se koriste u različitim tehnologijama.

JavaScript je dinamički programski jezik koji se koristi u izradi internetskih stranica, izradi internetskih aplikacija te u izradi igrice i još mnogo toga. JavaScript se interpretira kao objektno orijentirani programski jezik[6]. Dopušta implementaciju dinamičkih značajki u internetskim stranicama koje se ne mogu napraviti samo pomoću HTML-a i CSS-a.

JavaScript prepoznaje koji je tip varijable, često zbog toga dolazi do pogrešaka prilikom korištenja, zbog toga se koristi TypeScript jer se definiraju varijable i određuju tipovi. Tipovi koji se mogu definirati su: *string*, *boolean*, *number*, *undefined*, *null*.

Primjer koda:

```
const number = 21;
const string = "21"

function add100(number) {
  console.log(100 + number)
}

add100(number) // => 121
add100(string) // => "10021"
```

Definirana je varijabla *number* koja iznosi 21 i definirana je varijabla *string* koja iznosi „21“. Napisana je funkcija koja će zbrojiti broj 100 te definirane varijable. Prilikom ispisa ako se pozove varijabla *number* ispis će biti 121, a ako se pozove varijabla *string*, ispis će biti „10021“ jer je JavaScript prepoznao da je to *string*, a prilikom korištenja *stringa*, „+“ predstavlja spajanje dva *stringa* pa je s toga ispis „10021“.

4.4. Instalacija React-a

Za instalaciju React potrebno je imati instaliran Visual Studio Code i Node.js. Visual Studio Code je uređivač otvorenog koda koji je izradio Microsoft za Windows, Linux i macOS. Koristi se u različitim programskim jezicima uključujući Javu, JavaScript, Go, Node.js, Python i C++. Temelji se na Electricron okviru koji se koristi za razvoj web aplikacija.

Node.js omogućuje programerima da koriste JavaScript za pisanje alata naredbenog retka i za skriptiranje na strani poslužitelja. Pokretanje skripti na strani poslužitelja za stvaranje dinamičkog sadržaja web stranice prije nego što se stranica pošalje korisničkom web pregledniku.

Nakon instaliranog Visual Studio Code-a i Node.js potrebno je pokrenuti isti i u konzoli ukucati naredbu

```
npm create-react-app my-app --template typescript
```

nakon toga, potrebno je pričekati da se svi paketi instaliraju te kad se instaliraju, naredbom

```
npm start
```

se pokreće React i otvara se internetski preglednik i na linku *localhost:3000* vidi se početni program koji dolazi implementiran.

Sam proces, prilikom kucanja naredbe za pokretanje odvija se lokalno i samo je vidljiv osobama koji su spojeni na istu mrežu. React je inicijaliziran i spreman je za kucanje vlastitog projekta.

4.5. Korištenje dodatnih paketa u Reactu

Pored osnovnih paketa u React-u postoje dodatni paketi koji se mogu instalirati i ubaciti u postojeći projekt. Dodavanje paketa vrši se na isti način kao i instalacija React-a. Dodatni paketi koji su korišteni u izradi projekta su: firebase, redux, react-redux, react-helmet, react-router-dom. Instalirani paketi pozivaju se tako što se iznad cijelog koda upiše naredba

```
import {  
  BrowserRouter as Router,  
  Route,  
  Link,  
  Switch
```

```
} from 'react-router-dom';
```

Kod ispod pokazuje primjer korištenja react-router-dom paketa.

```
<Route exact path='/about' component={About}></Route>
```

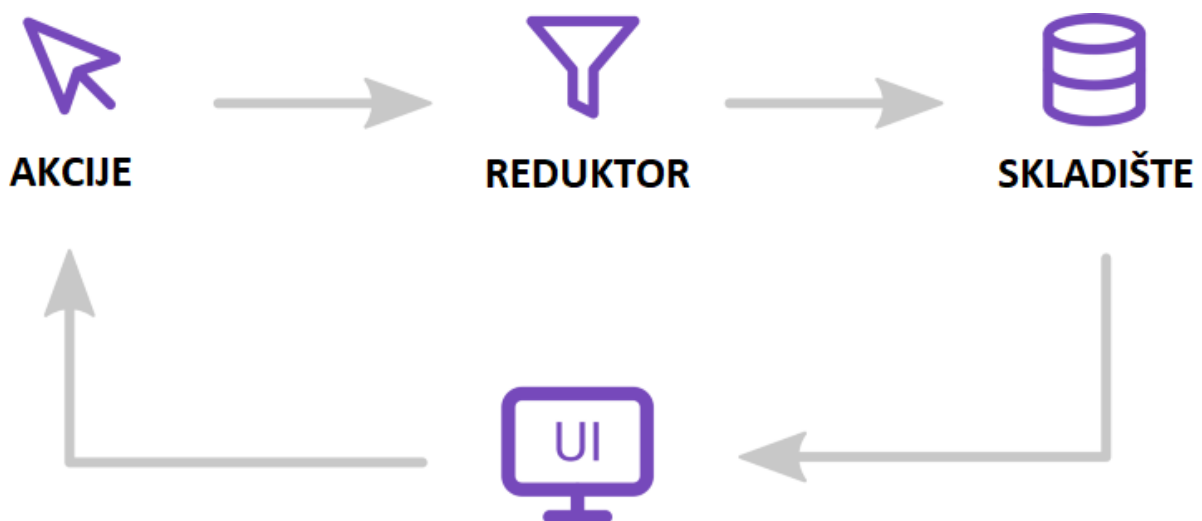
Komponenta *path* služi za kreiranje linka na koji će program dovesti korisnika, a na toj stranici pomoću komponenta *component* prikazati određenu funkciju koja se u ovom slučaju naziva *About*. Kada korisnik upiše *localhost:3000/about*, uz pomoć paketa *Route* pokazat će mu se funkcija koja se naziva *About* [7].

4.6. Redux

Za korištenje Redux-a potrebno je imati instalirani paket: *redux*, *react-redux*.

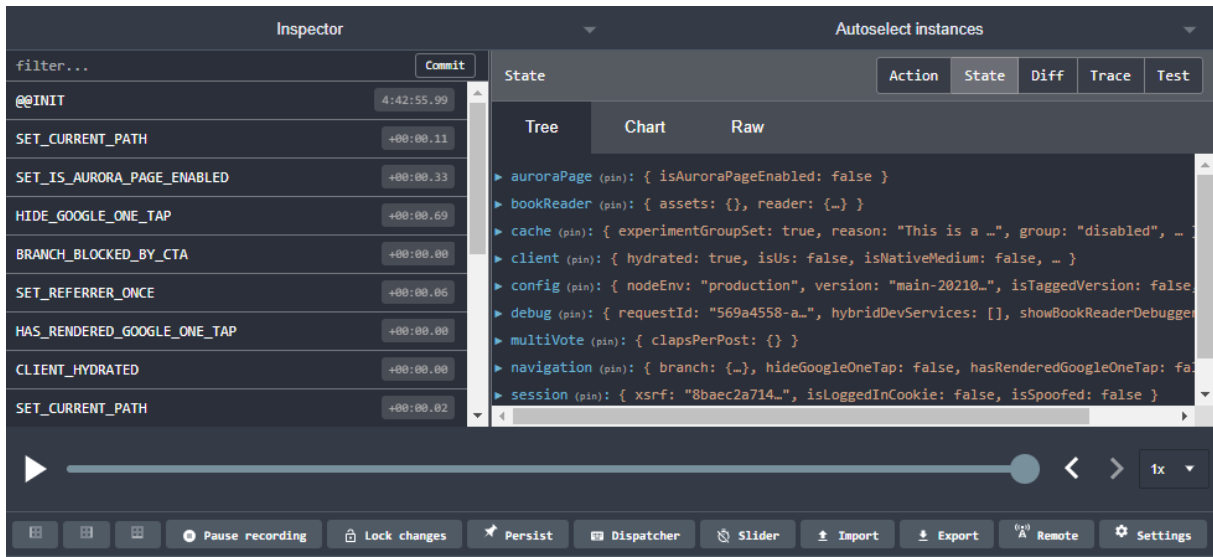
Redux služi za spremanje podataka lokalno na svakom pregledniku. Sastoji se od akcija i reduktora. Redux uveliko olakšava rad u React-u i prijenos podataka iz komponente u komponente.

Primjer korištenja Redux-a, ako želimo u nekoj aplikaciji imati podatke o korisnicima, njihovu električnu poštu, korisnikovo ime, sliku. Prilikom prijavljivanja na aplikaciju i ako je prijava točna u tome trenutku se podaci šalju na akciju te nakon toga na reduktor pa u skladište (*engl. store*) slika 4.2. [3].



Slika 4.2. Način rada Reduxa[8]

Promatranje onoga što se nalazi u samom skladištu i akcija može se pogledati u Chrome pregledniku. Potrebno je instalirati paket: *react-dev-extension*. Nakon instaliranog paketa, paket se implementira u program i u *DeveloperToolsu* se odabere Redux i mogu se pratiti promjene. Slika 4.3. prikazuje izgled Redux-a u *DeveloperToolsu*. Alat je vrlo koristan, jer se tako prate akcije i promjene u samom skladištu što dosta olakšava programerima posao.



Slika 4.3. Prikaz Redux-a unutar Chrome preglednika

Akcije su jedini način slanja podataka iz aplikacije u skladište. Akcije su običan JavaScript objekt s obaveznom svojstvom tipa. Mogu se poslati iz komponente kao odgovor na aktivnost korisnika.

Akcija može biti bilo koja vrijednost objekta koja ima tip. Može se slati podatke zajedno s reduktorom iz aplikacije u skladište. Pokretanja akcije vrši se na primjer, klikom na dugme, zahtjev može biti pokrenut ili dovršen [9].

Svojstva akcija:

- akcije su obični JS objekti i moraju imati svojstvo tipa koji označava vrstu radnje koja se izvodi,
- akcije mogu imati podatke koje se stavljaju u skladište,
- akcije mogu biti sinkrone i asinkrone.

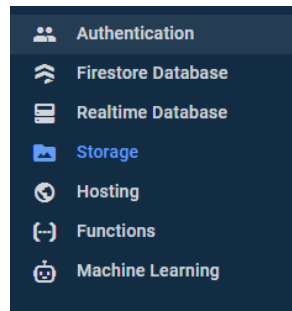
Reduktor je funkcija odgovorna za primanje trenutnog stanja, slanje akcije kao argumenta i vraćanje novog stanja. Više reduktora mogu se spojiti zajedno za stvaranje konačnog stanja primjene [9].

4.7. Platforma Firebase

Za korištenje Firebase u React-u potrebno je instalirati paket: firebase.

Firebase je Googlova platforma vezana za pozadinski servise (*engl. Backend services*). Omogućava programerima jednostavno korištenje i stvaranje fantastičnog korisničkog sustava. Pri korištenju Firebase nije potrebno pisanje API-ja, sve bitne stvari su već implementirane. Za korištenje Firebase potrebno je kreirati nalog na stranici: <https://firebase.google.com/>. Kreirati svoj projekat te nakon toga je spremno za korištenje [10].

Neki od osnovnih alata koji su dostupni na samoj stranici nakon kreiranog projekta su: autentifikacija, baza podataka, skladište i usluge poslužitelja (*engl. hosting*). Slika 4.4. prikazuje dostupne alate nakon kreiranja projekta.



Slika 4.4. Prikaz dostupnih alata nakon kreiranja projekta

Autentifikacija se sastoji od davatelja usluge (*engl. provider*) što je prikazano na slici 4.5.. Autentifikacija sadrži: Apple, Phone, Yahoo, Google, Facebook, Twiter, GitHub...itd. Implementacijom davatelja usluge omogućuje se korisniku da ako ima neki od gore navedenih naloga pristupi aplikacije bez pravljenja novog naloga. Ako korisnik želi napraviti novi nalog, samo na toj aplikaciji, korisnik to može učini pomoću prijave pomoću *email/password*.

Provider	Status
Email/Password	Enabled
Phone	Disabled
Google	Enabled
Play Games	Disabled
Game Center	Disabled
Facebook	Disabled
Twitter	Disabled
GitHub	Disabled
Yahoo	Disabled
Microsoft	Disabled
Apple	Disabled
Anonymous	Disabled

Slika 4.5. Prikaz davatelja usluge

Firestore Database je baza podataka u koju je omogućen upis podataka, a *Storage* je namijenjen za slike, pjesme, tekstualne datoteke, datoteke.

Firebase je našao primjenu za učenje, ali i za rad ozbiljnih projekata. Većina alata je besplatna te s toga je savršena za učenje.

4.8. Biblioteka CanvasJS

CanvasJS je HTML5 i Javascript biblioteka grafikona jednostavna za korištenje. Radi na svim uređajima, uključujući iPhone, Android, stolna računala... Omogućava stvaranje bogatih nadzornih ploča koje rade na različitim uređajima bez kompromisa u pogledu održavanja ili funkcionalnosti [11].

Omogućuje stvaranje dinamičkih grafikona koji se ažuriraju u određenom intervalu. Dinamički grafikonu potrebni su kada se prikazuju podaci koji se mijenjaju s vremenom, poput cijene dionica, temperature i slično. Oni se stvaraju na isti način kao i bilo koja druga vrsta grafikona, osim što se podatkovne točke dodaju ili uklanjaju u definiranom intervalu.

5. IZRADA INTERAKTIVNOG SUČELJA ZA UZLAZNI ISTOSMJERNI PRETVARAČ NAPONA

U ovom poglavlju opisan je način izrade jednog pretvarača. Objašnjeni su dijelovi koda i svi dijelovi koji idu zajedno s njim. Dodan je vizualni stil u vidu CSS-a, implementiran je CanvasJS. Prikazan kod bit će za uzlazni pretvarač napona, analogno tome, vrijedi i za sve ostale pretvarače. Zbog toga je prvo napravljena analiza rada pretvarača u trećem poglavlju.

5.1. Izrada komponenti koje su upotrebljive na više mjesta

Prilikom pisanja koda potrebno je što više imati komponenata koje se mogu više puta iskoristiti te tako omogućiti pregledniji kod i ako se želi nešto promijeniti u toj komponenti promijenit će se na svim mjestima gdje se komponenta koristi. Slika 5.1. predstavlja prikaz pisanja komponente za unos koja je upotrebljiva na više mjesta. Pored komponente za upis polja, korištene su još komponente za dugme, polja za potvrdne okvire.

Linija *Kod*

```
1:        /*Custom Input*/
2:        interface InputProps extends React.HTML<HTMLInputElement> {
3:        label:string;
4:        id:string;
5:        message:string;
6:        }
7:        export const CustomInput: React.FC<InputProps> =({
8:        label,
9:        id,
10:        Message,
11:        ...inputProps
12:        })=>{
13:        return (
14:            <div className="input">
15:            <div className="input--primary">
16:            <label className="input--label">{label}</label>
17:            <input
18:                className="input--input"
19:                id={id}
20:                {...inputProps}
```

```

21:         onWheel={ (event) =>{
22:             Event.currentTarget.blur();
23:         }}
24:     />
25:     <p className="input--message">{message}</p>
26: </div>
27: </div>

```

Slika. 5.1. Prikaz komponente za unos

Pozivanje komponenata odvija se tako da funkciji *CustomInput* predaje se kao argumenti. Na slici 5.2. prikazan je način korištenja komponenti koje se koriste na više mjesta. Kao argument predaje se koji je tip, tip je broj što znači da u ulazno polje se može napisati samo brojevi. Za tip postoje još: datum (date), tekst (text), potvrdni okvir (checkbox), domet(range) [12]. Slika 5.3. prikazuje konačan izgled za funkciju *CustomInput*.

Linija Kod

```

1:     /*Projekt test*/
2:     <CustomInput
3:     type="number"
4:     onChange={handleChange}
5:     message="Preporučena unesena vrijednost mora biti od 10V do 25V"
6:     label="Ulazni napon [V]"
7:     placeholder="E..."
8:     />

```

Slika. 5.2. Pozivanje komponente *CustomInput*

Slika 5.3. prikazuje prikazuje konačan izgled za funkciju *CustomInput*

Ulazni napon [V]

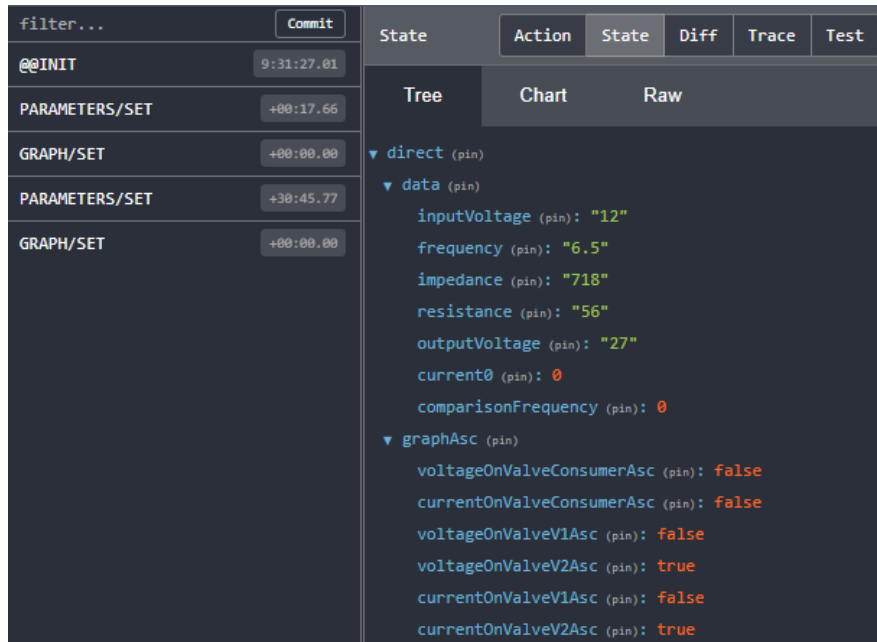
Preporučena unesena vrijednost mora biti od 10V do 25V

Sl. 5.3. Izgled *CustomInput* komponente

5.2. Kreiranje skladišta, akcija i reduktora

Podaci koji su zapisani u skladištu su dostupni globalno u bilo kojoj komponenti što znači da prijenos podataka iz komponente u komponentu je olakšan. Za izradu aplikacije, Redux se

koristio za prijenos parametara i za prikazivanje pojedinačnih valnih oblika. Slika 5.4. prikazuje izgled skladišta i način na koji su parametri spremljeni i valni oblici.



Slika 5.4. Prikaz skladišta za uzlazni pretvarač

U *data* nalaze se parametri pretvarača, a pod *graphAsc* nalaze se *boolean* vrijednosti za grafove. Za slučaj koji je prikazan na slici 5.4. prikazat će se valni oblik napona na ventilu V2 i valni oblik struje kroz ventil V2. Ako je vrijednost *true* valni oblik će se prikazati, a ako je vrijednost *false* valni oblik se neće prikazati. Za korištenje Redux-a, potrebno je prvo kreirati skladište. Slika 5.5. prikazuje kako se skladište kreira.

Linija Kod

```

10:   export const configureStore = () => {
11:     const rootReducer = {
12:       direct:directConverterReducer,
13:       indirect:indirectConverterReducer,
14:       unmanageable:unmanageableConverterReducer,
15:       autonomous:autonomousConverterReducer,
16:       manageable:manageableConverterReducer,
17:     },
18:     return createStore(
19:       combineReducers(rootReducer),
20:       composeWithDevTools(applyMiddleware(thunk))
21:     );}
22:   const storeState = configureStore().getState;
23:   export type RootState = ReturnType<typeof storeState>
  
```

Slika 5.5. Izrada unosa parametara i logike

Varijabla *rootReducer* sadrži sve reduktore koji se koriste u aplikaciji. Funkcija *combineReducer* spaja sve reducere u jedan, a *composeWithDevTools* služi za prikaz skladišta u pregledniku.

Kreiranje akcija prikazano je na slici 5.6.

Linija Kod

```
4:     export type DirectConverterActions =
5:       | {
6:         type: typeof DirectConverterTypes.Set;
7:         payload :Parameters;
8:       }
9:       | {
10:        type: typeof DirectConverterTypes.SetGraph;
11:        payload :Graph;
12:      };
```

Slika. 5.6. Kreiranje akcije

U *payload* definirano je sučelje, odnosno koji se tip varijable očekuje.

Linija Kod

```
28:     export const directConverterReducer = (
29:       state = INITIAL_STATE,
30:       action: DirectConverterActions
31:     ):ParState => {
32:       Switch (action.type){
33:         case DirectConverterTypes.Set:
34:           return {...state, data: action.payload};
35:         case DirectConverterTypes.SetGraph:
36:           return {...state, graph: action.payload};
37:         default:
38:           return state;
39:       }
40:     };
```

Slika. 5.7. Kreiranje reduktora

Kod pisanje reduktora, potrebno je imati definirano početno stanje varijabli i definirane akcije. Početno stanje služi da se vrijednosti dodjele svakom od tipova, kada se ne bi definirale početne vrijednosti u najgorem slučaju dolazi do pada same aplikacije i nemogućnost korištenja iste.

5.3. Unos parametara i upotreba Redux-a

Za unos parametara korišten je hooks pod nazivom *useState*.

Linija **Kod**

```
1:     const [parameters, setParameters] = useState<
2:     Omit<Parameters, „inductanceLd“>
3:     >({
4:     inputVoltage: 0,
5:     frequency:0,
6:     inductanceLe:0,
7:     resistance:0,
8:     outputVolgtage:0,
9:     })
```

Slika. 5.8. Definiranje useState

useState() definira se na način kako je prikazano na slici 5.8. Za korištenje *useState()* potrebno je definirati početno stanje varijabli i definirati tip podataka koje se očekuju koji će korisnik upisati. Tip podataka definiran je unutar sučelja *Parameters* i u njemu je definirano za svaku posebnu varijablu koji će tip imati. Tip koji je korišten za sve varijable prilikom izrade je *number*. Na taj način, definirano je da će korisnik unijeti broj, a ne znak.

Omit je dio osnovnog poznavanja *typescript-a* koji ima funkciju da iz postojećeg sučelja, u našem slučaju *Parameters* izbaci *inductanceLd*, jer prilikom pisanja koda za uzlazni pretvarač napona ne postoji parametar induktivitet *Ld*. Tako se dobiva preglednost koda, i moguće je sučelje koristiti na više mjesta u kodu, a ne mora se za svaki slučaj posebno pisati. [13]

Promjenu stanja se vrši preko funkcije *setParameters* (slika 5.9.), a rezultat promjene stanja dostupan je u varijabli *parameters*.

Linija **Kod**

```
21:     function handleChange(event: React.ChangeEvent<HTMLInputElement>) {
22:     setParameters({...parameters, [event.target.id]:event.target.value})
23:     }
```

Slika. 5.9. Unos parametara pomoću funkcije *setParameters*

Svaki od unosnih polja ima svoj *id*. *Id* mora biti istog naziva kao i parametri koji su definirani u sučelju. Funkcija *handleChange* se poziva svaki put kad dođe do promjene u unosnom polju. Slika 5.2. u liniji 4 prikazuje kako se poziva funkcija *handleChange*.

Event prima svojstva od *React.ChangeEvent<HTMLInputElement>* u kojemu se nalaze sva svojstva vezana za unosna polju. *Event* je objekt, za funkciju *setParameters* potreban je *event.target.id* što predstavlja sam *id* od unosnog polja, a *event.target.value* predstavlja unesenu vrijednost. Funkcija *handleChange* se poziva svaki put kad se vrijednost promijeni u unosnom polju i sprema ju u *parameters*.

Slika 5.10. prikazuje kako se podaci spremaju u Redux te kako se obavlja prelazak na novu stranicu

Linija Kod

```
35:     const history = useHistory();
36:     const dispatch = useDispatch();
37:     function onClick() {
38:       dispatch({ type:"PARAMETERS/SET", payload: parameters});
39:       dispatch({ type:"GRAPHS/SET", payload: graph });
40:       history.push(„/izravni-istosmjerni-pretvarac-napona/uzlazni-pretvarac-graf“)
41:     }
```

Slika. 5.10. Spremanje podataka u redux i prelazak na novu stranicu

useHistory() je funkcija koja se uvozi iz paketa *react-router-dom*, a funkcija *useDispatch* iz paketa *react-redux*. Nakon što korisnik popuni odgovarajuća polja, i klikne na dugme, izvršit će se funkcija *onClick* koja će pokrenuti funkciju *dispatch* i omogućiti spremanje u skladište pod odgovarajućem tipom i kao *payload* podatak koji se šalje.

Nakon što se *dispatch* funkcija izvrši, izvršit će se *history* koja će korisnika navigirati prema linku.

5.4. Izrada grafa karakterističnih valnih oblika

Nakon što je podatak spremljen u skladište, potrebno je tome istom podatku i pristupiti. Pristup skladištu, izvršava se preko funkcije *useSelector*. Funkcija *useSelector* se uvozi iz paketa *react-redux*. Slika 5.11. prikazuje pristupanje podacima koji se nalaze u skladištu

Linija Kod

```
35:     const { data, graph } =
36:     useSelector(directDcDcConvertorSelector.getState)
37:     const L = data.impedance * 1e-6;
```

```

37:   const E = data.inputVoltage;
38:   Const f = data.frequency *1e3;
39:   Const Ud = data.outputVoltage;
40:   Const Rd = data.resistance;

```

Sl. 5.11. Preuzimanje podataka iz skladišta

Nakon što su podaci preuzeti, potrebno je prilagoditi frekvenciju u kHz (linja 39) i induktivitet u μH (linija 36).

Linija Kod

```

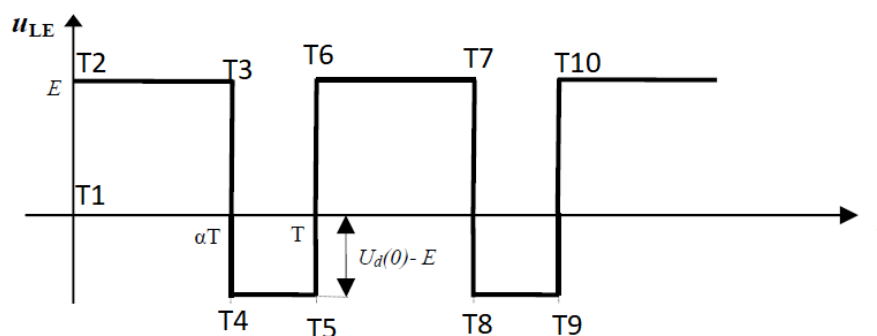
41:   const period = 1/f;
42:   const alpha = (Ud-E)/Ud;
43:   const tv1 = period * alpha;
44:   Const id1 = Ud/Rd;
45:   Const ie = id1/(1-alpha);
46:   Const deltaI = E*tv1/L;

```

Slika. 5.12. Formule potrebe za simulaciju

Sa slike 3.3. iz karakterističnih valnih oblika za kontinuirani način rada uzlaznog pretvarača, zapisuje se iznos svake točke zasebno za sve valne oblike.

Slika 5.13. prikazuje iznose za pojedine točke na grafu.



Slika 5.13. Potrebne točke za simulaciju

Iznosi pojedinih točaka date su u tablici 5.1. Analogno tome, za ostale valne oblike pretvarača, ispravljača i izmjenjivača napona napravljeno je slično.

Tablica 5.1. Iznos pojedinih točaka

Točka	t -os	u_{LE} -os
1	0	0
2	0	E
3	αT	E
4	αT	$-(U_d(0) - E)$
5	T	$-(U_d(0) - E)$
6	T	E
7	$T + \alpha T$	E
8	$T + \alpha T$	$-(U_d(0) - E)$
9	$2T$	$-(U_d(0) - E)$
10	$2T$	E

Sada kada su sve potrebne točke za simulaciju određene, vrijeme je da podatke iz tablice 5.1. zapišu kao kod. Za iscrtavanje valnih oblika korišten je CanvasJS. Slika 5.14. predstavlja način zapisa iznosa točaka.

Linija Kod

```

51:     let voltageImpedance ;
52:     voltageImpedance = {
53:       animationEnabled:true,
54:       title: {
55:         text: „uLe(t)“
56:       },
57:       axisX: {
58:         title:„Vrijeme [μs]“,
59:         minimum:0,
60:         maximum:2*period*1e6,
61:       },
62:       axisY: {
63:         title:„uLe [V]“,
64:       },
65:     data:[
66:       {
67:         type:"line",
68:         dataPoints: [
69:           { x:0, y:0},
70:           { x:0, y:E},

```



```

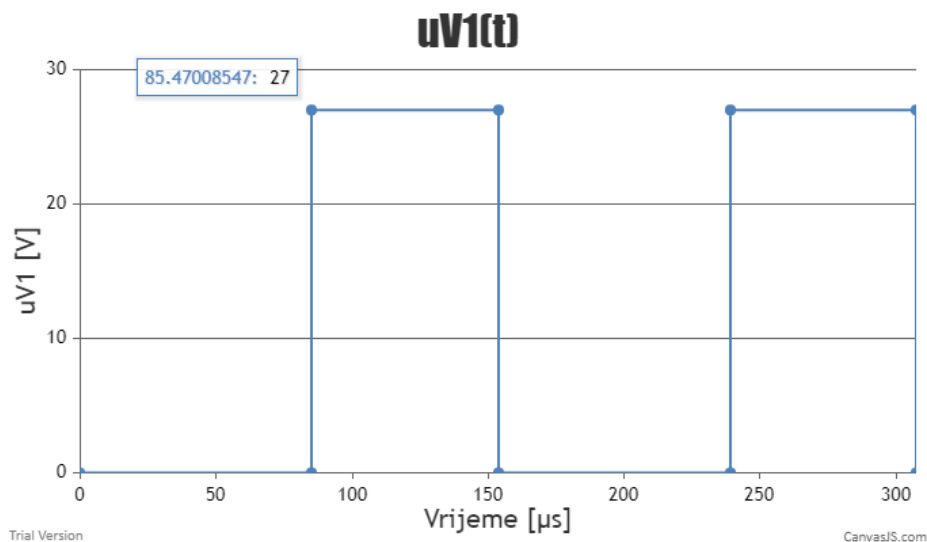
71:         { x:tv1*1e6, y:E},
72:         { x:tv1*1e6, y:-(Ud-E)},
73:         { x:period*1e6, y:-(Ud-E)},
74:         { x:period*1e6, y:E},
75:         { x:(period+tv1) *1e6, y:E},
76:         { x:(period+tv1) *1e6, y: -(Ud-E)},
77:         { x:2*period*1e6, y:-(Ud-E)},
78:         { x:2*period*1e6, y:E},
79:     ],
80: },
81: ],
82: },

```

Slika. 5.14. Zapis podataka iz tablice 5.1.

Slika 5.15. prikazuje konačan izgled za varijablu *voltageImpedance*. Naslov grafa, ordinata i apscisa može promijeniti tako što će se u *title* napisati naziv, a rezultat toga bit će prikazan na grafu. Moguće je pored mijenjanja imena osi i naslova mijenjati i minimum i maksimum osi. Tako se može odrediti točno koliko perioda želimo postići. Na svakom grafu, za sve pretvarače prikazana su po dvije periode valnog oblika.

Omogućeno je očitavanje vrijednosti pojedinih točaka s grafa, tako da korisnici mogu sami zaključiti kako pojedini parametri utječu na sami izgled karakterističnih valnih oblika te iz tih točaka izračunati efektivnu, srednju vrijednost valnog oblika.



Slika 5.15. Prikaz varijable *voltageImpedance*

6. REZULTATI INTERAKTIVNOG WEB-SUČELJA

6.1. Korištenje interaktivnog web-sučelja

Na slici 6.1. prikazan je konačni izgled web-sučelja. Korisnik pod kategorijom Vježbi ima pristup svim vježbama koji se obavljaju u sklopu laboratorijski vježbi. Za svaku od vježbi korisnik dobiva kratak opis pretvarača te simbol pretvarača. Kako bi korisnik imao u vidu koje parametre mora unijeti pripadajuća shema pretvarača prikazuje koji se parametri pretvarača unose, a analiza rada pretvarača dana je u trećem poglavlju ovoga rada.

NEUPRAVLJIVI ISPRAVLJAČI

Ispravljači su električni uređaji koji povezuju izmjeničnu poju mrežu i istosmjerno trošilo, pri čemu se energija prenosi od izmjenične pojne mreže prema istosmjernom trošilu, slika 3.1.. Pretvarачke komponente neupravljivih ispravljača jesu diode. Ovisno o broju faza pojne izmjenične mreže, neupravljivi se ispravljači dijele na jednofazne ili višefazne. Za trošila veće snage najčešće se koriste trofazni neupravljivi ispravljači, a za trošila manje snage jednofazni neupravljivi ispravljači.

Istosmjerna trošila dijele se na energetska i elektronička [1, 2]. Tipični primjeri energetskih trošila napajanih iz neupravljivih ispravljača uzbude su istosmjernih motora, kotve magnetna i akumulatorske baterije pri punjenju. Napajanje elektroničkih sklopova uređaja (osobna i procesna računala, instrumentacijska oprema, te audio i video uređaji) smatraju se elektroničkim trošilima. Elektronička trošila zahtijevaju kontinuirani napon, te se takva trošila modeliraju paralelnim spojem kapaciteta i otpora, zbog čega se smatraju kapacitivnim trošilima. Energetskim trošilima treba osigurati kontinuiranu struju, te se modeliraju serijskim spojem induktiviteta i otpora, zbog čega se smatraju induktivnim trošilima.

Sl. 4.1. Simbol jednofaznog ispravljača

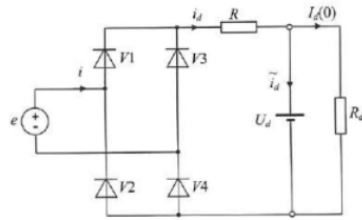
U vježbi treba usporediti karakteristične valne oblike dobivene analizom matematičkog modela i mjerenjima na fizičkom modelu.

Jednofazni neupravljivi mosni spoj

Slika 6.1. Izgled web-sučelja

Na slici 6.2. prikazan je način unosa podataka za ispravljač u jednofaznom mosnom spoju kapacitivno opterećen. Podaci se unose tako što se klikom na odgovarajuće polje unose veličine. Sve veličine moraju biti unesene jer inače web-sučelje neće raditi. Prilikom ispisa grafova, korisnik odabire što želi od grafova da se pokaže. U ovisnosti od pretvarača prikazuju se različiti valni oblici struja i napona. Kako bi korisnik lakše uvidio kako frekvencija utječe na režime rada pretvarača, omogućen je unos različite frekvencije (samo za uzlazni istosmjerni pretvarač napona). Na istom grafu, ako je opcija uključena, pojavit će se dva grafa, ali različite boje, svi uneseni parametri bit će isti osim frekvencije. Ako je frekvencija ispravno odabrana, pretvarač će raditi u drugom režimu rada te će korisnik moći usporediti dva karakteristična valna oblika.

Jednofazni neupravljivi mosni spoj



Sl. 3.2. Nadomjesna shema kapacitivno opterećenog jednofaznog mosnog spoja

Unos podataka

Vršna vrijednost ulaznog napona [V]

U...

Preporučena unesena vrijednost mora biti od 320V do 400V

Napon trošila [V]

Ud...

Preporučena unesena vrijednost mora biti od 280V do 310V

Otpor R [kΩ]

R...

Preporučena unesena vrijednost mora biti od 1kΩ do 3.3kΩ

Frekvencija [Hz]

f...

Preporučena unesena vrijednost mora biti od 30Hz do 100Hz

Odaberi grafove

- Napon trošila
- Napon mreže
- Struja mreže
- Struja kroz trošilo

Riješi

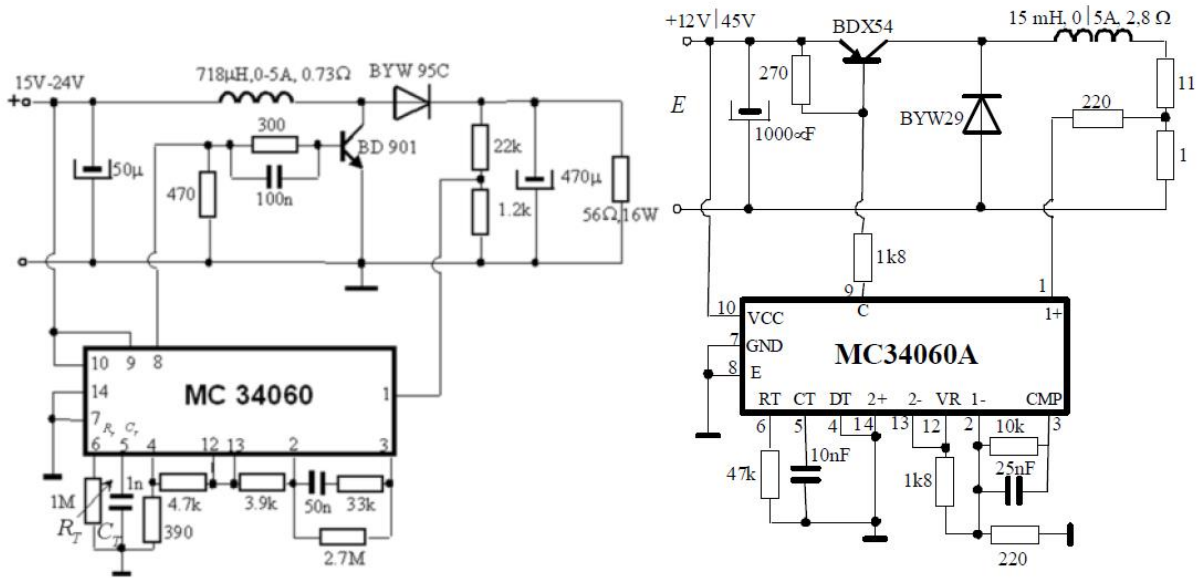
Slika 6.2. Prikaz unosa podataka i odabira grafova

6.2 Usporedba rezultata simulacije i mjerenja

U drugoj laboratorijskoj vježbi obrađuju se izravni istosmjerni pretvarači napona. U tu kategoriju spadaju pretvarači:

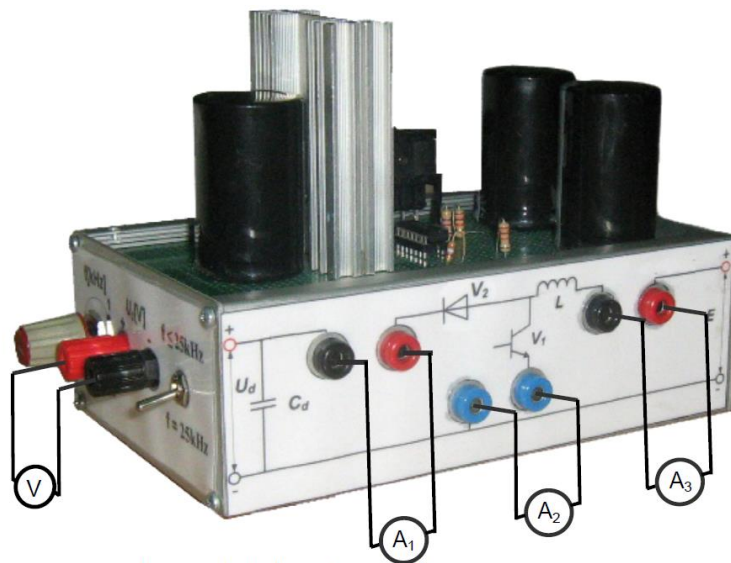
- istosmjerni uzlazni pretvarač napona
- istosmjerni silazni pretvarač napona

Slika 6.3. prikazuje shemu spoja fizičkih modela izravnih istosmjernih pretvarača napona.



Slika 6.3. Istosmjerni uzlazni pretvarač napona 27 V/1 A (lijevo), istosmjerni silazni pretvarač napona 5 V/2 A (desno)[1]

Mjernu opremu potrebno je spojiti prema slici 6.4.



Slika 6.4. Shema spoja istosmjernog uzlaznog pretvarača[1]

Na ulaz potrebno je dovesti napon od 12 V te pospajati mjerne instrumente. Nakon toga potrebno je uključiti pretvarač (sklopku postaviti u položaj „ON“), te preko osciloskopa, promatrati valne oblike. Izabrani pretvarač, pri frekvenciji od 6.5 kHz radi u kontinuiranom načinu rada dok pri frekvenciji od 2.4 kHz radi u diskontinuiranom režimu rada. Voltmetar mjeri izlazni napon, ampermetar A1 mjeri srednju vrijednost struje na neupravljivom ventilu

V2 (diodi), ampermetar A2 mjeri srednju vrijednost struje na upravljivom ventilu V1 (tranzistoru), dok ampermetar A3, mjeri srednju vrijednost struje kroz induktivitet L_e .

Za potrebe simulacije za uzlazni pretvarač napona, sa slike 6.3. preuzimaju se parametri za potrebe simulacije. Svi parametri potrebni za simulaciju nalaze se u tablici 6.1.

Tablica 6.1 Parametri potrebni za simulaciju

Parametri	Prvi set	Drugi set
Ulazni napon E [V]	12	12
Izlazni napon Ud [V]	27	27
Frekvencija [kHz]	6.5	2.4
Induktivitet L_e [μ H]	718	718
Otpor R_d [Ω]	56	56

Postoje dva seta parametara za uzlazni pretvarač napona jer se studentima u vježbi žele pokazati različiti režimi rada. Nakon što se parametri unesu dobivaju se karakteristični valni oblici.

Slika 6.5. prikazuje unos za prvi set parametara.

Unos podataka

Ulazni napon [V]

Preporučena unesena vrijednost mora biti od 10V do 30V

Frekvencija [kHz]

Preporučena unesena vrijednost mora biti od 2kHz do 10kHz

Izlazni napon [V]

Preporučena unesena vrijednost mora biti od 20V do 50V

Induktivitet L_e [μ H]

Preporučena unesena vrijednost mora biti od 500 μ H do 1000 μ H

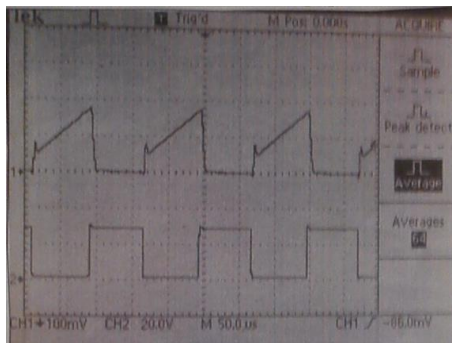
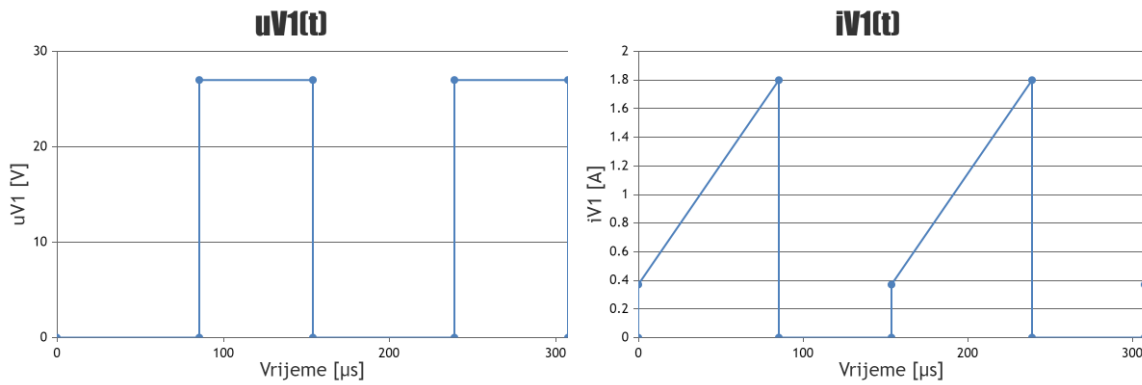
Otpor R_d [Ω]

Unesena vrijednost mora biti od 20 Ω do 100 Ω

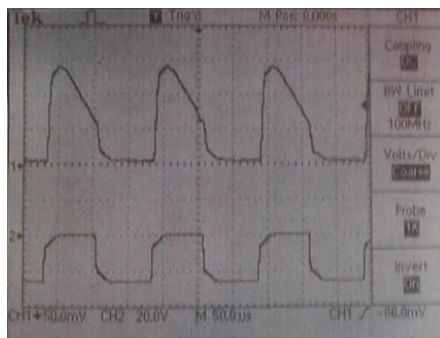
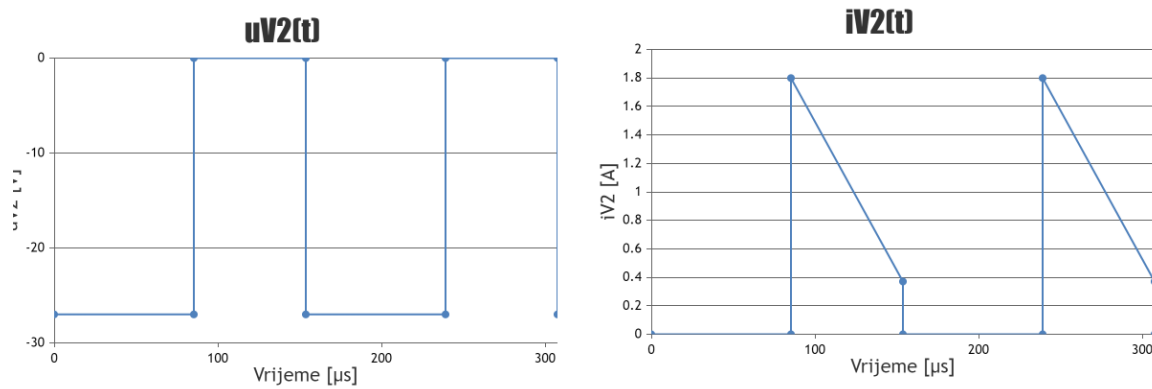
Odaberi grafove

- Napon na induktivitetu
- Napon na ventilu V1
- Napon na ventilu V2
- Struja na ventilu V1
- Struja na ventilu V2
- Struja kroz induktivitet

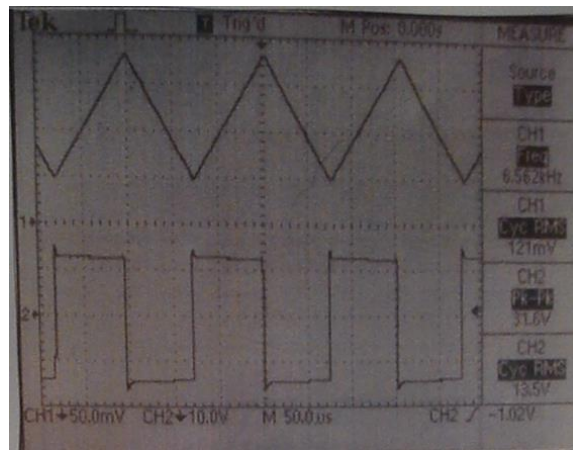
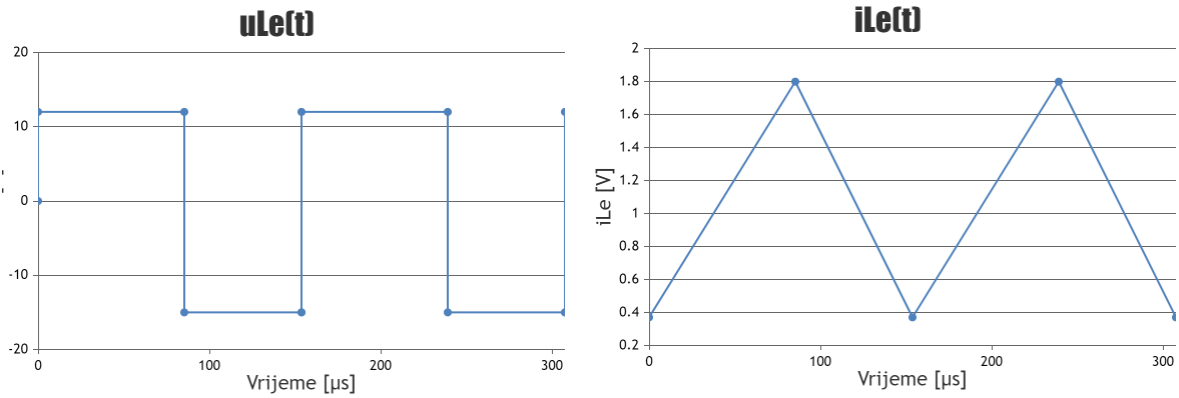
Slika 6.5. Prvi set parametara



Slika 6.6. Rezultati simulacije (gore), rezultati mjerenja (dolje) za struju i napon tranzistora kontinuirani način rada



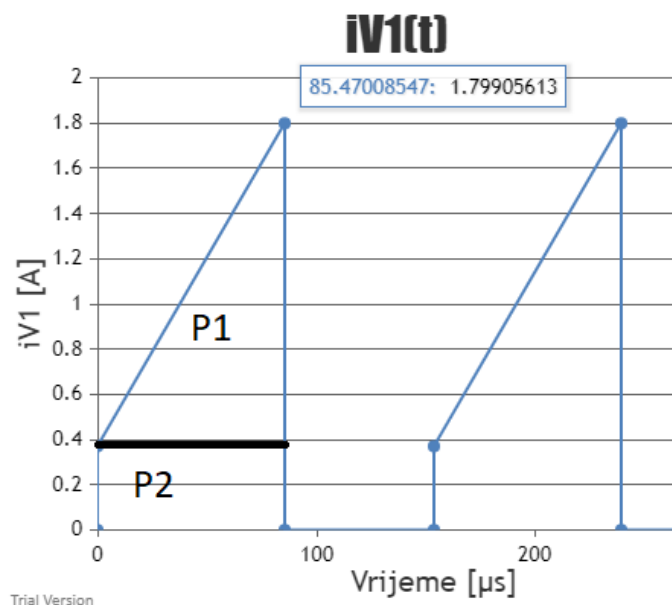
Slika 6.7. Rezultati simulacije (gore), rezultati mjerenja (dolje) za struju i napon diode kontinuirani način rada



Slika 6.8. Rezultati simulacije (gore), rezultati mjerenja (dolje) za napon i struju prigušnice kontinuirani način rada

Pri simulaciji pretvaračke komponente su modelirane idealnim sklopkama, vrijeme uklopa diode i tranzistora je trenutno te nema parazitskih svojstava za razliku od realnog slučaja pri testiranju pretvarača. Iz toga razloga rezultati dobiveni simulacijom se razlikuju od rezultata dobivenih mjerenjem. Pogreška nastaje i na mjernim instrumentima jer svaki instrument ima svoju propisanu pogrešku u kojoj rezultat odstupa od stvarnog.

Vrijednosti za pojedine točke (vrijeme vođenja ventila, vršne vrijednosti, perioda, početne vrijednosti struje) mogu se iščitati klikom na točku te iz toga je moguće analizirati rezultate. Mjerni instrumenti mjere srednju vrijednost te stoga je potrebno iz rezultata simulacije izračunati vrijednosti za pojedine slučaje (slika 6.9.).



Slika 6.9. Prikaz izračuna srednje vrijednosti struje

Površina ispod grafa je vrijednost koju mjere instrumenti pa je struja kroz ventil V1:

$$I_{V1(0)} = P1 + P2 = \left(\frac{(1.79 - 0.37) \cdot 85.47}{2} \right) \frac{1}{153.85} + 85.47 \cdot 0.37 \frac{1}{153.85} = 0.59A$$

Rezultati mjerenja za kontinuirani način rada uzlaznog pretvarača predstavljeni su u tablici 6.2.

Tablica 6.2 Rezultati mjerenja i simulacije za kontinuirani rad

	Izlazni napon Ud	Struja kroz ventil V2 (dioda)	Struja kroz ventil V1 (tranzistor)	Struja kroz induktivitet L_e
Mjerenje	27 V	0.52 A	0.62 A	1.14 A
Simulacija	27 V	0.48 A	0.59 A	1.08 A
Odstupanje	0 %	8.33 %	5.08 %	5.55 %

Tablica 6.2. prikazuje odstupanje između dobivene srednje vrijednosti struje i izračunate iz simulacije. Razlozi odstupanja su idealne sklopke, zanemarenje otpora prigušnice te znatno pojednostavljenje matematičkog modela.

Slika 6.10. prikazuje unos za drugi set parametara.

Unos podataka

Ulazni napon [V]

12

Preporučena unesena vrijednost mora biti od 10V do 30V

Frekvencija [kHz]

2.4

Preporučena unesena vrijednost mora biti od 2kHz do 10kHz

Izlazni napon [V]

27

Preporučena unesena vrijednost mora biti od 20V do 50V

Induktivitet L_e [μH]

718

Preporučena unesena vrijednost mora biti od 500 μH do 1000 μH

Otpor R_d [Ω]

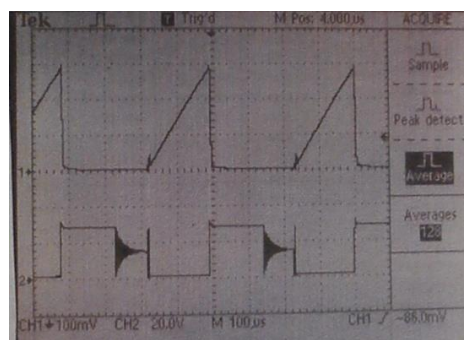
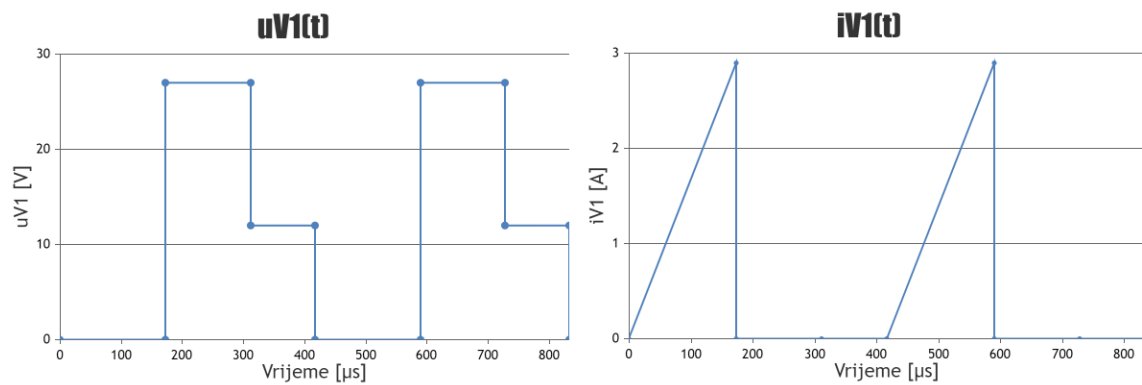
56

Unesena vrijednost mora biti od 20 Ω do 100 Ω

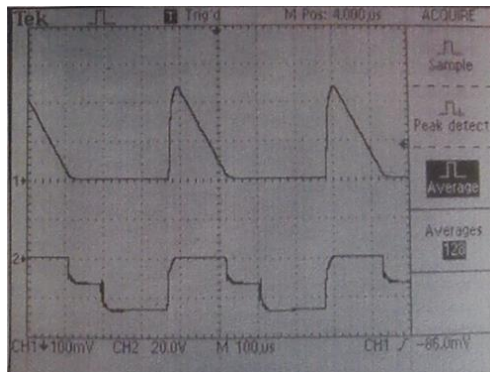
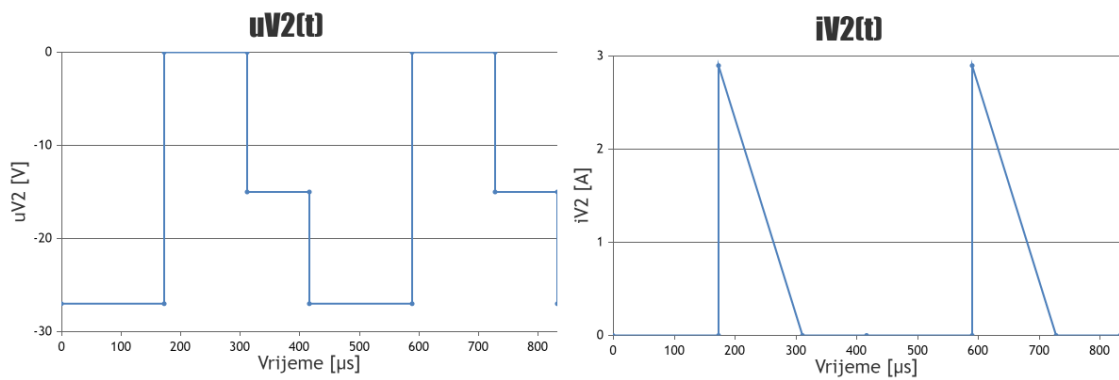
Odaberi grafove

- Napon na induktivitetu
- Napon na ventilu V1
- Napon na ventilu V2
- Struja na ventilu V1
- Struja na ventilu V2
- Struja kroz induktivitet

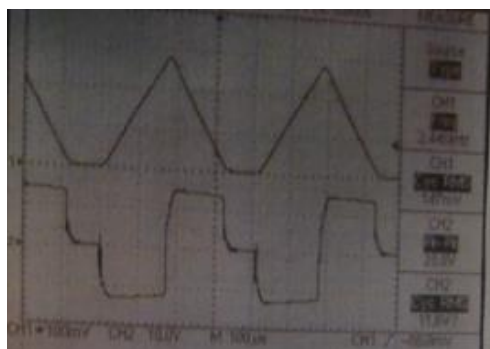
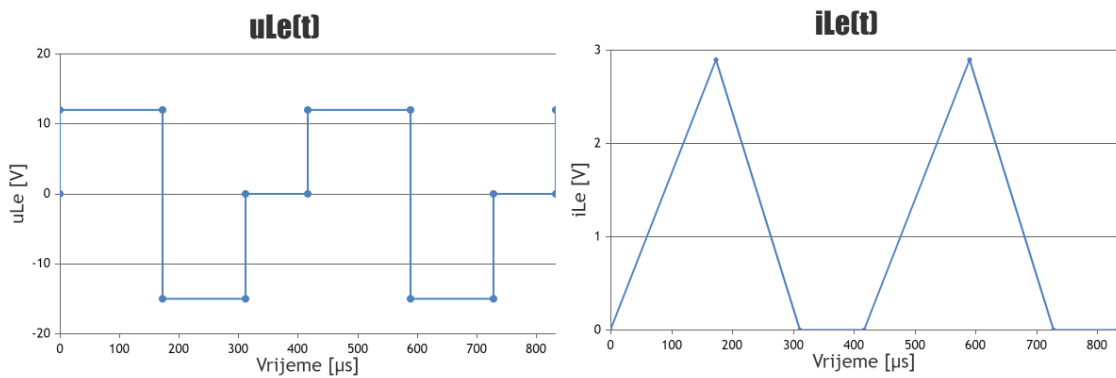
Slika 6.10. Drugi set parametara



Slika 6.11. Rezultati simulacije (gore), rezultati mjerenja (dolje) za napon i struju tranzistora diskontinuirani način rada



Slika 6.12. Rezultati simulacije (gore), rezultati mjerenja (dolje) za napon i struju diode diskontinuirani način rada



Slika 6.13. Rezultati simulacije (gore), rezultati mjerenja (dolje) za napon i struju prigušnice diskontinuirani način rada

Rezultati mjerenja za diskontinuirani način rada uzlaznog pretvarača prikazani su u tablici 6.3.

Tablica 6.3 Rezultati mjerenja i simulacije za diskontinuirani način rada

	Izlazni napon U_d [V]	Struja kroz ventil V2 (dioda) [A]	Struja kroz ventil V1 (tranzistor) [A]	Struja kroz induktivitet L_e [A]
Mjerenje	27 V	0.53 A	0.63 A	1.12 A
Simulacija	27 V	0.48 A	0.6 A	1.08 A
Odstupanje	0 %	10.42%	5 %	3.71 %

7. ZAKLJUČAK

Zadatak diplomskog rada bio je napraviti interaktivno web-sučelje koje će budućim generacijama studenata uveliko olakšati shvaćanje matematičkih modela istosmjernih pretvarača napona, ispravljača i izmjenjivača.

Za svaki pretvarač, definiran je interval u kojemu se preporučuje da budu unesene vrijednosti parametara. Tako se omogućuje da red veličine struja i napona bude kontrolabilan jer iznos parametara bitno utječu na vrijednost struja i napona karakterističnih veličina. Cilj je da modeli pretvarača imaju iste parametre kao pretvarači koji se koriste na laboratorijskim vježbama iz kolegija Osnove energetske elektronike.

Iako zbog znatnog pojednostavljenja matematičkog modela rezultati simulacije i rezultati mjerenja za kontinuirani način rada uzlaznog istosmjernog pretvarača pokazali su da srednje vrijednosti struja kroz ventil V1 se razlikuje za 5.08% od rezultata mjerenja. Za diskontinuirani način rada rezultati simulacije se razlikuju za 5%. Za prvu aproksimaciju karakterističnih vrijednosti struja pretvarača ovo je prihvatljivo.

Područje primjene dobivenih rezultata na osnovu simulacije i mjerenja je omogućiti studentima zaključivanje kako fizikalno radi pretvarač na osnovu matematičkog modela.

LITERATURA

- [1] D. Pelin, K. Miličević, D. Vulin: „Priručnik za praktikum iz osnova energetske elektronike“, Elektrotehnički fakultet Osijek, Osijek, 2013.
- [2] I. Flegar: „Elektronički energetske pretvarači“, KIGEN, Zagreb, 2010.
- [3] React, dostupno na: https://www.w3schools.com/whatis/whatis_react.asp [Preuzeto: 10.09.2021.]
- [4] DeveloperMozilla, dostupno na: <https://developer.mozilla.org/en-US/docs/Web/HTML> [Preuzeto: 10.09.2021.]
- [5] DeveloperMozilla – CSS Basics, dostupno na: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics [Preuzeto: 10.09.2021.]
- [6] DeveloperMozilla -- JavaScript, dostupno na: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [Preuzeto: 10.09.2021.]
- [7] React-router, dostupno na: <https://reactrouter.com/web/guides/quick-start> [Preuzeto: 10.09.2021.]
- [8] Redux, dostupno na: <https://javascript.plainenglish.io/the-only-introduction-to-redux-and-react-redux-youll-ever-need-8ce5da9e53c6> [Preuzeto: 10.09.2021]
- [9] Flexiple, dostupno na: <https://flexiple.com/react/learn-redux-and-its-usage-with-react-by-building-books-finder-app/> [Preuzeto: 10.09.2021.]
- [10] Firebase, dostupno na: <https://firebase.google.com/docs/build> [Preuzeto: 10.09.2021.]
- [11] CanvasJS, dostupno na: <https://canvasjs.com/docs/charts/intro/> [Preuzeto: 10.09.2021.]
- [12] W3Schools, dostupno na: https://www.w3schools.com/html/html_form_input_types.asp [Preuzeto: 02.09.2021.]
- [13] Typescript, dostupno na: <https://www.typescriptlang.org/docs/handbook/utility-types.html> [Preuzeto: 02.09.2021.]

SAŽETAK

Napravljena je analiza svih pretvarača koji se koriste na laboratorijskim vježbama iz kolegija Osnove energetske elektronike. Proučen je osnovni način rada s React-om, koje uključuje programski jezik HTML, CSS i JavaScript. Izvršena je simulacija svih pretvarača. Provedena je detaljna analiza za jedan pretvarač i napravljena usporedba između rezultata simulacije i mjerenja.

Ključne riječi: React, HTML, CSS, JavaScript, pretvarači

TITLE: Interactive web interface for converter operation analysis

ABSTRACT

An analysis of all converters used in laboratory exercises from the Basics of Power Electronics course was conducted. The basic way of working with React, which includes the programming language HTML, CSS, and JavaScript, has been studied. Simulation of all converters was performed. A detailed analysis for one converter was performed and a comparison was made between the simulation and measurement results.

Key words: React, HTML, CSS, JavaScript, converters

ŽIVOTOPIS

Miloš Mišljenović rođen je 18.12.1996. godine u Vukovaru u Republici Hrvatskoj. Osnovnu školu pohađao je u mjestu Bobota. Dugi niz godina bavio se sportom ,a posebice nogomet i teretanu. Nakon osnovne škole upisuje „Tehničku školu Nikola Tesla Vukovar“, smjer Tehničar za računarstvo. Sudjelovao na raznim natjecanjima. Nakon uspješne položene mature prvi odabir studija bio mu je Sveučilišni studij elektrotehnike na Elektrotehničkom fakultetu u Osijeku (sada Fakultet elektrotehnike, računarstva i informacijskih tehnologija). Na drugoj godini studija opredjeljuje se za izborni blok Elektroenergetike. Nakon uspješno završenog preddiplomskog studija upisuje diplomski studij smjer Industrijska elektroenergetika.

PRILOG

Na poveznici <https://energetskipretvaraci.web.app/> dostupna je aplikacija koja je odrađena kroz diplomski rad