

# Harmonijska analiza istosmjernog silaznog pretvarača s pomoću virtualnog spektralnog analizatora

---

**Firić, Antonio**

**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:200:602399>

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

*Download date / Datum preuzimanja:* **2025-04-01**

*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

**Harmonijska analiza istosmjernog silaznog pretvarača s  
pomoću virtualnog spektralnog analizatora**

Diplomski rad

Antonio Firić

Osijek, 2021.

## SADRŽAJ

<b>1. UVOD .....</b>	<b>1</b>
<b>2. PREGLED PODRUČJA TEME.....</b>	<b>2</b>
<b>3. KOMUNICIRANJE IZMEĐU RAČUNALA I OSCIOSKOPA.....</b>	<b>3</b>
<b>4. RAZRADBA GRAFIČKOG SUČELJA.....</b>	<b>4</b>
<b>4.1. Migriranje grafičkog sučelja.....</b>	<b>4</b>
<b>4.2. Ispravljanje pogrešaka migracije .....</b>	<b>6</b>
<b>5. TESTIRANJE GRAFIČKOG SUČELJA .....</b>	<b>13</b>
<b>6. ZAKLJUČAK.....</b>	<b>25</b>
<b>LITERATURA .....</b>	<b>26</b>
<b>SAŽETAK.....</b>	<b>27</b>
<b>ABSTRACT .....</b>	<b>27</b>
<b>ŽIVOTOPIS.....</b>	<b>28</b>
<b>PRILOG .....</b>	<b>29</b>
<b>PRILOG A: Upute za rukovanje virtualnim spektralnim analizatorom.....</b>	<b>29</b>
<b>PRILOG B: Kod unutar MATLAB-a.....</b>	<b>33</b>

## 1. UVOD

U ovom diplomskom radu prezentirat će se sposobnost kontrole osciloskopa TDS 224 proizvođača „Tektronix“ pomoću grafičkog sučelja. Grafičko sučelje je napravljeno unutar MATLAB-a u aplikaciji GUIDE. Već napravljenu aplikaciju je potrebno migrirati u noviju i ažuriranu MATLAB aplikaciju App Designer i potom ju poboljšati. Razlika između App Designer-a i GUIDE-a je u tome što je unutar App Designer-a programiranje objektno-orientirano što uvodi značajnije razlike u usporedbi s programiranjem u GUIDE aplikaciji. Nakon migracije potrebno je testirati grafičko sučelje na istosmjernom silaznom pretvaraču.

Ovaj rad se sastoji od pet poglavlja. Prvo poglavlje je uvod. Nakon uvoda slijedi drugo poglavlje i u njemu se nalazi pregled područja teme rada. U trećem poglavlju se nalazi opis načina komunikacije između računala i osciloskopa. Potom se u četvrtom poglavlju nastoji opisati postupak migracije iz GUIDE aplikacije u App Designer aplikaciju te sve probleme koji su se dogodili prilikom migracije. Peto poglavlje vezano je za testiranje grafičkog sučelja u kojem ćemo napraviti harmonijsku analizu istosmjernog silaznog pretvarača upravljanjem osciloskopa preko napravljenog grafičkog sučelja. Šesto poglavlje predstavlja zaključak.

## 2. PREGLED PODRUČJA TEME

F. Pitinac [1] pokazao je kako se odvija komunikacija između osciloskopa i računala. Također je pokazao kako napraviti sučelje unutar GUIDE aplikacije unutar MATLAB okruženja.

Pomoću videa [2] na službenom kanalu MATLAB-a moguće je saznati kako uspješno migrirati GUIDE aplikaciju u App Designer aplikaciju.

Pomoću videa [3] službenog MATLAB kanala je predstavljeno objektno orijentirano programiranje unutar MATLAB okruženja na primjeru aplikacije. Video je ključan u razumijevanju rada App Designera i programiranju unutar njega.

M. Čengić [4] je matematičkim postupkom dokazao rastav pravokutnog signala u Fourierov red koji nam pomaže u analitičkom postupku obrade pravokutnog valnog oblika napona.

I. Flegar je u knjizi [5] detaljno objasnio matematički model silaznog istosmjernog pretvarača napona i izveo faktor vođenja silaznog pretvarača preko matematičkog modela koja je pomogla pri testiranju virtualnog spektralnog analizatora.

### 3. KOMUNICIRANJE IZMEĐU RAČUNALA I OSCILOSKOPA

Za povezivanje osciloskopa TDS 224 s računalom potrebno je imati RS-232 kabel, adapter za prelazak s RS 232 na USB i USB kabel. Isto tako za povezivanje osciloskopa s računalom potrebno je skinuti i instalirati driver CH340 koji omogućuje rad integriranom krugu CH340. U [6] je navedeno kako integrirani krug CH340 služi za konvertiranje signala iz USB sučelja u TTL(serijsko) sučelje i obratno. Još jedna primjena drajvera CH340 je i pri povezivanju nekih vrsta Arduina s računalom.

Prema [1] za upravljanje osciloskop se koristi SCPI naredbeni jezik. Unutar SCPI naredbenog jezika postoje mnoge naredbe i upiti koji se šalju prema osciloskopu. Naredbe mijenjaju postavke osciloskopa ili izvršavaju neku radnju, dok upiti uzrokuju vraćanje podataka s osciloskopa na računalo ovisno o upućenoj upitnoj riječi. Popis svih naredbi i upita koje je moguće poslati na osciloskop nalazi se na [7].

Osciloskop TDS 224 je odabran jer se u njemu nalazi integrirani krug CH340 i uz pomoć njega je provedena komunikacija. Ako osciloskop nema integrirani krug CH340 nemoguć je rad između računala i osciloskopa. Ako se koristi drugi osciloskop potrebno je provjetriti manual osciloskopa kako bi se provjerilo potrebne naredbe i promijeniti je ako je potrebno. Na slici 3.1. prikazan je osciloskop „Tektronix“ TDS 224.



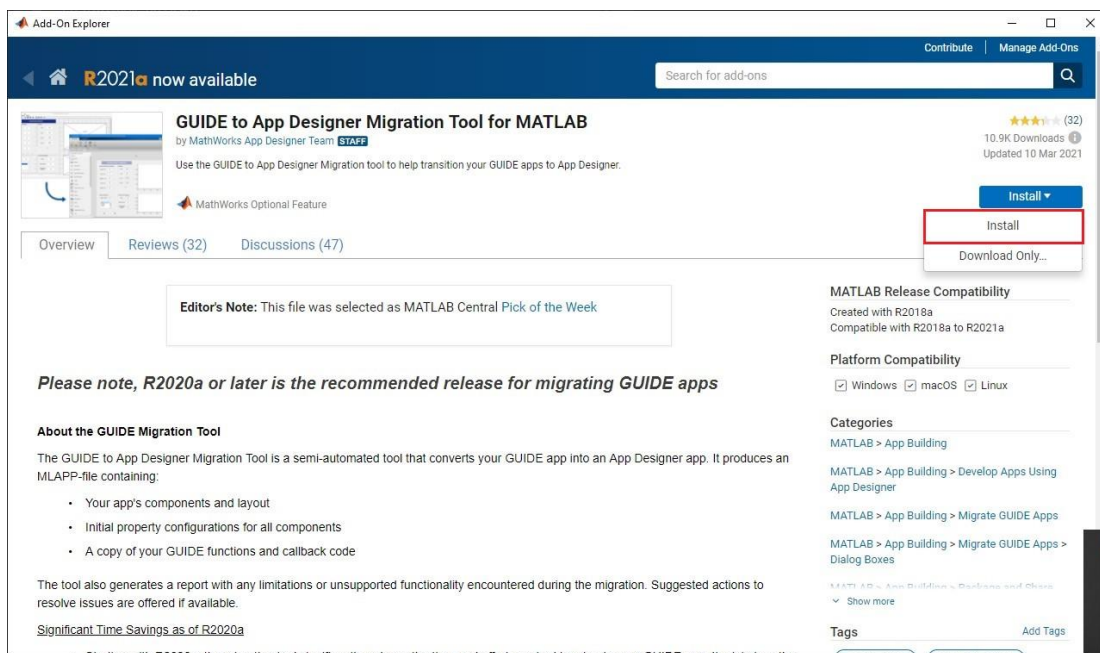
Slika 3.1. Osciloskop „Tektronix“ TDS 224

## 4. RAZRADBA GRAFIČKOG SUČELJA

Razrada grafičkog sučelja će se odvijati u dva koraka. Prvi korak je migriranje već napravljenog grafičkog sučelja u aplikaciji GUIDE u aplikaciju App Designer. Pravljenje grafičkog sučelja u aplikaciji GUIDE je opisano u [1]. Drugi korak je ispravljanja svih grešaka migracije i uspješno testiranje grafičkog sučelja na poznatim valnim oblicima. Osim što je App Designer modernija i jednostavnija aplikacija, glavna prednost joj je što se unutar nje koristi objektno orijentirano programiranje. Objektno orijentirano programiranje je programiranje s pomoću kojeg je moguće definirati strukture koje se nazivaju objekti i kombiniranje pridodanih svojstava objektu s operacijama koje se vrše na tom objektu. Više o objektno orijentiranom programiranju u MATLAB - u se može saznati na [8].

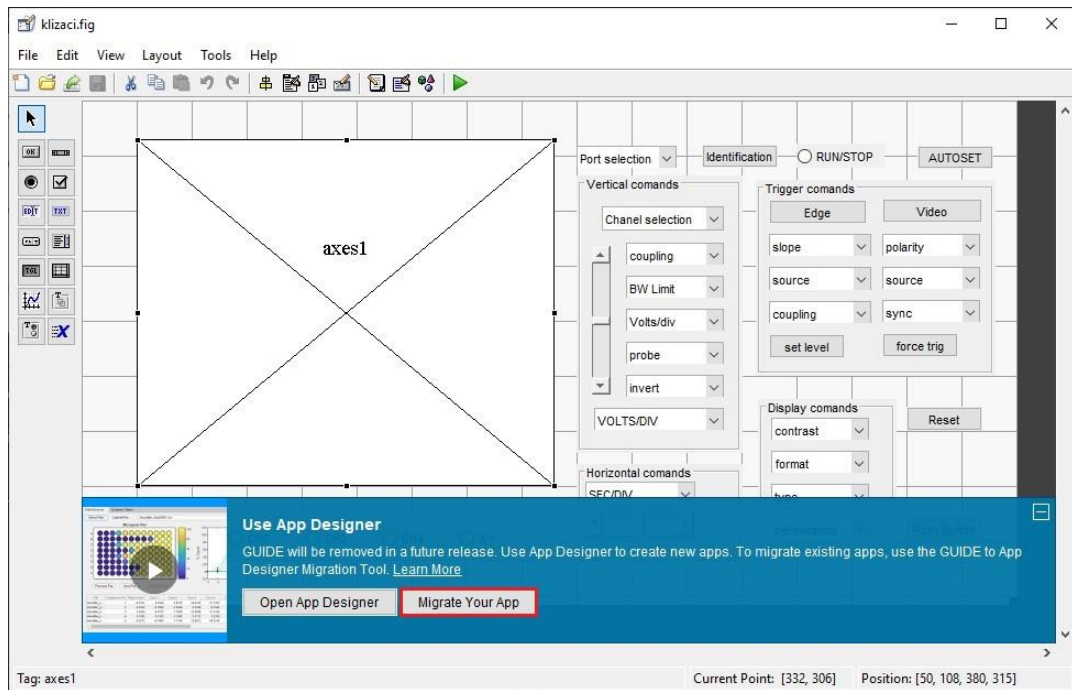
### 4.1. Migriranje grafičkog sučelja

Prvi korak migriranja je preuzimanje GUIDE to App Designer Migration Tool for MATLAB aplikacije s [9]. Za preuzimanje GUIDE to App Designer Migration Tool for MATLAB potrebno je imati MathWorks račun. Na slici 4.1. prikazano je izgled sučelja za instaliranje GUIDE to App Designer Migration Tool for MATLAB aplikacije gdje je crvenim pravokutnikom označen gumb za instaliranje. Nakon unosa podataka za MathWorks račun potrebno je pričekati da instalacija završi.



Slika 4.1. Instaliranje aplikacije za migriranje

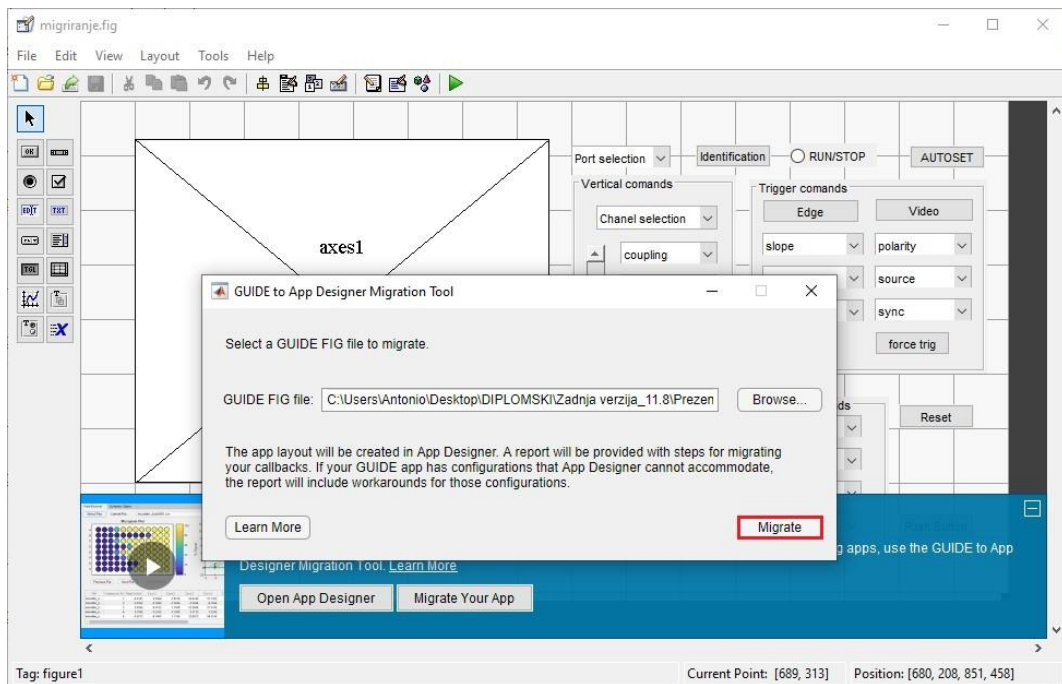
Drugi korak migriranja aplikacije je otvaranja GUIDE-a pomoću naredbe guide u Command Window-u MATLAB-a. Nakon otvaranja GUIDE-a potrebno je odabrati datoteku za migriranje te kliknuti na opciju Migrate Your App kako je prikazano na slici 4.2. Datoteka za migriranje ima nastavak .fig na kraju imena datoteke.



Slika 4.2. Izgled sučelja prilikom započinjanja migriranja

Treći korak je ujedno i završni. Potrebno je kliknuti na gumb Migrate nakon odabiranja datoteke za migriranje. Treći korak je prikazan na slici 4.3.



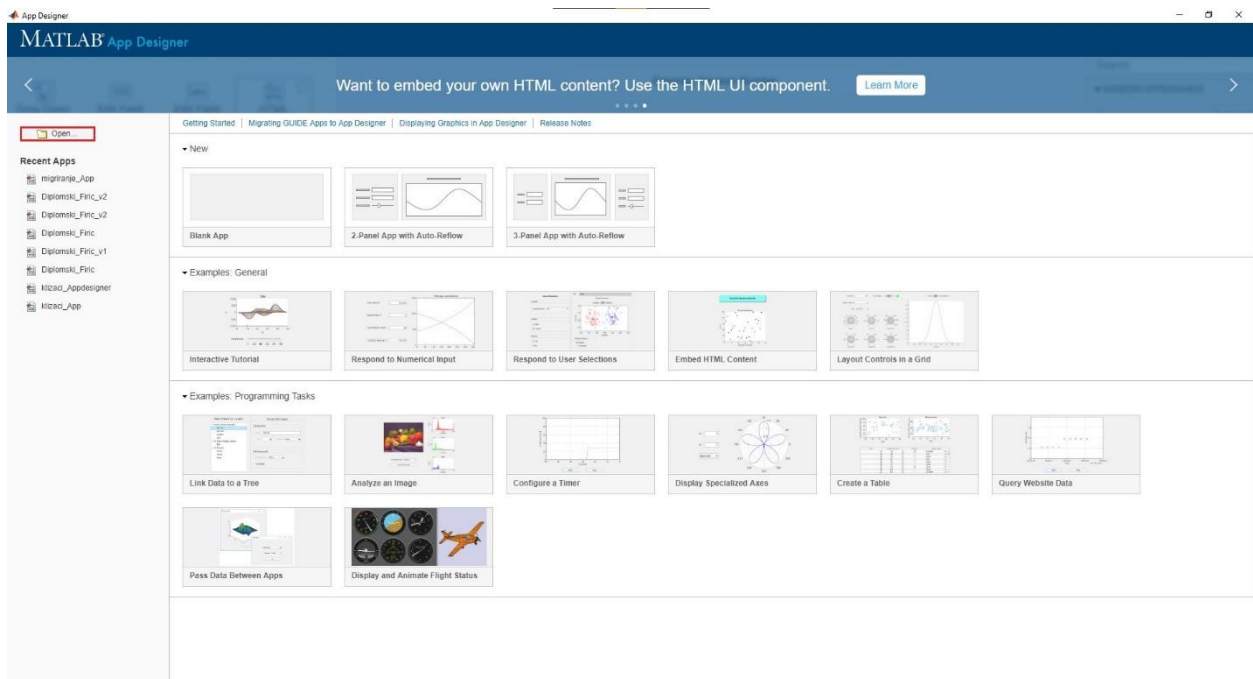


Slika 4.3. Sučelje za potvrdu migracije

Ako je migriranje uspješno unutar mape u kojem se nalazi i datoteka koju je potrebno migrirati nalaziti će se i nova migrirana datoteka s nastavkom .mlapp. Postupak migracije u obliku videa može se pronaći i na službenom YouTube kanalu MATLAB-a [2].

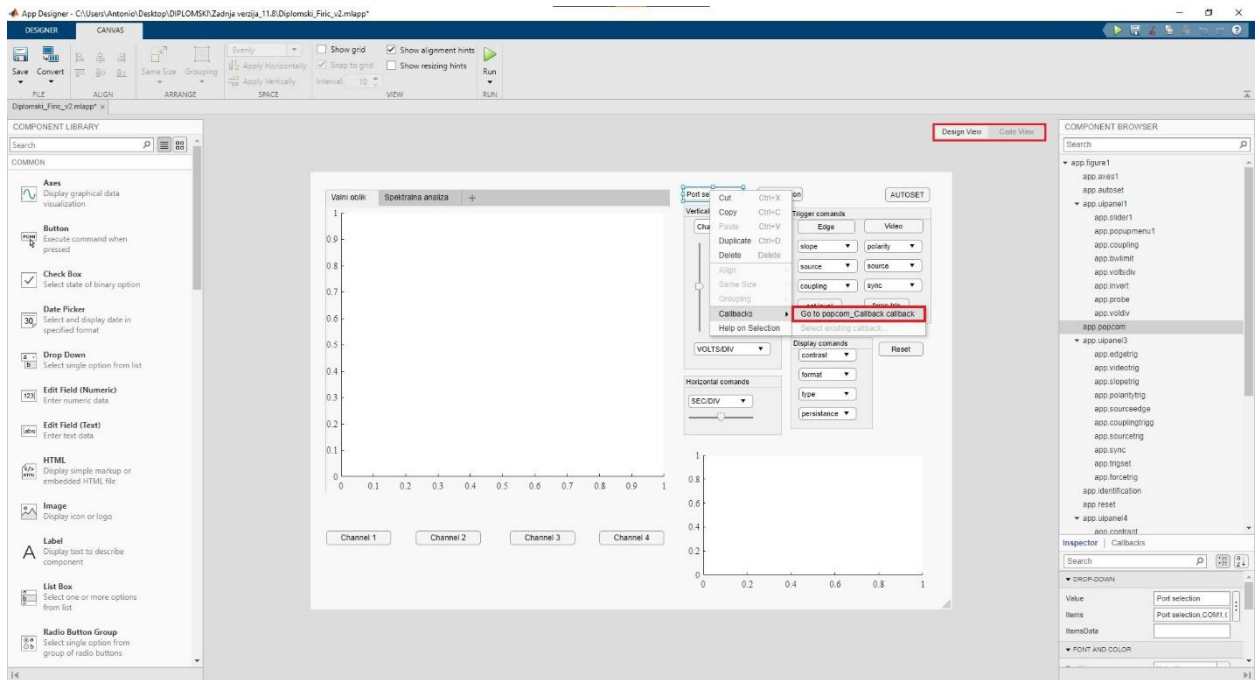
## 4.2. Ispravljanje pogrešaka migracije

Za uređivanje grafičkog sučelja prvo je potrebno pokrenuti aplikaciju App Designer unutar MATLAB-a. To se radi upisom naredbe `appdesigner` unutar Command Window MATLAB sučelja. Nakon ulaska u aplikaciju App Designer potrebno je otvoriti datoteku koju je potrebno urediti pomoću gumba `open`. Postupak je prikazan na slici 4.4.



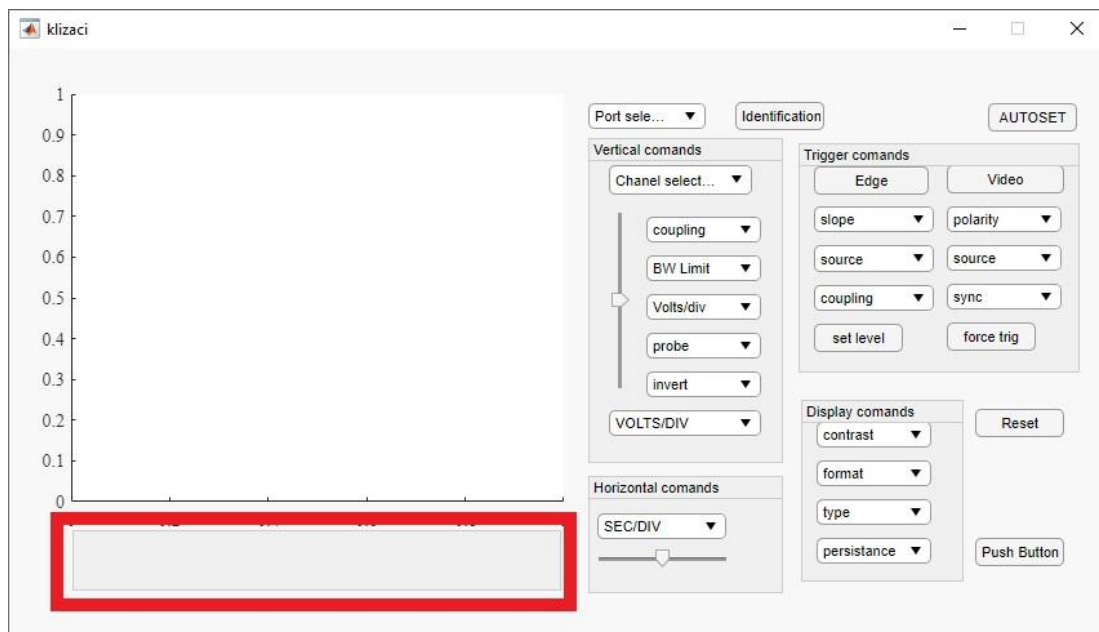
Slika 4.4. Sučelje prilikom ulaska u App Designer

Kako bi bilo moguće vidjeti i uređivati kod potrebno je kliknuti na gumb Code view, kako je prikazano na slici 4.5. Vraćanje na Design view je moguće klikom na pripadajući gumb. Ako je potrebno prikazati dio koda samo za određeni gumb, potrebno je pritisnuti desni klik na gumb i odabrati opciju go to callback.

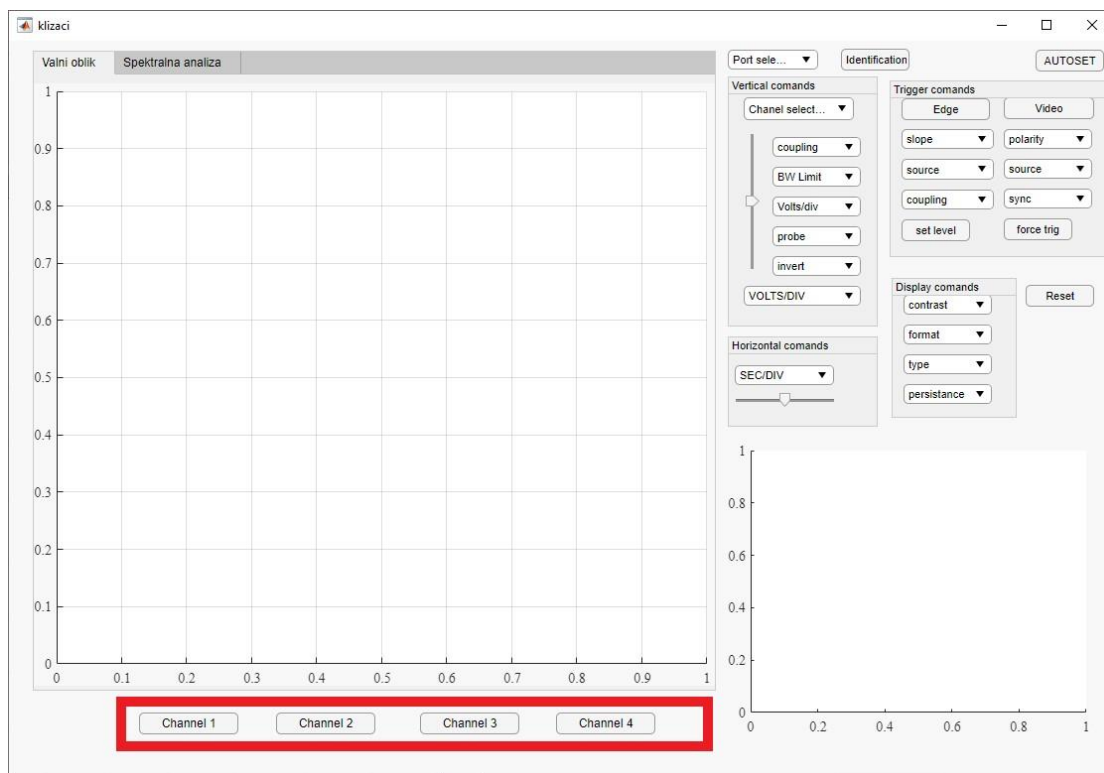


Slika 4.5. Sučelje App Designera

Prvi problem koji se pojavio pri migraciji je nepodržavanje radio button-a. Radio button-i imaju funkciju odabir kanala i „pozivanje“ slike s osciloskopa. Problem je riješen dodavanjem novih gumbova, kopiranjem koda i njegovim ispravljanjem. Na slici 4.6. prikazano je sučelje virtualnog spektralnog analizatora gumbova odmah nakon migracije. Slika 4.7. prikazuje sučelje virtualnog spektralnog analizatora u kasnijim verzijama nakon njihovog popravljnja. Na obje slike crvenim pravokutnikom označeno je mjesto gdje se gumbovi nalaze.



Slika 4.6. Sučelje virtualnog spektralnog analizatora nakon migracije



Slika 4.7. Sučelje virtualnog spektralnog analizatora u kasnijim verzijama

Na slici 4.8. se nalazi usporedba koda nakon migracije i u zadnjoj verziji za gumb koji invertira signal.

```

% Value changed function: invert
function invert_callback(app, event)
    % hObject    handle to invert (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Hints: contents = cellstr(get(hObject,'String')) returns invert contents as cell array
    %         contents{get(hObject,'Value')} returns selected item from invert
    global channelchoice;

    s = serial(app.popcom.Value);

    fopen(s)

    contents = cellstr(get(hObject,'String'));
    invert = contents{get(hObject,'Value')};
    fprintf(s,[channelchoice ':INVERT ' invert]);
    fclose(s)
    delete(s)
end

```

a)

```

% Value changed function: invert
function invert_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s,[app.popupmenu1.Value ':INVERT ' app.invert.Value]);
    fclose(s)
    delete(s)
end

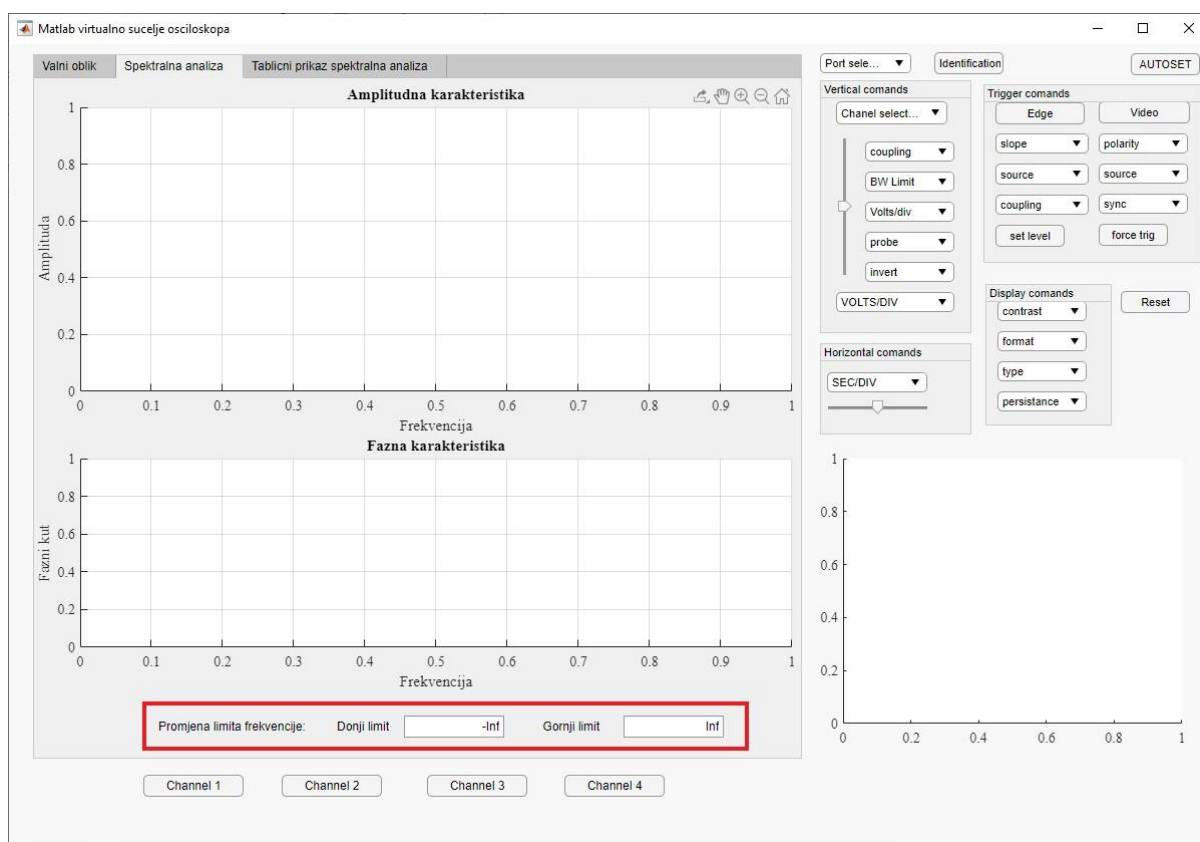
```

b)

Slika 4.8. a) Kod za gumb invertiranja signala nakon migracije

b) Kod za gumb invertiranja signala u zadnjoj verziji

Prilikom testiranja virtualnog spektralnog analizatora bilo je potrebno dodati još funkcija postojećem virtualnom spektralnog analizatora kako bi on bio funkcionalan i lakši za upotrebu. Jedna od funkcija koja je dodana je upisivanje limita na x-osi kod stupčastog prikaza rezultata spektralne analize jer je prikaz frekvencije bio postavljen na automatsko. Stoga je bilo teško očitati harmonike zbog njihove brojnosti. Na slici 4.9. je vidljivo mjesto za upisivanje limita unutar sučelja i označeno je crvenim pravokutnikom.



Slika 4.9. Mjesto za upisivanje limita frekvencije

Nakon što su promijenjeni limiti sljedeća potreba je bila smanjiti debljinu stupaca na stupčastom dijagramu. Smanjivanje debljine stupaca se postiže dodavanjem argumenta u funkciju za crtanje stupčastog dijagrama. Na slici 4.10. prikazan je kod za crtanje stupčastog dijagrama.

```

data_time_domain = [t.' a1];
br_harm = 20;
[fft_napon_out, fft_kut_out, frekvencije_x_os] = SnagaNaFrekv(data_time_domain , br_harm);

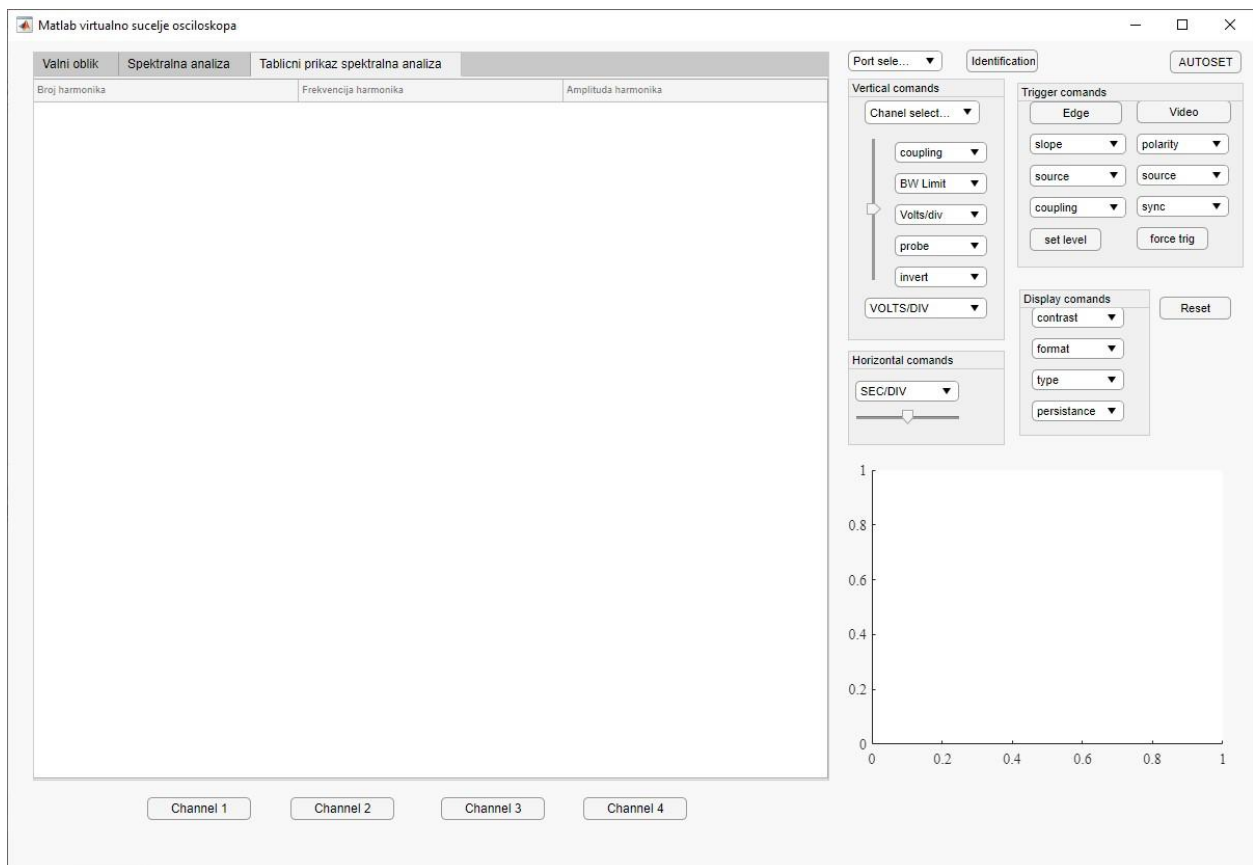
bar(app.UIAxes, frekvencije_x_os, fft_napon_out, 0.2); % 0.2 je širina stupca
grid(app.UIAxes, 'on');

bar(app.UIAxes2, frekvencije_x_os, fft_kut_out, 0.2);
grid(app.UIAxes2, 'on');

```

Slika 4.10. Kod za crtanje stupčastog dijagrama kod spektralne analize

Isto tako bi bilo vrlo korisno i olakšavajuće moći vidjeti harmonike u obliku tablice. Ta funkcija je dodana i na slici 4.11. se nalazi njezino mjesto unutar sučelja, a na slici 4.12. je vidljiv kod koji omogućuje tu funkciju.



Slika 4.11. Sučelje za tablični prikaz spektralne analize

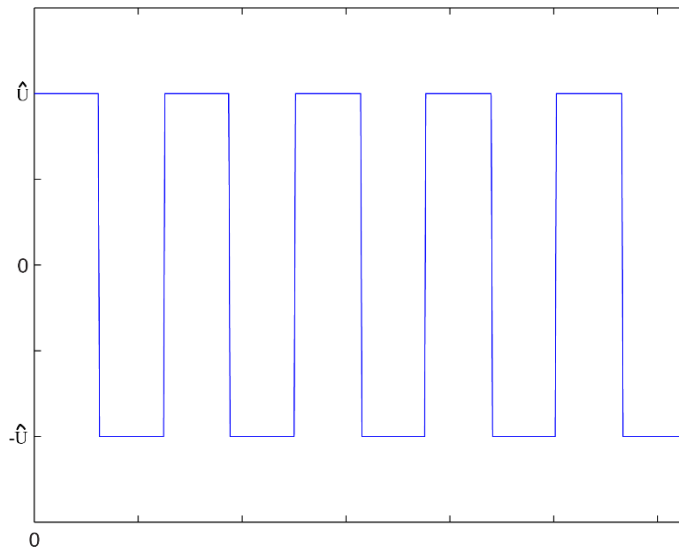
```
c = 0:1:br_harm;  
  
T = table(c', frekvencije_x_os', fft_napon_out');  
app.UITable.DisplayData = T;
```

Slika 4.12. Kod za prikazivanje tabličnog prikaza spektralne analize

U sljedećem poglavlju sučelje će biti testirano na silaznom pretvaraču. Jedna od najvećih pomoći u razumijevanju rada App Designera bio je video objavljen s službenog kanala MATLAB-a [3].

## 5. TESTIRANJE GRAFIČKOG SUČELJA

Za dolazak do zaključka jesu li rezultati spektralne analize točni potrebno je uzeti naponski signal i njega matematički obraditi, potom rezultat usporediti s dobivenim unutar sučelja. Kao poznati naponski signal uzet je signal pravokutnog valnog oblika napona, a valni oblik napona na zavojnici silaznog pretvarača će služiti za testiranje virtualnog spektralnog analizatora. Na slici 5.1. prikazan je pravokutni signal valnog oblika napona.



Slika 5.1. Pravokutni signal valnog oblika napona

Prema [4] rastav pravokutnog signala u Fourierov red glasi:

$$u(t) = \sum_{k=1}^{\infty} \left( \frac{\hat{U}}{\pi n} (-2 \cos(\pi n) + 2) \right) \sin kt \quad n = 1, 2, 3, 4, 5, \dots \quad (5-1)$$

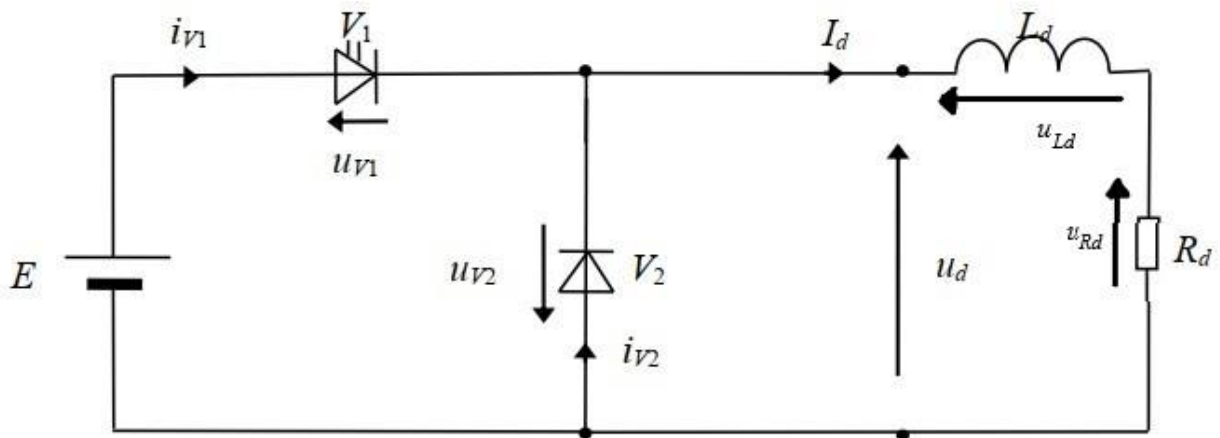
Razvojem funkcije  $u(t)$  za prvih pet članova dobiveno je:

$$u(t) = \frac{4\hat{U}}{\pi} \sin t + \frac{4\hat{U}}{3\pi} \sin 3t + \frac{4\hat{U}}{5\pi} \sin 5t + \frac{4\hat{U}}{7\pi} \sin 7t + \frac{4\hat{U}}{9\pi} \sin 9t + \dots \quad (5-2)$$

Rad virtualnog spektralnog analizatora će biti testiran na silaznom istosmjernom pretvaraču napona (engl. Buck converter). Silazni istosmjerni pretvarač napon je odabran jer se mijenjanjem



ulaznog napona silaznog pretvarača mijenjaju ustaljena stanja tzv. put u kaos udvostručavanjem periode što je korisno za testiranje virtualnog spektralnog analizatora. Na slici 5.2. nalazi se nadomjesna shema spoja silaznog istosmjernog pretvarača napona.



Slika 5.2. Nadomjesna shema silaznog pretvarača napona

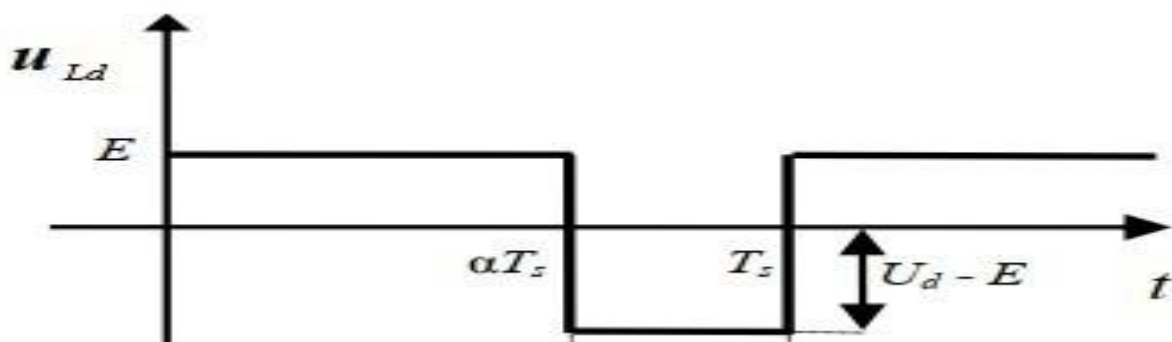
Za pretpostavljenu shemu spoja istosmjernog silaznog pretvarača prema slici 5.2. vrijede sljedeće jednačbe mreže:

$$u_{Ld} = E - u_{V1} - u_{Rd} \quad (5 - 3)$$

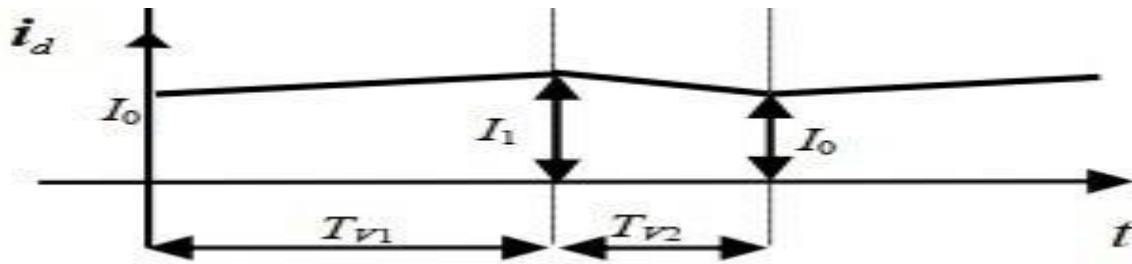
$$u_{Ld} = -u_{V2} - u_{Rd} \quad (5 - 4)$$

$$I_d = i_{V1} + i_{V2} \quad (5 - 5)$$

Na slici 5.3. prikazan je karakterističan valni oblik napona na trošilu, a na slici 5.4. prikazan je karakterističan valni oblik struje na trošilu.



Slika 5.3. Karakterističan valni oblik napona



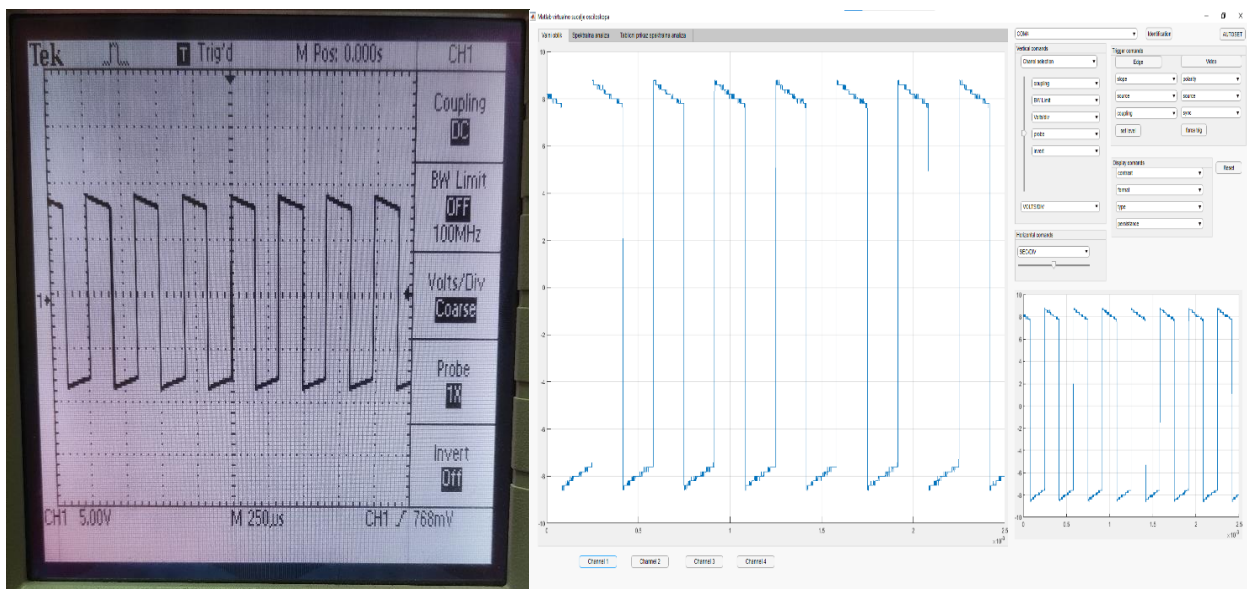
Slika 5.4. Karakterističan valni oblik struje

Za faktor vođenja prema [5] se dobije:

$$\frac{U_d(0)}{E} = \alpha \quad (5 - 6)$$

Za usporedbu pravokutnog valnog oblika s valnim oblikom napona na zavojnici silaznog pretvarača potrebno je da faktor vođenja  $\alpha$  bude 0.5 kako bi se rezultati spektralne analize valnog oblika napona mogli usporediti s pravokutnim valnim oblikom napona.

Na slici 5.3. pod a) prikazan je valni oblik napona snimljen na zavojnici silaznog pretvarača, a pod b) prikazan je isti valni oblik unutar sučelja virtualnog spektralnog analizatora. Srednja vrijednost ulaznog napona silaznog pretvarača iznosi 16.45 V, a srednja vrijednost ulazne struje iznosi 0.298 A. Na slici 5.4. prikazana je spektralna analiza valnog oblika s slike 5.1. pomoću stupčastog dijagrama, a na slici 5.5. rezultati spektralne analize su prikazani tablično.



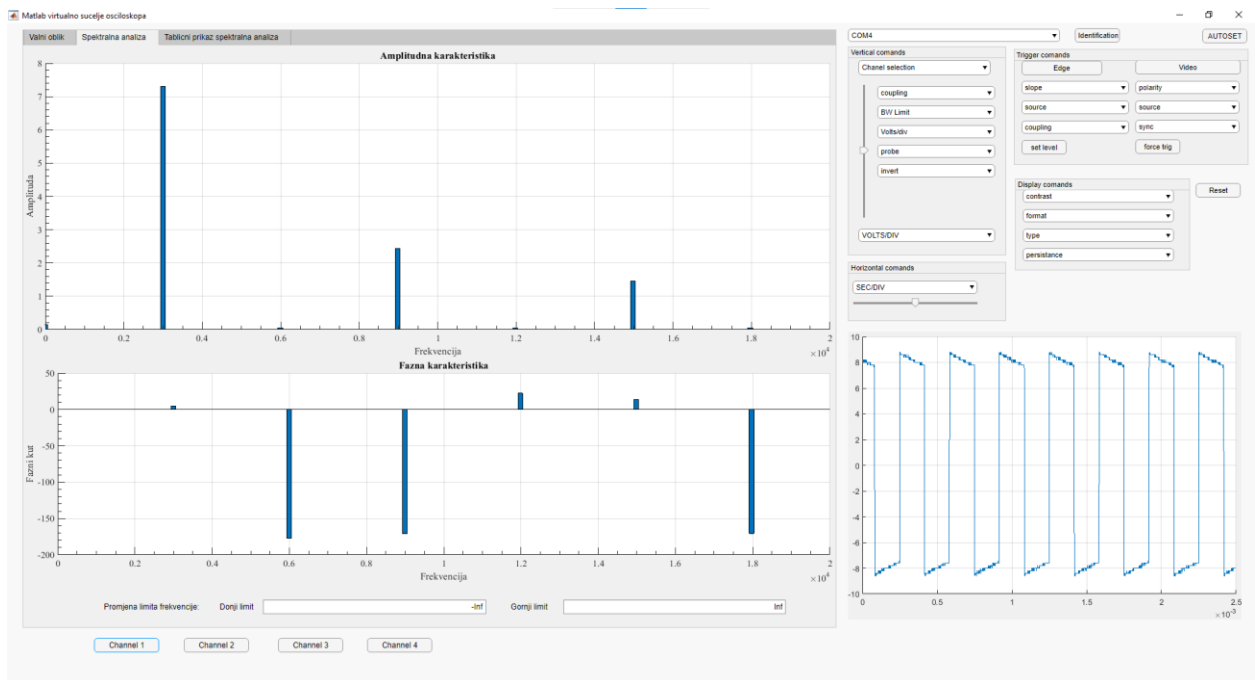
a)

b)

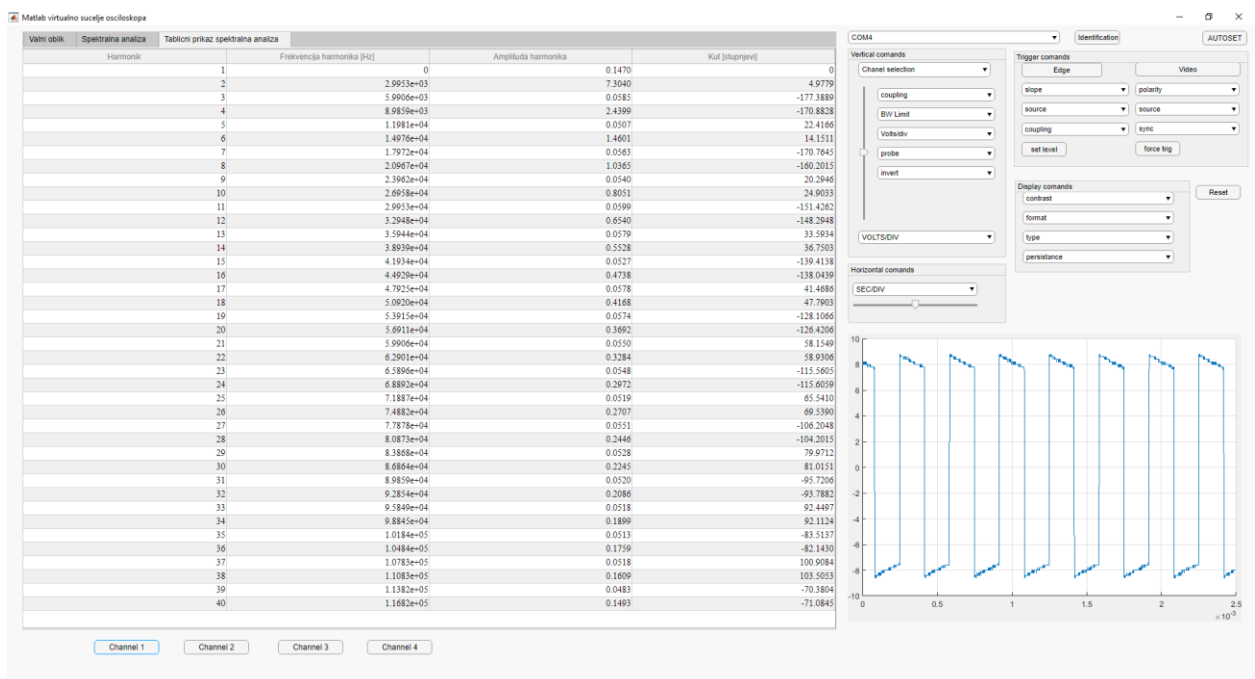
Slika 5.3. Snimljeni valni oblik;  $E = 16.45 \text{ V}$

a) Valni oblik na osciloskopu

b) Valni oblik snimljen pomoću virtualnog spektralnog analizatora



Slika 5.4. Spektralna analiza valnog oblika prikazana stupčastim dijagramom



Slika 5.5. Spektralna analiza valnog oblika napona prikazana tablično

U tablici 5.2. se nalaze rezultati spektralne analize iz slike 5.5.

**Tablica 5.2.** Rezultati spektralne analize

Harmonik	Frekvencija harmonika [Hz]	Amplituda, $U[V]$	Postotak osnovnog harmonika [%]
1	2995.3	7.30	100
3	8985.9	2.44	33
5	14976	1.46	20
7	20967	1.04	14
9	26958	0.81	11
11	32498	0.65	9

Iz tablice 5.2. je vidljivo da se osnovni harmonik poklapa s frekvencijom rada od oko 3000 Hz. Nakon prvih testiranja uvidjela se sporost crtanja grafova zbog slabe brzine prijenosa podataka između računala i osciloskopa. Grafovi unutar virtualnog spektralnog analizatora se učitaju tek nakon 5 sekundi i još sporije se učitavaju stupčasti i tablični dijagrami spektralne analize kao i tvorba Excel datoteke s vrijednostima spektralne analize kako bi se vrijednosti spektralne analize potencijalno mogle upotrijebiti u drugom programu.

Prema jednadžbi 5-2 moguće je izračunati amplitude harmonika pravokutnog valnog oblika. Znajući vršnu vrijednost napona moguće je doći do amplitude harmonika. Vršna vrijednost napona na je 8.225 V. Postupak je prikazan za prvi harmonik:

$$U(1) = \frac{4 \cdot 8.225}{\pi} \cdot \frac{\sqrt{2}}{2} = 7.4 \text{ V} \quad (5-3)$$

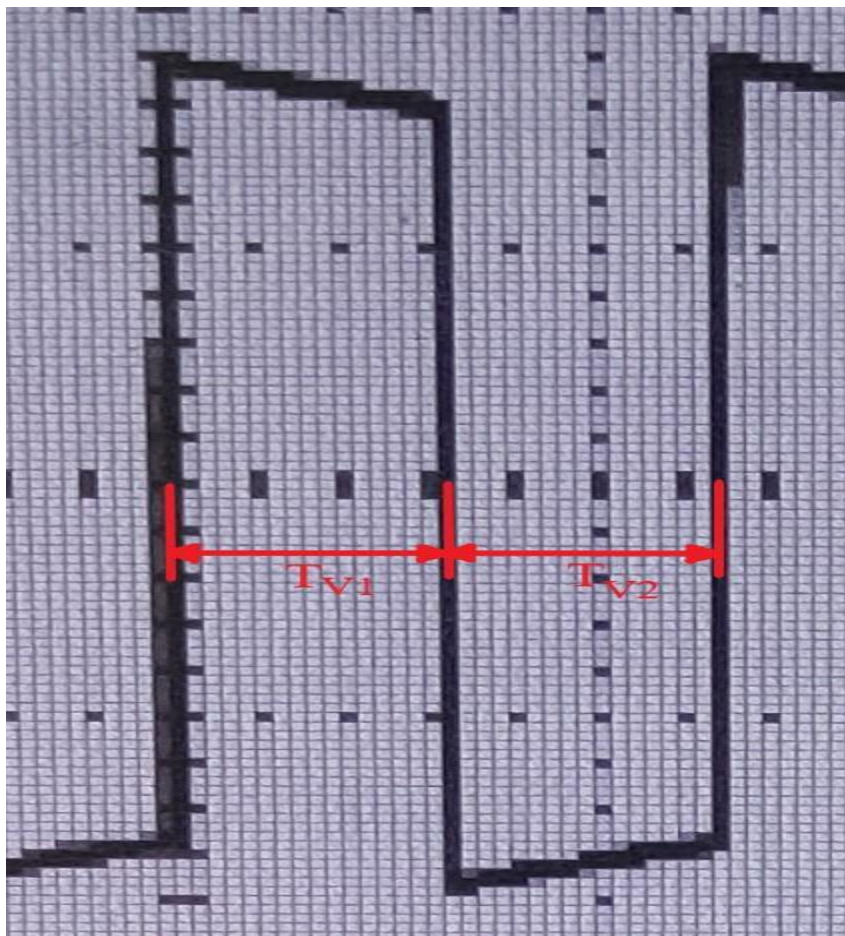
U tablici 5.1. prikazana je usporedba harmonika između analitičke metode i harmonika iz virtualnog spektralnog analizatora i postotak njihove razlike. Podatci za virtualni spektralni analizator su preuzeti iz tablice 5.2.

**Tablica 5.1.** Usporedba analitičke metoda s virtualnim spektralnim analizatorom

Harmonik	Analitički, $U[V]$	Virtualni spektralni analizator, $U[V]$	Postotak razlike [%]
1	7.4	7.3	1.35
3	2.47	2.44	1.2
5	1.48	1.46	1.35
7	1.06	1.04	1.88

Iz tablice je vidljivo da je razlika između analitičkog izračuna i izračuna virtualnog spektralnog analizatora prihvatljiva, odnosno manja je od 2%.

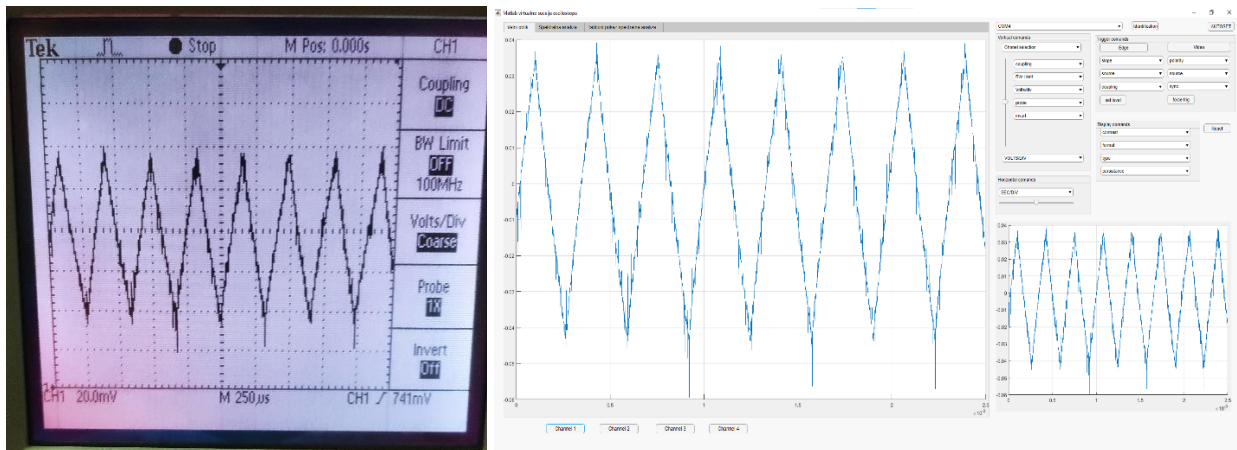
Na slici 5.6. moguće je vidjeti vrijeme vođenja ventila  $V1(T_{V1})$  i vrijeme vođenja ventila  $V2(T_{V2})$ .



Slika 5.6. Vrijeme vođenja ventila

Iz slike 5.7. moguće je zaključiti da su vremena vođenja ventila  $T_{V1}$  i  $T_{V2}$  jednaka. Odnosno da je faktor vođenja  $\alpha = 0.5$ .

Nakon valnog oblika ulaznog napona i njegove obrade, potrebno je obraditi valni oblik struje zavojnice koji je vidljiv na slici 5.7. Na slici 5.8. prikazana je spektralna analiza signala s slike 5.7. u obliku stupčastog dijagrama, a na slici 5.9. spektralna analiza prikazana je u obliku tablice.



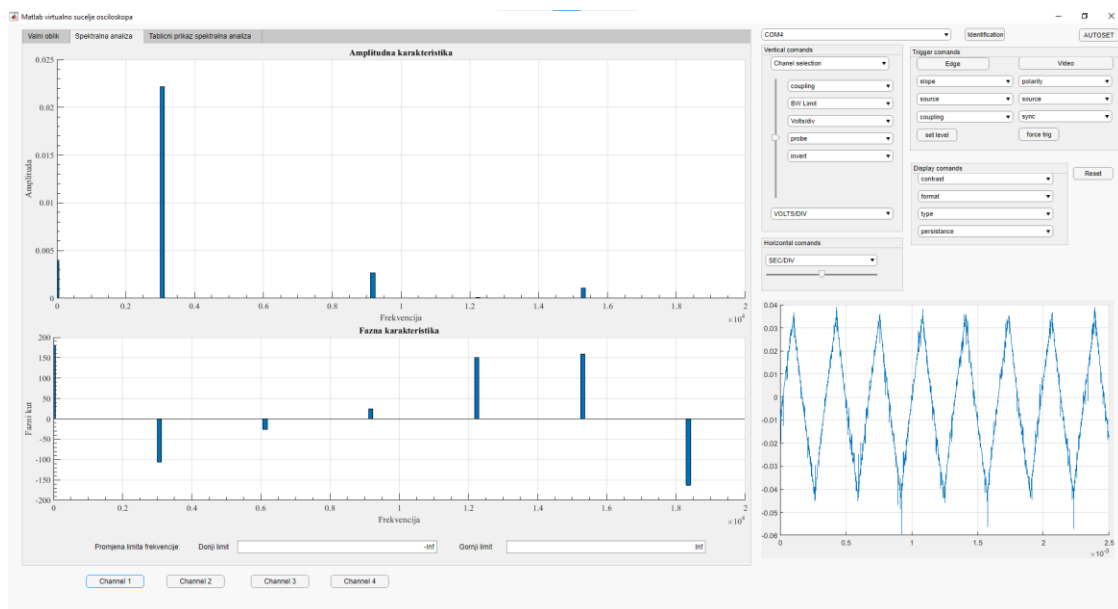
a)

b)

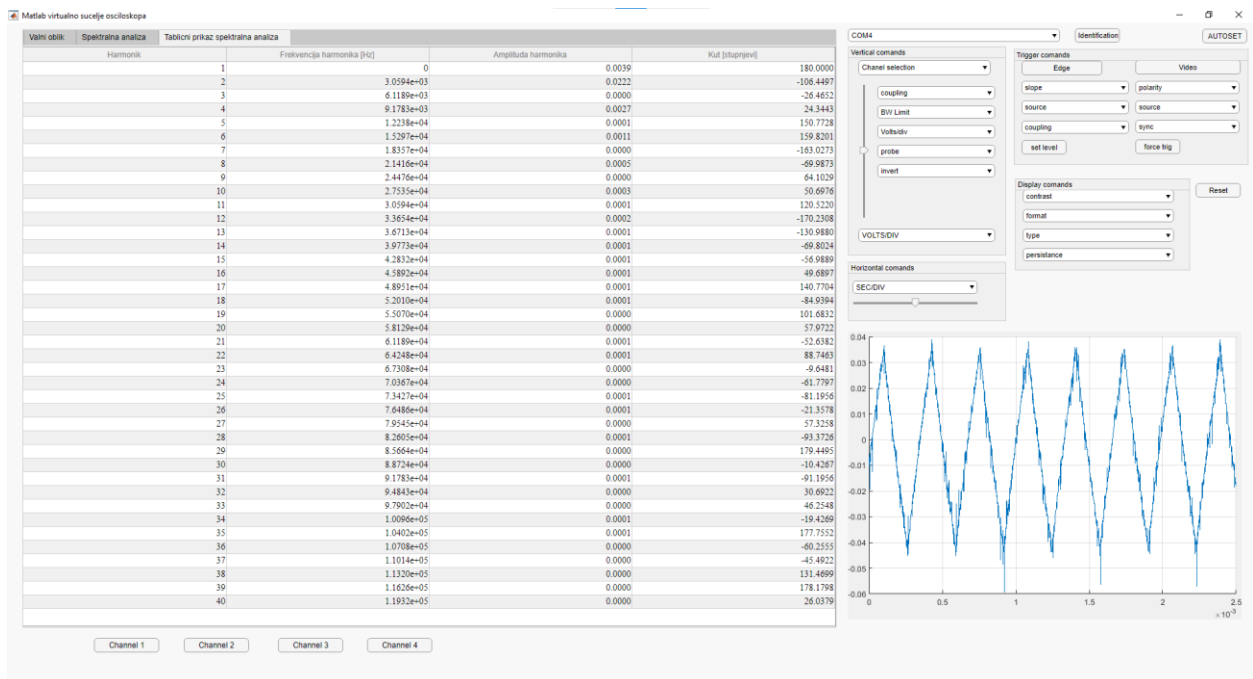
Slika 5.7. Snimljeni valni oblik struje zavojnice;  $E = 16.45 \text{ V}$

a) Valni oblik snimljen na osciloskopu

b) Valni oblik snimljen pomoću sučelja virtualnog spektralnog analizatora



Slika 5.8. Spektralna analiza valnog oblika struje prikazana stupčastom dijagramu



Slika 5.9. Spektralna analiza valnog oblika prikazana tablično

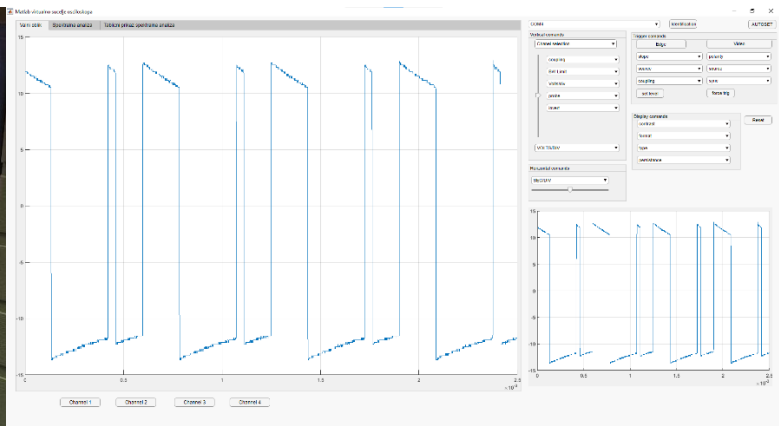
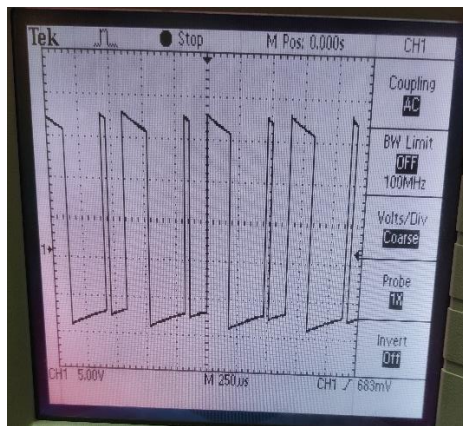
U tablici 5.3. vidljivi u rezultati spektralne analize za srednju vrijednost struje zavojnice.

**Tablica 5.3.** Rezultati spektralne analize

Harmonik	Frekvencija harmonika [Hz]	Amplituda, I[A]	Postotak osnovnog harmonika [%]
1	3059.4	0.0222	100
3	9178.3	0.0027	12
5	15297	0.0011	5
7	21416	0.0005	2
9	27535	0.0003	1
11	33654	0.0002	1

Nakon osnovnog periodičkog rješenja testirati će se rad virtualnog spektralnog analizatora na promjenu frekvencije rada. Sljedeće ustaljeno stanje koje će se analizirati je dvostruko periodičko rješenje. Dvostruko periodičko rješenje se kao i ostala periodička rješenje dobiva promjenom ulaznog napona. Promjena ustaljenih stanja znači i promjenu periode, odnosno frekvencije.

Na slici 5.10. prikazan je valni oblik napona zavojnice koji se mjerio pri ulaznom naponu 24.42 V i ulaznoj struji od 0.281 A. Na slici 5.11. prikazana je spektralna analiza navedenog signala u obliku stupčastog dijagrama, a na slici 5.12. u obliku tablice.



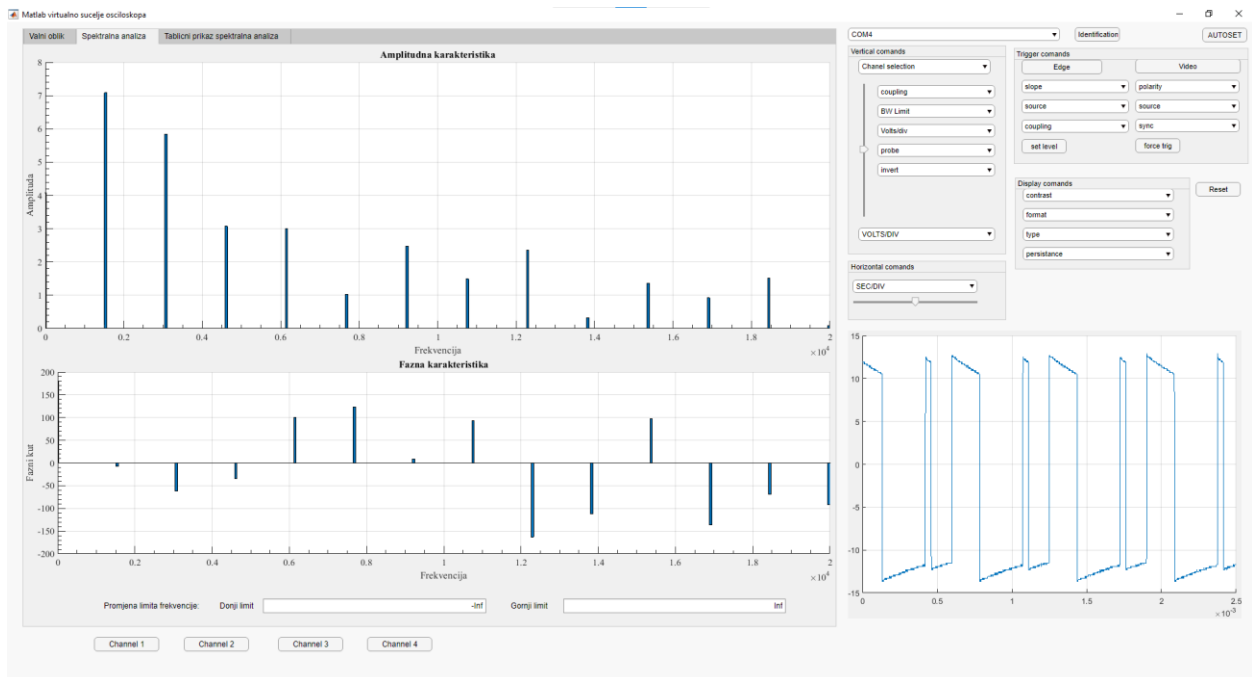
a)

b)

Slika 5.10. Snimljeni valni oblik napona dvostrukog periodičkog rješenja;  $E = 24.42 \text{ V}$

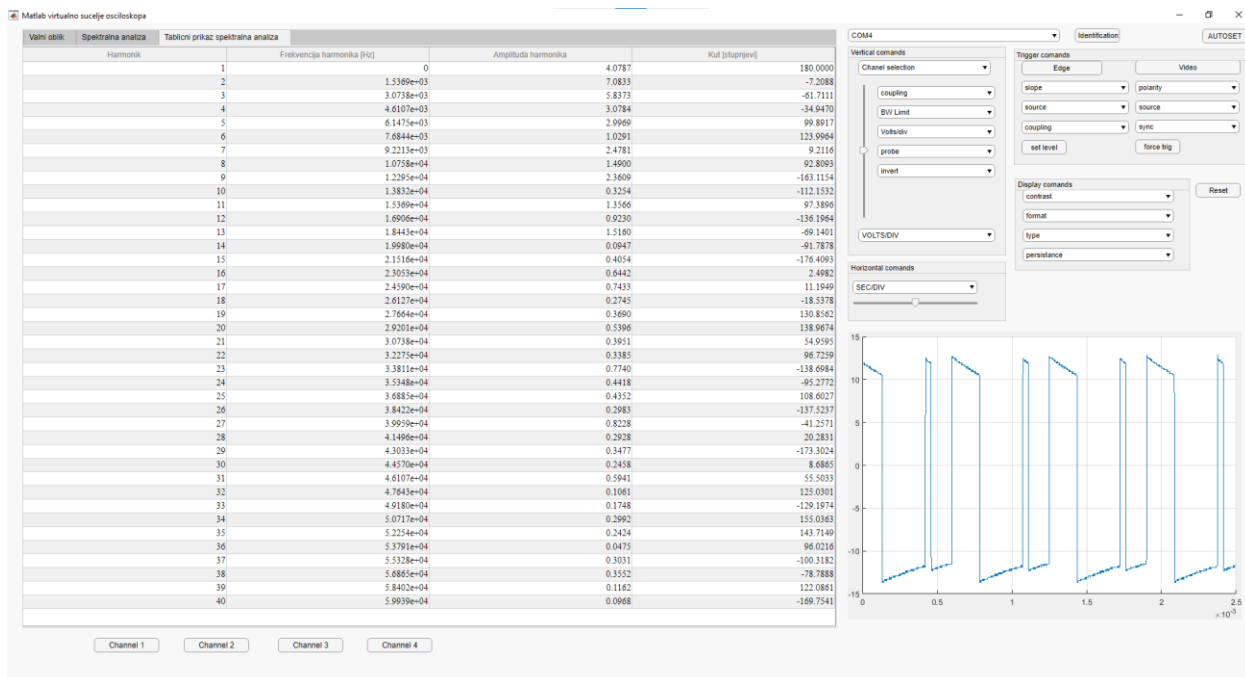
a) Valni oblik na osciloskopu

b) Valni oblik snimljen pomoću virtualnog spektralnog analizatora



Slika 5.11. Spektralna analiza valnog oblika napona prikazana stupčastim dijagramom





Slika 5.12. Spektralna analiza valnog oblika napona prikazana tablicom

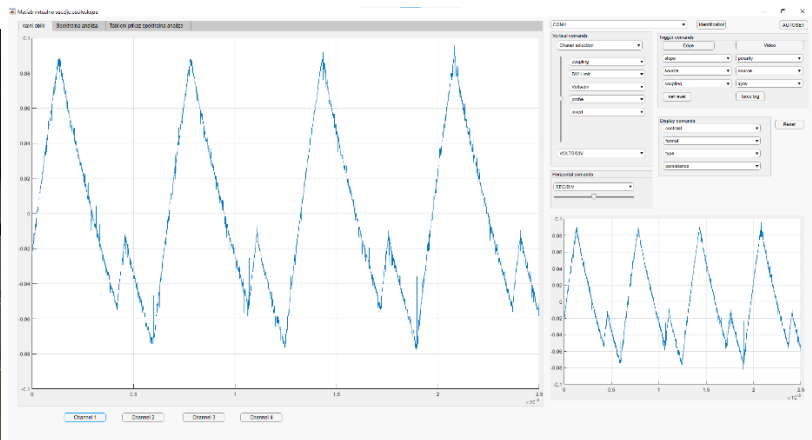
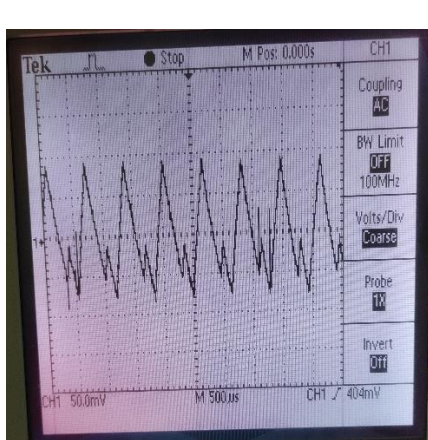
U tablici 5.4. prikazani su rezultati spektralne analize valnog oblika napona na zavojnici.

**Tablica 5.4.** Rezultati spektralne analize

Harmonik	Frekvencija harmonika [Hz]	Amplituda, I[A]	Postotak osnovnog harmonika [%]
0	0	4.0787	58
1	1537	7.0833	100
2	3074	5.8373	82
3	4610	3.0784	43
4	6148	2.9970	42
5	7684	1.0291	15
6	9221	2.4781	35

Iz tablice 5.4. je vidljivo da se pojavio polti harmonik koji je 1537 Hz, odnosno, pola frekvencije od osnovnog harmonika osnovnog periodičkog rješenja.

Analogno valnom obliku napona dvostrukog periodičkog rješenja na slici 5.13. nalazi se valni oblik struje dvostrukog periodičkog rješenja, dok se na slici 5.14. nalazi spektralna analiza prikazana stupčastim dijagramom i na slici 5.15. prikazana tablično.



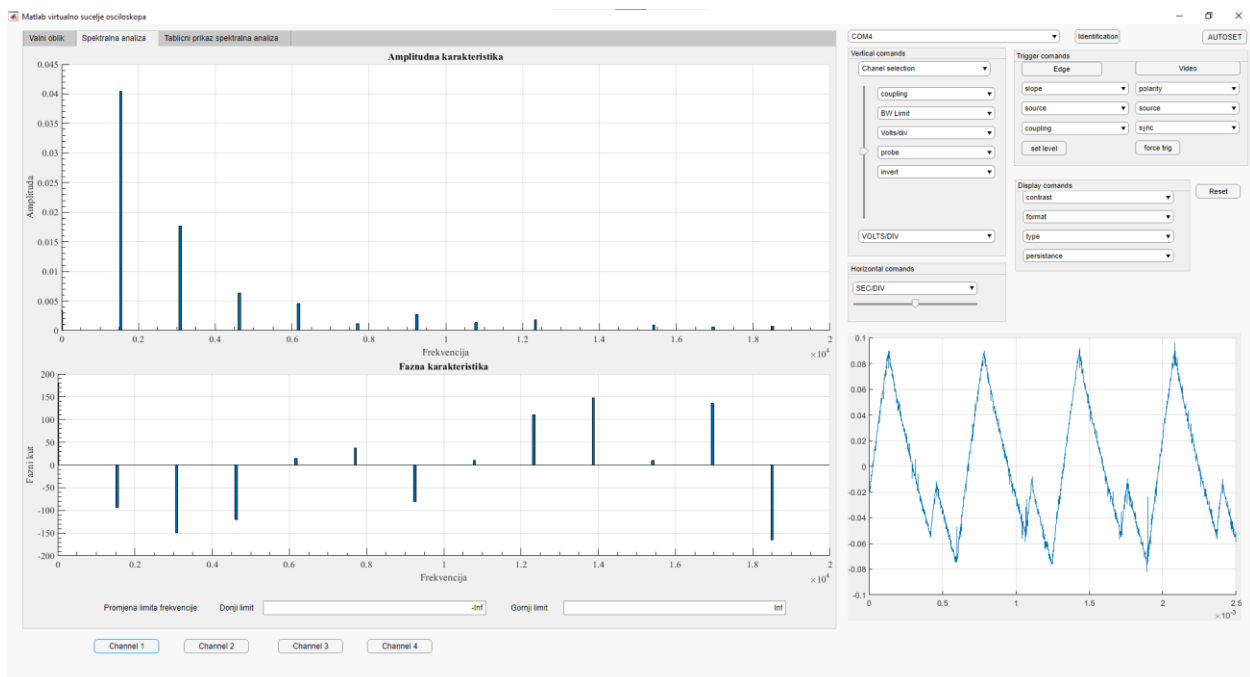
a)

b)

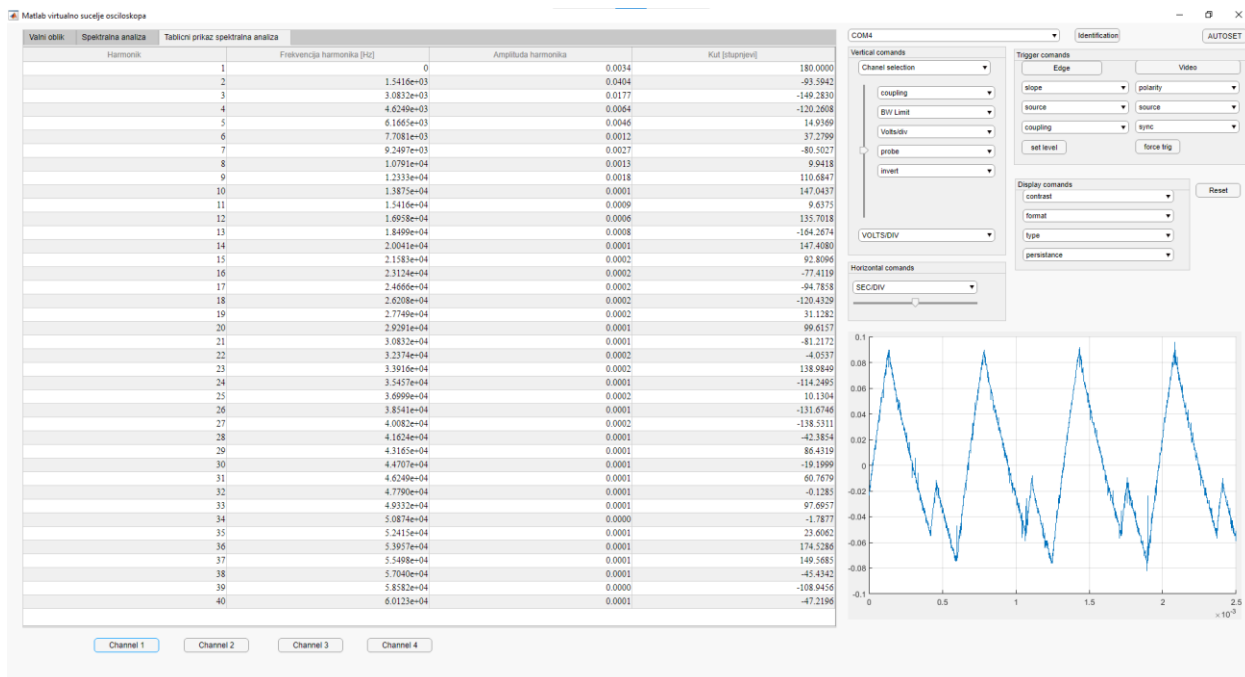
Slika 5.13. Snimljeni valni oblik struje dvostrukog periodičkog rješenja;  $E = 24.42 \text{ V}$

a) Valni oblik na osciloskopu

b) Valni oblik snimljen pomoću virtualnog spektralnog analizatora



Slika 5.14. Spektralna analiza valnog oblika struje prikazan stupčastim dijagramom



Slika 5.15. Spektralna analiza valnog oblika struje prikazana tablično

U tablici 5.5. prikazani su rezultat i spektralne analize srednje vrijednosti struje zavojnice.

**Tablica 5.5.** Rezultati spektralne analize

Harmonik	Frekvencija harmonika [Hz]	Amplituda, I[A]	Postotak osnovnog harmonika [%]
0	0	0.0034	8
1	1541	0.0404	100
2	3083	0.0177	43
3	4625	0.0064	16
4	6167	0.0046	11
5	7708	0.0012	3
6	9250	0.0027	7

Tablica 5.5. pokazuje da se osim poltog harmonika javljaju i parni i neparni harmonici. Nakon osnovnog harmonika najveći je drugi harmonik.

## 6. ZAKLJUČAK

Ovim radom prikazan je proces migriranja grafičkog sučelja iz GUIDE-a u App Designer unutar MATLAB grafičkog sučelja i dodavanje novih funkcija već postojećem sučelju unutar App Designer aplikacije. Funkcije su dodane kako bi se olakšalo očitavanje rezultata krajnjem korisniku. Isto tako omogućeno je upravljanje osciloskopa putem računala.

Detaljno je opisano migriranje aplikacije iz GUIDE-a u App Designer, proces dodavanja dodatnih funkcija kao što su: tablični prikaz spektralne analize, stupčasti dijagram spektralne analize, dodavanje limita osima i mogućnost kreiranja Excel datoteke s tabličnim rezultatima spektralne analize čija se potreba uvidjela pri testiranju. Sve nove funkcije su napisane unutar App Designera u MATLAB okruženju.

Testiranje rada virtualnog spektralnog analizatora provedeno je usporedbom pravokutnog valnog oblika napona i valnog oblika napona zavojnice silaznog pretvarača. Valni oblik napona na zavojnici je zbog usporedbe s pravokutnim valnim oblikom napona morao imati faktor vođenja  $\alpha = 0.5$ . Testiranjem valnog oblika napona zavojnice je pokazano da virtualni spektralni analizator radi spektralnu analizu s odstupanjem manjim od 2% u odnosu na analitički proračun. Prvi harmonik ima frekvenciju od 3 kHz. U dvostrukom periodičkom rješenju pojavio harmonik s frekvencijom od 1.6 kHz, što odgovara pretpostavljenoj frekvenciji rada.

Rezultat rada je potpuno funkcionalno virtualno grafičko sučelje s mogućnošću kontrole osciloskopa, pregleda valnog oblika unutar sučelja, pregleda spektralne analize navedenog valnog oblika i tablični prikaz harmonika navedenog valnog oblika. Jedna od mana je duže vrijeme čekanja kod učitavanja valnog oblika na grafičko sučelje iz razloga što je brzina prijenosa podataka mala, a svaki valni oblik ima puno točaka i stoga podataka za prenijeti.

Kao poboljšanja u budućim diplomskim radovima vezanima za ovu temu preporučam pokušavanje postizanje brže brzine prijenosa podataka povećanjem broja bauda unutar izbornika osciloskopa. Baud je mjerna jedinica za brzinu prijenosa podataka koja se koristi u komunikacijama.

## LITERATURA

- [1] F. Pitinac, Grafičko sučelje u MATLABu za osciloskop TDS 224, Osijek, 2020., diplomski rad
- [2] Službeni YouTube video o migraciji aplikacije:  
<https://www.youtube.com/watch?v=wOSSjgvG-es> (11.8.2021.)
- [3] Kako napraviti GUI u MATLAB-u koristeći App Designer:  
<https://www.youtube.com/watch?v=nb0jHVXKY2w&t=284s> (13.7.2021.)
- [4] M. Čengić, Harmonijska analiza snage pretvaračkih komponenata različitih pretvarača, Osijek, 2016., diplomski rad
- [5] I. Flegar, Elektronički energetski pretvarači, Zagreb 2010.
- [6] Osnovno o CH340, preuzeo s sparkfun-a:  
<https://cdn.sparkfun.com/datasheets/Dev/Arduino/Other/CH340DS1.PDF> (11.8.2021.)
- [7] TDS 224 manual:<https://download.tek.com/manual/071049301.pdf> (11.8.2021.)
- [8] Materijali o objektno orijentiranom programiranju u MATLABu:<https://www.mathworks.com/products/MATLAB/object-oriented-programming.html> (11.8.2021.)
- [9] Migriranje GUIDE aplikacije u App Designer aplikaciju:  
<https://www.mathworks.com/MATLABcentral/fileexchange/66087-guide-to-app-designer-migration-tool-for-MATLAB> (10.7.2021.)

## SAŽETAK

Ovim radom prikazan je proces migracije grafičkog sučelja virtualnog analizatora iz GUIDE-a u App Designer unutar MATLAB sučelja. Stečen je uvid u izradu dodatnih funkcija kao što je grafički prikaz spektralne analize u obliku tablice i stupčastog dijagrama. Izrađeni program daje krajnjem korisniku mogućnost kontrole osciloskopa, prikaz valnog oblika s osciloskopa, spektralne analize s mogućnošću prikazivanja rezultata unutar stupčastog dijagrama ili tablica. Rad programa je uspješno testiran na silaznom pretvaraču napona.

**Ključne riječi:** osciloskop, App Designer, MATLAB, virtualni spektralni analizator

Title: Harmonic analysis of a buck converter using a virtual spectrum analyzer

## ABSTRACT

This paper presents the process of migration of virtual graphical user interface from GUIDE to App Designer inside of MATLAB environment. An insight into the syntax of creating additional functions like graphical representation of spectral analysis in shape of table or bar graph was gained. The developed program enables the user to control oscilloscope, spectral analysis with the possibility of showing results in shape of bar graph or table. Program was successfully tested on a buck converter.

Keywords: oscilloscope, App Designer, MATLAB, virtual spectral analysis

## ŽIVOTOPIS

Rođen je 5. prosinca 1996. godine u Osijeku. U Osijeku završava 8. razred osnovne škole Svete Ane.

Nakon završene osnovne škole upisuje Elektrotehničku i prometnu školu Osijek, smjer elektrotehničar.

Nakon završetka srednje škole upisuje preddiplomski studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek, smjer elektrotehnika. Nakon završetka preddiplomskog studija upisuje diplomski studij, smjer Industrijska elektrotehnika na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek.

Antonio Firic

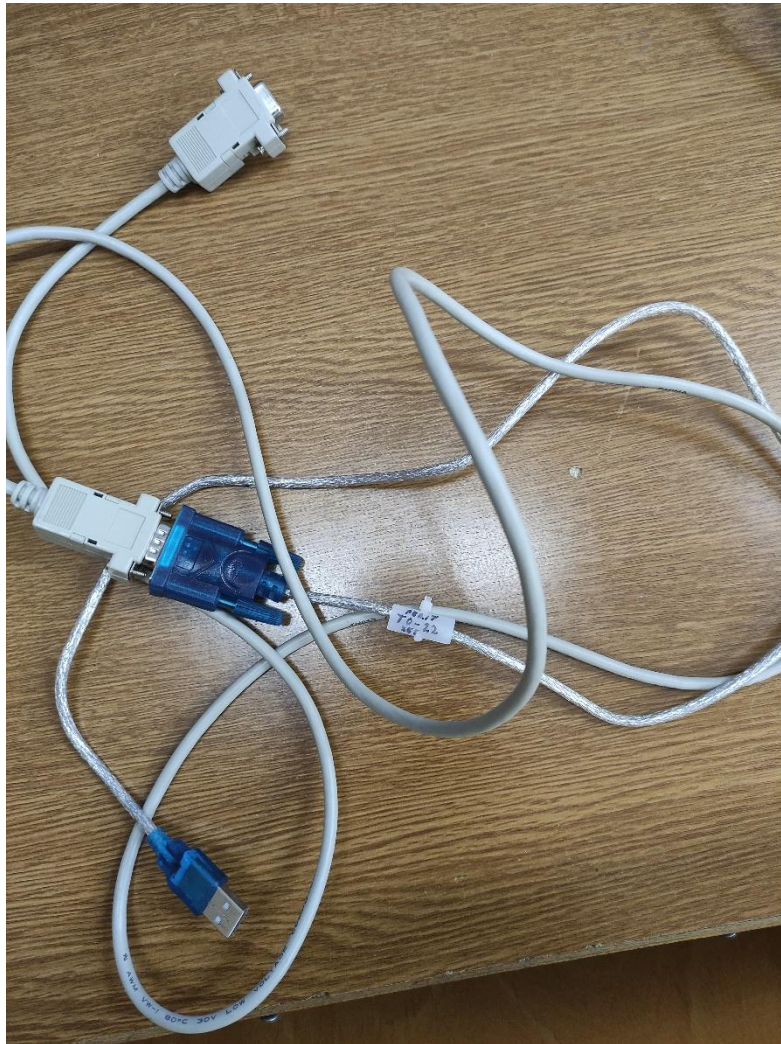
Potpis autora

## PRILOG

### PRILOG A: Upute za rukovanje virtualnim spektralnim analizatorom

#### Upute za virtualni spektralni analizator

1. Instalirati CH340 driver kako bi osciloskop mogao komunicirati s računalom
2. Spojiti RS232 priključak na osciloskop i USB na računalo. Kabel je prikazan na slici 1.

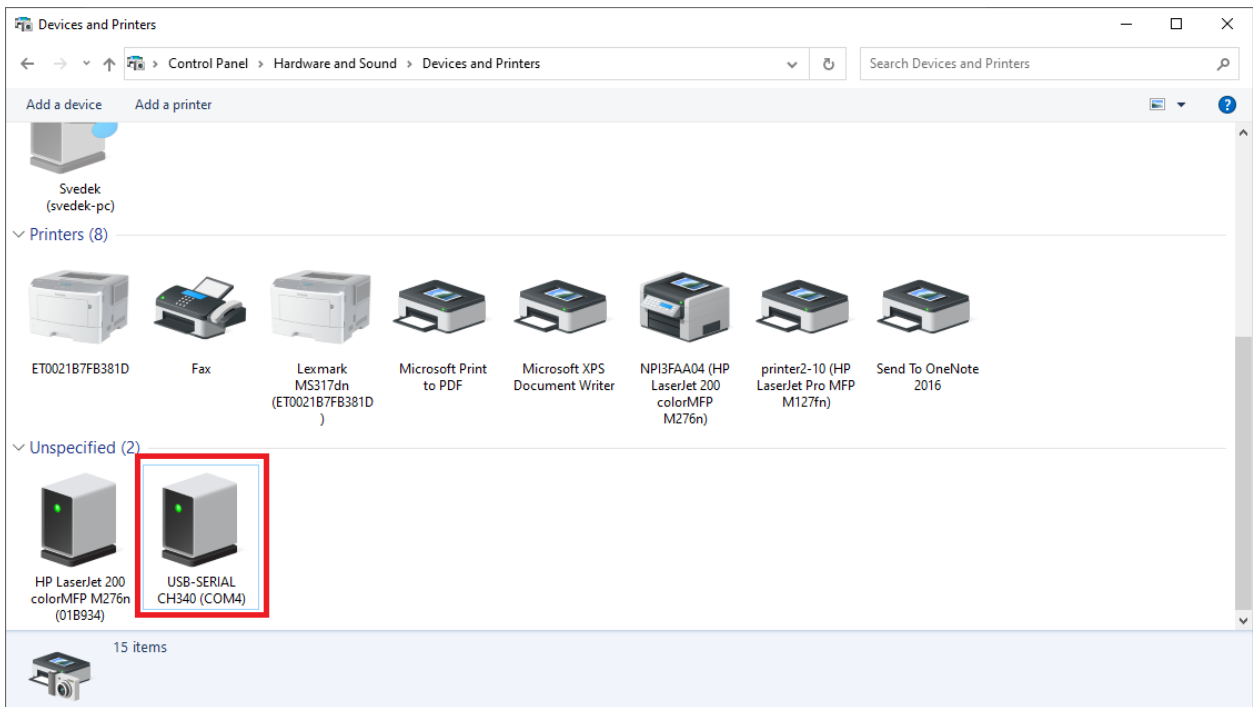


Slika 1.

3. Otvoriti MATLAB i pokrenuti aplikaciju Zadnja\_verzija\_1\_9 (ekstenzija je .mlapp). Potrebno je unutar foldera imati SnagaNaFrekv.m i OdrediPeriod\_v1.m MATLAB skripte kako bi sučelje pravilno radilo.

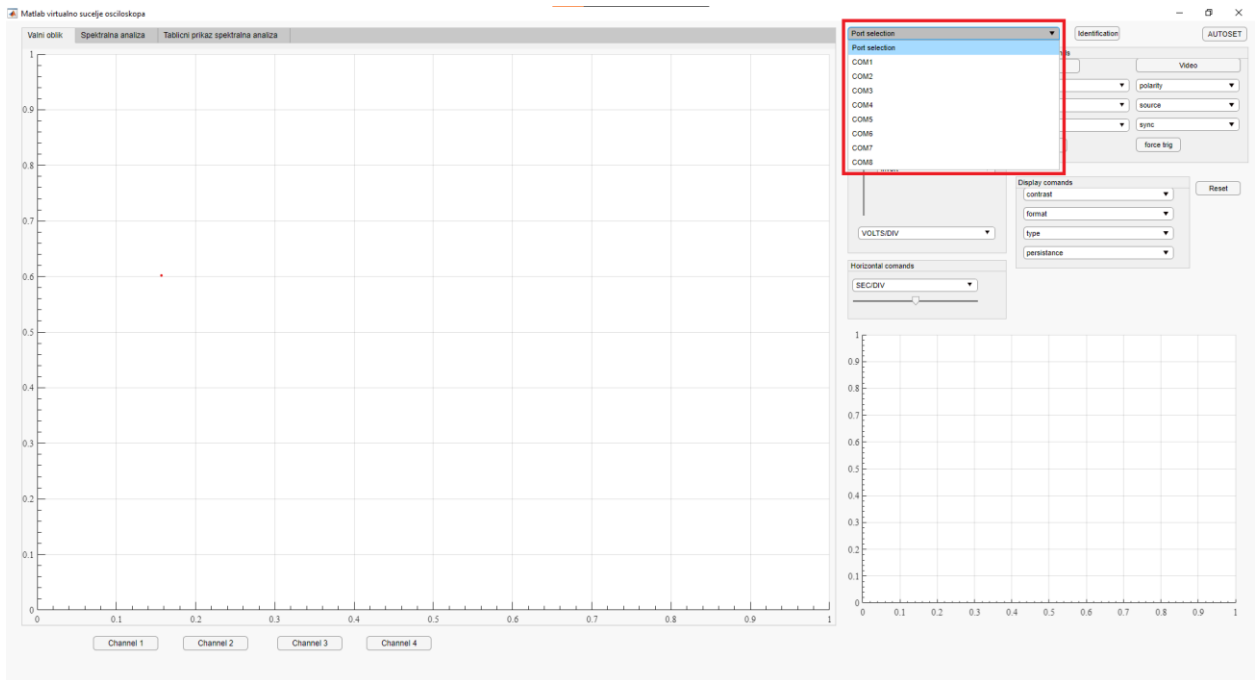


4. Za komunikaciju između računala i osciloskopa potrebno je odabrati pravilan priključak na računalu. priključak se nasumično zadaje kada priključimo RS232-USB kabel. Kako bi saznali koji je priključak zadan potrebno je otići na control panel -> device manager. Korak je prikazan na slici 2.



Slika 2.

Nakon što se sazna pravilan priključak potrebno ga je odabrati unutar sučelja virtualnog analizatora. Korak je prikazan na slici 3.



Slika 3.

Ako je odabran pravilan kanal pritiskom na tipku „Identification“ pojavit će se tekst kao na slici 4. unutar MATLAB Command Window-a.

```

Command Window
New to MATLAB? See resources for Getting Started.

Classroom License -- for classroom instructional use only.

>> appdesigner
Warning: MATLAB has disabled some advanced graphics rendering features by switching to software
here.

ans =

    'TEKTRONIX,TDS 224,0,CF:91.1CT FV:v1.06 TDS2CM:CMV:v1.04
    '
fx >> |

```

Slika 4.

5. Namjestiti sliku na osciloskopu tako da je na slici barem 4 periode. U suprotnom MATLAB skripta neće raditi.

6. Pritisnuti na gumb Channel 1 (Trenutno samo prvi kanal radi). Potrebno je sačekati neko vrijeme.
7. Ako je potrebno, podesiti postavke kao što su limiti na x-osi spektralnog analizatora. Moguće je i kontrolirati osciloskop pomoću aplikacije.

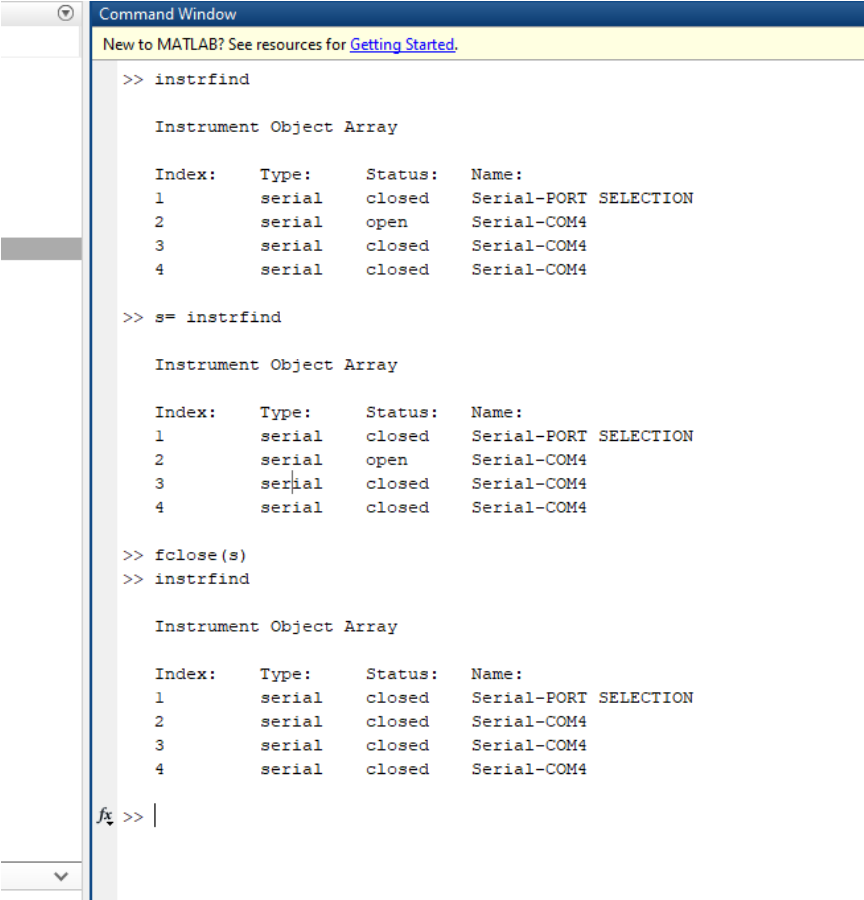
## Error-i

### Error zbog otvorenog kanal

Kako bi aplikacija funkcionirala kanal se u svakoj funkciji mora otvoriti i zatvoriti. Error nastane kad se dogodi greška u kodu i kod se ne izvrši do kraja, stoga se kanal ne zatvori. Kako bi ručno zatvorili kanal potrebno je upisati kod unutar Command Window-a koji je na slici 5. Odnosno kod je:

```
s = instrfind;
```

```
fclose(s)
```



```
Command Window
New to MATLAB? See resources for Getting Started.

>> instrfind

Instrument Object Array

Index:   Type:   Status:  Name:
1        serial  closed   Serial-POR SELECTION
2        serial  open     Serial-COM4
3        serial  closed   Serial-COM4
4        serial  closed   Serial-COM4

>> s = instrfind

Instrument Object Array

Index:   Type:   Status:  Name:
1        serial  closed   Serial-POR SELECTION
2        serial  open     Serial-COM4
3        serial  closed   Serial-COM4
4        serial  closed   Serial-COM4

>> fclose(s)
>> instrfind

Instrument Object Array

Index:   Type:   Status:  Name:
1        serial  closed   Serial-POR SELECTION
2        serial  closed   Serial-COM4
3        serial  closed   Serial-COM4
4        serial  closed   Serial-COM4

fx >> |
```

## Slika 5.

### PRILOG B: Kod unutar MATLAB-a

```
classdef Zadnja_verzija_1_9 < MATLAB.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        figure1 MATLAB.ui.Figure
        axes1 MATLAB.ui.control.UIAxes
        autose MATLAB.ui.control.Button
        uipanel1 MATLAB.ui.container.Panel
        slider1 MATLAB.ui.control.Slider
        popupmenu1 MATLAB.ui.control.DropDown
        coupling MATLAB.ui.control.DropDown
        bwlimit MATLAB.ui.control.DropDown
        voltsdiv MATLAB.ui.control.DropDown
        invert MATLAB.ui.control.DropDown
        probe MATLAB.ui.control.DropDown
        voldiv MATLAB.ui.control.DropDown
        popcom MATLAB.ui.control.DropDown
        uipanel3 MATLAB.ui.container.Panel
        edgetrig MATLAB.ui.control.Button
        videotrig MATLAB.ui.control.Button
        slopetrig MATLAB.ui.control.DropDown
        polaritytrig MATLAB.ui.control.DropDown
        sourceedge MATLAB.ui.control.DropDown
        couplingtrigg MATLAB.ui.control.DropDown
        sourcetrig MATLAB.ui.control.DropDown
        sync MATLAB.ui.control.DropDown
        trigset MATLAB.ui.control.Button
        forcetrig MATLAB.ui.control.Button
        identification MATLAB.ui.control.Button
        reset MATLAB.ui.control.Button
        uipanel4 MATLAB.ui.container.Panel
        contrast MATLAB.ui.control.DropDown
        format MATLAB.ui.control.DropDown
        disptype MATLAB.ui.control.DropDown
        persistence MATLAB.ui.control.DropDown
        uipanel5 MATLAB.ui.container.Panel
        popvrijeme MATLAB.ui.control.DropDown
        slider2 MATLAB.ui.control.Slider
        TabGroup MATLAB.ui.container.TabGroup
        ValnioblikTab MATLAB.ui.container.Tab
        axes1_2 MATLAB.ui.control.UIAxes
        SpektralnaanalizaTab MATLAB.ui.container.Tab
        UIAxes MATLAB.ui.control.UIAxes
        UIAxes2 MATLAB.ui.control.UIAxes
        GornjilimitEditFieldLabel MATLAB.ui.control.Label
        GornjilimitEditField MATLAB.ui.control.NumericEditField
        DonjilimitEditFieldLabel MATLAB.ui.control.Label
        DonjilimitEditField MATLAB.ui.control.NumericEditField
        PromjenalimitafrekvencijeLabel MATLAB.ui.control.Label
        Tablicniprikazspektralnaanaliza MATLAB.ui.container.Tab
        UITable MATLAB.ui.control.Table
        Channell1Button MATLAB.ui.control.Button
    end
end
```

```

Channel2Button          MATLAB.ui.control.Button
Channel3Button          MATLAB.ui.control.Button
Channel4Button          MATLAB.ui.control.Button
end

properties (Access = private)
    Donji_limit
    Gornji_limit % Description
end

% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function klizaci_OpeningFcn(app)

end

    % Button pushed function: autoset
    function autoset_Callback(app, event)
        s = serial(app.popcom.Value);
        fopen(s)
        fprintf(s, 'AUTOSSET EXECUTE');
        fclose(s)
        delete(s)
    end

    % Value changed function: bwlimit
    function bwlimit_Callback(app, event)
        s = serial(app.popcom.Value);
        fopen(s)
        fprintf(s, [app.popupmenu1.Value ':BANDWIDTH '
app.bwlimit.Value]);
        fclose(s)
        delete(s)
    end

    % Value changed function: contrast
    function contrast_Callback(app, event)
        ConBaza = [{'5%'} {'10%'} {'15%'} {'20%'} {'25%'} {'30%'} ...
{'35%'} {'40%'} {'45%'} {'50%'} {'55%'} {'60%'} {'65%'} ...
{'70%'} {'75%'} {'80%'} {'85%'} {'90%'} {'95%'} {'100%'}];

        ConBaza_1 = [{'5'} {'10'} {'15'} {'20'} {'25'} {'30'} ...
{'35'} {'40'} {'45'} {'50'} {'55'} {'60'} {'65'} ...
{'70'} {'75'} {'80'} {'85'} {'90'} {'95'} {'100'}];

        ConBaza_tablica = [ ConBaza ; ConBaza_1];
        s = serial(app.popcom.Value);
        set(s, 'InputBufferSize', 8*2500, 'Terminator', 'LF', 'Timeout', 30);
        fopen(s)
        for str_count = 1 : size(ConBaza_tablica , 2)
            if (strcmp(app.contrast.Value, ConBaza_tablica(1 ,
str_count)))
                fprintf(s, ['Display:CONTRast ' ConBaza_tablica{2 ,
str_count}]);
            end
        end
    end
end

```

```

        fclose(s)
        delete(s)
    end

    % Value changed function: coupling
    function coupling_Callback(app, event)
        s = serial(app.popcom.Value);
        fopen(s)
        fprintf(s, [app.popupmenu1.Value ':COUPLING '
app.coupling.Value]);
        fclose(s)
        delete(s)
    end

    % Value changed function: couplingtrigg
    function couplingtrigg_Callback(app, event)
        s = serial(app.popcom.Value);
        fopen(s)
        fprintf(s, ['TRIGger:MAIn:EDGE:COUPling '
app.couplingtrigg.Value]);
        fclose(s)
        delete(s)
    end

    % Value changed function: disptype
    function disptype_Callback(app, event)
        s = serial(app.popcom.Value);
        fopen(s)
        fprintf(s, ['DISplay:STYle VECTORS ' app.disptype.Value]);
        fclose(s)
        delete(s)
    end

    % Button pushed function: edgetrig
    function edgetrig_Callback(app, event)
        s = serial(app.popcom.Value);
        fopen(s)
        fprintf(s, 'TRIGger:MAIn:TYPe EDGE');
        fclose(s)
        delete(s)
    end

    % Button pushed function: forcetrig
    function forcetrig_Callback(app, event)
        s = serial(app.popcom.Value);
        fopen(s)
        fprintf(s, 'TRIGger force');
        fclose(s)
        delete(s)
    end

    % Value changed function: format
    function format_Callback(app, event)
        s = serial(app.popcom.Value);
        fopen(s)
        fprintf(s, ['DISplay:FORMat ' app.format.Value]);
        fclose(s)
        delete(s)
    end
end

```

```

% Button pushed function: identification
function identification_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, '*IDN?');
    fscanf(s)
    fclose(s)
    delete(s)
end

% Value changed function: invert
function invert_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, [app.popupmenu1.Value ':INVERT ' app.invert.Value]);
    fclose(s)
    delete(s)
end

% Value changed function: persistence
function persistence_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, ['DISplay:PERSist ' app.persistence.Value]);
    fclose(s)
    delete(s)
end

% Value changed function: polaritytrig
function polaritytrig_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, ['TRIGger:MAIn:VIDeo:POLarity '
app.polaritytrig.Value]);
    fclose(s)
    delete(s)
end

% Value changed function: popcom
function popcom_Callback(app, event)
end

% Value changed function: popupmenu1
function popupmenu1_Callback(app, event)
    value = app.popupmenu1.Value;
end

% Value changed function: popvrijeme
function popvrijeme_Callback(app, event)
    VremBaza = [{'5 ns'} {'10 ns'} {'25 ns'} {'50 ns'} {'100 ns'}
{'250 ns'} {'500 ns'} ...
{'1 us'} {'2.5 us'} {'5 us'} {'10 us'} {'25 us'} {'50 us'}
{'100 us'} {'250 us'} {'500 us'} ...
{'1 ms'} {'2.5 ms'} {'5 ms'} {'10 ms'} {'25 ms'} {'50 ms'}
{'100 ms'} {'250 ms'} {'500 ms'}...
{'1 s'} {'2.5 s'} {'5 s'}];
    VremBaza_1 = [{'5e-9'} {'10e-9'} {'25e-9'} {'50e-9'} {'100e-9'}
{'250e-9'} {'500e-9'} ...

```

```

        {'1e-6'} {'2.5e-6'} {'5e-6'} {'10e-6'} {'25e-6'} {'50e-6'}
{'100e-6'} {'250e-6'} {'500e-6'} ...
        {'1e-3'} {'2.5e-3'} {'5e-3'} {'10e-3'} {'25e-3'} {'50e-3'}
{'100e-3'} {'250e-3'} {'500e-3'}...
        {'1'} {'2.5'} {'5'}]];
VremBaza_tablica = [ VremBaza ; VremBaza_1];
s = serial(app.popcom.Value);
set(s, 'InputBufferSize', 8*2500, 'Terminator', 'LF', 'Timeout', 30);
fopen(s)
for str_count = 1 : size(VremBaza_tablica , 2)
    if (strcmp(app.popvrijeme.Value, VremBaza_tablica(1 ,
str_count)))
        fprintf(s, ['HORizontal:MAIn:SCALE ' VremBaza_tablica{2 ,
str_count}]);
    end
end
fclose(s)
delete(s)
end

% Value changed function: probe
function probe_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, [app.popupmenu1.Value ':PROBE ' app.probe.Value]);
    fclose(s)
    delete(s)
end

% Button pushed function: reset
function reset_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, '*RST');
    fclose(s)
    delete(s)
end

% Value changed function: slider1
function slider1_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, 'CH1?');
    fscanf(s)
    fprintf(s, [app.popupmenu1.Value ':POSition ' app.slider1.Value]);
    fclose(s)
    delete(s)
end

% Value changed function: slider2
function slider2_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fscanf(s)
    fprintf(s, ['HORizontal:POSition ' app.slider2.Value]);
    fclose(s)
    delete(s)
end

% Value changed function: slopetrig

```



```

function slopetrig_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, ['TRIGger:MAIn:EDGE:SLOpe ' app.slopetrig.Value]);
    fclose(s)
    delete(s)
end

% Value changed function: sourceedge
function sourceedge_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, ['TRIGger:MAIn:EDGE:SOURce ' app.sourceedge.Value]);
    fclose(s)
    delete(s)
end

% Value changed function: sourcetrig
function sourcetrig_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, ['TRIGger:MAIn:VIDEO:SOURce ' app.sourcetrig.Value]);
    fclose(s)
    delete(s)
end

% Value changed function: sync
function sync_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, ['TRIGger:MAIn:VIDEO:SYNC ' app.sync.Value]);
    fclose(s)
    delete(s)
end

% Button pushed function: trigset
function trigset_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, 'TRIGger:MAIn SETLevel');
    fclose(s)
    delete(s)
end

% Button pushed function: videotrig
function videotrig_Callback(app, event)
    s = serial(app.popcom.Value);
    fopen(s)
    fprintf(s, 'TRIGger:MAIn:TYPe VIDEo');
    fclose(s)
    delete(s)
end

% Value changed function: voldiv
function voldiv_Callback(app, event)
    NapBaza = [{'10 mV'} {'20 mV'} {'50 mV'} {'100 mV'} {'200 mV'}
{'500 mV'} ...
{'1 V'} {'2 V'} {'5 V'}];
    NapBaza_1 = [{'10e-3'} {'20e-3'} {'50e-3'} {'100e-3'} {'200e-3'}
{'500e-3'} ...

```

```

        {'1'} {'2'} {'5'}]];
NapBaza_tablica = [ NapBaza ; NapBaza_1];

s = serial(app.popcom.Value);
set(s, 'InputBufferSize', 8*2500, 'Terminator', 'LF', 'Timeout', 30);
fopen(s)
for str_count = 1 : size(NapBaza_tablica , 2)
    if (strcmp(app.popvrijeme.Value, NapBaza_tablica(1 ,
str_count)))
        fprintf(s, [app.popupmenu1.Value ':SCALE '
NapBaza_tablica{2 , str_count}]);
        end
    end
    fclose(s)
    delete(s)
end

% Value changed function: voltsdiv
function voltsdivValueChanged(app, event)
    value = app.voltsdiv.Value;

end

% Button pushed function: ChannellButton
function ChannellButtonPushed(app, event)
    s = serial(app.popcom.Value);
    set(s, 'InputBufferSize', 8*2500, 'Terminator', 'LF', 'Timeout', 30);
    fopen(s)

    fprintf(s, 'DATA:SOURce CH1');
    fprintf(s, 'DATA:encdg ascii');
    fprintf(s, 'DATA:width 2');
    fprintf(s, 'DATA:START 1');
    fprintf(s, 'DATA:STOP 2500');
    fprintf(s, 'WFMPRe?');
    form=fscanf(s);
    fprintf(s, 'CURVe?');

    formaY='';
    for i=1:2500
        formaY=[formaY '%d,'];
    end
    formaY(end)='';
    out=fscanf(s, formaY);

    fprintf(s, 'horizontal:scale?');
    divX=fscanf(s, '%f');
    t=[0:2499];
    t=t/250*divX;

    fprintf(s, 'CH1:scale?');
    divY=fscanf(s, '%f');
    a1=10*out*divY/(2^16-1);

%
    axes(handles.axes1);
    plot(app.axes1 , t, a1);
    grid(app.axes1, 'on');

    plot(app.axes1_2 , t, a1);
    grid(app.axes1_2, 'on');

```

```

        data_time_domain = [t.' a1];
        br_harm = 40;
        [fft_napon_out, fft_kut_out, frekvencije_x_os] =
SnagaNaFrekv(data_time_domain , br_harm);

        bar(app.UIAxes, frekvencije_x_os, fft_napon_out, 0.04); % 0.04 je
širina stupca
        grid(app.UIAxes, 'on');

        bar(app.UIAxes2, frekvencije_x_os, fft_kut_out, 0.04);
        grid(app.UIAxes2, 'on');

        a = 1:1:br_harm;
        T = table(a', frekvencije_x_os', fft_napon_out', fft_kut_out');
        writetable(T, 'Tablica spektralna.xls')
        app.UITable.Data = T;

        fclose(s)
        delete(s)

%         pl=evalin('base','p1');
%         axes(handles.axes1);
%         delete(p1);
end

% Button pushed function: Channel2Button
function Channel2ButtonPushed(app, event)
    s = serial(app.popcom.Value);
    set(s, 'InputBufferSize', 8*2500, 'Terminator', 'LF', 'Timeout', 30);
    fopen(s)

    fprintf(s, 'DATA:SOUrce CH2');
    fprintf(s, 'DATA:encdg ascii');
    fprintf(s, 'DATA:width 2');
    fprintf(s, 'DATA:START 1');
    fprintf(s, 'DATA:STOP 2500');
    fprintf(s, 'WFMPRe?');
    form=fscanf(s);
    fprintf(s, 'CURVe?');

    formaY='';
    for i=1:2500
        formaY=[formaY '%d,'];
    end
    formaY(end)='';
    out=fscanf(s, formaY);

    fprintf(s, 'horizontal:scale?');
    divX=fscanf(s, '%f');
    t=[0:2499];
    t=t/250*divX;

    fprintf(s, 'CH2:scale?');
    divY=fscanf(s, '%f');
    a1=10*out*divY/(2^16-1);

```

```

%         axes(handles.axes1);
plot(app.axes1 , t,a1);
grid(app.axes1, 'on');

plot(app.axes1_2 , t,a1);
grid(app.axes1_2, 'on');

data_time_domain = [t.' a1];
br_harm = 40;
[fft_napon_out, fft_kut_out, frekvencije_x_os] =
SnagaNaFrekv(data_time_domain , br_harm);

bar(app.UIAxes, frekvencije_x_os, fft_napon_out, 0.2); % 0.2 je
širina stupca
grid(app.UIAxes, 'on');

bar(app.UIAxes2, frekvencije_x_os, fft_kut_out, 0.2);
grid(app.UIAxes2, 'on');

T = table(br_harm', frekvencije_x_os', fft_napon_out');
app.UITable.DisplayData = T;

fclose(s)
delete(s)

%         p1=evalin('base','p1');
%         axes(handles.axes1);
%         delete(p1);
end

% Button pushed function: Channel3Button
function Channel3ButtonPushed(app, event)
    s = serial(app.popcom.Value);

    set(s, 'InputBufferSize',8*2500, 'Terminator', 'LF', 'Timeout',30);

    fopen(s)
    fprintf(s, '*IDN?');
    fscanf(s)

    fprintf(s, 'DATA:SOURce CH3');
    fprintf(s, 'DATA:encdg ascii');
    fprintf(s, 'DATA:width 2');
    fprintf(s, 'DATA:STArT 1');
    fprintf(s, 'DATA:STOP 2500');
    fprintf(s, 'WFMPRe?');
    form=fscanf(s);
    fprintf(s, 'CURVe?');
    formaY='';
    for i=1:2500
        formaY=[formaY '%d,'];
    end
    formaY(end)='';
    out=fscanf(s, formaY);

    fprintf(s, 'horizontal:scale?');
    divX=fscanf(s, '%f');
    t=[0:2499];
    t=t/250*divX;

```

```

fprintf(s, ['CH3:scale?']);
divY=fscanf(s, '%f');
a1=10*out*divY/(2^16-1)
axes(handles.axes1);
p1=plot(t,a1);
grid on;
hold on;
assignin('base','p3',p3)

fclose(s)
delete(s)

p3=evalin('base','p3');
axes(handles.axes1);
delete(p3);
end

% Button pushed function: Channel4Button
function Channel4ButtonPushed(app, event)
    s = serial(app.popcom.Value);

    set(s, 'InputBufferSize',8*2500, 'Terminator', 'LF', 'Timeout',30);

    fopen(s)
    fprintf(s, '*IDN?');
    fscanf(s)

    fprintf(s, 'DATa:SOUrce CH4');
    fprintf(s, 'DATa:encdg ascii');
    fprintf(s, 'DATa:width 2');
    fprintf(s, 'DATa:START 1');
    fprintf(s, 'DATa:STOP 2500');
    fprintf(s, 'WFMPRe?');
    form=fscanf(s);
    fprintf(s, 'CURVe?');
    formAY='';
    for i=1:2500
        formAY=[formAY '%d,'];
    end
    formAY(end)='';
    out=fscanf(s, formAY);

    fprintf(s, 'horizontal:scale?');
    divX=fscanf(s, '%f');
    t=[0:2499];
    t=t/250*divX;

    fprintf(s, 'CH4:scale?');
    divY=fscanf(s, '%f');
    a1=10*out*divY/(2^16-1)
    axes(handles.axes1);
    p4=plot(t,a1);
    grid on;
    hold on;
    assignin('base','p4',p4)

    fclose(s)
    delete(s)

```

```

        p4=evalin('base','p4');
        axes(handles.axes1);
        delete(p4);
    end

    % Value changed function: DonjilimitEditField
    function DonjilimitEditFieldValueChanged(app, event)
        value = app.DonjilimitEditField.Value;
        app.Donji_limit = value;
        app.UIAxes.XLim = [app.Donji_limit app.Gornji_limit];
        app.UIAxes2.XLim = [app.Donji_limit app.Gornji_limit];
    end

    % Value changed function: GornjilimitEditField
    function GornjilimitEditFieldValueChanged(app, event)
        value = app.GornjilimitEditField.Value;
        app.Gornji_limit = value;
        app.UIAxes.XLim = [app.Donji_limit app.Gornji_limit];
        app.UIAxes2.XLim = [app.Donji_limit app.Gornji_limit];
    end
end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

        % Create figure1 and hide until all components are created
        app.figure1 = uifigure('Visible', 'off');
        app.figure1.Color = [0.972549019607843 0.972549019607843
0.972549019607843];
        app.figure1.Position = [0 0 1200 800];
        app.figure1.Name = 'MATLAB virtualno sucelje osciloskopa';
        app.figure1.Scrollable = 'on';

        % Create axes1
        app.axes1 = uiaxes(app.figure1);
        app.axes1.XMinorTick = 'on';
        app.axes1.YMinorTick = 'on';
        app.axes1.NextPlot = 'replace';
        app.axes1.XGrid = 'on';
        app.axes1.YGrid = 'on';
        app.axes1.BackgroundColor = [0.972549019607843 0.972549019607843
0.972549019607843];
        app.axes1.Position = [810 102 370 294];

        % Create autoset
        app.autoset = uibutton(app.figure1, 'push');
        app.autoset.ButtonPushedFcn = createCallbackFcn(app,
@autoset_Callback, true);
        app.autoset.FontSize = 11;
        app.autoset.Position = [1120 770 69 22];
        app.autoset.Text = 'AUTOSSET';

        % Create uipanel1
        app.uipanel1 = uipanel(app.figure1);
        app.uipanel1.Title = 'Vertical comands';
        app.uipanel1.FontSize = 11;
        app.uipanel1.Position = [810 513 151 252];

```

```

% Create slider1
app.slider1 = uislider(app.uipanel1);
app.slider1.Limits = [-4 4];
app.slider1.MajorTicks = [];
app.slider1.Orientation = 'vertical';
app.slider1.ValueChangedFcn = createCallbackFcn(app,
@slider1_Callback, true);
app.slider1.MinorTicks = [];
app.slider1.FontSize = 11;
app.slider1.Position = [23 58 3 136];

% Create popupmenu1
app.popupmenu1 = uidropdown(app.uipanel1);
app.popupmenu1.Items = {'Chanel selection', 'CH1', 'CH2', 'CH3',
'CH4'};
app.popupmenu1.ValueChangedFcn = createCallbackFcn(app,
@popupmenu1_Callback, true);
app.popupmenu1.FontSize = 11;
app.popupmenu1.BackgroundColor = [1 1 1];
app.popupmenu1.Position = [16 208 111 23];
app.popupmenu1.Value = 'Chanel selection';

% Create coupling
app.coupling = uidropdown(app.uipanel1);
app.coupling.Items = {'coupling', 'DC', 'AC', 'GND'};
app.coupling.ValueChangedFcn = createCallbackFcn(app,
@coupling_Callback, true);
app.coupling.FontSize = 11;
app.coupling.BackgroundColor = [1 1 1];
app.coupling.Position = [45 171 89 20];
app.coupling.Value = 'coupling';

% Create bwlimit
app.bwlimit = uidropdown(app.uipanel1);
app.bwlimit.Items = {'BW Limit', 'ON', 'OFF'};
app.bwlimit.ValueChangedFcn = createCallbackFcn(app,
@bwlimit_Callback, true);
app.bwlimit.FontSize = 11;
app.bwlimit.BackgroundColor = [1 1 1];
app.bwlimit.Position = [45 141 89 20];
app.bwlimit.Value = 'BW Limit';

% Create voltsdiv
app.voltsdiv = uidropdown(app.uipanel1);
app.voltsdiv.Items = {'Volts/div '};
app.voltsdiv.ValueChangedFcn = createCallbackFcn(app,
@voltsdivValueChanged, true);
app.voltsdiv.FontSize = 11;
app.voltsdiv.BackgroundColor = [1 1 1];
app.voltsdiv.Position = [45 111 89 20];
app.voltsdiv.Value = 'Volts/div ';

% Create invert
app.invert = uidropdown(app.uipanel1);
app.invert.Items = {'invert', 'ON', 'OFF'};
app.invert.ValueChangedFcn = createCallbackFcn(app,
@invert_Callback, true);
app.invert.FontSize = 11;
app.invert.BackgroundColor = [1 1 1];

```

```

app.invert.Position = [45 51 89 20];
app.invert.Value = 'invert';

% Create probe
app.probe = uidropdown(app.uipanel1);
app.probe.Items = {'probe', '1', '10', '100', '1000'};
app.probe.ValueChangedFcn = createCallbackFcn(app,
@probe_Callback, true);
app.probe.FontSize = 11;
app.probe.BackgroundColor = [1 1 1];
app.probe.Position = [45 81 89 20];
app.probe.Value = 'probe';

% Create voldiv
app.voldiv = uidropdown(app.uipanel1);
app.voldiv.Items = {'VOLTS/DIV', '2 mV', '5 mV', '10 mV', '20
mV', '50 mV', '100 mV', '200 mV', '500 mV', '1 V', '2 V', '5 V'};
app.voldiv.ValueChangedFcn = createCallbackFcn(app,
@voldiv_Callback, true);
app.voldiv.FontSize = 11;
app.voldiv.BackgroundColor = [1 1 1];
app.voldiv.Position = [16 21 118 20];
app.voldiv.Value = 'VOLTS/DIV';

% Create popcom
app.popcom = uidropdown(app.figure1);
app.popcom.Items = {'Port selection', 'COM1', 'COM2', 'COM3',
'COM4', 'COM5', 'COM6', 'COM7', 'COM8'};
app.popcom.ValueChangedFcn = createCallbackFcn(app,
@popcom_Callback, true);
app.popcom.FontSize = 11;
app.popcom.BackgroundColor = [1 1 1];
app.popcom.Position = [810 772 91 20];
app.popcom.Value = 'Port selection';

% Create uipanel3
app.uipanel3 = uipanel(app.figure1);
app.uipanel3.Title = 'Trigger comands';
app.uipanel3.FontSize = 11;
app.uipanel3.Position = [973 583 218 178];

% Create edgetrig
app.edgetrig = uibutton(app.uipanel3, 'push');
app.edgetrig.ButtonPushedFcn = createCallbackFcn(app,
@edgetrig_Callback, true);
app.edgetrig.FontSize = 11;
app.edgetrig.Position = [12 138 89 22];
app.edgetrig.Text = 'Edge';

% Create videotrig
app.videotrig = uibutton(app.uipanel3, 'push');
app.videotrig.ButtonPushedFcn = createCallbackFcn(app,
@videotrig_Callback, true);
app.videotrig.FontSize = 11;
app.videotrig.Position = [115 139 91 22];
app.videotrig.Text = 'Video';

% Create slopetrig
app.slopetrig = uidropdown(app.uipanel3);
app.slopetrig.Items = {'slope', 'rise', 'fall'};

```



```

        app.slopetrig.ValueChangedFcn = createCallbackFcn(app,
@slopetrig_Callback, true);
        app.slopetrig.FontSize = 11;
        app.slopetrig.BackgroundColor = [1 1 1];
        app.slopetrig.Position = [12 109 93 20];
        app.slopetrig.Value = 'slope';

        % Create polaritytrig
        app.polaritytrig = uideropdown(app.uipanel3);
        app.polaritytrig.Items = {'polarity', 'NORMAL', 'INVERT'};
        app.polaritytrig.ValueChangedFcn = createCallbackFcn(app,
@polaritytrig_Callback, true);
        app.polaritytrig.FontSize = 11;
        app.polaritytrig.BackgroundColor = [1 1 1];
        app.polaritytrig.Position = [115 109 89 20];
        app.polaritytrig.Value = 'polarity';

        % Create sourceedge
        app.sourceedge = uideropdown(app.uipanel3);
        app.sourceedge.Items = {'source', 'CH1', 'CH2', 'CH3', 'CH4',
'LINE'};
        app.sourceedge.ValueChangedFcn = createCallbackFcn(app,
@sourceedge_Callback, true);
        app.sourceedge.FontSize = 11;
        app.sourceedge.BackgroundColor = [1 1 1];
        app.sourceedge.Position = [12 78 93 21];
        app.sourceedge.Value = 'source';

        % Create couplingtrigg
        app.couplingtrigg = uideropdown(app.uipanel3);
        app.couplingtrigg.Items = {'coupling', 'AC', 'DC', 'HFRej',
'NOISErej'};
        app.couplingtrigg.ValueChangedFcn = createCallbackFcn(app,
@couplingtrigg_Callback, true);
        app.couplingtrigg.FontSize = 11;
        app.couplingtrigg.BackgroundColor = [1 1 1];
        app.couplingtrigg.Position = [12 48 93 20];
        app.couplingtrigg.Value = 'coupling';

        % Create sourcetrig
        app.sourcetrig = uideropdown(app.uipanel3);
        app.sourcetrig.Items = {'source', 'CH1', 'CH2', 'CH3', 'CH4'};
        app.sourcetrig.ValueChangedFcn = createCallbackFcn(app,
@sourcetrig_Callback, true);
        app.sourcetrig.FontSize = 11;
        app.sourcetrig.BackgroundColor = [1 1 1];
        app.sourcetrig.Position = [115 79 89 20];
        app.sourcetrig.Value = 'source';

        % Create sync
        app.sync = uideropdown(app.uipanel3);
        app.sync.Items = {'sync', 'LINE', 'FIELD'};
        app.sync.ValueChangedFcn = createCallbackFcn(app, @sync_Callback,
true);
        app.sync.FontSize = 11;
        app.sync.BackgroundColor = [1 1 1];
        app.sync.Position = [115 49 89 20];
        app.sync.Value = 'sync';

        % Create trigset

```

```

        app.trigset = uibutton(app.uipanel3, 'push');
        app.trigset.ButtonPushedFcn = createCallbackFcn(app,
@trigset_Callback, true);
        app.trigset.FontSize = 11;
        app.trigset.Position = [12 16 69 22];
        app.trigset.Text = 'set level';

        % Create forcetrig
        app.forcetrig = uibutton(app.uipanel3, 'push');
        app.forcetrig.ButtonPushedFcn = createCallbackFcn(app,
@forcetrig_Callback, true);
        app.forcetrig.FontSize = 11;
        app.forcetrig.Position = [115 17 69 22];
        app.forcetrig.Text = 'force trig';

        % Create identification
        app.identification = uibutton(app.figure1, 'push');
        app.identification.ButtonPushedFcn = createCallbackFcn(app,
@identification_Callback, true);
        app.identification.FontSize = 11;
        app.identification.Position = [924 771 69 22];
        app.identification.Text = 'Identification';

        % Create reset
        app.reset = uibutton(app.figure1, 'push');
        app.reset.ButtonPushedFcn = createCallbackFcn(app,
@reset_Callback, true);
        app.reset.FontSize = 11;
        app.reset.Position = [1110 533 69 22];
        app.reset.Text = 'Reset';

        % Create uipanel4
        app.uipanel4 = uipanel(app.figure1);
        app.uipanel4.Title = 'Display comands';
        app.uipanel4.FontSize = 11;
        app.uipanel4.Position = [975 421 126 141];

        % Create contrast
        app.contrast = uidropdown(app.uipanel4);
        app.contrast.Items = {'contrast', '5%', '10%', '15%', '20%',
'25%', '30%', '35%', '40%', '45%', '50%', '55%', '60%', '65%', '70%', '75%',
'80%', '85%', '90%', '95%', '100%'};
        app.contrast.ValueChangedFcn = createCallbackFcn(app,
@contrast_Callback, true);
        app.contrast.FontSize = 11;
        app.contrast.BackgroundColor = [1 1 1];
        app.contrast.Position = [12 104 89 20];
        app.contrast.Value = 'contrast';

        % Create format
        app.format = uidropdown(app.uipanel4);
        app.format.Items = {'format', 'YT', 'XY'};
        app.format.ValueChangedFcn = createCallbackFcn(app,
@format_Callback, true);
        app.format.FontSize = 11;
        app.format.BackgroundColor = [1 1 1];
        app.format.Position = [12 74 89 20];
        app.format.Value = 'format';

        % Create disptype

```

```

    app.disptype = uidropdown(app.uipanel4);
    app.disptype.Items = {'type', 'VECTORS', 'DOTS'};
    app.disptype.ValueChangedFcn = createCallbackFcn(app,
@disptype_Callback, true);
    app.disptype.FontSize = 11;
    app.disptype.BackgroundColor = [1 1 1];
    app.disptype.Position = [12 44 89 20];
    app.disptype.Value = 'type';

    % Create persistence
    app.persistence = uidropdown(app.uipanel4);
    app.persistence.Items = {'persistence', 'OFF', '1', '2', '5',
'INFINITE', ''};
    app.persistence.ValueChangedFcn = createCallbackFcn(app,
@persistence_Callback, true);
    app.persistence.FontSize = 11;
    app.persistence.BackgroundColor = [1 1 1];
    app.persistence.Position = [12 14 89 20];
    app.persistence.Value = 'persistence';

    % Create uipanel5
    app.uipanel5 = uipanel(app.figure1);
    app.uipanel5.Title = 'Horizontal comands';
    app.uipanel5.FontSize = 11;
    app.uipanel5.Position = [810 412 151 91];

    % Create popvrijeme
    app.popvrijeme = uidropdown(app.uipanel5);
    app.popvrijeme.Items = {'SEC/DIV', '5 ns', '10 ns', '25 ns', '50
ns', '100 ns', '250 ns', '500 ns', '1 us', '2.5 us', '5 us', '10 us', '25
us', '50 us', '100 us', '250 us', '500 us', '1 ms', '2.5 ms', '5 ms', '10
ms', '25 ms', '50 ms', '100 ms', '250 ms', '500 ms', '1 s', '2.5 s', '5 s'};
    app.popvrijeme.ValueChangedFcn = createCallbackFcn(app,
@popvrijeme_Callback, true);
    app.popvrijeme.FontSize = 11;
    app.popvrijeme.BackgroundColor = [1 1 1];
    app.popvrijeme.Position = [7 42 100 20];
    app.popvrijeme.Value = 'SEC/DIV';

    % Create slider2
    app.slider2 = uislider(app.uipanel5);
    app.slider2.Limits = [-1 1];
    app.slider2.MajorTicks = [];
    app.slider2.ValueChangedFcn = createCallbackFcn(app,
@slider2_Callback, true);
    app.slider2.MinorTicks = [];
    app.slider2.FontSize = 11;
    app.slider2.Position = [8 25 99 3];

    % Create TabGroup
    app.TabGroup = uitabgroup(app.figure1);
    app.TabGroup.Position = [25 89 767 702];

    % Create ValnioblikTab
    app.ValnioblikTab = uitab(app.TabGroup);
    app.ValnioblikTab.Title = 'Valni oblik';

    % Create axes1_2
    app.axes1_2 = uiaxes(app.ValnioblikTab);
    app.axes1_2.XMinorTick = 'on';

```

```

app.axes1_2.YMinorTick = 'on';
app.axes1_2.NextPlot = 'replace';
app.axes1_2.XGrid = 'on';
app.axes1_2.YGrid = 'on';
app.axes1_2.BackgroundColor = [0.9725 0.9725 0.9725];
app.axes1_2.Position = [1 2 765 667];

% Create SpektralnaanalizaTab
app.SpektralnaanalizaTab = uitab(app.TabGroup);
app.SpektralnaanalizaTab.Title = 'Spektralna analiza';

% Create UIAxes
app.UIAxes = uiaxes(app.SpektralnaanalizaTab);
title(app.UIAxes, 'Amplitudna karakteristika')
xlabel(app.UIAxes, 'Frekvencija')
ylabel(app.UIAxes, 'Amplituda')
app.UIAxes.XLim = [0 20000];
app.UIAxes.XMinorTick = 'on';
app.UIAxes.YMinorTick = 'on';
app.UIAxes.XGrid = 'on';
app.UIAxes.YGrid = 'on';
app.UIAxes.Position = [1 323 765 350];

% Create UIAxes2
app.UIAxes2 = uiaxes(app.SpektralnaanalizaTab);
title(app.UIAxes2, 'Fazna karakteristika')
xlabel(app.UIAxes2, 'Frekvencija')
ylabel(app.UIAxes2, 'Fazni kut')
app.UIAxes2.XLim = [0 20000];
app.UIAxes2.XMinorTick = 'on';
app.UIAxes2.YMinorTick = 'on';
app.UIAxes2.XGrid = 'on';
app.UIAxes2.YGrid = 'on';
app.UIAxes2.Position = [1 68 765 255];

% Create GornjilimitEditFieldLabel
app.GornjilimitEditFieldLabel =
uilabel(app.SpektralnaanalizaTab);
app.GornjilimitEditFieldLabel.HorizontalAlignment = 'right';
app.GornjilimitEditFieldLabel.Position = [503 21 62 22];
app.GornjilimitEditFieldLabel.Text = 'Gornji limit';

% Create GornjilimitEditField
app.GornjilimitEditField = uieditfield(app.SpektralnaanalizaTab,
'numeric');
app.GornjilimitEditField.ValueChangedFcn = createCallbackFcn(app,
@GornjilimitEditFieldValueChanged, true);
app.GornjilimitEditField.Position = [589 21 100 22];
app.GornjilimitEditField.Value = Inf;

% Create DonjilimitEditFieldLabel
app.DonjilimitEditFieldLabel = uilabel(app.SpektralnaanalizaTab);
app.DonjilimitEditFieldLabel.HorizontalAlignment = 'right';
app.DonjilimitEditFieldLabel.Position = [297 21 58 22];
app.DonjilimitEditFieldLabel.Text = 'Donji limit';

% Create DonjilimitEditField
app.DonjilimitEditField = uieditfield(app.SpektralnaanalizaTab,
'numeric');

```

```

        app.DonjilimitEditField.ValueChangedFcn = createCallbackFcn(app,
@DonjilimitEditFieldValueChanged, true);
        app.DonjilimitEditField.Position = [370 21 100 22];
        app.DonjilimitEditField.Value = -Inf;

        % Create PromjenalimitafrekvencijeLabel
        app.PromjenalimitafrekvencijeLabel =
uilabel(app.SpektralnaanalizaTab);
        app.PromjenalimitafrekvencijeLabel.Position = [125 21 152 22];
        app.PromjenalimitafrekvencijeLabel.Text = 'Promjena limita
frekvencije: ';

        % Create Tablicniprikazspektralnaanaliza
        app.Tablicniprikazspektralnaanaliza = uitab(app.TabGroup);
        app.Tablicniprikazspektralnaanaliza.Title = 'Tablicni prikaz
spektralna analiza';

        % Create UITable
        app.UITable = uitable(app.Tablicniprikazspektralnaanaliza);
        app.UITable.ColumnName = {'Harmonik'; 'Frekvencija harmonika
[Hz]'; 'Amplituda harmonika'; 'Kut [stupnjevi]'};
        app.UITable.RowName = {};
        app.UITable.FontName = 'Times New Roman';
        app.UITable.FontSize = 14;
        app.UITable.Position = [0 2 766 674];

        % Create Channel1Button
        app.Channel1Button = uibutton(app.figure1, 'push');
        app.Channel1Button.ButtonPushedFcn = createCallbackFcn(app,
@Channel1ButtonPushed, true);
        app.Channel1Button.Position = [135 51 100 22];
        app.Channel1Button.Text = 'Channel 1';

        % Create Channel2Button
        app.Channel2Button = uibutton(app.figure1, 'push');
        app.Channel2Button.ButtonPushedFcn = createCallbackFcn(app,
@Channel2ButtonPushed, true);
        app.Channel2Button.Position = [273 51 100 22];
        app.Channel2Button.Text = 'Channel 2';

        % Create Channel3Button
        app.Channel3Button = uibutton(app.figure1, 'push');
        app.Channel3Button.ButtonPushedFcn = createCallbackFcn(app,
@Channel3ButtonPushed, true);
        app.Channel3Button.Position = [418 51 100 22];
        app.Channel3Button.Text = 'Channel 3';

        % Create Channel4Button
        app.Channel4Button = uibutton(app.figure1, 'push');
        app.Channel4Button.ButtonPushedFcn = createCallbackFcn(app,
@Channel4ButtonPushed, true);
        app.Channel4Button.Position = [555 51 100 22];
        app.Channel4Button.Text = 'Channel 4';

        % Show the figure after all components are created
        app.figure1.Visible = 'on';
    end
end

% App creation and deletion

```

```

methods (Access = public)

    % Construct app
    function app = Zadnja_verzija_1_9

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.figure1)

        % Execute the startup function
        runStartupFcn(app, @klizaci_OpeningFcn)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.figure1)
    end
end
end
end

```