

# Izrada grafičkog korisničkog sučelja u Matlabu za potrebe projektiranja digitalnih sustava upravljanja složenih struktura

---

**Polak, Siniša**

**Undergraduate thesis / Završni rad**

**2008**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:456678>

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

*Download date / Datum preuzimanja:* **2024-07-16**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



*Sveučilište u Zagrebu*  
***Fakultet strojarstva i brodogradnje***

# **ZAVRŠNI RAD**

*Siniša Polak*

*Zagreb, 2008.*

*Sveučilište u Zagrebu*  
***Fakultet strojarstva i brodogradnje***

# **ZAVRŠNI RAD**

*Voditelj rada:*

*Prof. dr. sc. Davor Zorc*

*Siniša Polak*

*Zagreb, 2008.*

## **Izjava**

Službeno Izjavljujem da sam završni rad radio sam . Zahvaljujem se višem asistentu Dr.sc. Danijel Pavković na suradnji i savjetima tijekom izrade završnog rada.

# Sadržaj

1. Uvod .....	7
2. Nastavna pomagala za predmete: Elektromotorni servo pogoni, Mikroprocesorsko upravljanje i Neizrazito digitalno upravljanje .....	8
2.1. Ukratko o radu s programskim paketom Matlab/Simulink .....	9
2.2. Primjeri nastavnih pomagala izrađenih u Matlab/Simulink okruženju ..	11
2.2.1 Primjer vježbi iz elektromotornih servo pogona .....	12
2.2.2 Primjer vježbi iz neizrazitog digitalnog upravljanja .....	13
2.2.3 Primjer vježbi iz mikroprocesorskog upravljanja .....	15
3. Matlab GUI – Graphical User Interface: korisničko grafičko sučelje .....	17
3.1 Guide control panel- kontrolni prozor GUI-a .....	18
3.2 Guide Alignment Tool- GUI alat za linijsko usklađivanje elemenata .....	19
3.3 Graphics Property Editor – preglednik grafičkih svojstava .....	20
3.4 Primjer izrade grafičkog korisničkog sučelja .....	22
3.5 Izrada korisničkog sučelja bez GUI alata .....	26
3.5.1 Korisničko sučelje poziva m-skriptu (callback) .....	27
3.5.2 Unutar korisničkog sučelja izračun i crtanje grafa .....	28
4. Prilagodba postojećih nastavnih pomagala primjenom GUI-a .....	29
4.1. Primjer prilagodbi za slučaj vježbi iz elektromotornih servo pogona: regulator brzine vrtnje	
4.2 Primjer prilagodbi za slučaj vježbi iz neizrazitog digitalnog upravljanja: analiza vremenski diskretnog signala ..	30
4.3. Primjer prilagodbi za slučaj vježbi iz mikroprocesorskog upravljanja: analiza vremenski diskretnih sustava .....	31
5. Zaključak .....	32
6. Literatura .....	33
7. Prilog .....	34

## Popis slika

Slika 1 Komandi prozor Matlaba	9
Slika 2. Simulink Library Browser(preglednik simulink biblioteka) i prozor novog simulacijskog modela	10
Slika 3. Unošenje varijabli preko naredbe input (vježba iz ESP)	11
Slika 4. Simulink shema regulatora brzine vrtnje istosmjernog motora (vježba iz ESP)	12
Slika 5. Prikaz rezultata (vježba iz ESP)	12
Slika 6. Prijenosna funkcija (vježba iz NDU)	13
Slika 7. Prikaz rezultata (vježba iz NDU)	14
Slika 8. Unošenje vrijednosti preko naredbe input (vježba iz MPU)	15
Slika 9. Simulink model sinusnog spektra (vježba iz MPU)	15
Slika 10. Prikaz rezultata (vježba iz MPU)	16
Slika 11. Control Panel – kontrolni prozor	18
Slika 12. Alat za linijsko usklađivanje elemenata	19
Slika 13. Prikaz prozora za promjenu opcija elemenata	20
Slika 14. Prikaz svih opcija za element StaticText	21
Slika 15. Prozor za proračun spektra vremenski kontinuiranog i diskretnog signala dobivenog otiskavanjem odziva oscilatornog P2 člana	22
Slika 16. Podešavanje svojstva units, odabrana je vrijednost normalized	23
Slika 17. Svojstvo Callback(pozivna rutina)	24
Slika 18. Svojstvo String(opisni tekst)	25
Slika 19. a) M-skripta, b) korisničko sučelje	26
Slika 20. Primjer 2-pozivanje m-skripte preko funkcije callback	27

Slika 21. a) m-skripta, b) korisničko sučelje	.....28
Slika 22. Primjer 2-pozivanje m-skripte preko funkcije callback	.....29
Slika 23. Primjer 3 Izrada korisničkog sučelja unutar m skripte pomoću switch case funkcije	.....30
Slika 24. ESP-simulacija brzine vrtnje elektromotora (graf 1.)	.....31

# 1. Uvod

Matlab je jezik visoke učinkovitosti u tehničkom računanju. On objedinjuje računanje, vizualizaciju i programiranje u prozorskom okolišu vrlo ugodnom za korisnika, gdje su problemi i rješenja izraženi uobičajenim matematičkim zapisom. Danas svojstva Matlab-a daleko prelaze originalni «matrični laboratorij». Uz osnovni paket postoje i brojni programski alati (toolboxes) koji pokrivaju gotovo sva područja inženjerske djelatnosti: obradu signala, slike, 2D i 3D grafičko oblikovanje, automatsko upravljanje, indentifikaciju sustava, statističke obrade, neuronske mreže, financijsku matematiku, simboličku matematiku i mnogih drugih. Paket SIMULINK je dodatak Matlab-u koji omogućuje simulaciju kontinuiranih i diskretnih sustava pomoću funkcijskih blok dijagrama i dijagrama stanja. Matlab je otvoren sustav u kojem korisnik može graditi svoje vlastite alate i biblioteke te modificirati postojeće, jer su dostupni u obliku izvornog koda.

Zašto Matlab - GUI (Graphical User Interface: korisničko grafičko sučelje) ? Zato što se može koristiti već gotov kod iz m-datoteka, odnosno lakše se integrirati u taj kod. Kod drugih GUI okruženja prvo bi trebali prevesti cijeli gotov kod nekog programa u taj jezik, a zatim još razviti i korisničko sučelje, Matlab ima gotov toolbox za izradu GUI-a.

U ovom radu je primijenjen GUI na već postojeće vježbe iz Mikroprocesorskog upravljanja, Elektromotornih servo pogona i Neizrazitog digitalnog upravljanja. M-skripte koje je napravio dr.sc. Danijel Pavković poslužile su kao materijal za izradu korisničkog sučelja. Umjesto da se dane skripte pokreću iz Matlab-ovog komadnog prozora (Command prompt), pokreću se pomoću grafičkog sučelja razvijenog GUI alatom.

U nastavku ovog rada bit će opisani problemi (mane) pokretanja m-skripte preko komadnog prozora i prednosti pokretanja skripte i pregledavanja grafova i unesenih varijabli preko korisničkog sučelja. Također će biti detaljnije opisan matlabov GUI alat.



## **2. Nastavna pomagala za predmete: Elektromotorni servo pogoni, Mikroprocesorsko upravljanje i Neizrazito digitalno upravljanje**

Vježbe iz elektromotornih servo pogona sastoje se iz 6 odvojenih cjelina.

1. Regulacija struje armature istosmjernog motora,
2. Regulacija brzine vrtnje istosmjernog motora,
3. Regulacija pozicije istosmjernog motora sa i bez predupravljanja,
4. Slijeđenje referente trajektorije,
5. Regulacija brzine vrtnje dvomasenog sustava s elastičnim vratilom,
6. Regulacija pogona s elastičnom transmisijom.

Vježbe iz mikroprocesorskog upravljanja sastoje se iz 4 odvojenih cjelina.

1. Izbor vremena uzorkovanja na kvalitetu rekonstrukcije vremenski kontinuiranog signala,
2. Inverzni postupak z-transformacije,
3. Regulacijski krug s jediničnim P regulatorom,
4. Određivanje parametara PID regulatora prema Takahashi-u za aperiodski proces i za integralni proces s oscilatornim članom i realnom nulom

Vježbe iz neizrazitog digitalnog upravljanja sastoje se iz 6 odvojenih cjelina.

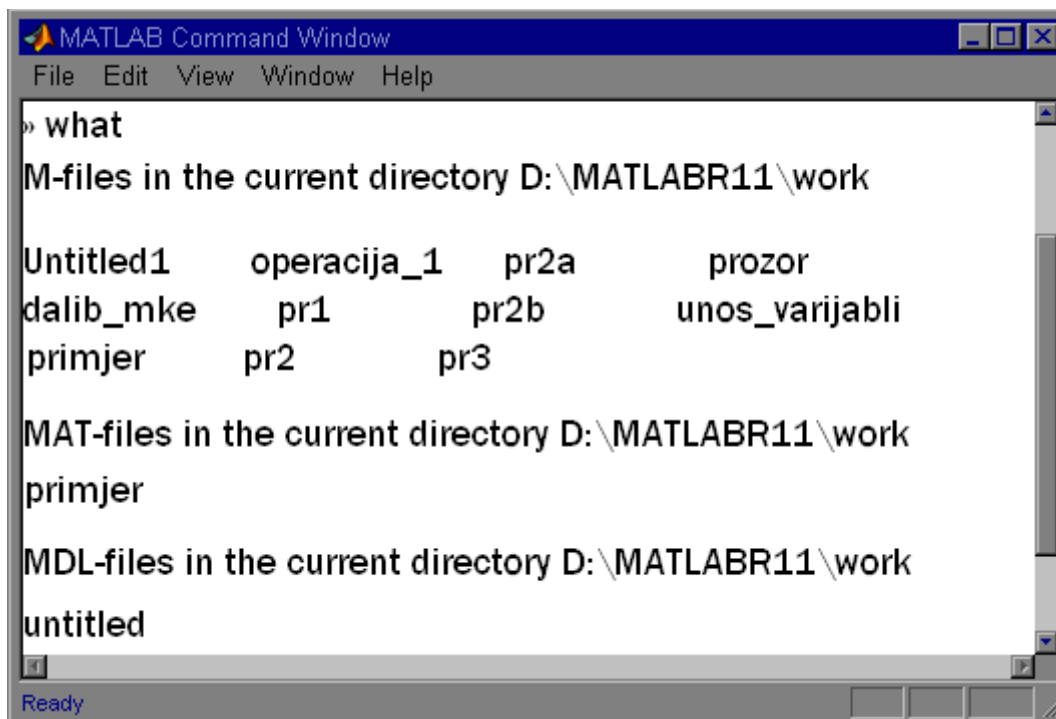
1. Z-transformacija i vremenski diskretna prijenosna funkcija, povezivanje Laplac-ove s i z domene, utjecaj polova i nula prijenosne funkcije u z području na brzinu odziva,
2. Postupak sinteze vremenski diskretnog PID regulatora,
3. Projektiranje regulatora stanja elektromotora opisanog dvomasenim modelom s elastičnom vezom,
4. Projektiranje regulatora stanja elektromotornog pogona sa složenom dinamikom mehanizma,
5. Projektiranje polinomskog regulatora brzine vrtnje elektromotornog pogona sa složenom dinamikom radnog mehanizma i Ottovog motora,
6. Postupci predikcije modela procesa zasnovane na minimiziranju srednjeg kvadratnog odstupanja predikcije modela.

Nastavni materijali za svaku vježbu(cjelinu) uobičajeno sadrže simulink model (u simulinku), te odgovarajuće M-datoteke(m-skripte) za inicijalizaciju simulacija i/ili za samostalne promjene. Primjeri M-skripti i simulacija dani su u prilogu. Navedene simulacije i programski kod za matlab izvode se 'ručno' zadavanjem naredbi u Matlab/Simulink okruženju. Cilj ovoga rada je maksimalno automatizirati rad s postojećim datotekama kako bi se olakšalo izvođenje nastave iz navedenih kolegija.

## 2.1.Ukratko o radu s programskim paketom Matlab/Simulink

U Windows okolišu Matlab se pokreće dvostrukim klikom na Matlab ikonu na radnoj površini (desktopu) ili izborom Matlab programa preko Start menu-a. Otvaranjem Matlab prozora, pojavljuje se radni prostor tzv. 'command window' u kojem Matlab ispisuje '»' kao upit (engl. prompt) i očekuje korisnikovu naredbu. U ovakvom interaktivnom načinu rada, korisnik upisuje naredbu koja se potom izvršava pritiskom na 'Enter' tipku. Nakon što Matlab izvrši (izračuna) naredbu, ispisuje se ili iscrtava odgovor, te ponovno pojavljuje prompt za nastavak komunikacije s korisnikom. U slučaju da naredba završava znakom ';' sprječava se ispis rješenja na zaslonu. Naredbe koje su duže od širine zaslona nastavljaju se u novom redu utipkavanjem triju točkica '...' u prethodnom redu, znak '%' služi kao komentar naredbe. Sve znakove desno od tog znaka Matlab neće interpretirati.

Kod poziva skript-datoteke, u naredbenoj se liniji napiše samo njeno ime (bez '.m' ekstenzije!), nakon čega se njezine naredbe izvršavaju jedna za drugom. Na zaslonu se pojavljuju samo rješenja koja naredba proizvede, a ne i same naredbe. Da bismo istodobno pratili i naredbe i njihovo izvršenje potrebno je uključiti ispis s naredbom echo on. Poništenje se postiže isključenjem iste naredbe – echo off. Bilo koji tekst koji je komentar u skripti ne ispisuje se na zaslon. Popis m-datoteka u trenutačnom imeniku (folderu, direktoriju) dobiva se pomoću naredbe what.




```
» what
M-files in the current directory D:\MATLABR11\work

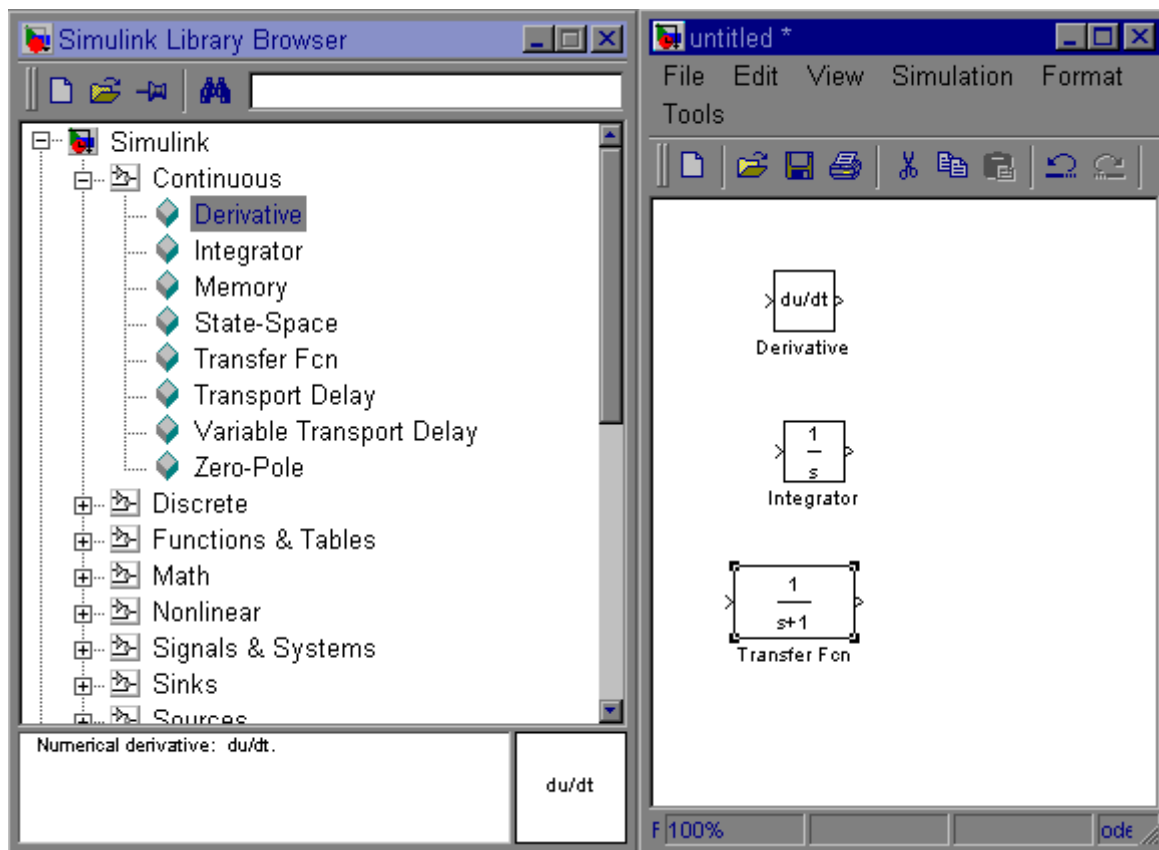
Untitled1    operacija_1  pr2a        prozor
dalib_mke    pr1          pr2b        unos_varijabli
primjer      pr2          pr3

MAT-files in the current directory D:\MATLABR11\work
primjer


MDL-files in the current directory D:\MATLABR11\work
untitled
```

Slika1. Komandni prozor Matlaba

U komandnom prozoru Matlaba utipka se naredba "simulink", pritisne enter i otvori se prozor "Simulink Library Browser". Pritiskom na ikonu  otvara se novi simulacijski model. U model se unose željeni simulacijski blokovi. Na slici.2 uneseni su u model blokovi skupine "Continuous", blok "Derivate"(derivacijski član), "Integrate"(integracijski član) i "Transfer Fcn"(prijenosna funkcija).



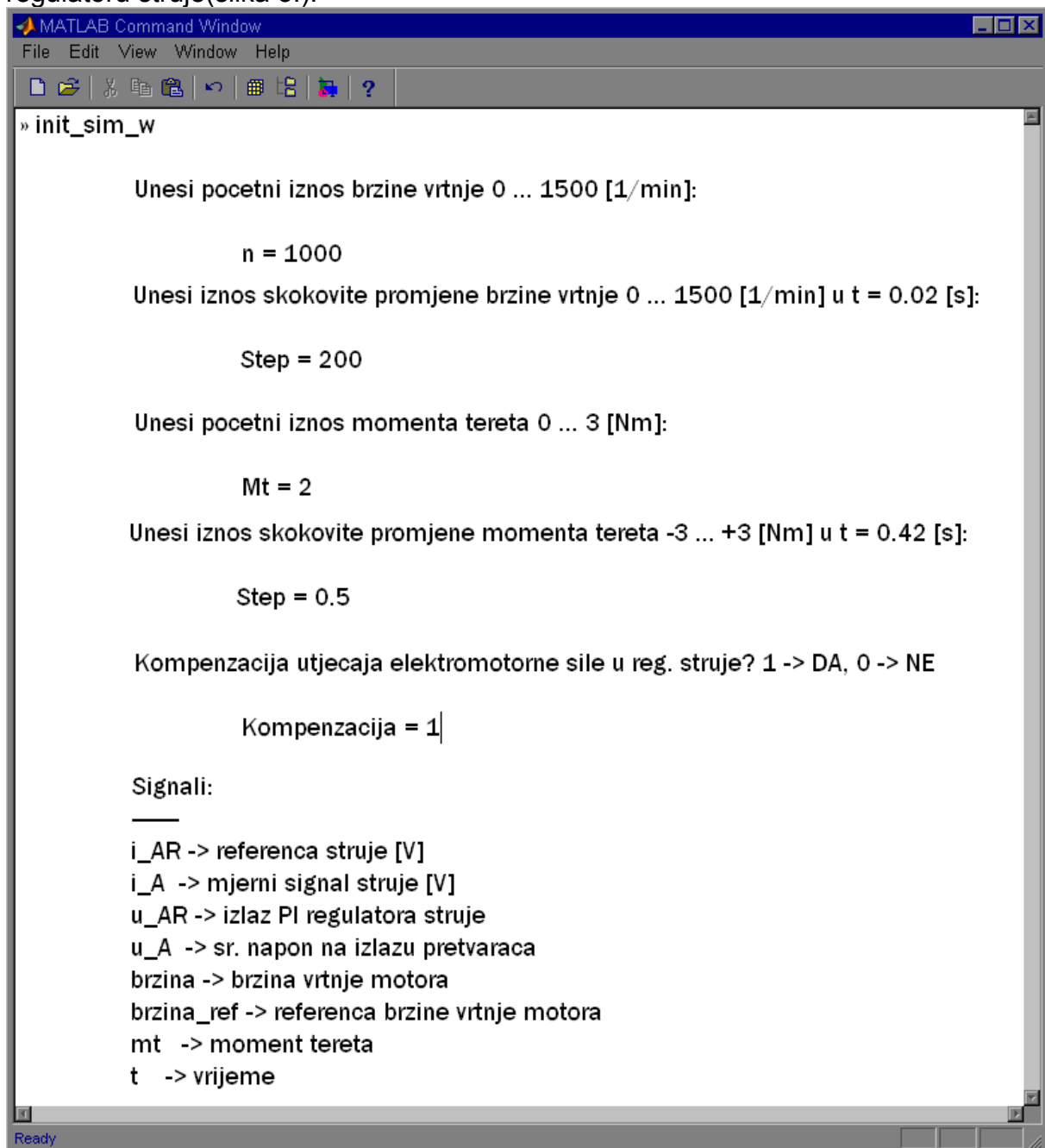
Slika 2. Simulink Library Browser(preglednik simulink biblioteka) i prozor novog simulacijskog modela

U komandnom prozoru Matlaba utipka se naredba "simulink", pritisne enter i otvori se prozor "Simulink Library Browser". Pritiskom na ikonu  otvara se novi simulacijski model. U model se unose odgovarajući simulacijski blokovi. Simulacijski blokovi koji se mogu koristiti u Simulinku nalaze se u više različitih biblioteka (tzv. Toolbox-a). Najčešće se koriste blokovi iz Simulink Toolbox-a (tzv. Opći blokovi). Njihove podskupine su : Continuous(vremenski-kontinuirani blokovi), Discrete(vremenski-diskretni blokovi), Functions & Tables(matematičke funkcije i pregledne tablice), Math(osnovne matematičke operacije), Signals & Systems(povezivanje/distribucija signala), Sinks(blokovi za spremanje i prikaz signala), Sources(izvori ispitnih signala, generiranje ispitnih signala).

## 2.2. Primjeri nastavnih pomagala izrađenih u Matlab/Simulink okruženju

### 2.2.1. Primjer vježbi iz elektromotornih servo pogona

Nakon što se utipka 'init\_sim\_w' u komandni prozor matlaba pokreće se m-skripta init\_sim\_w.m (simulacija regulacije brzine vrtnje-elektromotora, vidi prilog slika 4.). Preko naredbe input unosi se početni iznos brzine vrtnje, iznos skokovite promjene brzine vrtnje, početni iznos momenta tereta, iznos skokovite promjene momenta tereta, odabire se ili ne odabire (1 ili 0) kompenzacija utjecaja elektromotorne sile u regulatoru struje (slika 3.).



```
» init_sim_w

Unesi pocetni iznos brzine vrtnje 0 ... 1500 [1/min]:

      n = 1000
Unesi iznos skokovite promjene brzine vrtnje 0 ... 1500 [1/min] u t = 0.02 [s]:

      Step = 200

Unesi pocetni iznos momenta tereta 0 ... 3 [Nm]:

      Mt = 2
Unesi iznos skokovite promjene momenta tereta -3 ... +3 [Nm] u t = 0.42 [s]:

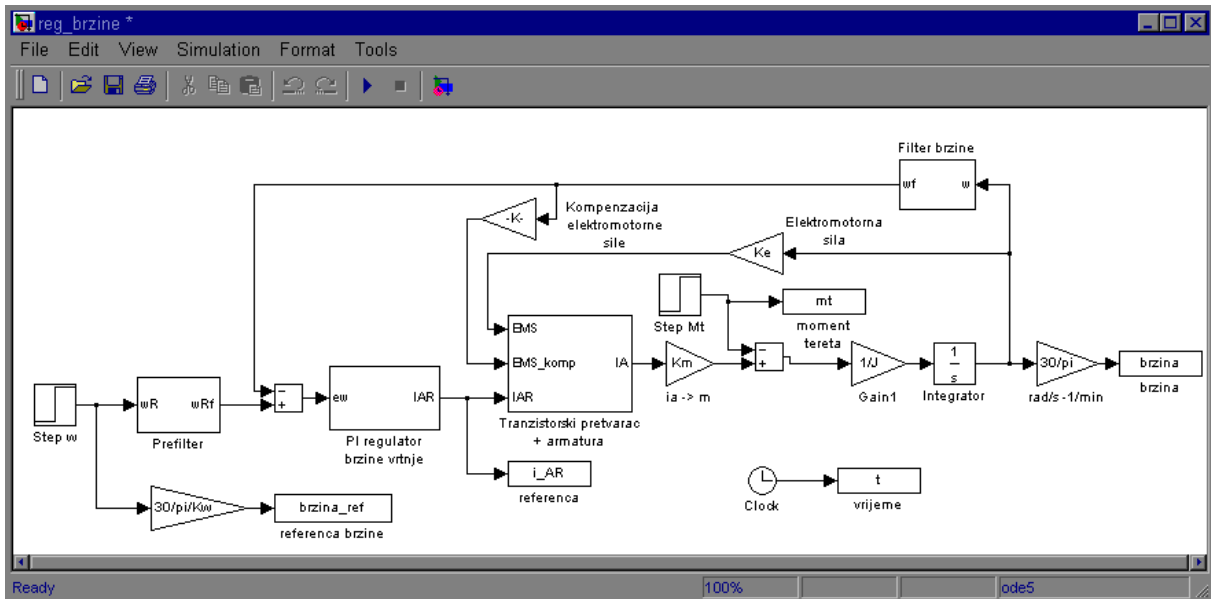
      Step = 0.5

Kompenzacija utjecaja elektromotorne sile u reg. struje? 1 -> DA, 0 -> NE

      Kompenzacija = 1|

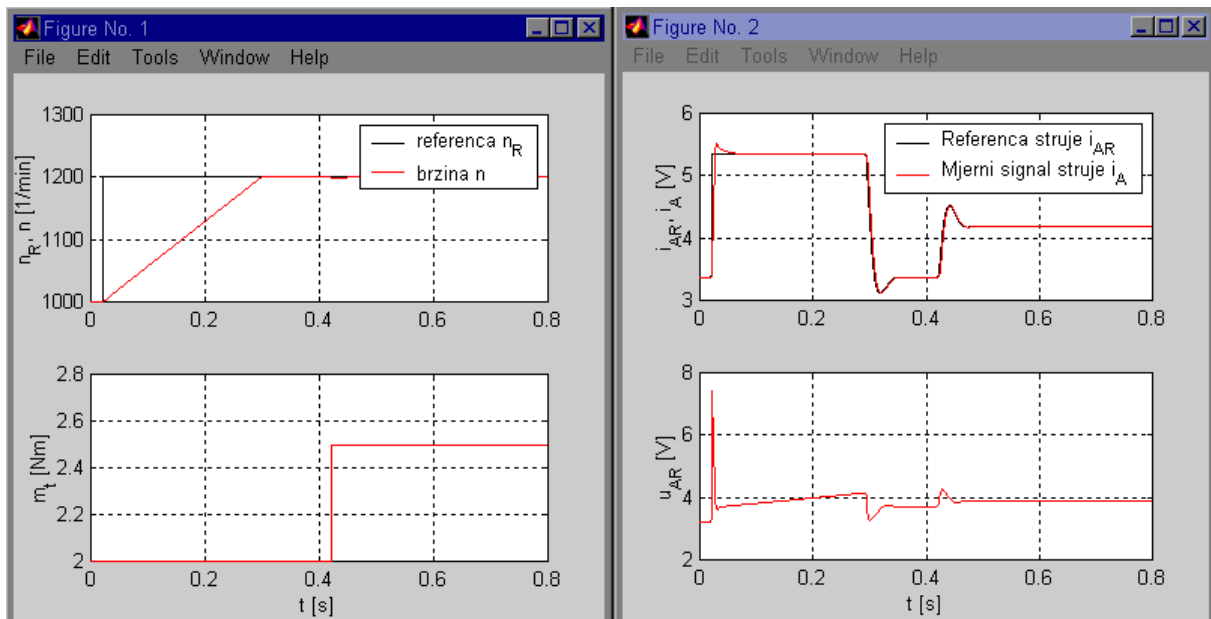
Signali:
_____
i_AR -> referenca struje [V]
i_A  -> mjerni signal struje [V]
u_AR -> izlaz PI regulatora struje
u_A  -> sr. napon na izlazu pretvaraca
brzina -> brzina vrtnje motora
brzina_ref -> referenca brzine vrtnje motora
mt    -> moment tereta
t     -> vrijeme
```

Slika 3. Unošenje varijabli preko naredbe input (vježba iz esp)



Slika 4. Simulink shema regulatora brzine vrtnje istosmjernog elektromotora (vježba iz esp)

Na slici 4. vidi se simulink shema regulatora brzine elektromotora, shema je malo pojednostavljena, jer je PI regulator brzine vrtnje stavljen u jedan blok, tranzistorski pretvarač i armatura je stavljen u jedan blok i filter brzine je stavljen u jedan blok.

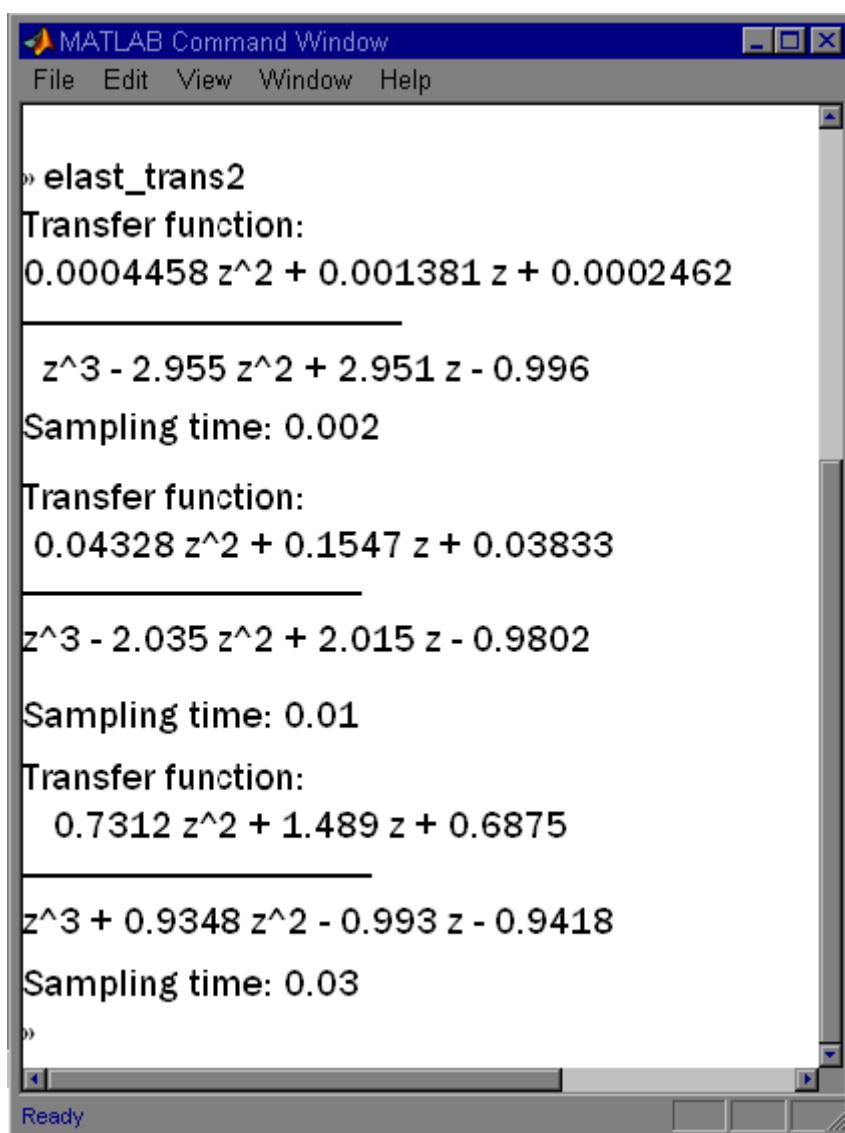


Slika 5. Prikaz rezultata (vježba iz esp)

Slika 5. prikazuje referencu brzine vrtnje i brzinu vrtnje u ovisnosti o vremenu, zatim moment tereta u ovisnosti o vremenu, referencu struje i mjerni signal struje u ovisnosti o vremenu, te napon armature u ovisnosti o vremenu.

Ako se želi promijeniti samo početni iznos brzine vrtnje treba ponovo pokretati program i ponovo upisivati sve vrijednosti, ali se mogu vidjeti istovremeno oba grafa.

## 2.2.2. Primjer vježbi iz neizrazitog digitalnog upravljanja



```
MATLAB Command Window
File Edit View Window Help

» elast_trans2
Transfer function:
0.0004458 z^2 + 0.001381 z + 0.0002462
-----
z^3 - 2.955 z^2 + 2.951 z - 0.996
Sampling time: 0.002

Transfer function:
0.04328 z^2 + 0.1547 z + 0.03833
-----
z^3 - 2.035 z^2 + 2.015 z - 0.9802
Sampling time: 0.01

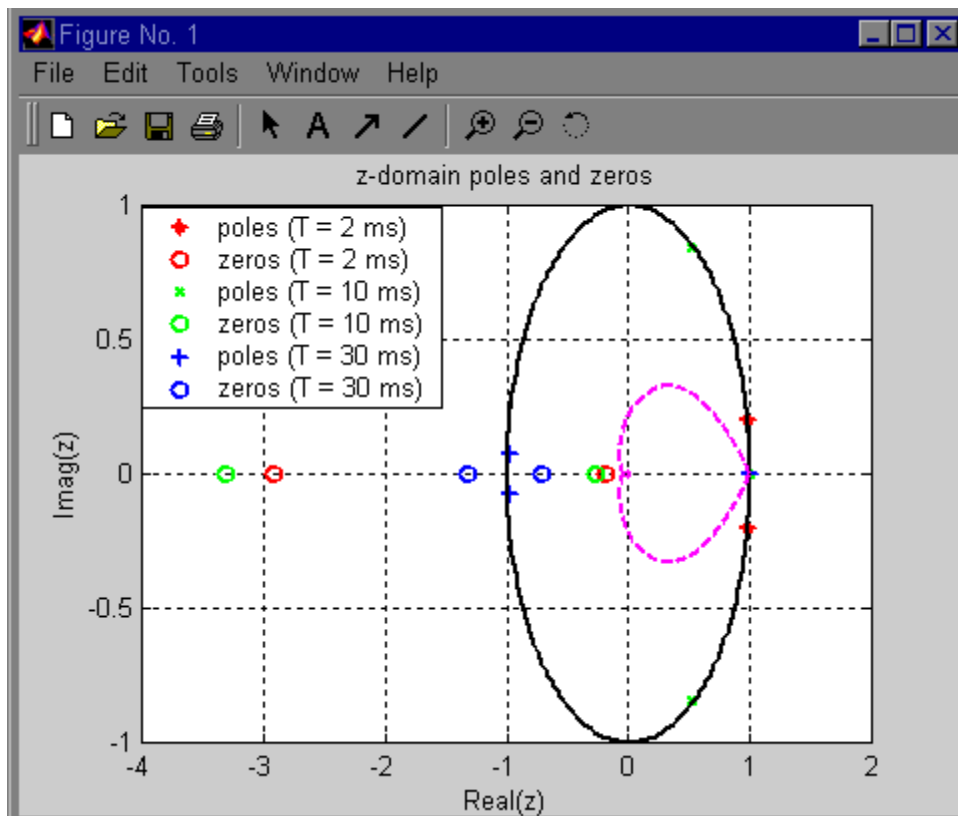
Transfer function:
0.7312 z^2 + 1.489 z + 0.6875
-----
z^3 + 0.9348 z^2 - 0.993 z - 0.9418
Sampling time: 0.03

»

Ready
```

Slika 6. Prijenosne funkcije (vježbe iz ndu)

Nakon što se utipka 'elast\_trans2' u komandni prozor matlaba pokreće se m-skripta elast\_trans2.m i kako korisnik pritišće tipku space ili enter na ekranu se ispisuju prijenosne funkcije s različitim vremenima uzorkovanja, a unutar drugog prozora se crtaju grafovi(slika 7.).

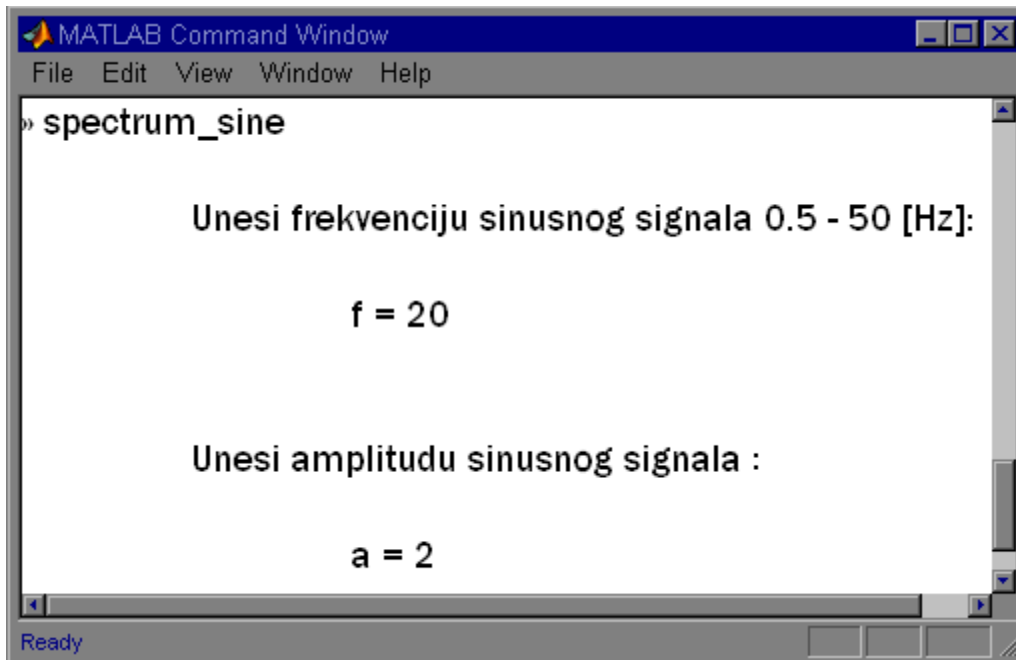


Slika 7. Prikaz rezultata (vježba iz ndu)

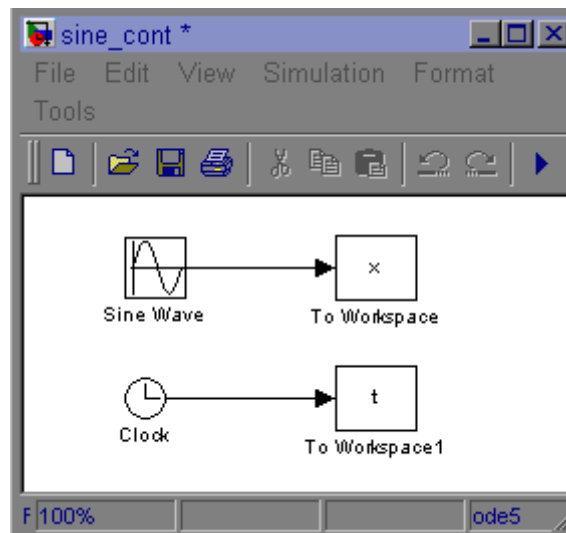
Na slici 7. je prikazan položaj polova i nula za dvomaseni sustav s elastičnom vezom. Slabo prigušeni polovi nalaze se vrlo blizu jedinične kružnice (rubu stabilnosti), s porastom vremena uzorkovanja polovi i nule pomiču se prema lijevom dijelu jedinične kružnice.

### 2.2.3. Primjer vježbi iz mikroprocesorskog upravljanja

Korisnik preko input menu-a unosi frekvenciju sinusnog signala i amplitudu, taj sinusni signal se diskretizira različitim frekvencijama (vidi prilog, slika 10.)



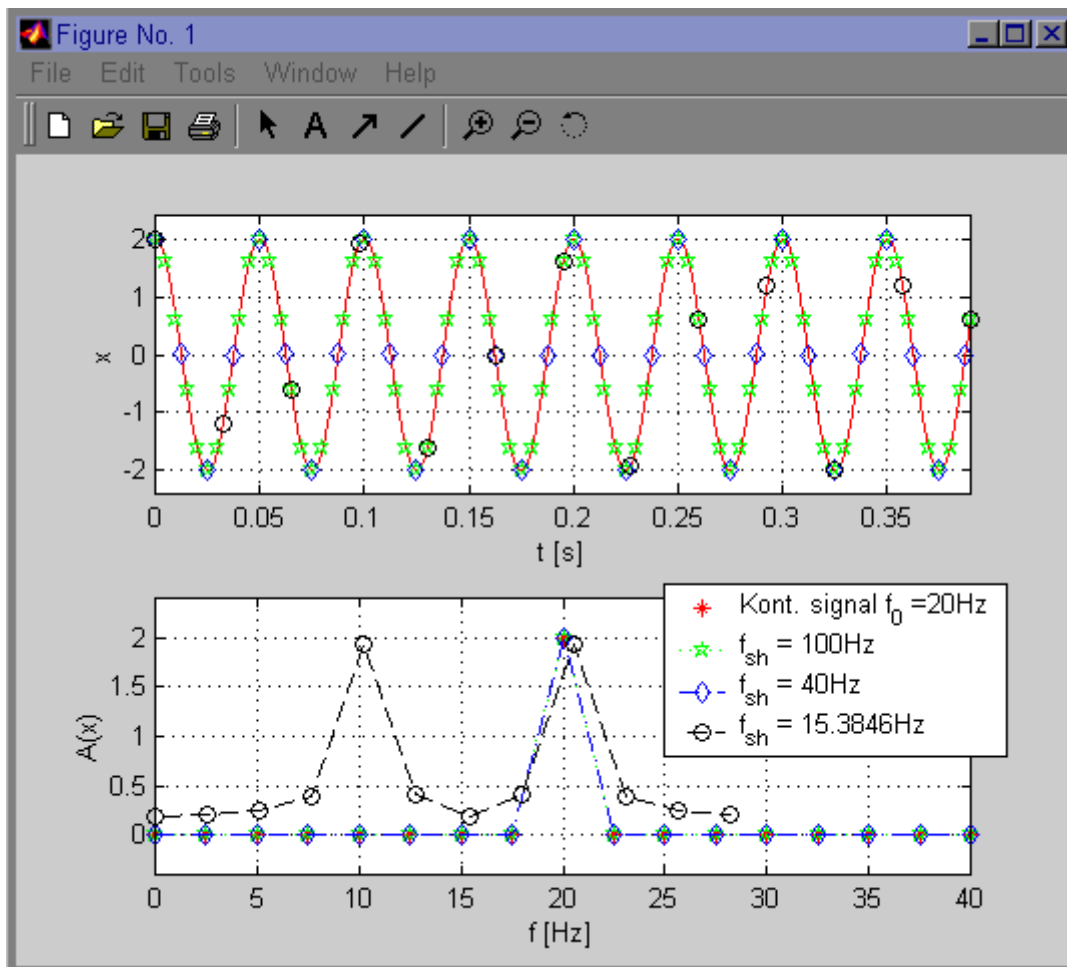
Slika 8. Unošenje vrijednosti preko naredbe input (vježba iz mpu)



Slika 9. Simulink model sinusnog spektra (vježba iz mpu)

Slika 9. prikazuje simulink model, sinusni val se šalje u workspace(radni prostor) kao varijabla x. U radni prostor se također šalje i vrijeme kao varijabla t. Bitno je da oba vektora budu jednake duljine odnosno veličine kako bi se mogli prikazati na grafu.





Slika 10. Prikaz rezultata (vježba iz mpu)

Gornji graf Slike 10. prikazuje kontinuirani signal frekvencije 20 Hz i vrijednosti amplitude 2, a donji graf prikazuje različite frekvencije uzorkovanja kontinuiranog signala.

Nakon što se preko komandnog prozora Matlaba i naredbe 'input' unese frekvencija i amplituda kontinuiranog signala, korisnik pritiskom tipke 'space' ili 'enter' iscrtava graf unutar prozora "Figure No.1".

### 3. Matlab GUI – Graphical User Interface : korisničko grafičko sučelje

Korisničko grafičko sučelje omogućuje izradu takvih programa za koje korisnik neće morati učiti programiranje, niti složene postupke njihove upotrebe: dovoljno je tipkom miša kliknuti na gumb, upisati podatke u okvir, izabrati neku ponudu i sl. GUI predstavlja samo grafički okoliš ugodan čovjeku za komunikaciju sa strojem, ali ne i program koji rješava neki problem. Postoji veza GUI s programom za koji je okoliš izgrađen. To se postiže definiranjem pozivnih rutina (eng. Callback routines) kao vrijednosnih svojstava posebnih UI objekata, a to su *Uicontrol*, *Uimenu*, *Uicontextmenu*.

*Uicontrol* su objekti za kontrolu kao što su gumbi, liste, tekst okviri, mjerila i sl. Svaki objekt očekuje neku interakciju s korisnikom, podatak, izbor ili klik mišem.

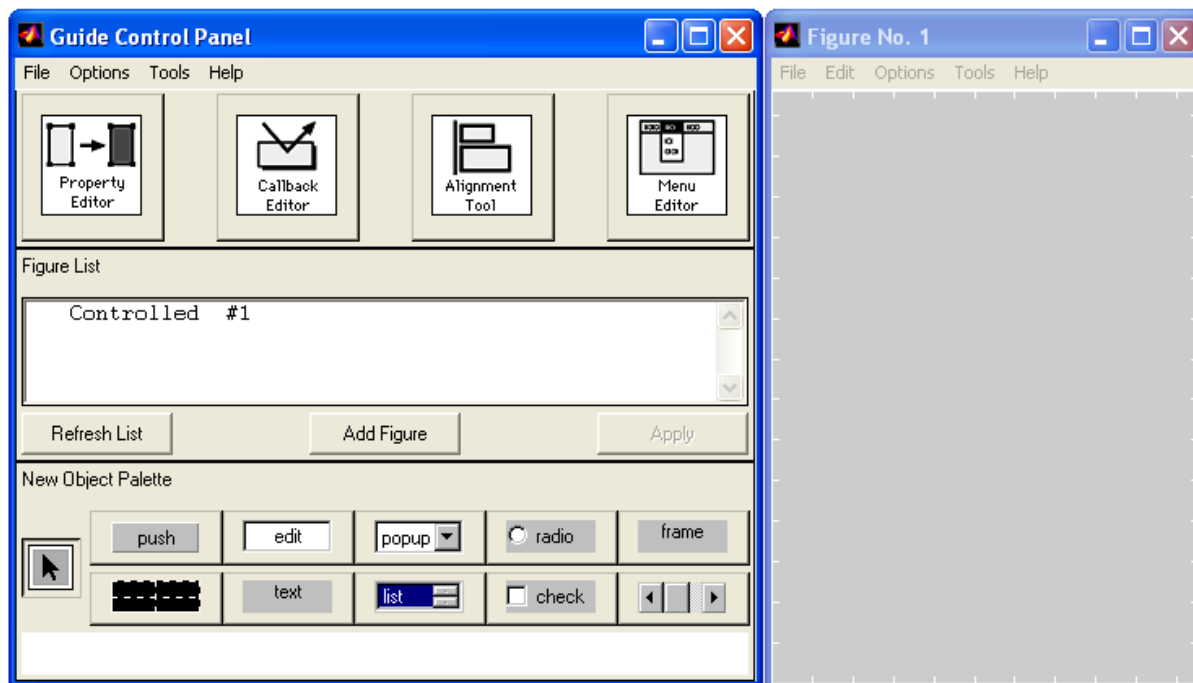
*Uimenu* objekti su padajuće izborne ponude koje izvršavaju pozitivne rutine na temelju korisnikovog izbora. *Uimenu* i *Uicontrol* su u hijerarhijskoj ljestvici sljedbenici slika (figure) i prema tome su neovisni o osima.

*Uicontextmenu* sa sintaksom

```
Uicontextmenu('svojstvo_1', vrijednost_1, 'svojstvo_2', vrijednost_2 ...)
```

stvora držalo koje se koristi u funkciji `uimenu()`, kako bi se izgradio sustav izbornih ponuda sa zadanim svojstvima.

### 3.1 Guide control panel – kontrolni prozor – GUI-a



Slika 11 Control Panel – kontrolni prozor

Kada se utipka GUIDE u komandni prozor MATLABA, otvara se kontrolni prozor i još jedan prazni prozor u koji se mogu stavljati ponuđene elemente.

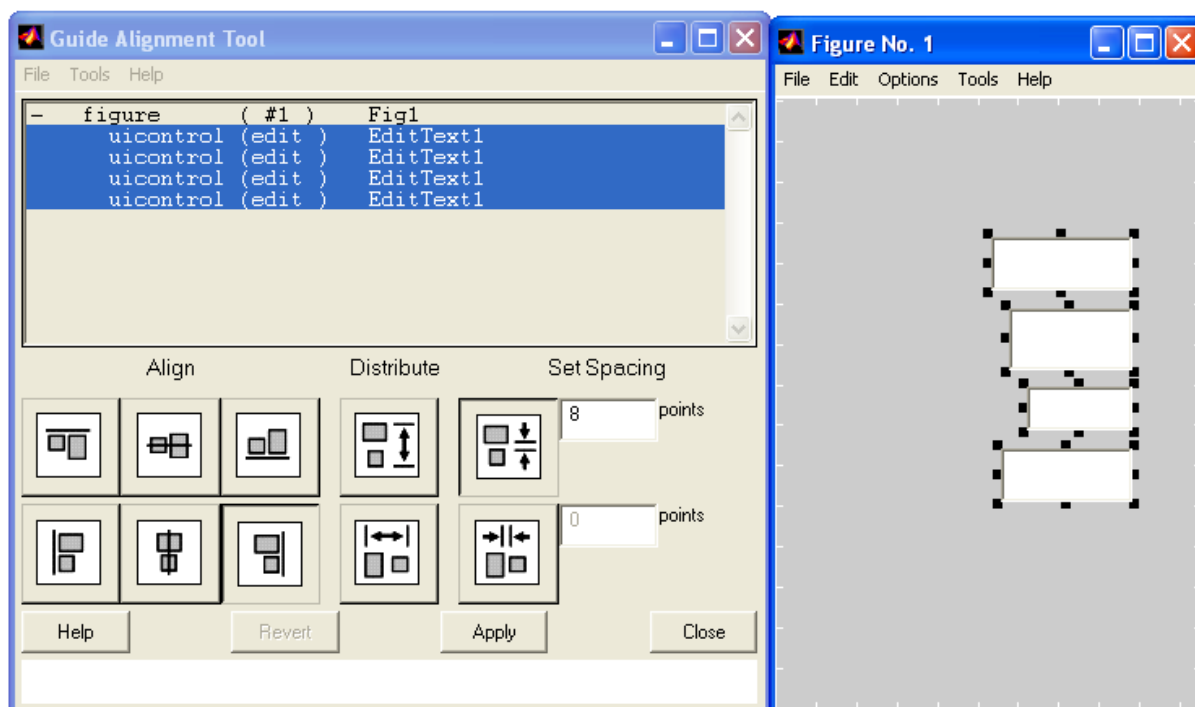
Ti elementi su ;

Push(gumb koji se aktivira pritiskom na lijevu tipku miša(klik)), edit(okvir za unos podataka), popup(padajući izbornik), radio(radio gumb), frame(okvir na koji se može smjestiti više elemenata), axes(osi za grafički prikaz), text(okvir za unos teksta), list(lista sa jednom ili više mogućnosti), check(selektivni gumb), slider(klizač).

Da bi se stavio element u prazni prozor, treba ga selektirati, a zatim pomaknuti pokazivač na željeno mjesto u praznom prozoru i uz pritisnutu lijevu tipku miša pomicati pokazivač kako bi se namjestila željena veličinu elementa i zatim se otpusti lijeva tipka miša. Ako se selektira element u prozoru (Figure) i pritisne desna tipka miša, može se kopirati selektirani element.

Kada se postave svi elementi u prazni prozor, mogu se po potrebi poravnati sa Alignment Tool (alat za poravnavanje), a sa Property Editorom mogu se prilagoditi opcije elemenata.

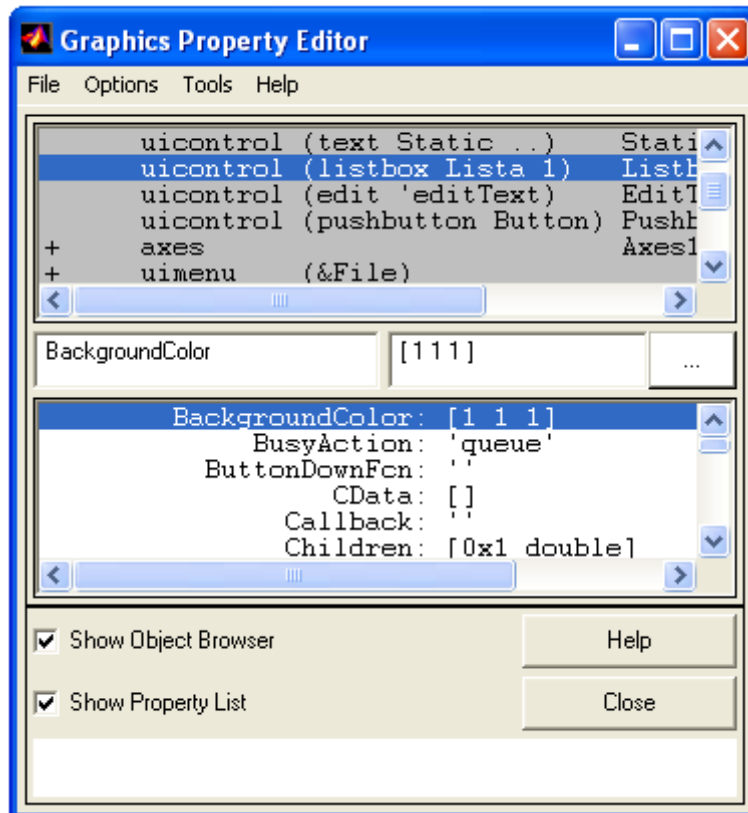
## 3.2 Guide Alignment Tool – GUI alat za linijsko usklađivanje elemenata



Slika 12. Alat za linijsko usklađivanje elemenata

Elementi se prvo selektiraju, pomoću pokazivača miša i tipke shift. Zatim se odabere jedna od mogućnosti usklađivanja elemenata, na slici je odabrano usklađivanje elemenata po desnom rubu s međusobnim razmakom od 8 točaka. Pritiskom na tipku Apply (primjena) dobije se rezultat prikazan na slici(12).

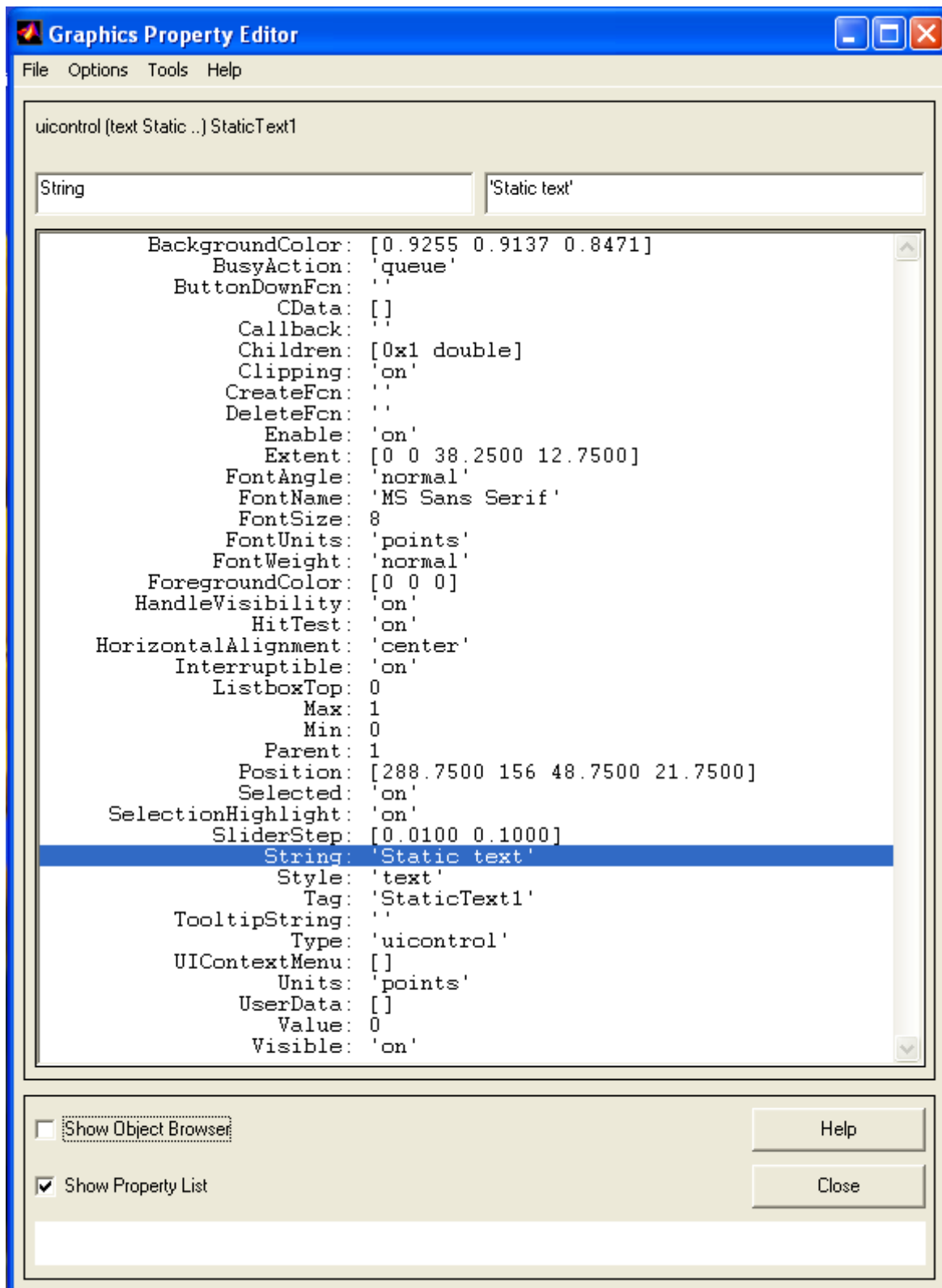
### 3.3 Graphics Property Editor – preglednik grafičkih svojstava



Slika 13. Prikaz prozora za promjenu opcija elemenata

Na slici 13. Graphics Property Editor (izbornik grafičkog prozora) koji je podijeljen na dva dijela; Object Browser(preglednik objekata) i Property List(lista opcija). U ovom prozoru može se odabrati jedan element i jedna od njegovih opcija i zatim se ta opcija može promijeniti. Na slici 3. je odabran objekt uicontrol (listbox Lista1) i opcija BackgroundColor(boja podloge), koja je postavljena na bijelu podlogu.

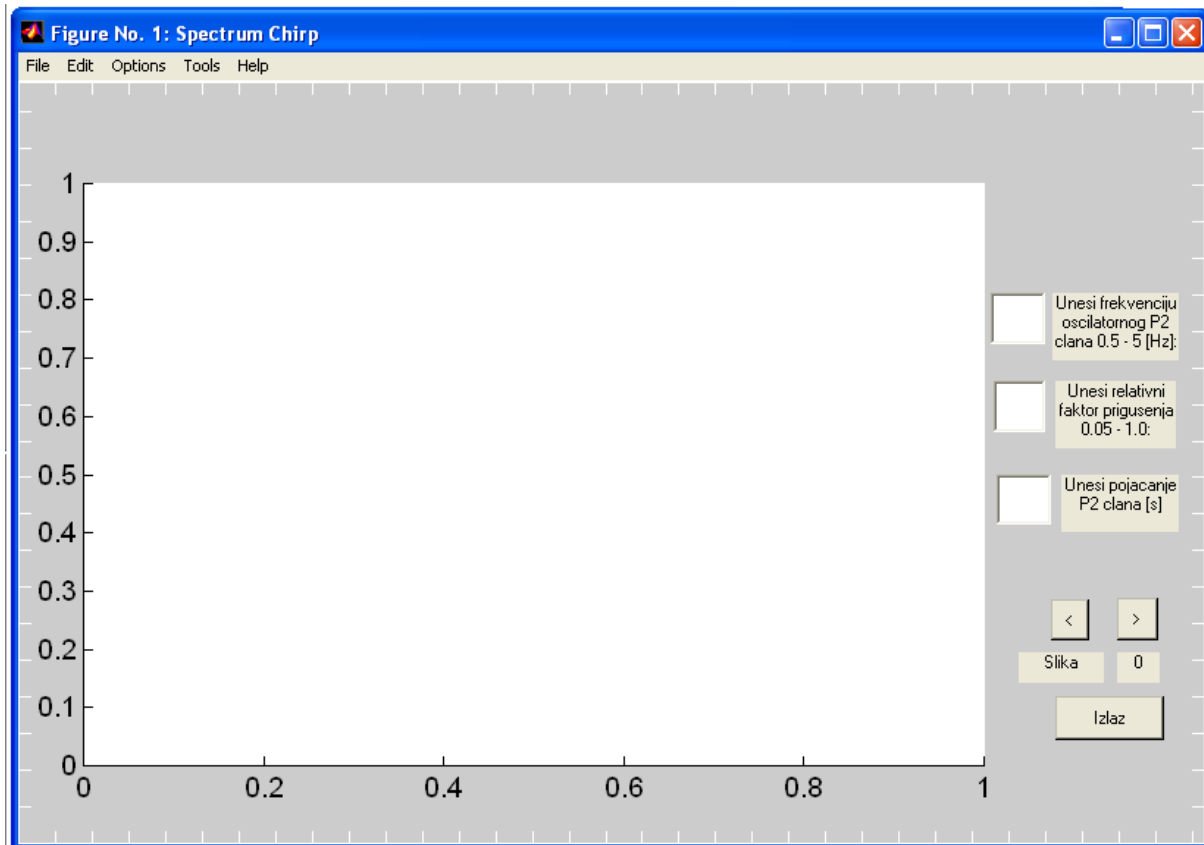
Ako u pregledniku objekata se odabere objekt uicontrol (text StaticText) i pritisne se lijeva tipka miša na Show Object Browser (kako bi isključili pregled objekata), dobije se lista od 39 opcija koje se mogu promijeniti po potrebi(slika 14.).



Slika 14. Prikaz svih opcija za element StaticText

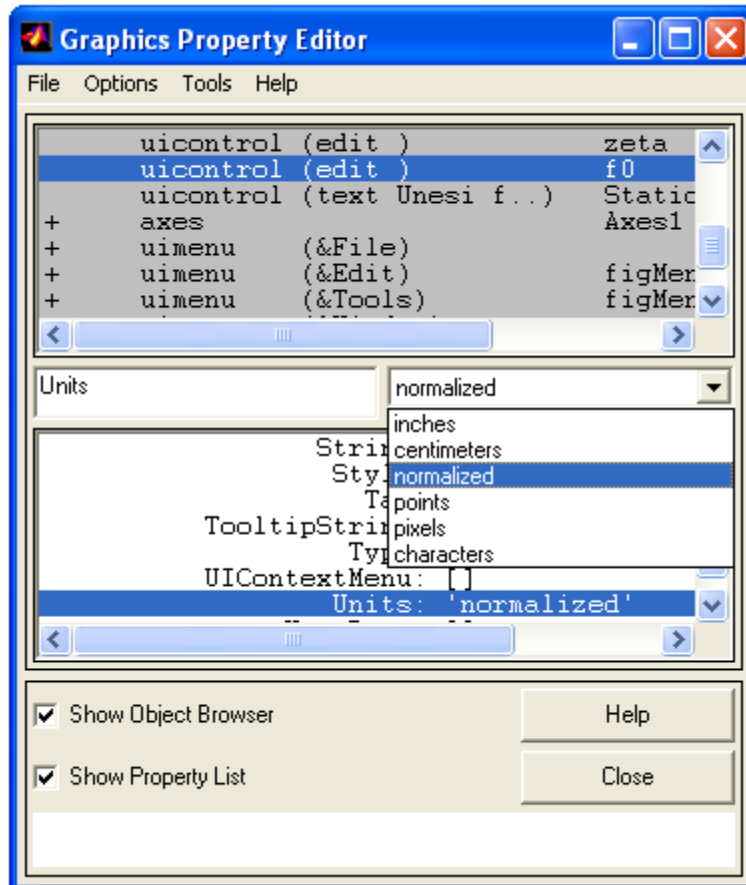
### 3.4 Primjer izrade grafičkog korisničkog sučelja (GUI)

Ovaj primjer opisuje izradu grafičkog korisničkog sučelja pomoću kojeg se može proračunati spektar vremenski kontinuiranog i vremenski diskretnog signala dobivenog otipkavanjem odziva dinamičkog P2 člana.



Slika 15. Prozor za proračun spektra vremenski kontinuiranog i diskretnog signala dobivenog otipkavanjem odziva oscilatornog P2 člana

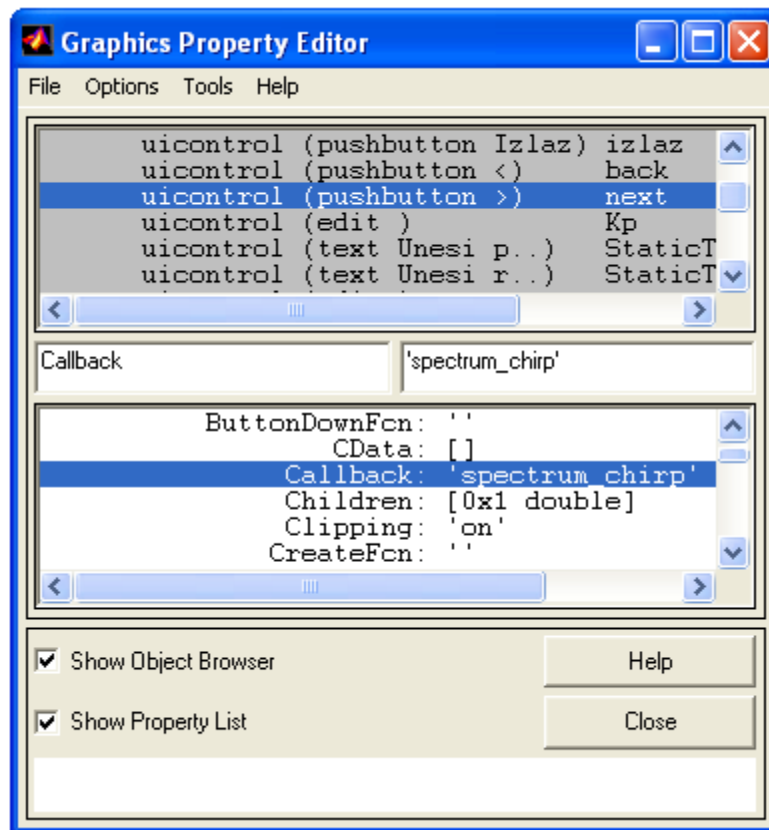
Prvo su unijeti svi potrebni elementi, a to su axes, za prikaz grafa, zatim okviri za unos podataka (EditText1, EditText2 i EditText3), pa okviri za prikaz teksta (StaticText1, StaticText2, StaticText3, StaticText4 i StaticText5) i tri gumba (Pushbutton1, Pushbutton2 i Pushbutton3). Korišten je Alignment Tool kako bi se svi elementi posložili i kako bi se dobio raspored koji se vidi na slici(15). Pomoću Graphics Property Editor postavljeni su parovi 'svojstva', vrijednosti elemenata. Svojstvo units, govori nam u kojim jedinicama se promatraju pozicije pojedinih elemenata unutar prozora i njihova veličina. Slika(16) prikazuje moguće vrijednosti svojstva unit.



Slika 16. Prozor za podešavanje svojstva units(jedinica veličine), odabrana je vrijednost normalized

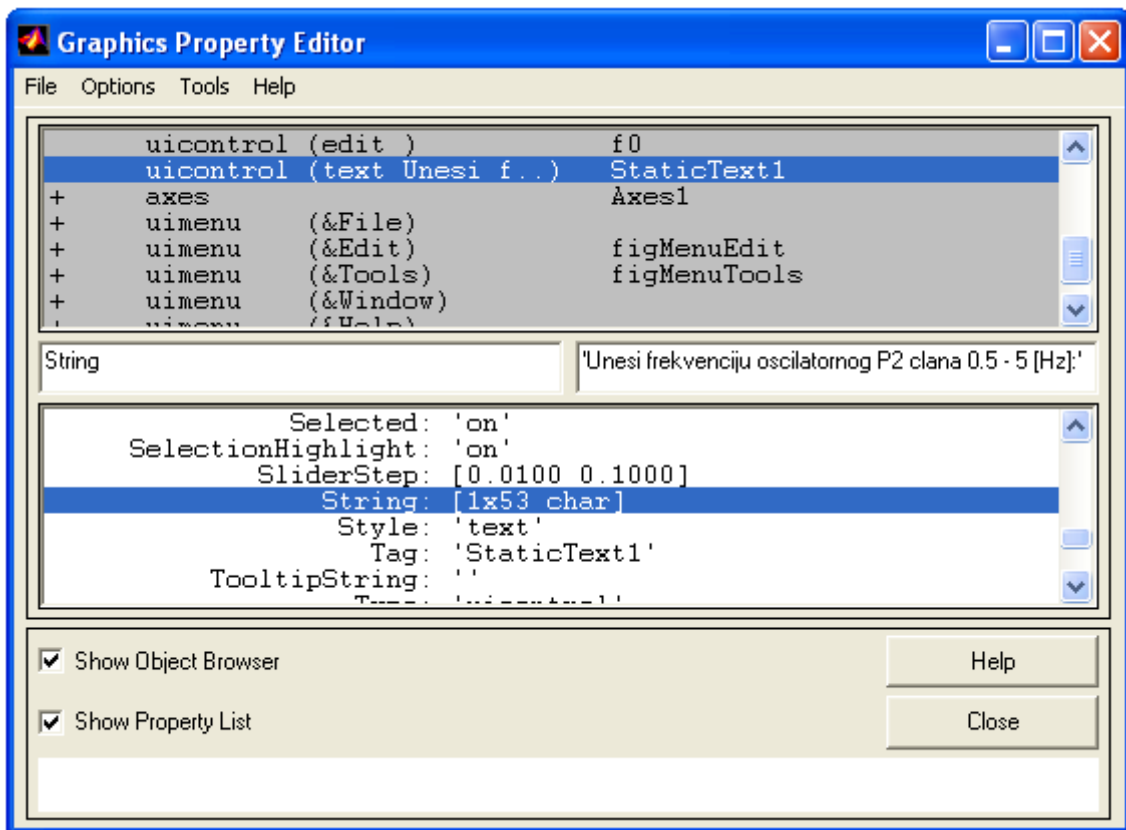
Odabrana je vrijednost normalized, kako bi se s promjenom veličine prozora, elementi uvijek zadržali na istom mjestu i kako bi se njihova veličina modificirala razmjerno veličini prozora. Veličina prozora i elemenata u njemu opisuje se svojstvom 'Position' [100 100 200 200], u ovom slučaju veličina se opisuje sa točkama ('units','points'), pa se donji lijevi ugao prozora nalazi na 100 točaka po x i y , a njegova širina i visina iznosi 200 točaka (Matlabu se ishodište nalazi u donjem lijevom uglu). Ako se promijeni rezolucija ekrana, prozor više neće biti na istom mjestu i njegova veličina neće biti ista. Ali ako se odabere normalized ('units','normalized'),a pozicija se zapiše na slijedeći način 'Position',[0.25 0.25 0.5 0.5]. Bez obzira na to koju rezoluciju imali donji lijevi ugao prozora će uvijek biti na četvrtini širine i visine, a širina i visina prozora će iznositi pola širine i visine ekrana. Isto vrijedi i za elemente unutar prozora, samo što se njihove vrijednosti gledaju u odnosu na prozor u kojem se nalaze, a ne na zaslon računala.





Slika 17. Prozor svojstva Callback(pozivna rutina)

Element pushbutton(gumb) , 'Callback','spectrum\_chirp'. Callback(pozivna rutina) kada se pritisne gumb unutar prozora, pozivna rutina poziva m-datoteku spectrum\_chirp. Unutar preglednika objekata (Object Browser) vidi se da piše uicontrol (pushbutton >) next . Znak '>' se nalazi na gumbu, on je unesen preko opcije ('String','>') ,a kako bi se lakše prepoznao gumb kasnije u kodu nazvan je next ('Tag','next'). Pritiskom na gumb next unutar prozora crtat će se sljedeći graf. Gumb back(uicontrol (pushbutton <) back) ima zadatak povratka na prethodni crtež. A gumb (uicontrol (pushbutton Izlaz) izlaz) zatvara prozor.



Slika 18. Svojstvo String(opisni tekst)

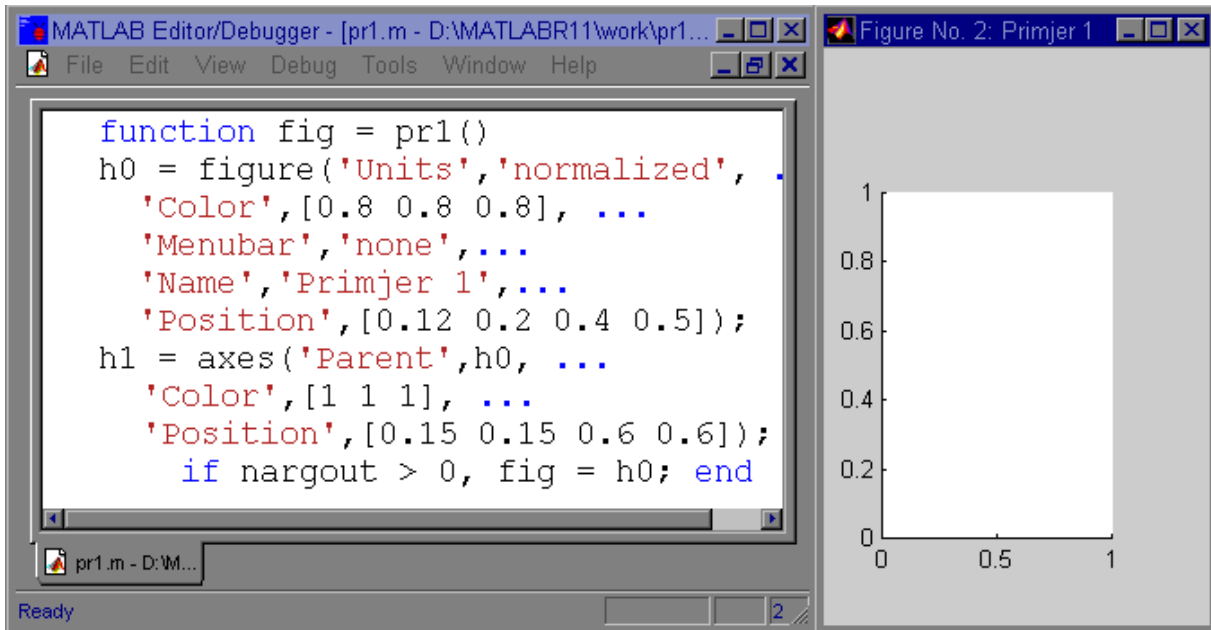
Ukupno postoji pet elemenata za unos teksta. Pomoću tih elemenata opisana su polja za unos podataka i dano je objašnjenje (uputa) o rasponu ulaznih parametara koji se unose preko GUI-a.

Nakon unosa svih elemenata i postavljanja odgovarajućih parova svojstvo-vrijednost, na alatnoj traci potrebno je odabrati izbornik "File" → "Save As" kako bi se cijeli prozor GUI sučelja spremio kao M-datoteka(s imenom SpChirp.m).

Pritom MATLAB automatski generira odgovarajući kod m-datoteke. Navedeni kod se nalazi u prilogu, zajedno s kodom spectrum\_chirp koje GUI-sučelje poziva.

### 3.5 Izrada korisničkog sučelja bez GUI alata

Korisničko sučelje se može izraditi u par redova unutar m skripte, jer automatskim generiranjem ispisuje se u m skriptu sav kod (vidi prilog, kod fig datoteke Sp\_Chirp), sve opcije, a u većini slučajeva dovoljno je par opcija. Glavne opcije su crtanje grafa, gumb i polje za unos vrijednosti. Te glavne opcije mogu se opisati sa par svojstava.



a)

b)

Slika 19. a) m-skripta, b) korisničko sučelje

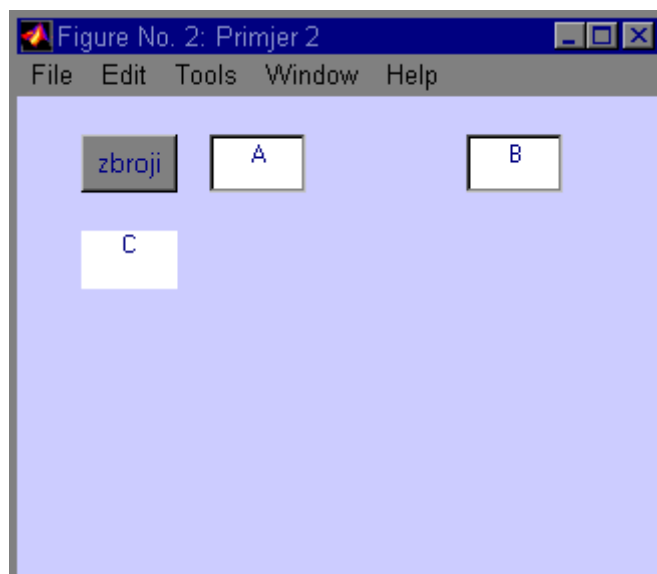
Sa samo par naredbi može se dobiti prozor u kojem su prikazane osi, ovaj jednostavni primjer nema nikakvu korisnu funkciju osim te da pokaže koje se naredbe koriste za pozivanje prozora s grafom.

Znači 'function fig = pr1()' je prva linija, pr1 je ime pod kojim spremamo m-skriptu i kada se u komadni prozor matlaba upiše 'pr1' pokrenut će se m-skripta 'pr1' i vidjet će se prozor s osima (sl.19.a).

Sljedećih pet linija dodjeljuje svojstva glavnom prozoru. Pozicija prozora ('Position') koja je dana u odnosu na veličinu ekrana jer je korišteno 'normalized' svojstvo ('units','normalized'), to je svojstvo koje postavlja početnu točku i širinu i visinu prozora u postocima veličine ekrana. Boja podloge prozora je dana preko kombinacija triju boja (RGB, 'Color'..). Ime prozora je dato preko 'Name' i preostaje traka s alatima, koja je isključena ('menubar','none').

Osi se nastavljaju na prozor, jer prozor je roditelj od osi. Definirana je boja podloge osi ('Color') i položaj osi unutar prozora ('Position'), položaj je definiran u postocima u odnosu na prozor, tako da kako se povećava veličina prozora mijenja se i veličina osi, ali položaj osi ostaje isti, to jest međuodnos osi i prozora je uvijek isti.

### 3.5.1 Korisničko sučelje poziva m skriptu (callback)



Slika 20. Primjer 2-pozivanje m-skripte preko funkcije callback

Dvije m skripte, pr2a i pr2b. Skripta pr2a iscrtava prozor na slici 11, korisnik unosi vrijednosti A i B, pritisne gumb zbroji i u polje c ispisuje se zbroj vrijednosti A i B. Kada se pritisne gumb skripta pr2a preko funkcije 'callback' poziva skriptu pr2b, skripta pr2b uzima vrijednosti A i B i ispisuje rješenje u polje C. Kod skripte pr2a dan je u prilogu, a kod skripte pr2b je naveden niže.

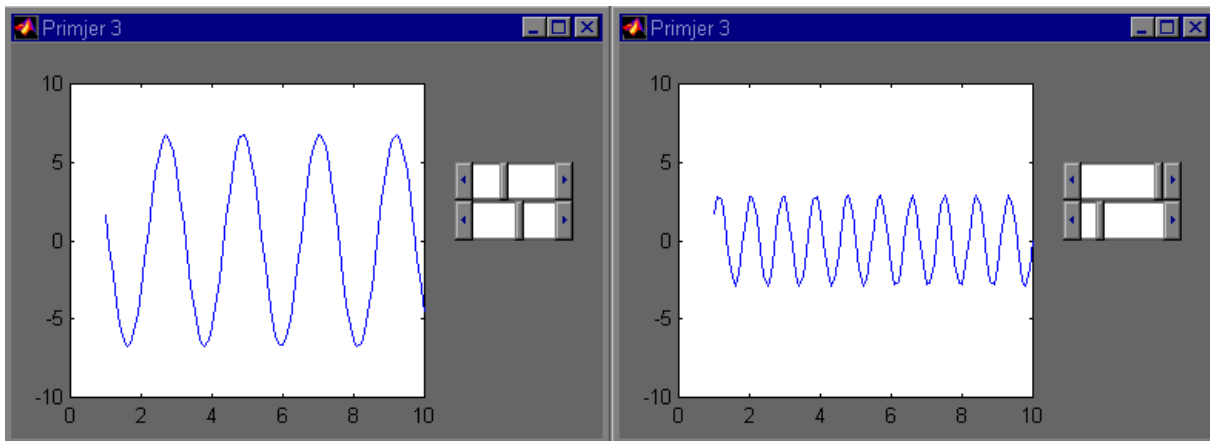
Kod iz skripte pr2b:

```
Clear
Clc
A=str2num(get(findobj('Tag','upisA'),'String'));
B=str2num(get(findobj('Tag','upisB'),'String'));
set(findobj('Tag','ispisC'),'String',A+B);
```

- Clear briše sve varijable iz radnog prostora,
- Clc briše sadržaj ekrana,
- 'str2num' pretvara string u broj,
- (get(findobj('Tag','upisA'),'String')) traži(findobj) unutar prozora objekt čija oznaka ('tag') je jednaka vrijednosti 'upisA', kada pronađe taj objekt vrijednost Stringa se pridružuje oznaci A,
- (get(findobj('Tag','upisB'),'String')) traži(findobj) unutar prozora objekt čija oznaka ('tag') je jednaka vrijednosti 'upisB', kada pronađe taj objekt vrijednost Stringa se pridružuje oznaci B,
- set(findobj('Tag','ispisC'),'String',A+B), traži se(findobj) objekt s oznakom 'ispisC', a zatim se pomoću naredbe 'set' postavlja vrijednost 'A+B' na mjesto stringa objekta 'ispisC'.

### 3.5.2 Izračun i crtanje grafa unutar GUI korisničkog sučelja

Struktura m-skripte je sljedeća : na početku je inicalizacija prozora, znači postavljaju se svojstva prozora, osi, gumbi, polja za unos teksta itd. Cijeli kod unutar m-skripte je pisan unutar jedne velike 'switch case' funkcije, tako da ovisno o tome što se aktivira unutar prozora 'switch' će prebaciti na neki od predviđenih, unaprijed unesenih slučajeva 'case'.



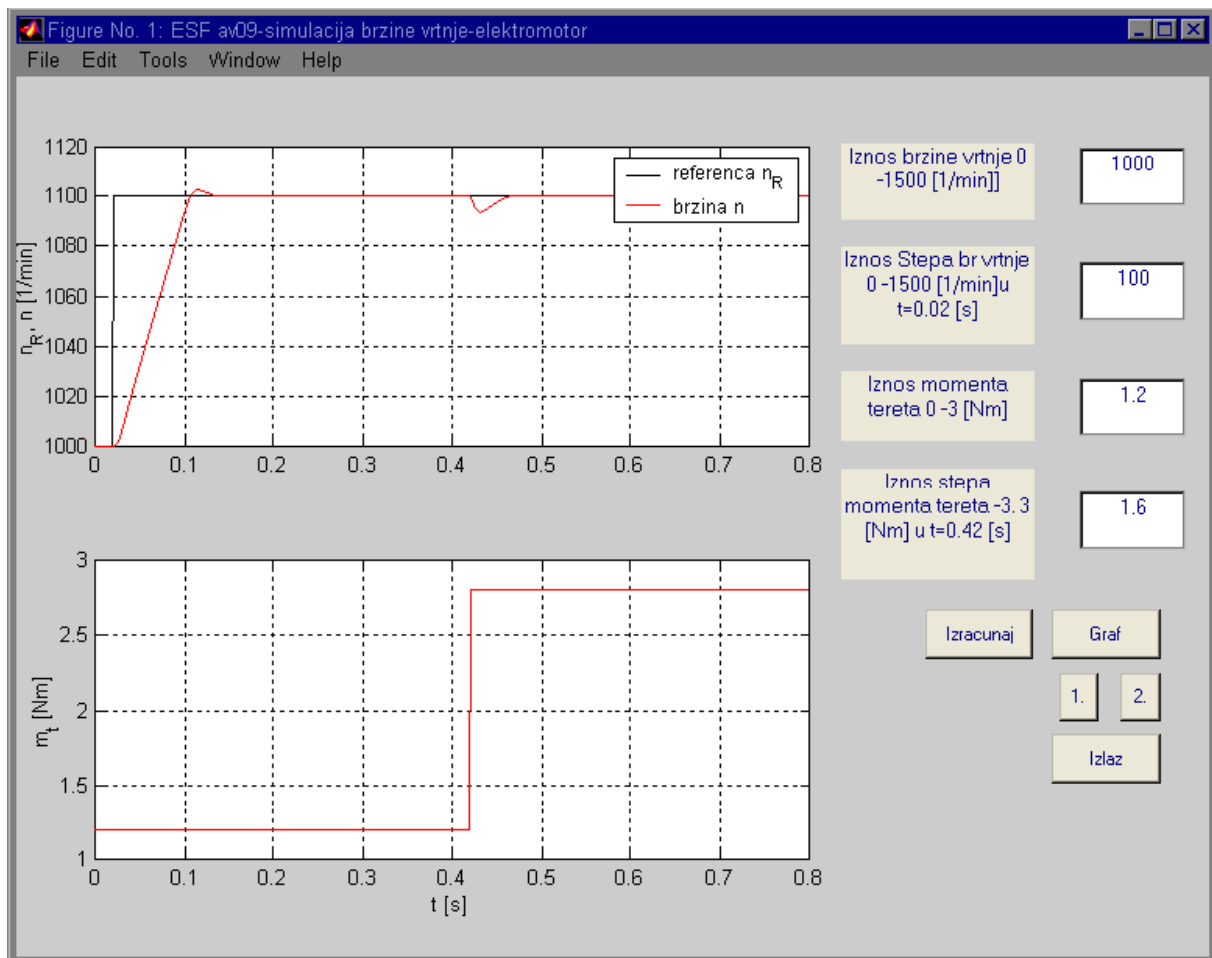
Slika 21 Primjer 3 Izrada korisničkog sučelja unutar m skripte pomoću switch case funkcije

Primjer 3 prikazuje jednostavno grafičko sučelje. Prozor ima dva klizača i graf, s jednim klizačem mijenjamo frekvenciju, a s drugim amplitudu sinusoide. Svaki pomak klizača automatski iscrtava novi graf. Kod je dodan u prilogu. Unutar koda može se vidjeti da kod definicije gumba klizač postoji funkcija `callback('Callback','pr3 klizac',...)`. Funkcija `callback` poziva ponovo istu skriptu (`pr3`), ali s tim da je varijabla akcija jednaka `klizac`, pa funkcija `switch` prebacuje na slučaj `klizac`, pa se uzimaju vrijednosti klizača, zatim se izračunava vrijednost frekvencije i amplitude i iscrtava se graf.

## 4. Prilagodba postojećih nastavničkih pomagala primjenom GUI-a

Prilagodba je napravljena preko Gui-a, povećana je preglednost nad unesenim varijablama, mogu se vidjeti unesene varijable, lako se može kretati između grafova i mogu se mijenjati parametri.

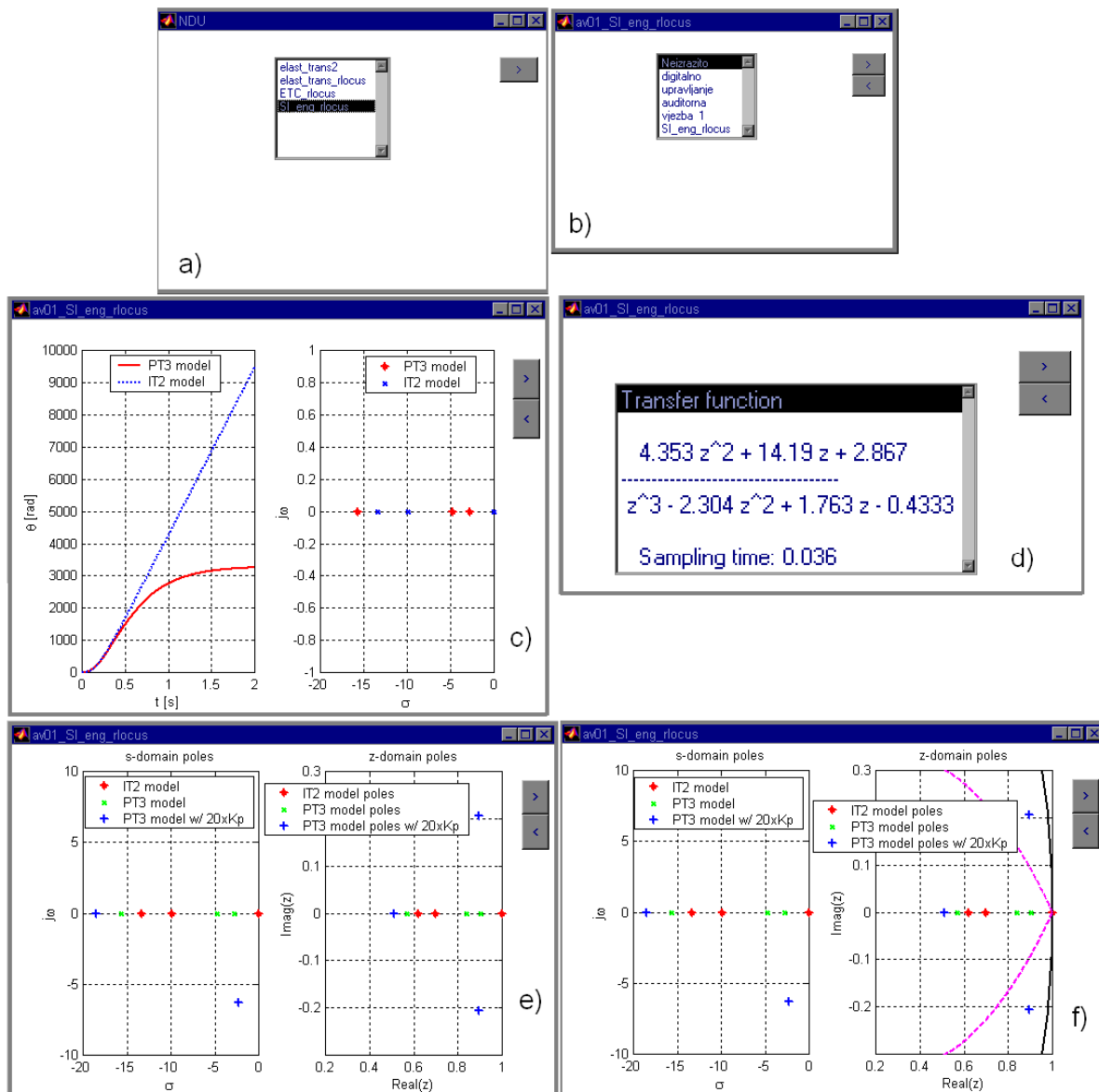
### 4.1. Primjer prilagodbi za slučaj vježbi iz elektromotornih servo pogona: regulacija brzine vrtnje



Slika 22. ESP-simulacija brzine vrtnje elektromotora (graf 1.)

Unos parametara simulacijskog modela provodi se primjenom korisničkog sučelja. Korisnik može vidjeti unesene varijable, pritiskom na gumb izračunaj(unutar skripte espav09 funkcija callback poziva skriptu init\_sim\_w) pokreće se m-skripta init\_sim\_w koja uzima unesene varijable(pomoću funkcije get i findobj), i izračunava podatke potrebne za crtanje grafa. Kada se pritisne tipka 'Graf' iscrtava se graf 1, pritiskom na tipku 1 ili 2 korisnik može vidjeti graf 1 ili 2. a pritiskom na tipku 'Izlaz', izlazi se iz aplikacije.

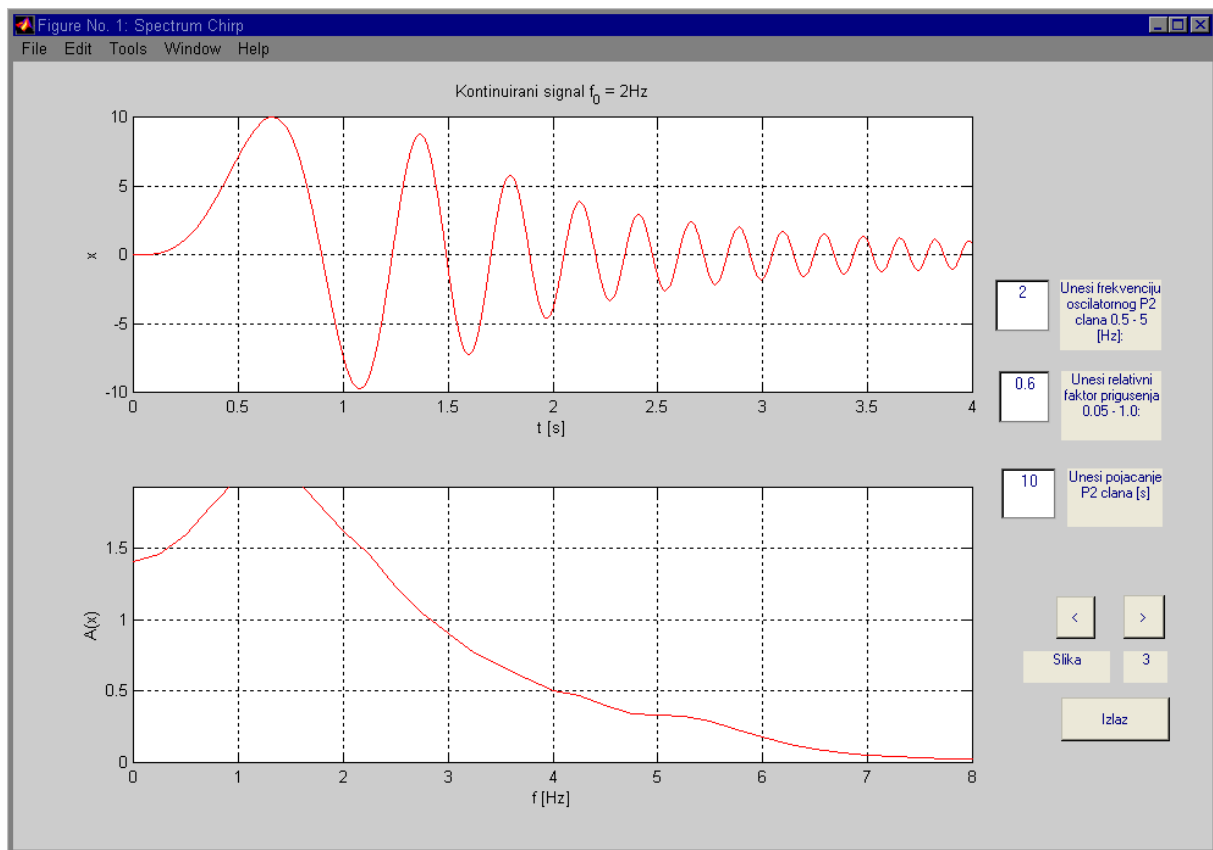
## 4.2. Primjer prilagodbi za slučaj vježbi iz neizrazitog digitalnog upravljanja: analiza vremenski diskretnih sustava



Slika 23. NDU- ETC-rlocus.m-prijenosna funkcija, Laplacova s i z domena

Slika 23. prikazuje korake od a do f koji se pokazuju na ekranu kada se klikne lijevom tipkom miša gumb '>'. Slika 23.a prikazuje početni izbornik gdje se odabire jedan od četiri ponuđena primjera. Slika 23.b daje opis (neizrazito digitalno upravljanje, vježba prva, primjer ETC-rlocus). Slika 23.c prikazuje odzive na skokovitu pobudu te položaje doninantnih i nedominantnih polova PT i IT modela zaklopke. Slika 23.d prikazuje jednadžbu prijenosne funkcije u z domeni s vremenom uzorkovanja od 0.036 sekundi. Slika 23.e prikazuje usporedno dva grafa, polove u s i z domeni. Slika 23.f prikazuje iste grafove kao slika 23.e, samo što ovdje imamo još dvije linije a to su linija granice dobre prigušenosti i linija jedinične kružnice, slabo prigušeni polovi nalaze se vrlo blizu linije jedinične kružnice (linije ruba stabilnosti).

### 4.3. Primjer prilagodbi za slučaj vježbi iz mikroprocesorskog upravljanja ; postupak frekvencijske analize



Slika 24. MPU-kontinuirani signali f0-2Hz

Preko korisničkog sučelja razvijenog pomoću matlab-ovog GUI alata, korisnik može vidjeti unesenu frekvenciju oscilatornog člana, relativni faktor prigušenja i pojačanje P2 člana i može se kretati između grafova jednostavnim klikom lijeve tipke miša na gumbе '<' i '>'. Na slici 15 se vidi kontinuirani signal frekvencije 2Hz, frekvenciju je odabrao korisnik, prvi graf prikazuje signal u vremenskoj, a drugi u frekvencijskoj domeni.



## 5. Zaključak

Primjena GUI korisničkog sučelja olakšava rad u slučaju kada se kombinira više različitih programskih okruženja u sklopu programskog paketa Matlab/Simulink. Naime, pri radu u komandnom prozoru Matlaba, osnovni je nedostatak nedovoljna preglednost i nepraktičnost unosa varijabli(parametara), naročito kada se na raj način inicijaliziranu Simulink blokovske sheme.

Kod programskog sučelja napravljenog pomoću GUI-a, dovoljno je promijeniti samo one varijable koje nas zanimaju i ponovo pokrenuti program.

Izrada korisničkog sučelja pomoću matlab-ovog GUI alata je jednostavna zbog toga što se željeni gumbi, grafovi, padajući izbornici lako unesu i postave na ekran, ali je kasnije nalaženje i promjena u m-skripti malo otežana jer kada matlab generira automatski kod on ubacuje puno podataka (parova svojstvo-vrijednost). Izrada korisničkog sučelja može se raditi i bez matlab-ovog GUI alata, unošenjem koda u m-skriptu, naravno treba poznavati osnovne parove svojstvo-vrijednosti objekta koji želimo postaviti unutar prozora, ali imamo na kraju veću preglednost unutar skripte.

Matlab-ov GUI alat je razmjerno jednostavan za korištenje i omogućuje visoku fleksibilnost u radu korisnicima koji ulažu nešto vremena za njegovo savladavanje.

## 6. Literatura

- [1] ... "using Matlab", MathWorks Inc,1999
- [2] ... "Building GUI's with Matlab",MathWorks Inc,1997
- [3] ... "Using Simulink",MathWorks Inc,1999
- [4] ... D. Pavković "Primjeri za auditorne vježbe iz Mikroprocesorskog upravljanja",FSB,2005
- [5]... D. Pavković "Primjeri za auditorne vježbe iz Elektromotornih servo-pogona,FSB,2005
- [6]... D. Pavković "Primjeri za auditorne vježbe iz Neizrazitog i digitalnog upravljanja,FSB,2007

## 7. Prilog

### Primjer 2a

```
function fig = pr2a()
h0 = figure('Units','normalized', ...
    'Color',[0.8 0.8 1], ...
    'Name','Primjer 2', ...
    'Position',[0.6 0.4 0.25 0.3]);

h1=icontrol('Parent',h0, ...
    'Callback','pr2b',...
    'FontSize',12,...
    'Units','normalized',...
    'Position',[0.1 0.8 0.15 0.12],...
    'String','zbroji',...
    'Tag','izracunaj',...
    'Style','Pushbutton');

h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontSize',0.3, ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.3 0.8 0.15 0.12], ...
    'String','A', ...
    'Style','edit', ...
    'Tag','upisA');

h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontSize',0.3, ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.7 0.8 0.15 0.12], ...
    'String','B', ...
    'Style','edit', ...
    'Tag','upisB');

h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontSize',0.3, ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.1 0.6 0.15 0.12], ...
    'String','C', ...
    'Style','text', ...
    'Tag','ispisC');
if nargin > 0, fig = h0; end
```

### Primjer 3

```
function pr3(action)

if nargin<1
    figure_h=figure;
    set(figure_h,'Units','normalized')
    set(figure_h,'UserData',figure_h);
    action='initialize';
end

switch action
case 'initialize';
    figure_h=get(gcf,'UserData');

    set(figure_h,...
        'Color',[0.4 0.4 0.4],...
        'Menubar','none',...
        'Name','Primjer 3',...
        'NumberTitle','off',...
        'Position',[0.12 0.6 0.3 0.32],...
        'Resize','on');

Slider_1 = uicontrol(figure_h,...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.75 0.5 0.2 0.1], ...
    'Style','Slider', ...
    'Tag','Slider_1', ...
    'Callback','pr3 klizac',...
    'Value',0);

Slider_2 = uicontrol(figure_h,...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.75 0.6 0.2 0.1], ...
    'Style','Slider', ...
    'Tag','Slider_2', ...
    'Callback','pr3 klizac',...
    'Value',0);

case 'klizac';
    Sl_1=get(findobj('tag','Slider_1'),'Value');
    Sl_2=get(findobj('tag','Slider_2'),'Value');

    t=1:0.1:10;
if Sl_2==0;
    f=1;
else
f=0.1+Sl_2; % frekvencija od 1 do 10 Hz..(Sl_2 -->0-1)
end
```

```

omega=2*pi*f;
A=1+((S1_1)*10); % amplituda od 1 do 10,1..
y=A*sin(omega*t);
x=t;

        cla;
        plot(x,y);ax_h1=gca;
        axis([0 10 -10 10])
set(ax_h1,'units','normalized')
set(ax_h1,'Position',[0.1 0.1 0.6 0.8])

otherwise
end

```

## Kod iz fig-datoteke spChirp

```

function fig = spChirp()
% This is the machine-generated representation of a Handle
Graphics object
% and its children. Note that handle values may change when
these objects
% are re-created. This may cause problems with any callbacks
written to
% depend on the value of the handle at the time the object was
saved.
% This problem is solved by saving the output as a FIG-file.
%
% To reopen this object, just type the name of the M-file at
the MATLAB
% prompt. The M-file and its associated MAT-file must be on
your path.
%
% NOTE: certain newer features in MATLAB may not have been
saved in this
% M-file due to limitations of this format, which has been
superseded by
% FIG-files. Figures which have been annotated using the plot
editor tools
% are incompatible with the M-file/MAT-file format, and should
be saved as
% FIG-files.
% Ovdje je predstavljen automatski izgeneriran kod
držala(Handle)
% nekog objekta i njegovog nasljednika. Primijetite da
vrijednost držala % objekta
% može biti promijenjena kada se objekti ponovo kreiraju. To
može % % % % prouzrokovati
% probleme s pozivima funkcija pisanih za točno određeni držač
funkcije u % vrijeme

```

```

% kada je objekt snimljen.
% Taj problem je riješen snimanjem izlaza kao fig-datoteke
%
% Kako bi ponovo otvorili isti objekt, samo utipakajte ime m
datoteke u % početni
% prozor Matlaba. M-datoteka i njezina MAT-datoteka moraju
biti u istom % % direktoriju.
%
% Bilješka: određene novije verzije u MATLABU možda neće biti
snimljene u % % M-datoteku
% zbog ograničenja formata, koji je potisnut od FIG-datoteka.
Prozori koji % su
% interpretirani korištenjem alata plot editora nisu
kompatibilni s M&MAT % formatom
% datoteka i trebali bi biti snimljeni kao FIG-datoteke

clear % briše sve varijable iz radnog prostora(eng. Workspace)
clc % briše sadržaj ekrana, odnosno command prompta
load spChirp % učitavanje MAT-datoteke spChirp
h0 = figure('Units','normalized', ... % postavljanje prozora,
normalizirana
% jedinica
'Color',[0.8 0.8 0.8], ... % postavljanje boje (tri
komponente R G B)
'Colormap',mat0, ... % mapa boja se učitava iz
MAT
% datoteke, mat0
'FileName','D:\MATLABR11\work\spChirp.m', ... % ime
datoteke, prikaz
%stabla
'Name','Spectrum Chirp', ... % ime prozora...
'PaperPosition',[18 180 576 432], ... % pozicija papira
'PaperUnits','points', ... %
'Position',[0.3796875 0.23125 0.60000000000000001 0.71], ...
% Pozicija
% prozora
'Tag','Fig1', ... % oznaka prozora je
Fig1
'ToolBar','none'); % traka s alatima je
% isključena

h1 = axes('Parent',h0, ... % osi, roditelj je
prozor h0
'FontUnits','normalized', ... % jedinica slova
je %
normalizirana
'CameraUpVector',[0 1 0], ... % vektor kamere
'CameraUpVectorMode','manual', ... % način rada
ručni

```

```

    'Color',[1 1 1], ...           % boja - bijela
    'ColorOrder',mat1, ...        % red boje se nalazi
u MAT-
    'FontSize',0.04535147392290249, ... % datoteci,mat1
slova                               % veličina
    'Position',mat2, ...         % pozicija se
očitava iz
    'Tag','Axes1', ...          % MAT-datoteke,mat2
    'UserData','[ ]', ...       % oznaka Axes1
su                                   % korisnički podaci
    'XColor',[0 0 0], ...       % prazni
    'YColor',[0 0 0], ...       %
    'ZColor',[0 0 0]);         %

h2 = text('Parent',h1, ...      % postavke za
tekst na
    'Color',[0 0 0], ...
    'HandleVisibility','off', ...
    'HorizontalAlignment','center', ...
    'Position',[0.4982817869415807 -0.06912442396313367
9.160254037844386], ...
    'Tag','Axes1Text4', ...
    'VerticalAlignment','cap');

set(get(h2,'Parent'),'XLabel',h2);
h2 = text('Parent',h1, ...
    'Color',[0 0 0], ...
    'HandleVisibility','off', ...
    'HorizontalAlignment','center', ...
    'Position',[-0.06701030927835053 0.4976958525345622
9.160254037844386], ...
    'Rotation',90, ...
    'Tag','Axes1Text3', ...
    'VerticalAlignment','baseline');
set(get(h2,'Parent'),'YLabel',h2);

h2 = text('Parent',h1, ...
    'Color',[0 0 0], ...
    'HandleVisibility','off', ...
    'HorizontalAlignment','right', ...
    'Position',mat3, ...
    'Tag','Axes1Text2', ...
    'Visible','off');
set(get(h2,'Parent'),'ZLabel',h2);

h2 = text('Parent',h1, ...
    'Color',[0 0 0], ...

```

```

    'HandleVisibility','off', ...
    'HorizontalAlignment','center', ...
    'Position',[0.4982817869415807 1.023041474654378
9.160254037844386], ...
    'Tag','Axes1Text1', ...
    'VerticalAlignment','bottom');
set(get(h2,'Parent'),'Title',h2);

h1 = uicontrol('Parent',h0, ...           % okvir za unos teksta
'
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'FontSize',0.2, ...
    'ListboxTop',0, ...
    'Position',[0.8723958333333333 0.6338028169014085
0.08463541666666666 0.0897887323943662], ...
    'String','Unesi frekvenciju oscilatornog P2 clana 0.5 - 5
[Hz]:', ...
    'Style','text', ...
    'Tag','StaticText1');

h1 = uicontrol('Parent',h0, ...           % okvir za unos brojevne
vrijednosti
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'FontSize',0.3809523809523809, ...
    'ListboxTop',0, ...
    'Position',[0.8190104166666666 0.6566901408450705
0.04557291666666666 0.06690140845070423], ...
    'Style','edit', ...
    'Tag','f0');% naziv, koji ću pozivati iz M-datoteke, kako
bih dobio

h1 = uicontrol('Parent',h0, ...           % vrijednost koju je unio
korisnik
    'Units','normalized', ...           % nema String objekta
    'FontUnits','normalized', ...       % jer sam ga ostavio
praznog
    'BackgroundColor',[1 1 1], ...     % string ću tražit iz m-
datoteke,a to
    'FontSize',0.3809523809523809, ... %je iznos koji je unio
korisnik

```



```

    'ListboxTop',0, ...
    'Position',[0.8216145833333333 0.5404929577464789 0.04296875
0.06690140845070423], ...
    'Style','edit', ...
    'Tag','zeta'); % druga vrijednost koju unosi korisnik

h1 = uicontrol('Parent',h0, ... % ovo je okvir za unos
teksta, opisujem
                                % zetu,to jest
    'Units','normalized', ...      % faktor prigušenja
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'FontSize',0.2, ...
    'ListboxTop',0, ...
    'Position',[0.8736979166666666 0.5193661971830986
0.08333333333333333 0.08802816901408452], ...
    'String','Unesi relativni faktor prigusenja 0.05 - 1.0:',
...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ... % još jedan okvir za unos
teksta, ovaj
                                % opisuje
    'Units','normalized', ...      % pojačanje P2 člana
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'FontSize',0.2, ...
    'ListboxTop',0, ...
    'Position',[0.87890625 0.4102112676056338 0.07942708333333333
0.07394366197183099], ...
    'String','Unesi pojačanje P2 člana [s]', ...
    'Style','text', ...
    'Tag','StaticText3');
h1 = uicontrol('Parent',h0, ... % okvir za unos pojačanja P2
člana
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'FontSize',0.3809523809523809, ...
    'ListboxTop',0, ...
    'Position',[0.82421875 0.4190140845070423
0.04557291666666666 0.06514084507042253], ...
    'Style','edit', ...
    'Tag','Kp');

h1 = uicontrol('Parent',h0, ... % ovo je okvir za gumb i to
je gumb koji poziva
    'Units','normalized', ...      % spectrum_chirp datoteku, i
prebavuje na sljedeći graf

```

```

    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'Callback','spectrum_chirp', ...
    'FontSize',0.3232323232323232, ...
    'ListboxTop',0, ...
    'Position',[0.9257003654080389 0.2673992673992674
0.0341047503045067 0.05494505494505494], ...
    'String','>', ...
    'Tag','next', ...
    'UserData','[ ]');
h1 = uicontrol('Parent',h0, ... % ovdje počinje okvir za gumb
koji vraća prethodni graf
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'Callback','spectrum_chirp', ...
    'FontSize',0.3232323232323232, ...
    'ListboxTop',0, ...
    'Position',[0.8695312500000001 0.2679558011049724 0.0328125
0.05386740331491713], ...
    'String','<', ...
    'Tag','back', ...
    'UserData','[ ]');
h1 = uicontrol('Parent',h0, ... % ovdje je gumb za izlaz
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'Callback','Close(gcf)', ...
    'FontSize',0.3232323232323232, ...
    'ListboxTop',0, ...
    'Position',mat4, ...
    'String','Izlaz', ...
    'Tag','izlaz', ...
    'UserData','[ ]');
h1 = uicontrol('Parent',h0, ... % ovo je okvir za tekst
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'FontSize',0.4637681159420289, ...
    'ListboxTop',0, ...
    'Position',[0.8424479166666666 0.2112676056338028
0.07161458333333333 0.04049295774647888], ...
    'String','Slika', ...
    'Style','text', ...
    'Tag','StaticText4');
h1 = uicontrol('Parent',h0, ... % ovo je okvir za tekst, a
vrijednost ovog stringa koji je u

```

```

    'Units','normalized', ... % početku nula, povećavat ću
                                za jedan, to mi je nešto poput
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ... %brojača
    'ListboxTop',0, ... % kako bih znao na
kojem sam trenutno grafu
    'Position',[0.92578125 0.2112676056338028
0.036458333333333333 0.04049295774647888], ...
    'String','0', ...
    'Style','text', ...
    'Tag','indeks');
if nargout > 0, fig = h0; end % ako su izlazni argumenti veći
od nule, fig=h0, kraj

```

### Kod iz M-datoteke spectrum\_chirp

```

% Proracunava spektar vremenski kontinuiranog i vremenski
diskretnog signala
% dobivenog otipkavanjem odziva P2 oscilatornog clana
% gco hvata držalo trenutnog objekta, objekta
koji je trenutno
gumb=get(gco,'Tag'); % aktiviran, , a get traži tag od tog
držala odnosno korisnik stisne
% gumb, a get(gco,'Tag') pridružuje varijabli
gumb ono što stoji uz
% tag stisnutog gumb to jest : back,next ili
izlaz
if strcmp(gumb,'next') % strcmp, string compare, uspoređuje
varijablu gumb(back,nextizlaz) i
    h_ind=findobj('type','uicontrol','tag','indeks'); % handle
od indeksa % 'next'
    i=str2num(get(h_ind,'String')); % findobj, vraća držalo
traženog objekta
    if i<7 % ovdje se provjerava koji
je trenutno graf nacrtan
        i=i+1; % svaki sljedeći graf povećava i
za jedan
    end % set (držalo,objekt,vrijednost
objekta)
    set(h_ind,'String',i) % postavljanje novog
broja, nove oznake
end

if strcmp(gumb,'back') % ako smo pritisnuli gumb back,
vraćamo se korak natrag
    h_ind=findobj('type','uicontrol','tag','indeks'); %
handle od indeksa

```

```

        i=str2num(get(h_ind,'String')); % str2num, pretvara
string u broj...
        if i>1
            i=i-1;
        end
        set(h_ind,'String',i)
end
%clear; close all; % brisanje svih varijabli ir radnog
prostora, zatvaranje otvorenih prozora
%warning off; echo off; % sve to neće se dogoditi jer je iza
komentara,
% ova m-datoteka je bila prije napravljena, napisao ju je
dr.sc. Danijel Pavković
% a ja sam stavio njegove dijelove koje ne koristim iza
komentara,
%fprintf(1,'\n\t Unesi frekvenciju oscilatornog P2 člana 0.5 -
5 [Hz]:\n\n'); % tako se ovdje
%fprintf(1,'\t\t');f0 = input(' f0 = ');fprintf(1,'\n'); %
prije unosio fo preko inputa s
f0=str2num(get(findobj('type','uicontrol','tag','f0'),'String'
)); % command prompta
if(isempty(f0) | f0 <= 0.0) % prvo findobj, pronađe držalo
od editTexta koji ima tag f0
    f0 = 1.0; % zatim get preko tog držala
uzima vrijednost koja je u Stringu
end % pa str2num, pretvara tu
vrijednost u broj
w0 = f0*2*pi; % [rad/s] % i nastaje varijabla f0
odnosno frekvencija ocilatornog člana

%fprintf(1,'\n\t Unesi relativni faktor prigusenja 0.05 -
1.0:\n\n');
%fprintf(1,'\t\t');zeta_ = input(' zeta_ = ');fprintf(1,'\n');
zeta_=str2num(get(findobj('type','uicontrol','tag','zeta'),'St
ring')); % ovdje se odvija dohvat
if(isempty(zeta_) | zeta_ <= 0.0) % ako korisnik ostavi
prazno polje, zeta će iznositi 0.3
    zeta_ = 0.3;
end

%fprintf(1,'\n\t Unesi pojačanje P2 člana [s]:\n\n');
%fprintf(1,'\t\t');Kp = input(' Kp = ');fprintf(1,'\n');
Kp=str2num(get(findobj('type','uicontrol','tag','Kp'),'String'
)); % pojačanje....
if(isempty(Kp) | Kp <= 0.0)
    Kp = 1.0;
end

% Brojnik i nazivnik prijenosne funkcije u s-podrucju
numc = Kp;
denc = [(1/w0^2) 2*zeta_/w0 1];

```

```

% Prijenosna funkcija - ispis na ekranu
tfs = tf(numc,denc)
%pause;

% Prijenosna funkcija - iscrtavanje frekvencijske
karakteristike
if i==1 % ako je pritisnut gumb next, i je
povećan za jedan i iznosi 1
    %figure(1), % iscrtava se prvi graf
    %hold on,grid on,
    bode(tfs);
    a=gca; % gca vraća držalo trenutnih osi i pridružuje ga
varijabli a
    grid on % uključivanje mreže na osima
    set(a,'Position',[0.04 0.05 0.76 0.85]) %
set(držalo,objekt,vrijednost), postavljanje pozicije
end % osi unutar prozora, kako
se ne bi prekrili gumbići
if i==2 % pa drugi graf itd....
    hold off % isključivanje mreže
    %figure(2),
    %hold on,grid on,
    step(tfs); % odziv na step od prijenosne funkcije
    grid on
end
% Prebacaj s obzirom na skokovitu pobudnu funkciju
sigma_m = exp(-zeta_*pi/sqrt(1-zeta_^2));

% Limit y-osi za vremenski odziv
ax1_lim = fix(10*(1+sigma_m)+0.5)/10;

% Rezonantno izdizanje
if(zeta_ < 0.7)
    Mr = Kp/(2*zeta_*sqrt(1-zeta_^2));
else
    Mr = 0.0;
end

% Osnovni simulacijski parametri
sim_end = 8/f0; fmax = 3.2*f0;

% Limit y-osi za frekvencijski spektar signala
ax2_limit = Mr/fmax + 0.3; % + 0.3 -> određeno empirijski

%pause;

% Simulacija kontinuiranog sustava u Simulink-u
step_size = 1e-3;

T = 0;
sim('chir_cont');

```

```

xs0 = xs;
t0 = t;

% Frekvencijski vektor za analizu spektra signala
NF = 2*fix(length(t0)/2);
fr0 = (1/step_size)*(0:NF-1)/NF;
if i==3
% Odziv na harmonicki test signal, tzv. "chirp" signal
%figure(3);
subplot(211),plot(t0,xs0,'r'),grid on,hold on
ax_h1=gca;
ylabel('x'),xlabel('t [s]') % ovdje
se koristi subplot(212) % znaći

crta se na dva grafa, pa
% Spektar odziva koristenjem FFT-a
% hvatam sa gca, držalo jednog % pa
ampls0 = abs(fft(xs0,NF))/(NF/2); % pa
drugog grafa, kakobi ih %
subplot(212),plot(fr0,ampls0,'r'),grid on,hold on %
mogao pozicionirati unutar %
ax_h2=gca; %
prozora i izbjegao preklapanje %
ylabel('A(x)'),xlabel('f [Hz]') %
gumba i grafa i okvira teksta
subplot(211),title(['Kontinuirani signal f_0 = ' num2str(f0)
'Hz'])
subplot(212),axis([0 4*fix(f0+0.5) 0 ax2_limit])

set(ax_h1,'Position',[0.1 0.58 0.7 0.35]) % korigiranje prvog
grafa
set(ax_h2,'Position',[0.1 0.11 0.7 0.35]) % korigiranje drugog
grafa
%pause;
end
% Simulacija kontinuiranog sustava u Simulink-u uz otipkavanje
(T = 0.1/f0)
T = 0.1/f0; T1 = T;
sim('chir_cont');

xs1 = xs;
t1 = t;

% Frekvencijski vektor za analizu spektra signala
NF = 2*fix(length(t1)/2);
fr1 = (1/T)*(0:NF-1)/NF;
if i==4
%figure(3);
subplot(211),plot(t1,xs1,'gp'),grid on,hold on
ax_h1=gca;

```

```

% Spektar odziva koristenjem FFT-a
ampls1 = abs(fft(xs1,NF))/(NF/2);

subplot(212),plot(fr1,ampls1,'g'),grid on,hold on
ax_h2=gca;
subplot(211),title(['Zeleno - otipkani signal T = ' num2str(T)
's' ' ' ' (f_s_h = ' num2str(0.5/T1) 'Hz)'])
subplot(212),axis([0 4*fix(f0+0.5) 0 ax2_limit])
set(ax_h1,'Position',[0.1 0.58 0.7 0.35])
set(ax_h2,'Position',[0.1 0.11 0.7 0.35])
%pause;
end
%figure(3);clf;

% Simulacija kontinuiranog sustava u Simulink-u uz
otipkavanje(T = 0.25/f0)
T = 0.25/f0; T2 = T;
sim('chir_cont');

xs2 = xs;
t2 = t;

% Frekvencijski vektor za analizu spektra signala
NF = 2*fix(length(t2)/2);
fr2 = (1/T)*(0:NF-1)/NF;
if i==5
%figure(3);
subplot(211),plot(t0,xs0,'r'),grid on,hold on
ylabel('x'),xlabel('t [s]')
subplot(211),plot(t2,xs2,'bd'),grid on,hold on
ax_h1=gca;
% Spektar odziva koristenjem FFT-a
ampls2 = abs(fft(xs2,NF))/(NF/2);

subplot(212),plot(fr0,ampls0,'r',fr2,ampls2,'b'),grid on,hold
on
ax_h2=gca;
ylabel('A(x)'),xlabel('f [Hz]')
subplot(211),title(['Plavo - otipkani signal T = ' num2str(T)
's' ' ' ' (f_s_h = ' num2str(0.5/T2) 'Hz)'])
subplot(212),axis([0 4*fix(f0+0.5) 0 ax2_limit])
set(ax_h1,'Position',[0.1 0.58 0.7 0.35])
set(ax_h2,'Position',[0.1 0.11 0.7 0.35])
%pause;
end
%figure(3),clf;

% Simulacija kontinuiranog sustava u Simulink-u uz otipkavanje
(T = 0.7/f0)
T = 0.65/f0; T3 = T;
sim('chir_cont');

```

```

xs3 = xs;
t3 = t;
% Frekvencijski vektor za analizu spektra signala
NF = 2*fix(length(t3)/2);
fr3 = (1/T)*(0:NF-1)/NF;
if i==6
%figure(3);
subplot(211),plot(t0,xs0,'r'),grid on,hold on
ax_h1=gca;
ylabel('x'),xlabel('t [s]')
subplot(211),plot(t3,xs3,'ko'),grid on,hold on

% Spektar odziva koristenjem FFT-a
ampls3 = abs(fft(xs3,NF))/(NF/2);

subplot(212),plot(fr0,ampls0,'r',fr3,ampls3,'k'),grid on,hold
on
ax_h2=gca;
ylabel('A(x)'),xlabel('f [Hz]')
subplot(212),axis([0 4*fix(f0+0.5) 0 ax2_limit])
subplot(211),title(['Crno - otipkani signal T = ' num2str(T)
's' ' ' ' (f_s_h = ' num2str(0.5/T3) 'Hz)'])
set(ax_h1,'Position',[0.1 0.58 0.7 0.35])
set(ax_h2,'Position',[0.1 0.11 0.7 0.35])
%pause;
end
%figure(3),clf;
if i==7
subplot(211),title('') %clear title
subplot(211),plot(t0,xs0,'r',t1,xs1,'gp',t2,xs2,'bd',t3,xs3,'k
o'),grid on,hold on
ax_h1=gca;
ylabel('x'),xlabel('t [s]')
subplot(212),plot(fr0,ampls0,'r',fr1,ampls1,'g:p',fr2,ampls2,'
b--',fr3,ampls3,'k:o'),grid on,hold on
ax_h2=gca;
subplot(212),axis([0 4*fix(f0+0.5) 0 ax2_limit])
ylabel('A(x)'),xlabel('f [Hz]')

subplot(211),legend(['Kont. signal f_0 = ' num2str(f0)
'Hz'],['f_s_h = ' num2str(0.5/T1) 'Hz'],['f_s_h = '
num2str(0.5/T2) 'Hz'],['f_s_h = ' num2str(0.5/T3) 'Hz'])
subplot(212),legend(['Kont. signal f_0 = ' num2str(f0)
'Hz'],['f_s_h = ' num2str(0.5/T1) 'Hz'],['f_s_h = '
num2str(0.5/T2) 'Hz'],['f_s_h = ' num2str(0.5/T3) 'Hz'])
set(ax_h1,'Position',[0.1 0.58 0.7 0.35])
set(ax_h2,'Position',[0.1 0.11 0.7 0.35])
end

```



## Kod iz fig datoteke ESPav09

```
function fig = ESPav09()
% This is the machine-generated representation of a Handle
Graphics object
% and its children. Note that handle values may change when
these objects
% are re-created. This may cause problems with any callbacks
written to
% depend on the value of the handle at the time the object was
saved.
% This problem is solved by saving the output as a FIG-file.
%
% To reopen this object, just type the name of the M-file at
the MATLAB
% prompt. The M-file and its associated MAT-file must be on
your path.
%
% NOTE: certain newer features in MATLAB may not have been
saved in this
% M-file due to limitations of this format, which has been
superseded by
% FIG-files. Figures which have been annotated using the plot
editor tools
% are incompatible with the M-file/MAT-file format, and should
be saved as
% FIG-files.
clear;
load ESPav09

h0 = figure('Units','normalized', ...
    'Name','ESP av09-simulacija brzine vrtnje-elektromotor',...
    'Color',[0.8 0.8 0.8], ...
    'Colormap',mat0, ...
    'FileName','D:\MATLABR11\work\ESPav09.m', ...
    'PaperPosition',[18 180 576 432], ...
    'PaperUnits','points', ...
    'Position',[0.12 0.18875 0.6 0.71], ...
    'Tag','Fig1', ...
    'ToolBar','none');
h1 = axes('Parent',h0, ...
    'FontUnits','normalized', ...
    'CameraUpVector',[0 1 0], ...
    'Color',[1 1 1], ...
    'ColorOrder',mat1, ...
    'FontSize',0.04535147392290249, ...
    'Position',mat2, ...
    'Tag','Axes1', ...
    'XColor',[0 0 0], ...
```

```

        'YColor',[0 0 0], ...
        'ZColor',[0 0 0]);
h2 = text('Parent',h1, ...
        'Color',[0 0 0], ...
        'HandleVisibility','off', ...
        'HorizontalAlignment','center', ...
        'Position',[0.499009900990099 -0.07349081364829391
9.160254037844386], ...
        'Tag','Axes1Text4', ...
        'VerticalAlignment','cap');
set(get(h2,'Parent'),'XLabel',h2);
h2 = text('Parent',h1, ...
        'Color',[0 0 0], ...
        'HandleVisibility','off', ...
        'HorizontalAlignment','center', ...
        'Position',[-0.0712871287128713 0.4960629921259844
9.160254037844386], ...
        'Rotation',90, ...
        'Tag','Axes1Text3', ...
        'VerticalAlignment','baseline');
set(get(h2,'Parent'),'YLabel',h2);
h2 = text('Parent',h1, ...
        'Color',[0 0 0], ...
        'HandleVisibility','off', ...
        'HorizontalAlignment','right', ...
        'Position',mat3, ...
        'Tag','Axes1Text2', ...
        'Visible','off');
set(get(h2,'Parent'),'ZLabel',h2);
h2 = text('Parent',h1, ...
        'Color',[0 0 0], ...
        'HandleVisibility','off', ...
        'HorizontalAlignment','center', ...
        'Position',mat4, ...
        'Tag','Axes1Text1', ...
        'VerticalAlignment','bottom');
set(get(h2,'Parent'),'Title',h2);
h1 = uicontrol('Parent',h0, ...
        'Units','normalized', ...
        'FontUnits','normalized', ...
        'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
        'FontSize',0.2882882882882883, ...
        'ListboxTop',0, ...
        'Position',[0.6918392204628502 0.836996336996337
0.1619975639464068 0.08791208791208791], ...
        'String','Iznos brzine vrtnje 0 -1500 [1/min]]', ...
        'Style','text', ...
        'Tag','StaticText1');
h1 = uicontrol('Parent',h0, ...
        'Units','normalized', ...

```

```

    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'FontSize',0.3809523809523809, ...
    'ListboxTop',0, ...
    'Position',[0.8915956151035323 0.8516483516483516
0.08891595615103531 0.06776556776556776], ...
    'Style','edit', ...
    'Tag','EditText1');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'FontSize',0.3809523809523809, ...
    'ListboxTop',0, ...
    'Position',[0.8915956151035323 0.7216117216117216
0.08891595615103531 0.06776556776556776], ...
    'Style','edit', ...
    'Tag','EditText2');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'FontSize',0.2318840579710145, ...
    'ListboxTop',0, ...
    'Position',[0.6918392204628502 0.6959706959706959
0.1619975639464068 0.1098901098901099], ...
    'String','Iznos Stepa br vrtnje 0 -1500 [1/min]u t=0.02
[s]', ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'FontSize',0.3232323232323232, ...
    'ListboxTop',0, ...
    'Position',[0.6918392204628502 0.5860805860805861
0.1619975639464068 0.07875457875457875], ...
    'String','Iznos momenta tereta 0 -3 [Nm]', ...
    'Style','text', ...
    'Tag','StaticText3');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'FontSize',0.3809523809523809, ...
    'ListboxTop',0, ...
    'Position',[0.8915956151035323 0.5915750915750916
0.08891595615103531 0.06593406593406594], ...

```

```

        'Style','edit', ...
        'Tag','EditText3');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'FontSize',0.2012578616352201, ...
    'ListboxTop',0, ...
    'Position',[0.6918392204628502 0.4285714285714286
0.1619975639464068 0.1263736263736264], ...
    'String','Iznos stepa momenta tereta -3..3 [Nm] u t=0.42
[s]', ...
    'Style','text', ...
    'Tag','StaticText4');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'FontSize',0.3809523809523809, ...
    'ListboxTop',0, ...
    'Position',[0.8915956151035323 0.4615384615384616
0.08891595615103531 0.06593406593406594], ...
    'Style','edit', ...
    'Tag','EditText4', ...
    'UserData','[ ]');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'Callback','init_sim_w', ...
    'FontSize',0.3232323232323232, ...
    'ListboxTop',0, ...
    'Position',[0.7612667478684531 0.3388278388278389
0.09135200974421437 0.05494505494505494], ...
    'String','Izracunaj', ...
    'Tag','start');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'Callback','init_sim_w', ...
    'FontSize',0.3232323232323232, ...
    'ListboxTop',0, ...
    'Position',[0.8684531059683313 0.3388278388278389
0.09135200974421437 0.05494505494505494], ...
    'String','Graf', ...
    'Tag','graf', ...
    'UserData','[ ]');

```

```

h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'Callback','init_sim_w', ...
    'FontSize',0.3232323232323232, ...
    'ListboxTop',0, ...
    'Position',[0.9257003654080389 0.2673992673992674
0.0341047503045067 0.05494505494505494], ...
    'String','2.', ...
    'Tag','next', ...
    'UserData','[ ]');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'Callback','init_sim_w', ...
    'FontSize',0.3232323232323232, ...
    'ListboxTop',0, ...
    'Position',[0.8745432399512789 0.2673992673992674
0.03288672350791717 0.05494505494505494], ...
    'String','1.', ...
    'Tag','back', ...
    'UserData','[ ]');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.925490196078431 0.913725490196078
0.847058823529412], ...
    'Callback','init_sim_w', ...
    'FontSize',0.3232323232323232, ...
    'ListboxTop',0, ...
    'Position',[0.8684531059683313 0.1978021978021978
0.09135200974421437 0.05494505494505494], ...
    'String','Izlaz', ...
    'Tag','izlaz', ...
    'UserData','[ ]');
if nargin > 0, fig = h0; end

```

## Kod iz m datoteke init\_sim.m

```
% Podaci motora:

P = 500; % [W]
U = 220; % [V]
n = 1500; % [1/min]
IA = 3.4; % [A]
RA = 16.35; % [Ohm]
KA = 1/RA; % [1/Ohm]
Ta = 18.3e-3; % [s]
LA = Ta*RA; % [H]
J = 0.0157; % [kg*m^2]
om = n*pi/30; % [rad/s]
M = P/om; % [Nm]
Km = M/IA; % [Nm/A]
Ke = (U-IA*RA)/om; % [Vs/rad]

% Parametri tranzistorskog pretvaraca

fch = 4e3; % [Hz]
Kch = 45; % [V/V]
Tch = 1/fch; % [s]

% Parametri filtra signala struje

Ki = 1.57; % [V/A]
Ti = 0.75e-3; % [s]

% Parametri regulatora struje
% Karakteristicni odnos optimuma dv. odnosa
D2i = 0.5;
Tsigi = Ti + Tch;

Tci = Ta;
Kci = Tci/Tsigi*D2i/Kch/KA/Ki;
Tei = Tci/(Kci*Kch*KA*Ki);

% Filter mjernog signala brzine vrtnje
Kw = 0.065; % [Vs/rad]
Tw = 2e-3; % [s]

% Proracun parametara regulatora brzine vrtnje
% Karakteristicni odnosi optimuma dvostrukog odnosa
D2w = 0.5; D3w = 0.5;
Tsigw = Tei + Tw;
Tcw = Tsigw/D2w/D3w;
Tew = Tcw;
Kcw = 1/(D2w*Tcw)*Ki*J/Kw/Km;

% Limit napona choppera
```

```

Uch_limit = U;

% Limit struje
IAR_limit = IA*Ki;

% Pocetni uvjeti
% *****

gumb=get(gco, 'Tag');

%fprintf(1, '\n\t Unesi pocetni iznos brzine vrtnje 0 ... 1500
[1/min]:\n\n');
%fprintf(1, '\t\t'); n0 = input(' n = '); fprintf(1, '\n');
%*****
if strcmp(gumb, 'start') % ako je start pritisnut
****++++
n0=str2num(get(findobj('Type','uicontrol','Tag','EditText1'),'
String')); %$$$
if isempty(n0)
n0 = 0.0;
end
w0 = n0*pi/30; % [rad/s]

%fprintf(1, '\n\t Unesi iznos skokovite promjene brzine vrtnje
0 ... 1500 [1/min] u t = 0.02 [s]:\n\n');
%fprintf(1, '\t\t'); Dn = input(' Step = '); fprintf(1, '\n');
Dn=str2num(get(findobj('Type','uicontrol','Tag','EditText2'),'
String')); %$$$
if isempty(Dn)
Dn = 100.0;
end

% U slucaju da zeljena promjena brzine vrtnje izlazi izvan
nazivnog raspona brzina motora ...
if(abs(n0+Dn) > 1500)
Dn = (1500-abs(n0))*sign(Dn);
%fprintf(1, '\n\t Prekoracenje raspona promjene brzine
vrtnje!\n');
msgbox('Prekoracenje raspona promjene brzine vrtnje!')
%fprintf(1, '\t Novi iznos skokovite promjene reference
brzine vrtnje = %f [1/min]\n', Dn);
msgbox('Novi iznos skokovite promjene reference brzine
vrtnje je...')
Dn_novi=num2str(Dn);
msgbox(Dn_novi)
end

DwR = Dn*pi/30; % [rad/s]

%fprintf(1, '\n\t Unesi pocetni iznos momenta tereta 0 ... 3
[Nm]:\n\n');

```

```

%fprintf(1,'\t\t');Mt = input(' Mt = ');fprintf(1,'\n');
Mt=str2num(get(findobj('Type','uicontrol','Tag','EditText3'),'String')); %$$$$
if isempty(Mt)
    Mt = 0.0;
end

%fprintf(1,'\n\t Unesi iznos skokovite promjene momenta tereta
-3 ... +3 [Nm] u t = 0.42 [s]:\n\n');
%fprintf(1,'\t\t');DMt = input(' Step = ');fprintf(1,'\n');
DMt=str2num(get(findobj('Type','uicontrol','Tag','EditText4'),'String')); %$$$$
if isempty(DMt)
    DMt = 1.0;
end

% U slucaju da zeljena promjena momenta tereta izlazi izvan
nazivnog raspona momenata motora ...
if(abs(Mt+DMt) > 3.0)
    DMt = (3.0-abs(Mt))*sign(DMt);
    msgbox('Prekoracenje raspona promjene momenta tereta!')
    %fprintf(1,'\n\t Prekoracenje raspona promjene momenta
tereta!\n');
    %fprintf(1,'\t Novi iznos skokovite promjene momenta tereta
= %f [Nm]\n',DMt);
    msgbox('Novi iznos skokovite promjene momenta tereta
je...')
    DMt_novi=num2str(DMt);
    msgbox(DMt_novi)
end

%fprintf(1,'\n\t Kompenzacija utjecaja elektromotorne sile u
reg. struje? 1 -> DA, 0 -> NE \n\n');
izbor=menu('kompenzacija utjecaja elektromotorne sile u
reg.struje','DA','NE'); % da je 1, ne je 2
%fprintf(1,'\t\t');enc = input(' Kompenzacija =
');fprintf(1,'\n');
if izbor==1
    enc=1;
else
    enc=0;
end

if isempty(enc)
    enc = 0.0;
end

end % end od ako je start pritisnut ****

if strcmp(gumb,'graf')

```



```

% Skokovita promjena reference brzine vrtnje nastupa u ...
tw_step = 0.02;
% Skokovita promjena momenta tereta nastupa u ...
tm_step = 0.42;
% Kraj simulacije
t_end = 0.8;

% Pojacanje za kompenzaciju EMS
Kcomp = enc*Ke/Kch/Kw;

% U stacionarnom stanju danom momentu tereta Mt odgovara napon
armature:
Uch0 = Ke*w0 + Mt/Km/KA;

% ... a koji se postize uz ulazni napon pretvaraca.
Uc0 = Uch0/Kch - Kcomp*w0*Kw;

% => pocetno stanje integratora u PI regulatoru struje
Ui0 = Uc0;

% Pocetni iznos reference struje -> pocetno stanje intergatora
u PI regulatoru brzine
IAR0 = Mt/Km*Ki; % [V]

% Pocetni iznos struje armature
IA0 = Mt/Km; % [A]

% Pokreni simulaciju
sim('reg_brzine');

%fprintf(1,'\n\t Signali:');
%fprintf(1,'\n\t ----- \n');
%fprintf(1,'\n\t i_AR -> referenca struje [V]');
%fprintf(1,'\n\t i_A -> mjerni signal struje [V]');
%fprintf(1,'\n\t u_AR -> izlaz PI regulatora struje');
%fprintf(1,'\n\t u_A -> sr. napon na izlazu pretvaraca');
%fprintf(1,'\n\t brzina -> brzina vrtnje motora');
%fprintf(1,'\n\t brzina_ref -> referenca brzine vrtnje
motora');
%fprintf(1,'\n\t mt -> moment tereta');
%fprintf(1,'\n\t t -> vrijeme');
%fprintf(1,'\n\n');

%figure(1),
subplot(211),plot(t,brzina_ref,'k',t,brzina,'r'),grid on
ax1_h=gca;
set(ax1_h,'Position',[0.065 0.58 0.6 0.34])
ylabel('n_R, n [1/min]'),legend('referenca n_R','brzina n')
subplot(212),plot(t,mt,'r'),grid on

```

```

ax2_h=gca;
set(ax2_h, 'Position', [0.065 0.11 0.6 0.34])
ylabel('m_t [Nm]'), xlabel('t [s]')

end % od ako je pritisnut graf

if strcmp(gumb, 'next')
%figure(2),
subplot(211), plot(t, i_AR, 'k', t, i_A, 'r'), grid on
ax1_h=gca;
set(ax1_h, 'Position', [0.065 0.58 0.6 0.34])
ylabel('i_A_R, i_A [V]'), legend('Referenca struje
i_A_R', 'Mjerni signal struje i_A')
subplot(212), plot(t, u_AR, 'r'), grid on
ax2_h=gca;
set(ax2_h, 'Position', [0.065 0.11 0.6 0.34])
ylabel('u_A_R [V]'), xlabel('t [s]')

end

if strcmp(gumb, 'back')
%figure 1
subplot(211), plot(t, brzina_ref, 'k', t, brzina, 'r'), grid on
ax1_h=gca;
set(ax1_h, 'Position', [0.065 0.58 0.6 0.34])
ylabel('n_R, n [1/min]'), legend('referenca n_R', 'brzina n')
subplot(212), plot(t, mt, 'r'), grid on
%XMIN_b=0; XMAX_b=0.8;
%if mt(1,1)<0
% YMIN_b=mt(1,1)-0.5;
%else
% YMIN_b=mt(1,1)+0.5;
%end
%if mt(8000,1)<0
% YMAX_b=mt(1,1)-0.5;
%else
% YMAX_b=mt(1,1)+0.5;
%end
%axis([XMIN_b XMAX_b YMIN_b YMAX_b])
ax2_h=gca;
set(ax2_h, 'Position', [0.065 0.11 0.6 0.34])
ylabel('m_t [Nm]'), xlabel('t [s]')
end

if strcmp(gumb, 'izlaz')
cd ..
close(gcf)
end

```