

ELEKTROTEHNIČKI FAKULTET OSIJEK		Električni strojevi
Vježba br. 3:	UVOD U MATLAB	
Student:		
Grupa:		

1.1. CILJ VJEŽBE

Cilj je ove vježbe upoznavanje s osnovnim operacijama programskog sustava Matlab. Te operacije podrazumijevaju definiranje varijabli (skalara, vektora , matrica) te aritmetičke operacije nad tim varijablama. Upoznat ćete se s naredbama ponavljanja, kao i osnovama grafičkog prikazivanja rezultata.

1.2. Što je MATLAB ?

MATLAB je multifunkcionalni programski sustav koji u jednom softverskom paketu i na jednom mjestu ujedinjuje funkcionalitete:

- numeričnih,
- simboličnih i
- grafičkih sustava.

Matlab je okružje i programski jezik. U srcu Matlaba nalazi se pojam matrice, o čemu govori i samo ime Matlab koje potječe od engleskih riječi MATrix LABoratory. Matrica je jednostavan matematički objekt, pravokutna tablica brojeva, koja se prirodno javlja u najrazličitijim područjima i situacijama, dok jezgru Matlaba čini skup funkcija za jednostavno, prirodno i efikasno manipuliranje matricama. Upravo iz tog razloga Matlab se sve više širi i u specijalizirana područja, o čemu govore brojni novi *toolbox-i*. Uobičajena je uporaba Matlab za

- matematiku i izračune,
- razvoj algoritama,
- modeliranje, simulaciju, analizu,
- analizu i obradu podataka, vizualizaciju,
- znanstvenu i inženjersku grafiku,
- razvoj aplikacija, uključujući i izgradnju GUI

Neke od prednosti Matlab su:

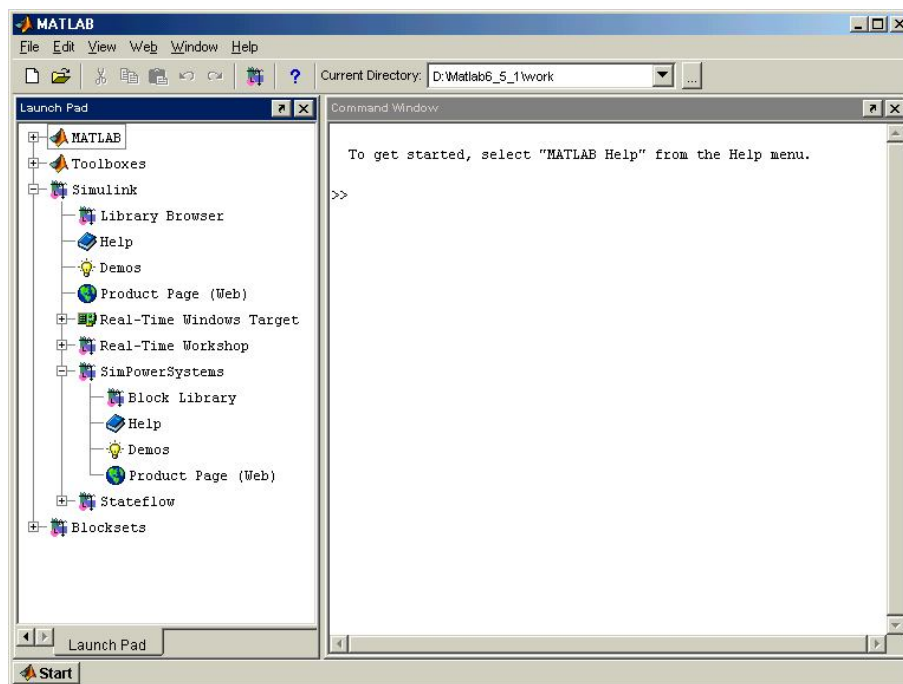
- interaktivno sučelje,
- brzo i lako programiranje,
- ugrađeni grafički podsustav omogućava jednostavnu, kvalitetnu i brzu vizualizaciju,
- programi pisani u Matlab-ovom jeziku su obične tekstualne datoteke i stoga su potpuno prenosive između različitih operacijskih sustava/platformi,
- mnogobrojni dodatni paketi (*toolbox -i*) za razna specijalna područja,
- mnogobrojne m-datoteke i čitavi paketi koje autori, ujedno i korisnici, stavljaju na slobodno raspolaganje putem Interneta.

Program pisan u Matlabu, koji je interpretiran, nužno se sporije izvršava od onih pisanih u npr. Fortranu ili C-u, koji su prevedeni.

Više o MATLAB – u na www.mathworks.com

1.3. MATLAB – radni prostor

Rad u Matlab – u odvija se putem radnog prozora, a unutar radnog prostora. Radni prozor služi za unos naredbi, te ispis rezultata i najvažniji je dio korisničkog sučelja. Radni prostor zamišljamo kao dio radne memorije dodijeljen Matlabu prilikom pokretanja, koji sadrži korisnikove varijable. Moglo bi se reći da se kroz radni prozor "gleda" u radni prostor.



Osnovni prozor Matlabu sadrži izbornik, redak s alatima (*toolbar*) i pet mogućih podprozora. Odgovarajući prozor aktivira se u izborniku *View*. *Launch Pad* prikazuje osnovnu strukturu Matlabu. *Workspace* je prozor koji omogućuje uvid u memorijski prostor, odnosno u tekući direktorij. Podprozor *Command History* sadrži popis svih do sada upotrebljenih naredbi. U *Command Window* se upisuju naredbe Matlabu iz kojih se dobivaju numerički rezultati. Redak teksta upućuje na *Help*, odnosno pomoć koju korisnik može zatražiti od programa. Znak `>>` predstavlja prompt Matlabu. Označava da je Matlab spreman za prihvatanje naredbi od korisnika. Iza prompta moguće je upisivati naredbe Matlabu, pokretati funkcije i izvršavati matematičke operacije.

1.4. MATLAB – struktura i varijable

MATLAB koristi slijedeće elementarne operacije i funkcije: aritmetičke, logičke i operacije izmjene podataka te matematičke i grafičke funkcije. One se izvode na objektima određenog tipa podataka, a to su matrice. Iz tog se razloga matricama ovdje pridodaje naziv MATLAB varijable. One mogu sadržavati realne i kompleksne brojeve te ASCII znakove. Funkcije rade nad varijablama koje se trenutno nalaze u radnom prostoru ili kreiraju nove varijable. Po izvršenju funkcije sve varijable ostaju u radnom prostoru i može ih se pregledati, crtati, te dalje koristiti.

Cjelokupni rad s Matlabom zasniva se na radu s varijablama. Na definirane ulazne varijable primjenjuju se matematičke operacije i funkcije, a kao rezultat dobiju se izlazne varijable. Varijabla se definira i unosi u komandni prozor preko tipkovnice uz slobodni izbor njenog naziva, kao na primjer:

```
» x = 7.25
```

```
x =
```

```
7.2500
```

Format ispisa rezultata ostvaruje se preko podešavanja odgovarajuće postavke iz menija *File/Preferences/Command Window/numeric format*; ili iz komandnog prozora naredbom *format*. Temeljni je format short koji ispisuje 4 značajna decimalna mjesta. To se može promijeniti odabirom opcije long ili naredbom *format long*.

U nastavku je prikazano nekoliko tipičnih primjera unosa varijabli:

```
» vektor = [4 2 -3]
```

```
uz odgovor
```

```
vektor =
```

```
4 2 -3
```

```
» matrica = [3 1+2*i 7; 4 0 -5]
```

```
uz odgovor
```

```
matrica =
```

```
3.0000 1.0000 + 2.0000i 7.0000  
4.0000 0 -5.0000
```

```
» jednostupcanamatrica = [3; 5; 7; 1-3i]
```

```
uz odgovor
```

```
jednostupcanamatrica =
```

```
3.0000  
5.0000  
7.0000  
1.0000 - 3.0000i
```

```
» [2, -4, 5; -8, 3, -2]
```

uz odgovor

```
ans =
```

```
     2   -4    5  
    -8    3   -2
```

Ne upiše li se naziv varijable Matlab sam dodaje naziv *ans* (odgovor). Iz ovih primjera proizlazi da su unosi jednog reda matrice međusobno odvojeni praznim mjestima, a jednog stupca s točka – zarezom. Točka – zarez na kraju naredbe označava da se rezultat ne ispisuje na ekranu. Pored toga ; služi za razdvajanje više naredbi u jednom redu. Ukoliko nam je naredba predugačka za jedan red dodavanjem na kraju tog reda ... ista se nastavlja u sljedećem redu.

Matlab varijable mogu sadržavati realne i kompleksne brojeve te ASCII znakove.

Varijable u Matlabu su matrice, različitog sadržaja i dimenzije. Skalarne se veličine tretiraju kao matrice 1×1, dok su vektori stupčane ili redne matrice. Kao i kod svih računalnih jezika postoje neka od osnovnih pravila kod imenovanja varijabli i datoteka:

- Potrebno je razlikovati upotrebu malih/velikih slova,
- Maksimalni broj znakova,
- Prvi znak mora biti slovo.

Matlab posjeduje i specijalne (simboličke) varijable (*pi*, *eps*, *ans*...) čiji su nazivi rezervirani. Simboličke varijable sadrže simbole koje se interpretiraju numerički.

Varijable Matlab se mogu podijeliti u više skupina (prema sadržaju elemenata matrice, prema dohvat, prema izvoru nastanka) a njihova svojstva ovise o pripadnosti pojedinim skupinama.

Sadrže li elementi matrice realne brojeve i varijabla se može nazvati realna. Kompleksne varijable sadrže kompleksne brojeve.

Prema vidljivosti varijable možemo podijeliti na lokalne i globalne. Ukoliko je potrebno određene varijable primijeniti u nekoliko funkcija, a želimo izbjeći njihovo pojavljivanje u popisu ulaznih varijabli tih funkcija, dovoljno je takve varijable deklarirati kao globalne. Globalne su varijable one varijable koje su vidljive iz više funkcija Matlab. Postoji li definirana varijabla u funkciji Matlab, nakon izvršenja funkcije u *Workspacu* varijabla neće postojati. Vidljiva je samo u funkciji gdje je definirana. Takve varijable tretiramo kao lokalne. Varijabla postaje globalna tako da se deklarira naredbom:

```
>> global ime_varijable
```

Prema izvoru nastanka varijable možemo podijeliti na interne i eksterne varijable. Interne su varijable one varijable koje definira sam Matlab kao što su *eps*, *realmin*, *realmax*, *pi*, *inf*, itd. Bilo koji od ovih varijabli interpretiraju se jednostavnim ukucavanjem naziva u komadni prozor iza znaka prompt.

Eksterne varijable su varijable definirane od korisnika ili varijable nastale kao rezultat matematičkim operacijama i funkcija izvedenih u Matlabu.

Definirane varijable moguće je vidjeti u tzv. *Workspace-u* (pregledniku radnog prostora) ili u komadnom prozoru primjenom naredbe *who* (ispis varijabli) i *whos* (detaljni ispis varijabli).

Tako je u našem primjeru:

» *who*

Your variables are:

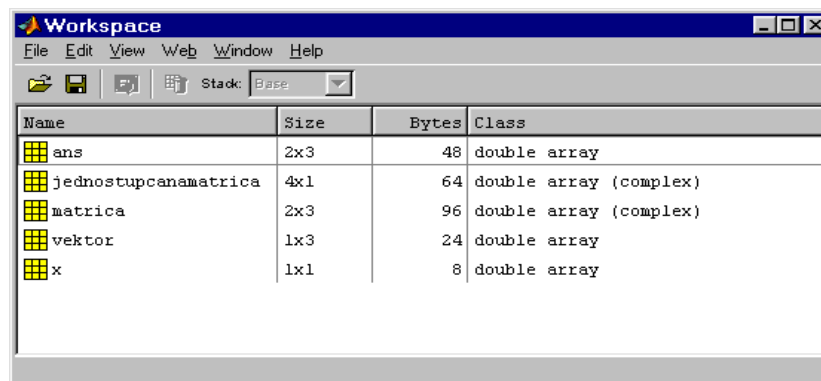
```
ans          matrica          x
jednostupcanamatrica  vektor
```

» *whos*

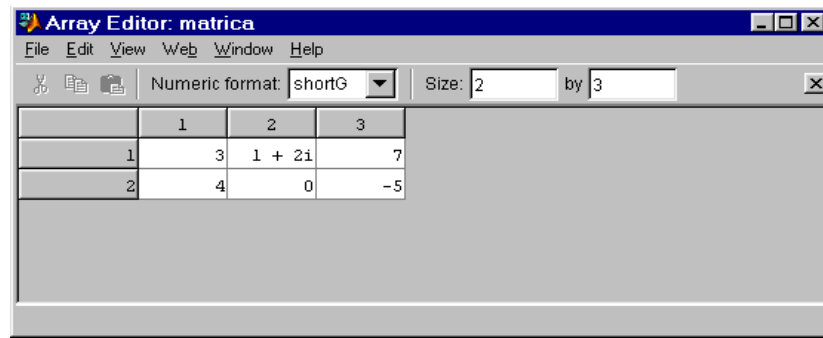
Name	Size	Bytes	Class
ans	2x3	48	double array
jednostupcanamatrica	4x1	64	double array (complex)
matrica	2x3	96	double array (complex)
vektor	1x3	24	double array
x	1x1	8	double array

Grand total is 20 elements using 240 bytes

Izborom naloga *View/Workspace* možemo te iste varijable vidjeti i u prozoru *Workspace*.



Dvostrukim klikom na bilo koju varijablu unutar tog podprozora otvara se *Array Editor* u kojem je moguće promijeniti različite parametre kao što su dimenzije matrice, format prikazivanja te pojedinačne unose.



Ukoliko neku od varijabli želimo izbrisati, možemo ju izbrisati nalogom *clear* na sljedeći način:

```
>> clear jednostupcanamatrixa
```

```
>> who
```

Your variables are:

```
ans      matrica      vektor      x
```

Samim nalogom *clear* bez dodatnih opcija izbrisali bi sve varijable iz radnog prostora. Ista mogućnost nam stoji na raspolaganju s preglednikom radnog prostora - pritiskom desne tipke miša možemo odabrati *Delete* za brisanje odabrane varijable (ili ikonicu za brisanje) te *Clear Workplace* za brisanje svih varijabli iz radnog prostora.

Sadržaj radnog prostora možemo spremiti s binarnim formatom u željenu datoteku *ime.mat* i to na nekoliko načina:

- iz menija *File/Save Workspace as*,
- iz menija *File/Import Data*,
- iz preglednika radnog prostora - pritiskom desne tipke miša odabirom *Save Workplace as*.
- naredbom u komandnom prozoru *save ime*

Dijalogni prozor *File/Import Data* sadrži popis datoteka različitih formata. Klikom na bilo koju datoteku otvara se prozor *Import Wizard*. Na taj način se mogu izvesti odgovarajuće izmjene u datoteci.

Matlab koristi više raznih vrsta datoteka i one se prepoznaju po nastavku unutar imena. Najčešći korišteni tipovi su:

- *ime.m* - je tekstualna datoteka koja sadrži program pisan u Matlabu.
- *ime.mat* - je binarna datoteka, tj. standardni Matlabov format za spremanje varijabli i njihovih vrijednosti.

Takve datoteke se direktno učitavaju u *Workspace* na sljedeći način:

```
>> load ime.mat.
```

- *ime.fig* - je binarna datoteka, tj. standardni format Matlaba za spremanje slika. Učitava se u Workspace naredbom:

```
>> open ime.fig.
```

Savjeti za rad

Ranije uneseni nalozi u radnu površinu ostaju pohranjeni i mogu se po potrebi rekonstruirati. To omogućava pojednostavljeno ponavljanje naloga ili njihovu izmjenu. U tu se svrhu mogu koristiti tipke u tipkovnici sa strelicama za gore i dolje, pri čemu se pojavljuju raniji nalozi koji se pritiskom na tipku *Enter*, u istom obliku ili korigirani, ponovo mogu izvesti.

Ukoliko pri izvršenju programa dođe do neočekivanih grešaka, Matlab nas upozorava porukom u komandnom prozoru i prekida daljnje izvršavanje programa. Bitno je napomenuti da se za prve korake u Matlab - u korektnom analizom poruke greške iz komandnog prozora može brže napredovati u radu. Npr. poruka pri grešci funkcije ili varijable bi bila:

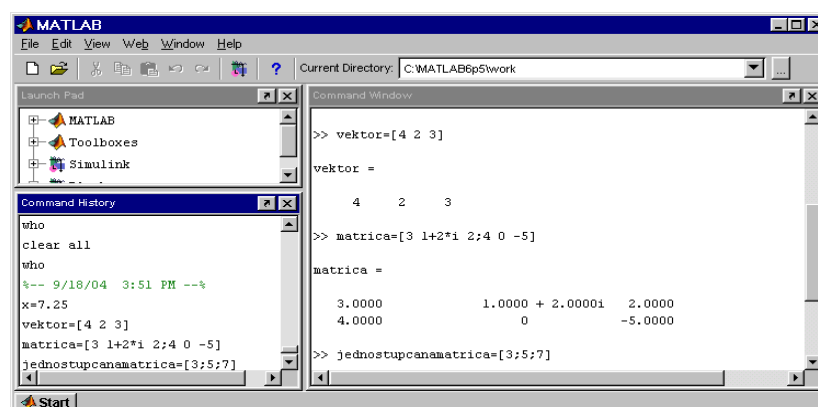
```
>> y=average(x)
??? Undefined function or variable 'average'.
```

Kao i većina drugih programskih jezika i u *MATLAB*-u moguće je od korisnika zatražiti određeni ulaz. Osnovni je oblik unos ulaza preko tipkovnice komandom *input*, kao npr.

```
>> m=input('unesi vrijednost za varijablu P[kW]: ')
```

Ako je u *Command Window* prisutno puno naloga, njihovo se traženje može pojednostaviti utipkavanjem iza promptnog znaka samo početnih slova naloga. Pritiskom na tipku sa strelicom prema gore pojavljuje se odmah traženi nalog.

Čitavu situaciju dodatno pojednostavljuje i prisutnost četvrtog podprozora pod nazivom *Command History*. U njemu su izlistani svi ranije korišteni nalozi uz navođenje datuma njihova nastanka. Kad se nađe traženi nalog, za njegovo ponovo izvođenje dovoljno je na njega samo dvaput kliknuti za učitavanje u radni prostor.



1.5. MATLAB – grafičke funkcije

Osim matricnog koncepta, druga uporišna točka Matlab-a je izvrstan grafički podsustav. Interne grafičke funkcije koje su odmah dostupne omogućavaju relativno lako crtanje slikovnih prikaza. S druge pak strane moguće je detaljno upravljanje svim aspektima prikaza kao što su veličina i položaj slike na papiru, položaj označavanja osi itd.

Za dvodimenzionalni prikaz potrebna su dva vektora iste duljine a za trodimenzionalni tri. Kad se želi istovremeni prikaz zavisnosti više funkcija o jednoj varijabli, onda je varijabla vektor a funkcije se prikazuju matricom. Uz to se linije grafa mogu prikazati na različite načine.

Najvažniju funkciju ima nalog *plot*, a unosom naloga *help plot* dobivaju se sve potrebne informacije o ovoj funkciji.

U svrhu grafičkog prikaza neka se varijabla x može, kao vektor odnosno jednorodna matrica, definirati i unijeti i na slijedeći način

```
» t=(0:2:10)
t =
    0    2    4    6    8   10
```

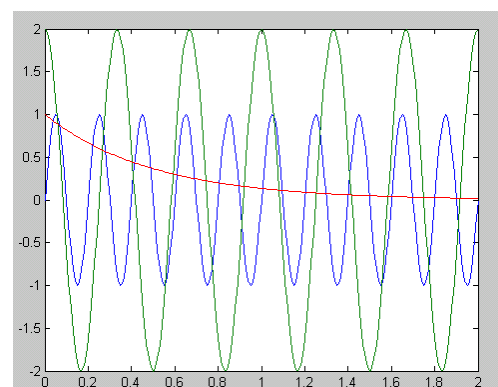
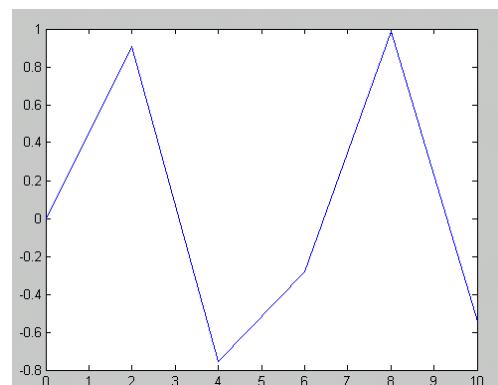
gdje je t vremenski vektor, prva znamenka u zagradi označuje prvu vrijednost varijable, srednja razmak pojedinačnih vrijednosti a zadnja konačnu vrijednost. One su međusobno odvojene dvotočkom. Zada li se ta varijabla na slijedeći način:

```
» t=(0:2:10);
```

nalog se ne izvršava odmah, nego tek iza naloga koji ne sadrži točku - zarez, što je za izvođenje simulacija veoma bitno. Želimo li prikazati graf $\sin t$, onda unos naloga ima izgled

```
» t=(0:2:10);
» s=sin(t);
» plot(t,s)
```

nakon čega se otvara prozor s grafom kao u slici 4.. Ovakav malo čudan izgled grafa pojavljuje se onda, ako su vrijednost na apscisi međusobno dosta razmaknute. Numeričke oznake u osima se pojavljuju automatski, a pomoćne linije, označavanje osi i naziv grafa moraju se unijeti dodatnim nalogima.



U idućem primjeru treba za vremenski vektor $t=(0:0.01:2)$ izračunati i prikazati u jednom dijagramu graf sinusa amplitude 1 i frekvencije 5 Hz, graf kosinusa amplitude 2 i frekvencije 3 Hz te graf eksponencijalne funkcije e^{-2t} .

```

» t=(0:0.01:2);
» sinfkt=sin(2*pi*5*t);
» cosfkt=2*cos(2*pi*3*t);
» expfkt=exp(-2*t);
» plot(t,[sinfkt ; cosfkt ; expfkt])

```

U grafovima se jednostavnim nalogima, baziranim na uputama u već spomenutom nalogu *help plot*, mogu mijenjati boje i vrste linija. U cilju dokumentiranja grafova nužno je uz to upisati oznake osi i naziv dijagrama, ucrtati mrežu pomoćnih linija te povećati pojedine isječke dijagrama.

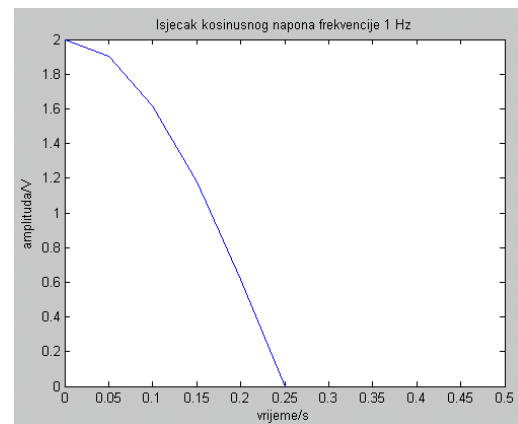
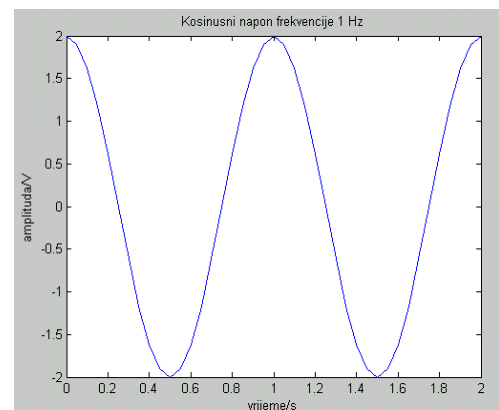
Za opise služe nalozi *xlabel*, *ylabel* i *title*, za mrežu pomoćnih linija nalog *grid*, a za povećanje isječka nalozi *zoom* i *axis* u kojima se mora navesti x,y područje koje se želi povećati. Idući primjer pokazuje neke od tih mogućnosti.

```

» t=(0:0.05:2);
» cosfkt=2*cos(2*pi*t);
» plot(t, cosfkt)
» xlabel('vrijeme/s')
» ylabel('amplituda/V')
» title('Kosinusni napon frekvencije 1 Hz')
» figure % Otvoriti novi prozor !
» plot(t, cosfkt)
» xlabel('vrijeme/s')
» ylabel('amplituda/V')
» axis([0,0.5, 0,2]) % Isjecak intervala [0,0,5] i [0,2]
» title('Isjecak kosinusnog napona frekvencije 1 Hz')

```

Definiranje gornjeg dijagrama završava nalogom *title('Kosinusni napon frekvencije 1 Hz')*, a novi se otvara nalogom *figure* i za njegovo definiranje moraju se zatim unijeti nalozi koji iza njega slijede (vidi gore).



Osim funkcije *plot* postoje i funkcije *semilogy*, *semilogx*, *loglog*, *plotyy* za crtanje u logaritamskom mjerilu, te mnoge druge funkcije za crtanje dvodimenzionalnih prikaza. Sve ove funkcije imaju različitu sintaksu za različite potrebe korisnika. Više o tim funkcijama možete doznati pozivanjem naloga *help ime_funkcije*.

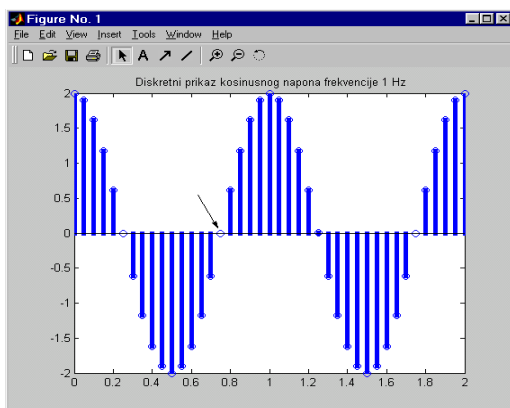
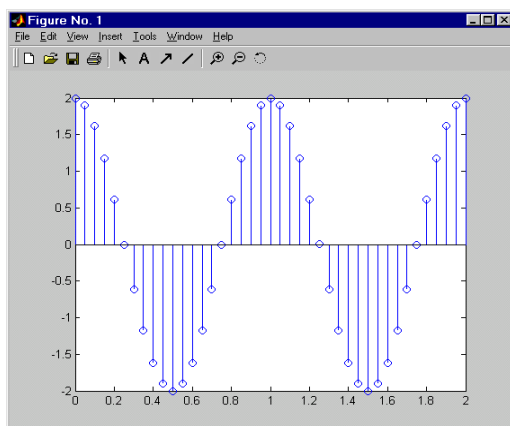
Naknadnim pozivanjem neke grafičke funkcije poput *plot* postojeća slika, će u pravilu, biti obrisana i zamijenjena novom. Ako se pak želi crtati preko postojeće slike treba koristiti naredbu *hold on*. Suprotan učinak ima naredba *hold off*.

Parametriranje dijagrama

Dijagrami definirani nalogima u radnoj površini mogu se dodatno doraditi. Za primjer uzmimo već prikazani dijagram Kosinusnog napona frekvencije 1 Hz, prethodno ga prikažimo na jedan drugi način i zatim doradimo. Prikažimo ga u obliku tipičnom za diskretne signale, korištenjem naloga *stem*.

```
» t=(0:0.05:2);
» cosfkt=2*cos(2*pi*t);
» stem(t,cosfkt)
```

Dobiveni dijagram se, isto kao i oni ranije, pojavljuje u Matlab grafičkom prozoru (slika dolje lijevo). U slici se vidi traka s nalogima koji omogućavaju grafičku doradu dijagrama, a ispod nje i traka sa simbolima od kojih je posebno zanimljiv zadnji desno koji omogućuje rotaciju trodimenzionalnih dijagrama. Doradimo prikazani dijagram.



Nalogom pomoću simbola A upišimo naziv dijagrama i centrirajmo ga iznad grafa, a pomoću simbola strelice naznačimo prijelaz iz negativnog u pozitivno područje. Dvostrukim klikom na graf otvorimo dijalogni prozor i u njemu promijenimo debljinu linije s 0.5 na 4.0.

Dvostrukim klikom na osi otvorimo drugi dijalogni prozor, izgleda kao na slici gore desno. On omogućuje promjene skaliranja i još nekih osobina osi, što se vidi u dijalognom prozoru. Upišimo u njemu oznake obojih osi. Dobili smo time doradjeni dijagram izgleda kao onaj ispod osnovnog.

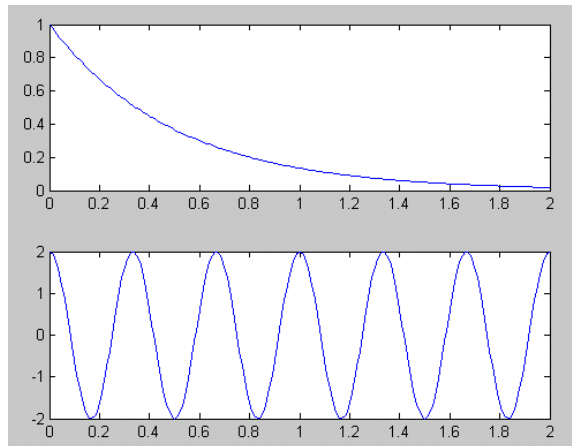
Već je navedeno kako napraviti više dijagrama odnosno više grafičkih prozora, ali kako na jednom grafičkom prozoru napraviti više dijagrama. To se ostvaruje naredbom *subplot*.

Pogledajmo jedan primjer:

```

» t=(0:0.01:2);
» cosfkt=2*cos(2*pi*3*t);
» expfkt=exp(-2*t);
» subplot(211)           % ucrtati lijevi graf
» plot(t,expfkt)
» subplot(212)           % ucrtati desni graf
» plot(t,cosfkt)

```



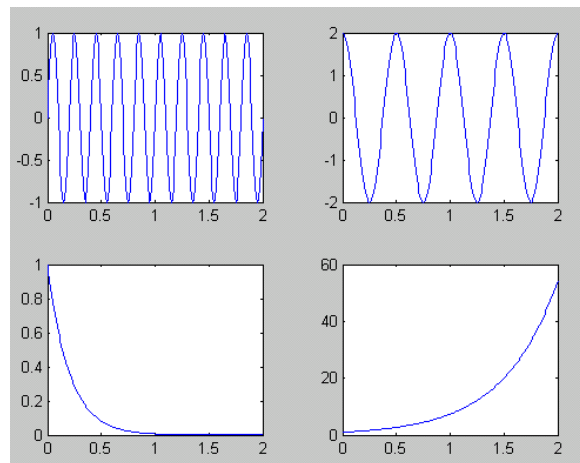
Rezultat ovih naloga prikazan je u gornjoj slici. Položaj osi drugog grafa usklađuje se s položajem osi prvog preko naloga *subplot*, a za prikaz obaju grafova odgovoran je nalog *plot*.

Parametri naloga *subplot* kažu koliko grafova odnosno polja okomito sadrži prikaz (prva znamenka u nalogu) i koliko vodoravno (druga znamenka). Treća znamenka kaže koliko grafova je predviđeno, brojeći od lijevo gore prema desno dolje. Primjerice nalog *subplot* (325) znači da se radi o petom grafu smještenom u prikazu s poljima 3x2. Pogledajmo jedan primjer u kojem treba u jednoj slici prikazati četiri grafa, dva gore i dva dolje.

```

» t=(0:0.01:2);
» sinfkt=sin(2*pi*5*t);
» subplot(221)           % ucrtati gornji lijevi graf
» plot(t,sinfkt)
» cosfkt=2*cos(2*pi*2*t);
» subplot(222)           % ucrtati gornji desni graf
» plot(t,cosfkt)
» expfkt1=exp(-5*t);
» subplot(223)           % ucrtati donji lijevi graf
» plot(t,expfkt1)
» expfkt2=exp(2*t);
» subplot(224)           % ucrtati donji desni graf
» plot(t,expfkt2)

```

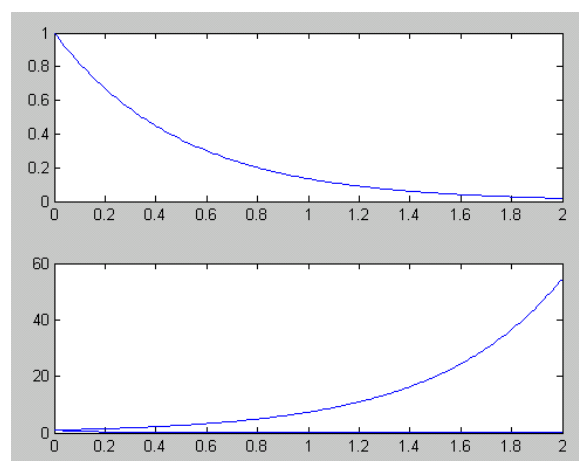


Druga je mogućnost samo okomiti prikaz grafova, jedan ispod drugog. U tu svrhu služi nalog *hold*, kako se vidi u idućem primjeru.

```

» t=(0:0.01:2);
» expfkt1=exp(-5*t);
» expfkt2=exp(2*t);
» plot(t,expfkt1)
» hold                   % ucrtati drugi graf ispod prvog
» plot(t,expfkt2)

```



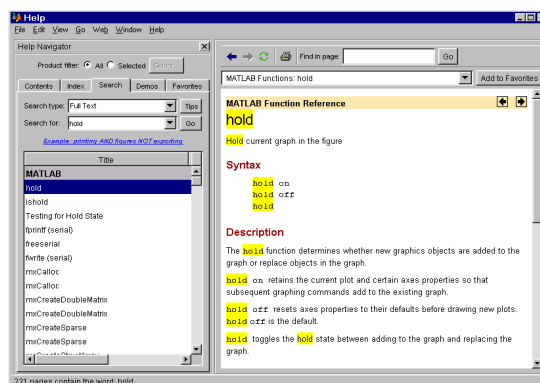
Svi ovi naponi grafičke prezentacije podataka bili bi uzaludni kada dobivene dijagrame ne bi mogli transformirati iz Matlab-a u željeni oblik: ispis na printeru ili u grafičku datoteku ili pak direktan prijenos (*copy-paste*) u neku drugu aplikaciju. Četiri su osnovne operacije kojima je moguće transformirati grafiku iz Matlab-a:

1. ispis na printer,
2. ispis u datoteku (*Postscript* datoteku),
3. eksportiranje u datoteku u odabranom grafičkom formatu,
4. eksportiranje u *Clipboard* (za daljnju uporabu u nekoj drugoj aplikaciji).

Osnovna naredba Matlab-a za ispis je *print* i ovisno o našoj potrebi primjenjuje se nekoliko opcija. Svaka je opcija primjenjiva direktno iz komandnog prozora. Način na koji se nalazi unose u radni prostor mogu se vidjeti pozivanjem naloga *help print*.

1.6. MATLAB - pomoć

U traženju pomoći pri definiranju i unosu naloga treba kliknuti na nalog *Help/MATLAB Help* u traci s nalogima te u otvorenom dijalognom prozoru u kartici *Search* upisati nalog za koji se pomoć traži. Lijevo u slici dolje vidi se dijalogni prozor s otvorenom karticom *Search* i u njoj primjerice upisani nalog *hold* za koji se traže dodatna objašnjenja. Desno u slici vide se objašnjenja koja se pojave nakon klika na nalog *Go* u kartici *Search* i dvostrukog klika na nalog *hold* u polju s nalogima ispod oznake *Title*.



Matlab je i okružje i programski jezik. Jedna od jačih strana Matlab-a je činjenica da njegov programski jezik omogućava izgradnju vlastitih alata za višekratnu uporabu. Možete lako sami kreirati vlastite funkcije i programe (poznate kao *m-datoteke*) u kodu Matlab-a.

Matlab posjeduje niz *demo* funkcija za prezentaciju mogućnosti Matlab-a.

Postoje funkcije visokog stupnja za 2D i 3D vizualizaciju podataka, obradu slike, animaciju. Također postoje i funkcije za izgradnju grafičkih sučelja za vaše Matlab aplikacije.

U ovoj vježbi bit će pokazane samo osnovne funkcije Matlab-a dostatne za početak samostalnog rada u Matlab-u. Za sve daljnje informacije proučite Matlab-ov help ili dodatne upute koje dolaze s instalacijom.