Univerzitet u Banjaluci
Elektrotehnički fakultet
Katedra za opštu elektrotehniku
Analogni i digitalni filtri

Vježba 1.
Uvod u MATLAB

Uvodne napomene, vektori, matrice

MATLAB se pokreće biranjem **Start→Programs→Matlab**. Nakon ovoga otvara se komandni prozor i pojavljuje MATLAB prompt >>.

Iz MATLAB-a se može izići kucanjem komande:

```
»quit
```

U MATLAB je ugrađen i obiman sistem pomoći koji se dobija komandom:

```
»help
```

Odnosno, ako znate ime funkcije, ali se ne možete sjetiti tačne sintakse:

```
»help ime_funkcije
```

Većina matematičkih funkcija je ugrađena u MATLAB i obično je korisno da se komanda `help` izvrši za željenu funkciju da bi se upoznali sa njenim mogućnostima.

Osnovna struktura podataka u MATLAB-u je matrica. Specijalni slučajevi su 1×1 matrica (skalar) i matrice čija je jedna dimenzija 1 – vektori.

Matrice se mogu unijeti:

1. nabrajanjem svih elemenata,
2. učitavanjem matrice iz fajla,
3. generisanje pomoću ugrađenih funkcija,
4. generisanje pomoću sopstvenih (korisničkih) funkcija.

```
»x = 5
»b = [1 3 8 12.1 7]
»A = [1, 7, 8; 2, 9, 13; 4, 8, 11; 2, 5, 3]
```

MATLAB nakon izvršenja svake komande ispisuje rezultat. To se može izbjeći ako se linija završi sa ;

Pojedinim elementima matrice može se pristupiti navođenjem njihovih indeksa:

```
» b(1)
ans =

     1
» A(1,2) = 1
» A(1,:) = [1, 7, 8]
» A(:,1) = [1; 2; 4; 2]
```

Koristan način za zadavanje vektora može biti operator dvotačka `:` čija je sintaksa

```
x = start:korak:kraj
```

Na primjer:

```
» x = 3:8
x =
     3 4 5 6 7 8
» x = 2:2:10
x =
     2 4 6 8 10
```

Konkatenacija nizova je veoma jednostavna:

```
>> A = [1 2 3 4];
>> B = [5 6 7 8];
» C = [A B]
C =
     [1 2 3 4 5 6 7 8]
```

Osnovne aritmetičke operacije +, -, *, / mogu da se koriste na isti način kao i u drugim višim programskim jezicima s tim što njihovi argumenti mogu biti i vektori, odnosno matrice. Isto vrijedi i za elementarne funkcije: `exp, log, sqrt, sin, cos` itd. Ako su argumenti ovih funkcija matrice onda se elementarne funkcije izračunavaju po elementima.

Dimenzije matrice dobijaju se pomoću funkcije `size`.

```
» size(A)
```

Transponovanje niza/matrice vrši se pomoću operatora `'`

```
» C = C'
C =
     1
     2
     3
     4
     5
     6
     7
     8
```

Da biste izvršili skalarnu operaciju na vektoru koristite obične matematičke operacije. Na primjer, da biste pomnožili svaki element vektora C sa 2 koristite komandu:

```
» C = 2*C
C =
     2     4     6     8    10    12    14    16
```

Pretpostavimo da želite da pomnožite svaki element vektora a odgovarajućim elementom vektora b. U tom slučaju koristićete operator `.` koji izvršava operacije po elementima. Na primjer:

```
» a = [2 2 2 2]
```

```
a =
     2     2     2     2
» b = [1 2 3 4]
b =
     1     2     3     4
» a.*b
ans =
     2     4     6     8
```

Ako biste koristili obično množenje * dobili biste:

```
» a*b
??? Error using ==> *
Inner matrix dimensions must agree.
```

zato što MATLAB pokušava da pomnoži matrice a i b, prema pravilima za množenje matrica.

Postoji i niz funkcija koje generišu karakteristične matrice. Dvije osnovne su

```
» A = zeros(n,m);
» B = ones (n,m);
```

`Zeros` generiše matricu dimenzija (n,m) čiji su svi elementi nule, a `ones` matricu istih dimenzija čiji su svi elementi jedinice. Ovo je vrlo korisno kod zadavanja signala.

Pored promjenljivih definisanih na pomenute načine u MATLAB-u postoje i neke specijalne predefinisane promjenljive:

Specijalne promjenljive	**Opis**
`ans`	Rezultat poslednje operacije
`pi`	Broj π
`eps`	Najmanji broj koji kada se doda jedinici, daje broj koji je na računaru veći od jedinice.
`flops`	Broj floating-point operacija
`inf`	Beskonačno (npr. 1/0).
`NaN` (or) `nan`	Nije broj – Not-a-Number (npr. 0/0).
`i` (and) `j`	i = j = √-1
`realmin`	Najmanji mogući realan broj
`realmax`	Najveći upotrebljiv pozitivan realni broj

Polinomi

Polinomi se u MATLAB-u predstavljaju pomoću nizova. Jedna mogućnost je da niz sadrži koeficijente polinoma od člana najvišeg do člana najnižeg stepena. Ako član ne postoji, unosi se vrijednost koeficijenta 0. Na primjer, ako je dat polinom:

$$y(x) = 9x^3 + 5x + 1,$$

njegova reprezentacija u MATLAB-u bila bi:

```
» y = [9 0 5 1]
```

Drugi način da se zadaju polinomi je pomoću njihovih korijena. Na primjer:

$$y(x) = (x+3)(x-5)(x+9) \,,$$

se u MATLAB-u predstavlja kao:

```
» y = [-3 5 -9]
```

Korisne funkcije pri radu sa polinomima su `roots` i `poly`. Funkcija `roots(y)` pronalazi korijene polinoma i kao rezultat vraća vektor korijena. Funkcija `poly(y)` radi suprotno. Ona uzima vektor korijena polinoma i kao rezultat vraća vektor koeficijenata.

Za množenje dva polinoma koristi se funkcija `conv(y,z)`, koja uzima dva vektora koeficijenata polinoma i kao rezultat vraća vektor koeficijenata polinoma koji je njihov proizvod. Ova funkcija se, jasno, koristi i za izračunavanje konvolucije nizova y i z.

Pisanje programa u MATLAB-u

Programi u MATLAB-u su tekstualni fajlovi koji sadrže MATLAB komande. Ekstenzija ovih fajlova je .m, pa se zbog toga oni ponekad zovu i m-fajlovi. Ovi fajlovi mogu da se kreiraju pomoću bilo kojeg editora teksta, a može da se i koristi MATLAB-ov ugrađeni editor/debager koji postoji od verzije 5.

Da biste izvršili program u MATLAB-u, pređite u direktorijum u kojem se nalazi m-fajl i otkucajte ime fajla bez ekstenzije. Na primjer, ako je ime fajla `lab1.m`, program pokrećete kucajući

```
» lab1
```

Ukoliko se u fajlu nalaze validne MATLAB komande i funkcije one će se izvršiti i promjenljive koje su kreirane nalaziće se u radnoj memoriji MATLAB-a i sa njima se može dalje raditi na uobičajen način.

Grafičko prikazivanje podataka

Osnovna funkcija za grafički prikaz podataka je `plot`. U zavisnosti od ulaznih argumenata dobijaju se različiti rezultati. Na primjer,

```
» x = [1, 7, 5, 4.3, 2, 9, 11, 8.8];
» plot(x)
```

će nacrtati tačke iz vektora x u funkciji njihovog indeksa i povezati ih pravolinijskim segmentima, a

```
» plot(x,'o')
```

će nacrtati tačke označene kružićima i neće ih povezati.

Primjer

Nacrtati funkciju $y = x^2$ na intervalu [-2, 2].

```
x = -2:0.01:2;
y = x.^2;
plot(x,y);
```

Moguće je na jednoj slici nacrtati više grafika i MATLAB ih crta različitim bojama iz predefinisanog skupa.

```
y1 = x.^2 – 4;
```

```
y2 = x.^2 + 4;
plot(x,y,x,y1,x,y2);
```

Boja, oblik tačkica, tip linija se mogu zadati eksplicitno, pogledati help stranicu za komandu `plot`.

Komanda `plot` automatski otvara novi prozor, ako ne postoji ni jedan već otvoren. Ako je neki grafički prozor otvoren, `plot` crta u njega, a prethodni sadržaj se briše. Novi prozor se otvara pomoću komande `figure` i on postaje aktivan. Sledeća `plot` komanda crta u njega. Već postojeći prozor se aktivira sa `figure(n)`, gdje je `n` broj prozora koji se nalazi u naslovnoj liniji, npr. Figure No.1

Pomenuto je da se crtanjem u već postojeći prozor njegov prethodni sadržaj briše. Ovakvo se ponašanje može promijeniti pomoću komande

```
>> hold on
```

koja zadržava postojeći grafik i sledeća komanda za crtanje crta preko njega.

Predefinisano ponašanje vraća se sa:

```
>> hold off
```

Graficima se lako mogu dodati naslov, kao i oznake osa.

```
t = -pi:pi/100:pi;
y = sin(t);
plot(t,y);
title('Grafik funkcije sinus');
xlabel('t');
ylabel('sin(t)');
```

Predstavljanje kontinualnih signala u MATLAB-u

Osnovna snaga MATLAB-a leži u njegovim algoritmima i bibliotekama koje implementiraju numeričke algoritme. Dakle, nije moguće operisati funkcijama (signalima) u njihovom analitičkom obliku. Ono što je moguće je generisanje vektora koji sadrže vrijednosti signala u određenim tačkama i zatim manipulisanje tim vektorima umjesto analitičkim oblikom signala. Jasno je da se u stvari vrši diskretizacija signala. Primjeri ovoga su već pokazani kod crtanja grafika funkcija. Može se pokazati da se signali, ukoliko su ispunjeni određeni uslovi, mogu vjerodostojno predstaviti na ovaj način. Jasno je da se pomenuti uslovi svode na interval u kojem se uzimaju vrijednosti signala (odmjerci). Poznata *teorema o odmjeravanju* tvrdi da je signal moguće jednoznačno rekonstruisati iz ovih odmjeraka ukoliko su oni uzeti na razmaku koji je manji ili jednak $2F_m$, gdje je F_m najveća frekvencija u spektru signala.

Osim toga, da bismo dobili i vjerodostojnije grafičke prikaze kontinualnih signala ovaj period odmjeravanja treba još smanjiti. Sa druge strane, premalen period odmjeravanja rezultuje jako dugim nizovima, tj. velikim utroškom memorije. Imajući sve ovo u vidu možemo empirijski odabrati period odmjeravanja od $10F_m$ za većinu naših primjena, uz mogućnost njegovog smanjivanja ili povećavanja u pojedinim slučajevima.

Linearni, vremenski nepromjenljivi sistemi

Jedan od načina zadavanja linearnih, vremenski nepromjenljivih sistema u MATLAB-u je pomoću njihove funkcije prenosa. Poznato je da funkcija prenosa ovakvih sistema ima oblik racionalne funkcije. Dakle, moguće je memorisati dva polinoma, N i D, koji predstavljaju brojnik i nazivnik funkcije prenosa.

Za određivanje frekvencijske karakteristike filtra koristi se funkcija `freqs` čija je sintaksa:

```
[h, w] = freqs(N, D);
```

gdje su `N` i `D` brojnik i nazivnik funkcije prenosa filtra, respektivno, a u vektoru `h` nalaze se vrijednosti frekvencijske karakteristike u frekvencijama koje se nalaze u vektoru `w`.

Pokušajte da iskoristite naredbu `freqs` bez izlaznih parametara, dakle samo `freqs(N, D)`. Kakav rezultat se dobija?

Impulsni odziv sistema dobija se funkcijom `impulse(N,D,tfinal)`, gdje su `N` i `D` brojnik i nazivnik funkcije prenosa filtra, respektivno, a `tfinal` je trajanje simulacije.

Simulacija linearnih, vremenski nepromjenljivih sistema može se izvršiti pomoću funkcije lsim, čija je sintaksa:

```
y = lsim(N, D, u, t);
```

Kao i do sada `N` i `D` su brojnik i nazivnik funkcije prenosa filtra, respektivno, a u vektorima `u` i `t` nalaze se ulazni signal i vrijeme. Vrijednosti izlaznog signala nalaze se u vektoru `y`.

Zadaci

Predstavljanje kontinualnih signala u MATLAB-u

1. Generisati dva sinusna signala

$$x_1(t) = 10\cos(2\pi \cdot 50t)$$
$$x_2(t) = 5\cos(2\pi \cdot 75t + \pi/3)$$

   Nacrtajte svaki od ovih signala na intervalu 0 do 80ms. Označite grafike i ose. Koliki ste period odmjeravanja izabrali? Koliko elemenata imaju dobijeni vektori?

2. Formirati novi signal:

$$y(t) = x_1(t) + x_2(t).$$

   Nacrtajte ovaj signal na intervalu 0 do 80ms.

3. Prenos signala kod AM radija vrši se amplitudskom modulacijom nosećeg signala. Jednačina po kojoj se noseći signal moduliše je:

$$f(t) = A[1 + m(t)]\cos(\omega_c t).$$

   Gdje je $m(t)$ signal koji se prenosi (obično govor ili muzika), ω_c je noseća frekvencija, i A je konstanta.

   Formirajte amplitudno modulisan signal prema sledećoj jednačini:

$$f(t) = [1 + m\cos(2\pi t)]\cos(2\pi \cdot 10t).$$

   Nacrtajte rezultat za $m = 0$, 0.5 i 1 za $0<t<3$ sekunde. Šta se dešava kada se m mijenja?

4. Dat je polinom $p(s) = -s^6 + \frac{1}{64}s^4 + \frac{1}{32}s^2 + \frac{81}{64}$.

   Pronaći korijene ovog polinoma. Izdvojiti one korijene koji se nalaze u lijevoj poluravni kompleksne ravni, te formirati novi polinom čiji su to korijeni.

   Pokušajte ovaj postupak u potpunosti automatizovati, tj. napišite program u MATLAB-u koji će kao ulaz imati koeficijente zadatog polinoma, a kao izlaz davati novi polinom dobijen opisanim postupkom.

Analiza linearnih, vremenski nepromjenljivih sistema

1. Zadat je sistem (filtar) sa funkcijom prenosa

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1}\ .$$

   Predstaviti ovaj sistem u MATLAB-u i korištenjem MATLAB-ovih funkcija uradite sledeće zadatke.

2. Nacrtati frekvencijsku karakteristiku ovog filtra.
3. Odrediti slabljenje datog filtra na učestanostima 0, 1, 10, 100 rad/s.
4. Nacrtati impulsni odziv ovog filtra.
5. Odrediti odziv filtra ukoliko se na njegov ulaz dovede signal

$$u(t) = \cos t\ .$$

   Komentarisati rezultat uzimajući u obzir rezultate iz tačke 3.