Semnale si sisteme

Universitatea Politehnica Bucuresti
Facultatea de Automatica si Calculatoare
laborator 1

October 11, 2006

1 Introducere

MATLAB este atat un limbaj de programare cat si un pachet software de vizualizare a datelor, care este in special utilizat in analiza si procesarea de semnale. Acest document este o scurta intordducere in MATLAB, care pune accentul pe caracteristicile relevante in cadrul cursului de Semnale si sisteme.

Mai multe informatii despre MATLAB pot fi gasite pe site-ul web al producatorului (The MathWorks, Inc.): `http://www.m`thworks.com.

2 Preliminarii

MATLAB contine cateva comenzi utile similare comenzilor UNIX (ex:`'ls'`, `'pwd'`,`'cd'`).Comenzile sunt utile pentru a lista directorul de lucru al MATLAB, verificarea caii catre directorul de lucru si schimbarea directorului de lucru. MATLAB cauta fisiere MATLAB in anumite directoare, care sunt controlate de comanda `'path'`.Comanda `'path'` listeaza directoarele din calea in care MATLAB cauta aceste fisiere. Un nou director poate fi adaugat la aceasta cale cu comanda `path(path,p)`sau `path(p,path)`,unde `p` este directorul nou (care poate contine de exemplu functii scrise de utilizator).

3 Help din MATLAB

Daca stim numele unei functii pe care am dori sa stim sa o utilizam folosim comanda `'help'`:

```
>> help  numefunctie
```

Aceasta comanda da o descriere a functiei si o lista de functii legate de ea. Daca nu va amintiti exact numele functiei, puteti folosi comanda `'lookfor'` si un cuvant cheie asociat functiei:

```
>> lookfor  cuvantcheie
```

Aceasta comanda va afisa o lista de functii care au acel cuvant cheie in descriere.

Alte comenzi utile sunt `'info'`,`'what'`si `'which'`.Descrierea acestor comenzi poate fi gasita cu ajutorul comenzii help. MATLAB mai contine si o serie de exemple care pot fi vizualizate cu ajutorul comenzii `'demo'`.

4 Variabile MATLAB - scalari, vectori si matrice

MATLAB stocheaza variabilele sub forma de matrice de dimensiuni $M * N$, unde M este numarul de linii si N este numarul de coloane. O matrice de dimensiune $1 * 1$ este un scalar; o matrice de dimensiune $1 * N$ este un vector linie, iar o matrice $M * 1$ este un vector coloana. Elementele unei matrice pot fi numere reale sau complexe; $\sqrt{-1}$ poate fi scris `'i'`sau `'j'`cu conditia sa nu fie redefinite de utilizator. O matrice este scrisa intre '[]' cu spatii care separa coloane adiacente si ';' care separa linii adiacente. De exemplu, se considera urmatoarele valori pentru variabila x

Scalar real	$>> x = 5$
Scalar complex	$>> x = 5 + 10j$ (or $>> x = 5 + 10i$)
Vector linie	$>> x = [1\ \ 2\ \ 3]$ (or $>> x = [1, 2, 3]$)
Vector coloana	$>> x = [1; 2; 3]$
Matrice 3*3	$>> x = [1\ \ 2\ \ 3; 4\ \ 5\ \ 6; 7\ \ 8\ \ 9]$

Observatii!

- Elementele complexe ale unei matrice nu trebuie sa fie scrise cu spatii (ex: '$-1 + 2j$' este corect ca element al matricei, dar '$-1\ +\ 2j$' nu este);
- De asemenea '$-1 + 2j$' este intrepretat corect, in schimb '$-1 + j2$' **nu este** (MATLAB interpreteaza **'**$j2$**'** ca numele unei variabile). Se poate scrie '$-1 + j * 2$'.

4.1 Operatii cu numere complexe

Mai jos sunt cateva operatii cu numere complexe:

Scalar complex	$>> x = 3 + 4j$	
Partea reala a lui x	$>> real(x)$	$\Longrightarrow 3$
Partea imaginara a lui x	$>> imag(x)$	$\Longrightarrow 4$
Modulul lui x	$>> abs(x)$	$\Longrightarrow 5$
Argumentul lui x	$>> angle(x)$	$\Longrightarrow 0.9273$
Complex conjugatul lui x	$>> conj(x)$	$\Longrightarrow 3 - 4i$

4.2 Generarea vectorilor

Vectorii pot fi generati folosind comanda ':'. De exemplu, pentru a genera un vector x care ia valori intre 0 si 10 cu pasul 0.5, se poate folosi urmatoarea comanda (genereaza o matrice de $1 * 21$)

```
>> x=[0:0.5:10];
```

Alte metode:

- **linspace** - genereaza un vector prin specificarea primului si ultimului element si a numarului de intervale egale dintre cele doua numere
- **logspace** - este asemanatoare comenzii 'linspace' numai ca intervalele sunt logaritmice.

4.3 Accesul la elementele unui vector

Elementele unei matrice sunt accesate prin specificarea liniei si a coloanei. De exemplu, in matricea A =[1 2 3; 4 5 6; 7 8 9], elementul de pe linia 1 si coloana 3 poate fi accesat scriind:

`>> x=A(1,3)` care va intoarce valoarea 3

Linia 2 poate fi obtinuta scriind

`>> y=A(2,:)` care va intoarce [4 5 6]
unde ':' inseamna "toate coloanele". O submatrice a lui A formata din liniile 1 si 2 si toate 3 coloanele poate fi scrisa

`>> z=A(1:2,1:3)` care intoarce [1 2 3;4 5 6]

5 Operatii cu matrice

MATLAB poate efectua operatii aritmetice, relationale si logice cu matrice.

5.1 Operatii aritmetice cu matrice

Operatiile de baza cu matrice (si bineinteles cu scalari, care sunt un caz special de matrice) sunt: adunare(+), scadere(-), inmultire(*), impartire la dreapta(/), impartire la stanga(\), ridicare la putere($\hat{}$), transpus conjugata(').

Daca dimensiunile matricelor implicate in operatie sunt incompatibile MATLAB intoarce un mesaj de eroare. Impartirea este definita in felul urmator: Solutia ecuatiei $A*x = b$ este $x = A\backslash b$ si solutia ecuatiei $x*A = b$ este $x = b/A$ numai daca A este inversabila si toate matricele sunt compatibile.

Adunarea si scaderea implica operatii element cu element; in schimb inmultirea si impartirea nu. MATLAB ofera si posibilitatea efectuarii operatiilor element cu element punand '.' naintea operatorului dupa cum urmeaza: inmultire(.*), impartire la dreapta(./), impartire la stanga(.\), ridicare la putere(.$\hat{}$), transpusa neconjugata(.').

Diferenta dintre inmultirea de matrice si inmultirea element cu element este vizibila in urmatorul exemplu:

$>> A = [1\ \ 2; 3\ \ 4]$
$A =$

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$$

$>> B = A * A$
$B =$

$$\begin{matrix} 7 & 10 \\ 15 & 22 \end{matrix}$$

$>> C = A. * A$
$C =$

$$\begin{matrix} 1 & 4 \\ 9 & 16 \end{matrix}$$

5.2 Operatori relationali

Sunt definiti urmatorii operatori relationali: mai mic (<), mai mic sau egal (<=), mai mare (>), mai mare sau egal (>=), egal (==), diferit (∼=).

Aceste operatii sunt operatii element cu element si intorc o matrice de unu ($1 = true$) si zero ($0 = false$).

Observatie! Atentie la diferenta dintre '=' si '=='.

5.3 Operatii in bucla si operatii logice

MATLAB contine structurile obisnuite de ex: **for**, **while**, **if** si operatorii logici, ex: \(and), |(or) si (not).

5.4 Functii matematice

MATLAB are un numar mare de functii incorporate, care opereaza pe matrice element cu element. Dintre acestea:
sin, **cos**, **tan**, **asin**, **acos**, **atan**, **exp**, **log**, **log10**, **sqrt**, **abs**, **sign**.

6 Fisiere MATLAB

Exista mai multe tipuri de fisiere MATLAB incluzand scripturi de comenzi MATLAB, fisiere care definesc functii create de utilizator, fisiere care includ rezultate numerice sau grafice.

6.1 Fisiere .m

MATLAB este un limbaj interpretativ, comenzile scrise sunt interpretate in sesiunea curenta a MATLAB. Totusi, este dificil sa scriem secvente lungi de comenzi de fiecare data cand vrem sa realizam un task. Exista doua modalitati de a extinde puterea MATLAB-ului - scripturi si functii. Ambele metode utilizeaza fisiere .m. Avantajul fisierelor .m este ca sunt salvate comenzile si pot fi usor modificate fara a mai rescrie intreaga lista de comenzi.

6.1.1 Scripturi

Scripturile MATLAB sunt secvente de comenzi scrise intr-un editor si salvate intr-un fisier .m. Comenzile se scriu in ordinea executiei. Instructiunile sunt excutate scriind numele fisierului in prompt-ul MATLAB

```
>> fisier
```

Executia fisierelor .m este echivalenta cu scrierea intregii liste de comenzi in prompt-ul MATLAB. Toate variabilele folosite in fisierul .m sunt aduse in mediul de lucru MATLAB (workspace). Workspace-ul, care este liber atunci cand MATLAB-ul este pornit, contine toate variabilele definite in sesiunea MATLAB.

6.1.2 Functii

Un al doilea tip de fisier .m este o functie care are urmatoarea forma:

```
function [iesire1,iesire2]=numefunctie(intrare1,intrare2)
Comanda MATLAB1;
Comanda MATLAB2;
Comanda MATLAB3;
```

Numele fisierului .m corespunzator acestei functii este $numefunctie.m$ si este executat din linia de comanda MATLAB sau dintr-un alt fisier .m prin comanda

```
>> [iesire1,iesire2]=numefunctie(intrare1,intrare2)
```

Observatii!
Comentariile se marcheaza cu '%'.
Afisarea rezultatelor este suprimata prin folosirea `';'`la sfarsitul liniei.
O linie poate fi extinsa pe o alta linie folosind `'...'`la sfarsitul primeia.

6.2 Fisiere .mat

Fisierele .mat sunt fisiere binare comprimate folosite pentru a stoca rezultate numerice. Aceste fisiere pot fi folosite pentru a salva rezultate generate printr-o secventa de intructiuni MATLAB. De exemplu, pentru a salva valorile a doua variabile, `var1` si `var2` in fisierul numefisier.mat vom scrie

```
>> save numefisier.mat var1 var2
```

Pentru a salva toate variabilele curente in acest fisier

```
>> save numefisier.mat
```

Un fisier .mat poate fi incarcat in MATLAB

```
>> load numefisier.mat
```

6.3 Fisiere postscript

Graficele generate in MATLAB pot fi salvate intr-un fisier postscript pentru a fi utilizate mai tarziu. De exemplu, pentru a salva graficul curent vom scrie

```
>> print -dps numefisier.ps
```

Pentru alte optiuni `'help print'`.

6.4 Fisiere diary

O inregistrare a sesiunii MATLAB poate fi tinuta cu comanda diary si salvata intr-un fisier diary. Pentru a incepe inregistrarea unui fisier diary si a-l salva ca un fisier cu numele $numefisier$ vom scrie

```
>> diary numefisier
```

Pentru a marca sfarsitul inregistrarii si a inchide fisierul vom folosi comanda

```
>> diary off
```

7 Grafice

MATLAB contine numeroase comenzi pentru crearea graficelor bidimensionale si tridimensionale. Cea mai des utilizata metoda este `'plot'`,care are mai multe argumente optionale. Un exemplu simplu de utilizare a comenzii este graficul unei functii variabile in timp

Figura1: Graficul functiei $x(t) = te^{-t}cos(2\pi 4t)$

```
t = linspace(0,8,401);   %Defineste un vector de timp de la 0 la 8 secunde cu 401 de puncte
x = t.*exp(-t).*cos(2*pi*4*t);   %Defineste un vector x
plot(t,x)   %Graficul lui x functie de t
xlabel('Timp(s)')   %Eticheta axei timpului
ylabel('Amplitudine')   %Eticheta axei amplitudinii
```

Acest script genereaza graficul din Figura 1.

7.1 Comenzi

Cateva comenzi care genereaza grafice 2D sunt

plot	Grafic in coordonate liniare ca o functie continua
stem	Grafic in coordonate liniare ca o functie discreta
loglog	Axe x si y logaritmice
semilogx	Axa y liniara si axa x logaritmica
semilogy	Axa x liniara si axa y logaritmica
hist	Histograma
polar	Coordonate polare

7.2 Personalizarea graficelor

Cateva dintre cele mai frecvente comenzi

xlabel	Eticheta a axei x
ylabel	Eticheta a axei y
title	Titlu al graficului
grid	Adauga un grid graficului
gtext	Permite pozitionarea textului cu mouse-ul
text	Permite pozitionarea textului la anumite coordonate ale graficului
axis	Permite schimbarea axelor x si y
figure	Creeaza o noua fereastra pentru un grafic
figure(n)	Face figura n figura curenta
hold on	Permite desenarea mai multor grafice pe acelasi sistem de axe de coordo
hold off	Release hold on current plot
close(n)	Inchide figura numarul n
subplot(a,b,c)	Creeaza o matrice de grafice $a * b$ cu c fiind figura curenta
orient	Specifica orientarea unei figuri

8 Semnale si sisteme - comenzi

Urmatoarele comenzi sunt strans legate de semnale si sisteme. Fiecare din aceste comenzi are optiuni care ii extind utilitatea.

8.1 Polinoame

Polinoamele apar frecvent in teoria sistemelor. MATLAB reprezinta polinoamele ca vectorii linie de coeficienti polinomiali. De exemplu, polinomul $s^2 + 4s - 5$ este reprezentat in MATLAB prin `>> p = [1 4 -5]`. In continuare vom lista cateva din cele mai importante comenzi folosite la manipularea polinoamelor.

roots(p)	Radacinile polinomului p
polyval(p,x)	Evaluarea polinomului p la valorile vectorului x
conv(p1,p2)	Produsul polinoamelor $p1$ si $p2$
deconv(p1,p2)	Rezultatul impartirii lui $p1$ la $p2$
poly2str(p,'s')	Afisarea polinomului ca o ecuatie in s
poly(r)	Calcului unui polinom fiind dar un vector coloana de radacini r

8.2 Transformatele Laplace si Z

Transformatele Laplace sunt o unealta importanta pentru analiza sistemelor dinamice continue, iar transformatele Z sunt importante pentru analiza sistemelor dinamice discrete. Cateva comenzi importante pentru manipularea acestor transformate sunt incluse in lista urmatoare.

lsim(SYS,u)	Calculeaza raspunsul in timp al sistemului SYS la intrarea u
step(SYS)	Calculeaza raspunsul la treapta a sistemului SYS
impulse(SYS)	Calculeaza raspunsul la impuls al sistemului SYS
pzmap(n,d)	Diagrama poli-zerouri a sistemului SYS
rlocus(SYS)	Calculeaza locul radacinilior pentru un sistem al carui corespondent in
dlsim(n,d,u)	Calculeaza raspunsul in timp la intrarea u al sistemului cu functia $n(z)$
dstep(n,d)	Calculeaza raspunsul la treapta al sistemului cu functia $n(z)/d(z)$
dimpulse(n,d)	Calculeaza raspunsul la impuls al sistemului cu functia $n(z)/d(z)$
zplane(z,p)	Diagrama poli-zerouri din vectori de poli si zerouri, p si z

Cateva din aceste comenzi functioneaza impreuna cu cateva specificatii ale sistemelor liniare invariante in timp. O asemenea specficatie este in termeni de functie de transfer caz in care `'SYS'`este inlocuit cu `'TF(num,den)'`,unde `'num'` si `'den'`sunt vectorii de coeficienti ai numaratorului si respectiv ai numitorului functiei de transfer.

8.3 Raspuns in frecventa

Exista mai multe functii pentru calculul si afisarea raspunsului in frecventa a sistemelor continue sau discrete date prin functii de transfer.

bode(n,d)	Diagrama Bode a unui sistem continuu cu functia de transfer o rationala $n($
freqs(n,d)	Raspunsul in frecventa pentru un sistem continuu cu functia de transfer $n(s$
freqz(n,d)	Raspunsul in frecventa pentru un sistem cu functia de transfer $n(z)/d(z)$

8.4 Transformate Fourier

fft(x)	Calculeaza transformata Fourier discreta a vectorului x
ifft(x)	Calculeaza transformata Fourier discreta inversa a vectorului x

9 Exemple

9.1 Calcului diagramei poli-zerouri, diagramei Bode si raspunsului la intrare treapta pe baza functiei de transfer

Se da functia de transfer

$H(s) = \frac{s}{s^2+2s+101}$.

MATLAB permite obtinerea diagramei poli-zerouri, diagramei Bode si a raspunsului la treapta.

9.1.1 O solutie simpla

Cea mai simpla metoda de a obtine grafice este de a lasat la latitudinea MATLAB alegerea scalelor, etichetelor si notatiilor cum se face in urmatorul script.

```
num = [1 0]; % Defineste numaratorul polinomial
den = [1 2 101]; % Defineste numitorul polinomial

%% Diagrama poli-zerouri
figure(1) % Creeaza figura 1
pzmap(num,den); % Deseneaza diagrama poli-zerouri in figura 1

%% Diagrama Bode
figure(2) % Creeaza figura 2
bode(num,den); % Deseneaza diagrama Bode in figura 2

%% Raspunsul la intrare treapta
figure(3) % Creeaza figura 3
step(num,den); % Deseneaza raspunsul la treapta in figura 3
```

Acest script produce graficele din Figura 2.

Figura 2. Diagrama poli-zeroruri, diagrama Bode, raspunsul la treapta

9.1.2 O solutie personalizata

Daca dorim sa avem un anumit grad de control asupra afisarii rezultatelor putem adauga etichete, titluri sau denumiri pentru axe, etc.

%% Variabile

t = linspace(0,5,201) % Defieste un vector de timp cu 201 puncte la distante egale de la 0 la 5 secunde

w = logspace(-1,3,201) % Defineste un vector de frecvente cu 201 puncte despartite logaritmic de la 10^{-1} la 10^{3} rad/s

num = [1 0]; % Defineste numaratorul polinomial

den = [1 2 101]; % Defineste numitorul polinomial

[poles,zeros] = pzmap(num,den); % Defineste poles ca un vector de poli si zeros ca un vector de zerouri ale functiei de transfer

[mag,angle] = bode(num,den,w); % Defineste mag si angle ca si amplitudinea si unghiul raspunsului in frecventa in punctul w

[y,x] = step(num,den,t); % Defineste raspunsul in timp y la momentul t

%% Diagrama poli-zeroruri

figure(1) % Creeaza figura 1

subplot(2,2,1) % Defineste figura 1 ca o matrice 2*2 de grafice, urmatorul grafic fiind pe pozitia (1,1)

plot(real(poles),imag(poles),'x',real(zeros),imag(zeros),'o') % Deseneaza diagrama poli-zerouri cu 'x' pentru poli si 'o' pentru zeoruri

title('Diagrama poli-zeroruri'); % Adauga titlu graficului

xlabel('Real'); % Eticheta a axei x

ylabel('Imaginar'); % Eticheta axei y

axis([-1.1 0.1 -12 12]); % Defineste axele x si y
grid; % Adauga un grid

%% Diagrama de amplitudine Bode
subplot(2, 2, 2); % Urmatorul grafic se va afla pe pozitia (1,2)
semilogx(w,20*log10(mag)); % Amplitudinea(decibeli) functie de frecventa in w (scara logaritmica)
title('Diagrama Bode de amplitudine');
ylabel(Amplitudine (dB));
xlabel(Frecventa (rad/s));
axis([0.1 1000 -60 0]);
grid;
subplot(2, 2, 4);
semilogx(w,angle); % Plot angle logarithmically in w and linearly in angle
title(Angle of Bode Diagram);
ylabel(Angle (deg));
xlabel(Radian Frequency (rad/s));
axis([0.1 1000 -90 90]);
grid;

%% Raspunsul la treapta
subplot(2,2,3);
plot(t,y);
title(Raspunsul la treapta);
xlabel(Timp (s));
ylabel(Amplitudine);
grid;

Acest script genereaza graficul din Figura 3.

Figura 3. Diagrama poli-zerouri, diagrama Bode, raspunsul la intrare treapta unitate

9.2 Locul radacinilor

In analiza sistemelor este util sa determinam locul radacinilor unui polinom in timp de un parametru se modifica. Un exemplu bun este urmarirea parametrilor unui sistem in bucla inchisa in timp ce amplificarea in bucla deschisa se modifica. De exemplu, pentru sistemul din figura 4 functia de transfer este

$$H(s) = \frac{Y(s)}{X(s)} = \frac{G(s)}{1 + KG(s)}$$

Daca G(s) este o functie rationala in s atunci poate fi exprimata

$$G(s) = \frac{N(s)}{D(s)}$$

unde $N(s)$ si $D(s)$ sunt polinoame. Vom obtine

Figura 4. Sistem cu reactie negativa si amplificare in bucla deschisa $G(s)$

$$H(s) = \frac{N(s)}{D(s) + KN(s)}$$

Problema de a gasi polii lui H(s) in timp ce K variaza se reduce la a gasi radacinile polinomului $D(s) + KN(s)$. MATLAB ofera o functie pentru a gasi locul radacinilor numita '**rlocus**'.

Urmatorul script genereaza locul radacinilor pentru $G(s)$

$$G(s) = \frac{s}{s^2 + 2s + 101}$$

```
num = [1 0];
den = [1 2 101];
figure(1);
rlocus(num,den)
```

Graficul este in Figura 5.

Comanda 'rlocus' poate fi folosita si in alte contexte. De exemplu, sa presupunem ca avem un circuit R, L, C serie cu admitanta

$$Y(s) = \frac{1}{L}\frac{s}{s^2 + Rs + 2}$$

si vrem sa obtinem un grafic al locului polilor (radacinile numitorului) cand rezistenta R variaza. Spargem numitorul in doua parti $N(s) = s$ si $D(s) = s^2 + 2$ si folosim 'rlocus' pentru a obtine graficul.

Figura 5. Locul radacinilor pentru polii lui $H(s)$ atunci cand K variaza

9.3 Raspunsul unui sistem liniar invariant in timp

Comanda MATLAB **lsim** calculeaza raspunsul unui sistem liniar invariant in timp cu functia de transfer $H(s)$ la o intrare $x(t)$.

Se da

$$H(s) = \frac{5s}{s^2 + 2s + 101}$$

si vrem sa calculam raspunsul sistemului la intrarea $x(t) = cos(2\pi t)u(t)$. Urmatorul script calculeaza si afiseaza raspunsul.

Figura 6. Raspunsul unui sistem liniar invariant in timp

```
figure(1);
num = [5 0];
den = [1 2 101];
t = linspace(0, 10, 401);
u = cos(2πt)
[y,x] = lsim(num,den,u,t);
plot(t,y,'r',t,u,'b');
xlabel('Timp(s)');
ylabel('Amplitudine');
```

Graficul este afisat in Figura 6.