Laborator 11. Funcţii definite de utilizator în Matlab

Bibliografie

1. G. Anastassiou, I. Iatan, "Intelligent Routines: Solving Mathematical Analysis with Matlab, Mathcad, Mathematica and Maple", Springer, 2013.
2. I. Iatan, "Îndrumător de laborator în Matlab 7.0", Ed. Conspress, Bucureşti, 2009.
3. R. Trandafir, I. Iatan, *Modelare- Simulare. Noţiuni teoretice şi Aplicaţii*, Ed. Conspress, Bucureşti, 2013.
4. D. Xue, Y. Chen, *Solving Applied Mathematical Problems with Matlab*, Taylor & Francis Group, 2009.

Un program Matlab 7.0 poate fi scris sub forma fişierelor *script* sau a fişierelor *function*.

Fişierele ce conţin instrucţiuni Matlab 7.0 poartă denumirea de fişiere- M datorită faptului că au extensia "m" .

Un fişier *script* este un fişier- M, ce conţine o secvenţă de comenzi Matlab. Pentru execuţia acestei secvenţe de comenzi, se scrie în linia de comandă numele fişierului script.

Fişierele funcţie sunt fişiere- M, care conţine în prima linie cuvântul cheie *function*. Spre deosebire de un script, o funcţie poate lucra cu argumente.

Sintaxa primei linii a unui fişier funcţie este

function [y1,…,yn]=nume_funcţie(x1,…,xn)

unde

- y1,…,yn constituie parametrii de ieşire (în lipsa acestora se elimină parantezele drepte şi semnul egal);
- x1,…,xn reprezintă parametrii de intrare (în cazul lipsei acestora se elimină parantezele rotunde).

După terminarea execuţiei unei funcţii, numai variabilele de ieşire ale acesteia vor rămâne în memoria calculatorului, în timp ce în cazul unui *script* rămân în memorie toate variabilele cu care acesta a operat.

Dintre instrucţiunile de control logic din Matlab 7.0 menţionăm: *if*, *else*, *elseif*, *end*, *while*, *for*, *break*.

Instrucţiunea condiţională *if* are forma generală

```
if expresie1
  grup1_instrucţiuni;
elseif expresie2
  grup2_instrucţiuni;
else
  grup3_instrucţiuni;
  end
```

Cele două instrucţiuni *elseif* şi *else*, asociate cu *if* sunt opţionale.

Expresia este de forma

expr1 op expr2

unde *op* este un operator relaţional din tabelul următor.

Operator relaţional	Semnificaţia
==	egal
~=	diferit
<	mai mic
>	mai mare
<=	mai mic sau egal
>=	mai mare sau egal

Se testează expresia 1; dacă aceasta este adevărată atunci se execută grup1_instrucţiuni. Altfel se testează expresia 2. Dacă aceasta este adevărată atunci se execută grup2_instrucţiuni iar în caz contrar se execută grup3_instrucţiuni.

Instrucţiunea repetitivă *for* este utilizată pentru repetarea unui grup de instrucţiuni, de un anumit număr de ori şi are sintaxa

```
for index=expr
  grup_instrucţiuni;
end
```

unde

- *index* este numele contorului,
- expr este o expresie de forma

iniţial:pas:final,

în care:

- *iniţial* este prima valoare a contorului,
- *pas* constituie pasul (implicit se consideră 1),
- *final* reprezintă cea mai mare valoare pe care o poate lua *index*.

Instrucţiunea repetitivă *while* se foloseşte în scopul repetării unui grup de instrucţiuni de un număr de ori, determinat de expresia specificată; are formatul

```
while expr
  grup_instrucţiuni;
end
```

Expresia este de forma

expr1 op expr2

unde *op* este un operator relaţional.

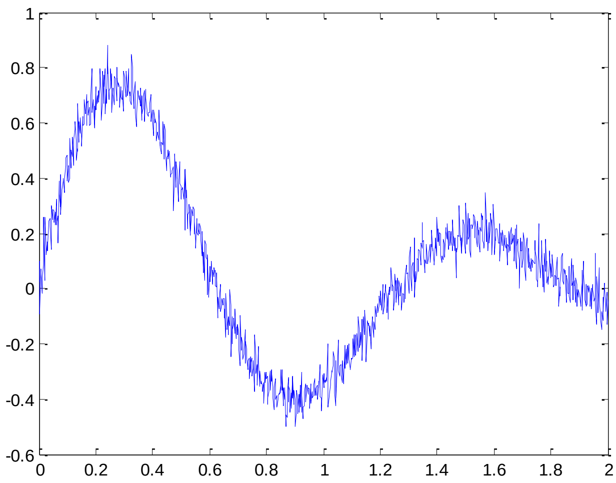
Grupul de instrucţiuni se execută cât timp expresia este adevărată.

Instrucţiunea *break* termină execuţia buclelor *for* şi *while*; în afara acestor bucle, instrucţiunea *break* nu este definită.

Instrucţiunea *end* încheie ciclurile *for*, *while* şi *if*.

Aplicaţii rezolvate

1. Se considera un semnal afectat de zgomot (care rezulta prin suprapunerea unui zgomot Gaussian $N(0, 0.05)$ peste semnalul $y(n) = e^{-n}\sin(5n)$):

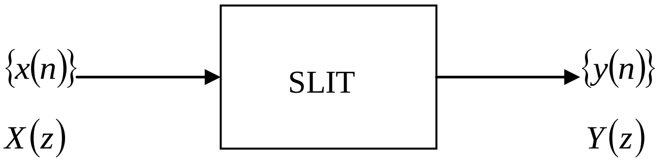


Proiectati un filtru Butterworth de ordinul 5 si apoi un filtru dat de:

$$H(z)=\frac{1.2296\cdot 10^{-6}\left(1+z^{-1}\right)^{7}}{\left(1-0.7265\,z^{-1}\right)\left(1-1.488\,z^{-1}+0.5644\,z^{-2}\right)\left(1-1.595\,z^{-1}+0.6769\,z^{-2}\right)\left(1-1.78\,z^{-1}+0.8713\,z^{-2}\right)}$$

pentru restaurarea semnalului alterat de zgomot.

Filtrele digitale sunt sunt sisteme discrete, liniare si invariante in timp (SLIT).



Raspunsul unui SLIT la un semnal de intrare $x(n)$ se determina:

A) in domeniul de timp

a) pe baza convolutiei dintre semnalele $x(n)$ si $h(n)$ (constituie raspunsul la impuls al filtrului), adica:

$$y(n)=\sum_{k=-\infty}^{\infty} h(k)x(n-k),\quad n\geq 0 \qquad (1)$$

b) prin ecuatia cu diferente

$$y(n)=\sum_{i=0}^{M} b_i\cdot x(n-i)-\sum_{i=1}^{N} a_i\cdot y(n-i),\quad n\geq 0 \qquad (2)$$

B) in domeniul Z

a) relatia (1) poate fi scrisa sub forma:

$$Y(z)=H(z)\cdot X(z). \qquad (3)$$

b) relatia (2) devine (aplicand transformata Z in ambii membri):

$$H(z)=\frac{\sum_{i=0}^{M} b_i\cdot z^{-i}}{1+\sum_{i=1}^{N} a_i\cdot z^{-i}}. \qquad (4)$$

In aa.m se scriu instructiunile:

x=0:0.002:2;

y=exp(-x).*sin(5*x);

r=0.05*randn(size(x));

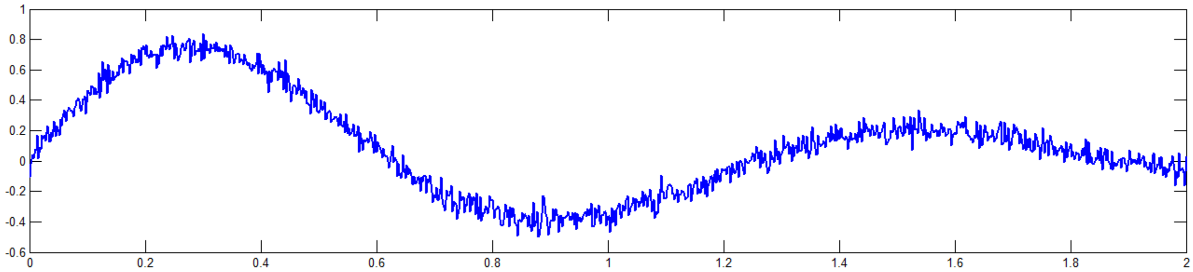
y1=y+r;

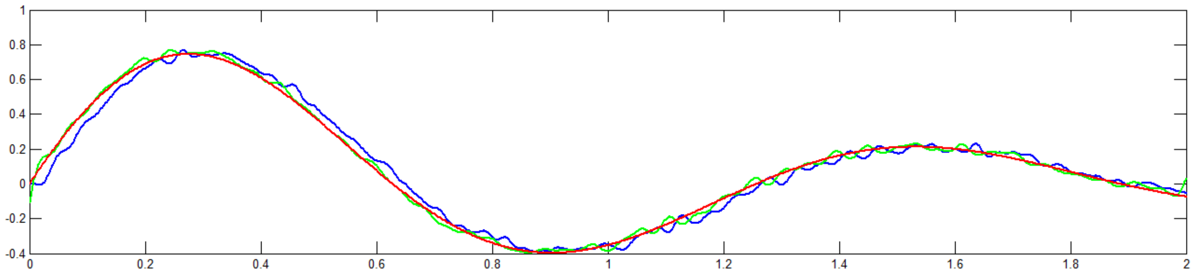
subplot(2,1,1);

```
plot(x,y1)
[b,a]=butter(5,0.1,'low');
y2=filter(b,a,y1);
b1=1.2296*10^-6*conv([1 4 6 4 1],[1 3 3 1]);
a1=conv([1,-0.7265],conv([1,-1.488,0.5644],conv([1,-1.595,0.6769],[1 -1.78 0.8713])));
y3=filtfilt(b1,a1,y1);
subplot(2,1,2);
plot(x,y2,x,y3,x,y)
```

In linia de comanda se lanseaza scriptul in executie:

>>aa





2. Scrieţi un fişier script în Matlab 7.0 care sa calculeze aria cardiodei

$$(C):\begin{cases} x = 2a\cos t - a\cos 2t \\ y = 2a\sin t - a\sin 2t \end{cases}, t \in [0, 2\pi].$$

Aria unei suprafete plane marginita de o curba C poate fi calculata folosind una din urmatoarele formule:

$$\mathrm{A} = \oint_C x\,\mathrm{d}\,y, \tag{1}$$

$$\mathrm{A} = -\oint_C y\,\mathrm{d}\,x, \tag{2}$$

$$A = \frac{1}{2}\oint_C x\,\mathrm{d}y - y\,\mathrm{d}x, \tag{3}$$

(sensul de parcurgere al conturului este sensul trigonometric).

<u>Etapa I</u>. Se selectează succesiv File->New->M-file şi se scriu următoarele instructiuni:

```
syms a t
x=2*a*cos(t)-a*cos(2*t); y=2*a*sin(t)-a*sin(2*t);
xt=diff(x,t); yt=diff(y,t);
int(x*yt,0,2*pi);
```

<u>Etapa II</u>. Se salvează fişierul cu aria.m apoi în linia de comanda se scrie:

```
>>aria
 ans =
 6*a^2*pi
```

3. Calculati cu ajutorul unui fisier de tip function realizat in Matlab 7.0, momentele de inertie in raport cu axele de coordonate pentru o placa de forma domeniului:

$$D = \{(x, y) \in \Re^2 \mid x + y \le 1,\ x \ge 0,\ y \ge 0\}$$

daca densitatea sa este: $\rho(x, y) = xy$.

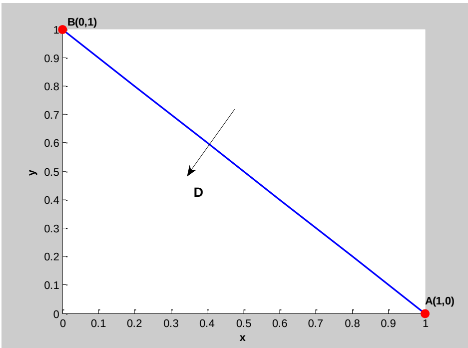
Momentele de inertie in raport cu axele de coordonate Ox si Oy pentru o placa plana, de forma domeniului D si avand densitatea $\rho(x, y)$ sunt respectiv:

$$I_x = \iint_D y^2 \rho(x, y)\,\mathrm{d}x\,\mathrm{d}y \tag{4}$$

$$I_y = \iint_D x^2 \rho(x, y)\,\mathrm{d}x\,\mathrm{d}y. \tag{5}$$

<u>Etapa I</u>. Secventa Matlab urmatoare permite reprezentarea domeniului D.

```
>> x=0:.2:1; y=1-x;
>> plot(x,y,1,0,'or',0,1,'or')
```



Etapa II. Se defineste functia densitate in fisierul ro.m:

```
function r=ro(x,y)
r=x*y;
end
```

Etapa III. Se selectează succesiv File->New->M-file şi se scriu următoarele instructiuni:

```
function [u,v]=momente(x,y)
u=int(int(y^2*ro(x,y),x,0,1-y),y,0,1);
v=int(int(x^2*ro(x,y),y,0,1-x),x,0,1);
end
```

Etapa IV. Se salvează fişierul cu momente.m apoi în linia de comanda se scrie:

```
>>syms x y
>> [Ix,Iy]=momente(x,y)
    Ix =
         1/120
    Iy =
         1/120
```

4. S-au observat un număr de 74 autovehicule pe durata a 25000 km, cât reprezintă perioada de garanţie. Datele, în ordinea producerii ieşirilor din funcţionare se prezintă în tabelul următor:

Autovehicul	t_i	Autovehicul	t_i
1	130	38	10410
2	200	39	11091
3	228	40	11300
4	412	41	11600
5	430	42	12010
6	540	43	12450
7	750	44	12450
8	750	45	12450
9	830	46	12600
10	965	47	12800
11	1000	48	13600
12	1434	49	14080
13	1635	50	15190
14	1981	51	15700
15	2040	52	15700
16	2040	53	15870
17	2280	54	15910
18	2400	55	15980
19	2830	56	16102
20	2901	57	16540

21	2901	58	16600
22	4380	59	17090
23	5140	60	17100
24	5400	61	17120
25	5611	62	17600
26	5900	63	17600
27	6012	64	19130
28	7200	65	19300
29	7200	66	19400
30	7510	67	19400
31	8200	68	21320
32	8430	69	22040
33	9000	70	22040
34	9060	71	22800
35	9111	72	23130
36	9613	73	23130
37	9710	74	23900

Să se estimeze fiabilitatea acestui sistem, utilizând media duratei rămase de funcţionare.

Algoritmul care determină fiabilitatea unui sistem, utilizând media duratei rămase de funcţionare este:

Pasul 1. Intrare: $t_1 \le t_2 \le ... \le t_n$, valorile observate; t^* momentul în care se estimează fiabilitatea.

Pasul 2. Determină o estimaţie a funcţiei media timpului rămas de funcţionare *L(t)*, evaluată la momentul $t = t_i$, pe baza datelor observate este dată de:

$$\hat{L}(t_i) = \hat{\mu}_{n-i} - t_i \ , \ i = \overline{1, n-1},$$

unde media de selecţie pentru ultimele (*n-i*) valori observate este

$$\hat{\mu}_{n-i} = \frac{1}{n-i}\sum_{j=i+1}^{n} t_j \ , \ i = \overline{1, n-1}. \qquad (6)$$

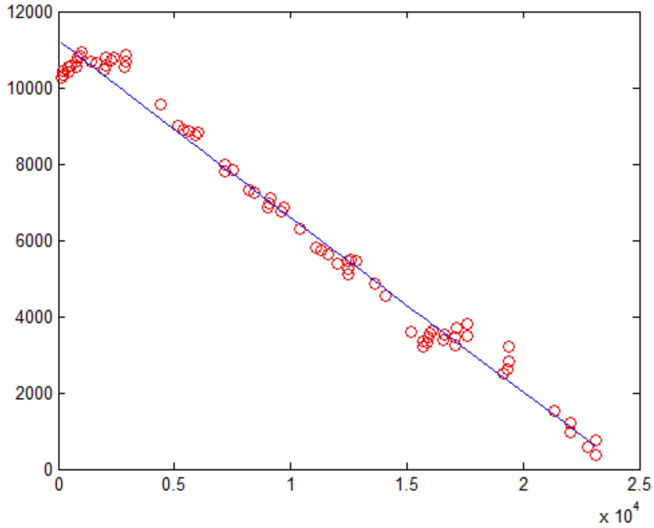
Pasul 3. Se determină expresia analitică a dependenţei $\left(t_i, \hat{L}(t_i)\right)$, $i = \overline{1, n-1}$, folosind metoda celor mai mici pătrate.

Pasul 4. Se estimează fiabilitatea după $t^* = 24000$ km, cu relaţia

$$\overline{F}(t^*) = e^{-\int_0^{t^*} \frac{1+L'(x)}{L(x)}dx}.$$

```
function miu = timpramas(n,t,i)
s=0;
for j=i+1:n
  s=s+t(j);
end
```

```
miu=s/(n-i);
end
>> n=74;
>>fid=fopen('datta.txt','r');
>>t=fscanf(fid,'%f\t',[1,n]);
>>fclose(fid)
>> for i=1:n-1
    miu(n-i)=timpramas(n,t,i);
    L(i)=miu(n-i)-t(i);
    end
    >> tt=t(1:n-1);
    >> coef=polyfit(tt,L,2);
    >> syms x ts
    >> LL=@(x) coef(1)*x.^2+coef(2)*x+coef(3);
    >> plot(tt,L,'ob',tt,LL(tt))
```




```
    >> F=@(ts) exp(-int((1+diff(LL(x),x))/LL(x),x,0,ts));
    >> vpa(F(24000),4)
     ans =
     0.008316
```

Observaţie. Probabilitatea de bună funcţionare a unui sistem este cu atât mai mare, cu cât durata de funcţionare este mai mică.