

Instrucțiuni de control logice.

Obiectivele lucrării de laborator:

- Prezentarea și descrierea instrucțiunilor de control: **for**, **if**, **while**, etc.
- Prezentarea și implementarea parametrului **Callback** – uicontrol, uimenu
- Realizarea unei interfețe grafice pe structura **Script** – **Function** – **Callback**

6.1 Tipuri de instrucțiuni logice

Trebuie menționat faptul că programul Matlab nu este doar un program care execută numai pași secvențiali, operațiile fiind realizate una după alta, ci, pune la dispoziția utilizatorului, la fel ca orice alt mediu de programare și instrucțiuni repetitive a unui set de instrucțiuni atâta timp cât o condiție este realizată sau a unui alt set de instrucțiuni dacă condiția nu este realizată, sau repetarea unor instrucțiuni de un anumit număr de ori. Tabelul 6.1 prezintă o serie de instrucțiuni ce se pot folosi în cazul în care se dorește utilizarea unei bucle de control logice.

Instrucțiunea	Proprietatea
for	Instrucțiune pentru cicluri cu număr specificat de pași
if	Instrucțiune condiționată
else	Instrucțiune asociată cu if
elseif	Instrucțiune asociată cu if
while	Instrucțiune cu condiție logică
break	Instrucțiune pentru forțarea ieșiri din buclă
return	Instrucțiune pentru returnarea execuției la funcția precizată
error	Instrucțiune pentru returnarea unui mesaj de eroare
end	Instrucțiune pentru încheierea ciclurilor for , if , while

Tabel 6.1. Instrucțiuni logice.

6.1.1 Instrucțiunea repetitivă *for*

Această instrucțiune permite repetarea unui grup de instrucțiuni din cadrul buclei, de un anumit număr de ori. O astfel de instrucțiune o putem numi și o structură iterativă cu contor.

Forma generală a unei bucle **for** este:

```
for variabilă = start : pas : final
    instrucțiuni
end
```

unde: *variabilă* reprezintă variabila de contorizare a buclei;

start reprezintă valoarea atribuită inițial variabilei de contorizare;

pas reprezintă pasul de incrementare al variabilei de contorizare;

final reprezintă valoarea finală atribuită variabilei de contorizare;

instrucțiuni reprezintă o succesiune de una sau mai multe comenzi MATLAB;

end marchează sfârșitul buclei **for**.

Exemple:

1. Să se genereze o matrice cu l linii și m coloane:

```
for l=1:3           % l contor de la 1 la 3
    for c=1:5       % c contor de la 1 la 5
        m(l,c)=1; % m matrice de l lini și c coloane
                    % cu toate elementele 1
    end             % inchidem primul for
end               % inchidem al doilea for
m
```

Rezultatul acestor instrucțiuni returnează o matrice cu 3 linii și 5 coloane cu toate elementele 1.

2. Dezvoltarea în serie Fourier a unui semnal este:

$$s = \frac{4 \cdot A}{\pi} \cdot \frac{1}{1} \cdot \sin(1 \cdot \omega t) + \frac{1}{3} \cdot \sin(3 \cdot \omega t) + \frac{1}{5} \cdot \sin(5 \cdot \omega t) + \dots + \frac{1}{2 \cdot n - 1} \cdot \sin((2 \cdot n - 1) \cdot \omega t)$$

Putem să scriem în Matlab un astfel de program sub forma :

```
A=3;           % amplitudinea semnalului
f=50;         % frecventa semnalului
T=1./f;      % perioada semnalului
w=2.*pi.*f;  % pulsatia
dt=0.001;    % pas fix
dt=T./100;   % pas dependent de T
t=0:dt:1.*T; % incarcam vectorul timp
nt=length(t); % nt= lungimea vectorului timp
k=1;         % dam lui k valoarea 1
Na=2;        % numarul de armonici

for m=1:nt;
    x(k,m)=(4.*A./pi).*(1./(2.*k-1)).*sin((2.*k-1).*w.*dt.*(m-1));
% semnalul exprimat in serie Fourier
End

plot(t,x,'-r');grid on;
title('Reprezentare semnal cu o armonica')
xlabel('timp [s]')
```

Rezultatul este prezentat în figura 6.1, unde vizualizăm semnalul descris în serie Fourieri cu o singura componenta armonică.

Dacă dorim să vizualizăm mai multe componente spectrale putem înlocui bucla **for** anterioară cu următoarea buclă:

```
for k=1:Na
    for m=1:nt;
        x(k,m)=(4.*A./pi).*(1./(2.*k-1)).*sin((2.*k-1).*w.*dt.*(m-1));
        % semnalul exprimat in serie Fourier
    end
end

plot(t,x(1,:), 'r', t,x(2,:), 'b');grid on;
title('Reprezentare semnal cu armonici')
xlabel('timp [s]')
```

Rezultatul reprezentării grafice este prezentat în figura 6.2.

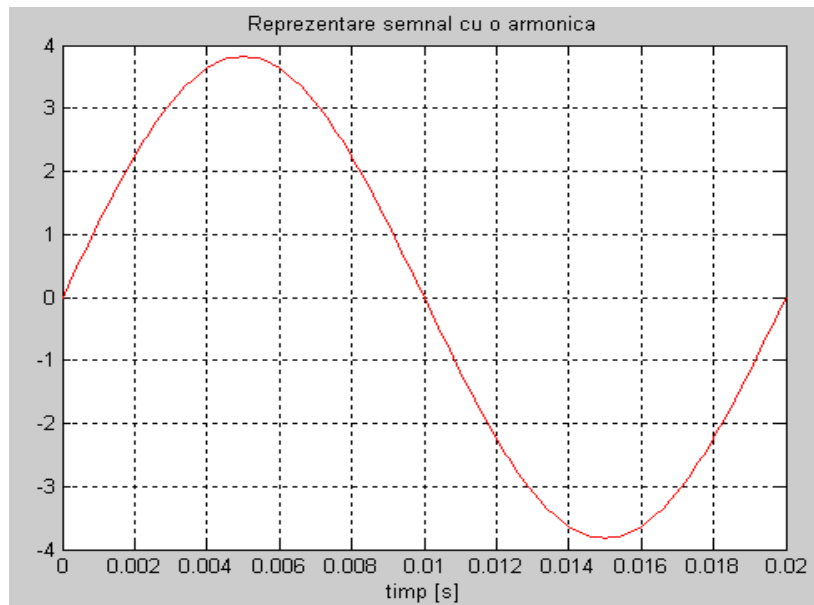


Figura 6.1. Reprezentarea grafică a unui semnal construit din seria fourier cu o armonică

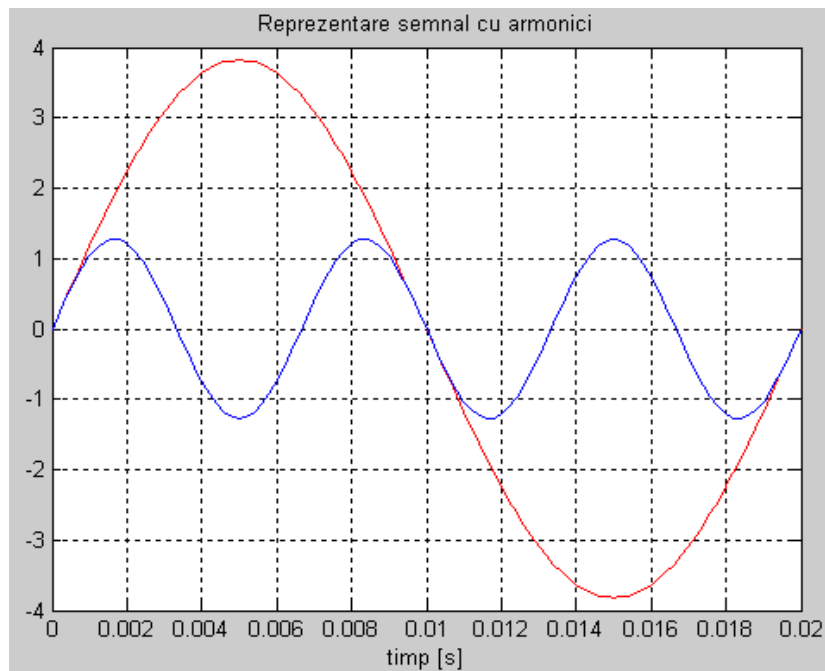


Figura 6.2. Reprezentarea grafică a unui semnal construit din seria fourier cu 2 armonici

6.1.2 Instrucțiunea conditionata *if*

Este o instrucțiune folosită în luarea unei decizii sau pentru luarea unor decizii multiple. Această instrucțiune este condiționată de valoarea de adevăr a unei decizii.

Forma generală a instrucțiunii **if** este

if condiție

 secvență instrucțiuni

end

unde: - **condiție** = expresie logică ce se evaluează.

- **secvență de instrucțiuni** = secvență ce urmează a fi efectuată dacă condiția este adevărată. Dacă condiția nu este adevărată atunci se execută comenzile de după **end**.

Această instrucțiune se poate asocia cu operatorii relaționali și logici pe care programul Matlab îi recunoaște așa cum se prezintă în tabelul 6.2:

Operatorii relaționali	Semnificație
<	Mai mic
<=	Mai mic sau egal
>	Mai mare
>=	Mai mare sau egal
= =	Identic
~ =	Diferit

Tabel 6.2. Operatorii relaționali recunoscuți de programul Matlab

Operatori logici	Simbol	Prioritatea
NU	~	1
SI	&	2
SAU		3

Tabel 6.3. Operatorii logici recunoscuți de programul Matlab

Exemple:

3. Să se evalueze expresia:

```
y=7;  
z=4;  
if z < 5  
    y=y+1  
end  
y
```

Ca rezultat această instrucțiune returnează $y = 8$.

O astfel de instrucțiune **if** poate fi inclusă în interiorul unei alte instrucțiuni **if** la fel ca în cazul instrucțiuni de tip **for** prezentată în exemplul 2.

```
y=7;  
z=4;  
if z < 5  
    y=y+1  
    if z-1>2  
        y=y+5  
    end  
end  
y
```

Rezultatul acestor instrucțiuni returnează $y = 13$. Instrucțiunea **if** evaluează prima condiție și incrementează y , apoi trece la evaluarea celei de-a doua condiții. Dacă condiția este adevărată execută secvența din cel de-al doilea **if**, dacă nu atunci se iese din buclă.

O instrucțiune de decizie tip **if** poate fi folosită și în luarea deciziilor multiple folosind instrucțiunile **if-else**, sau **if-elseif**

4. Considerând funcția:

$$f(x) = \begin{cases} 2x + 8, & \text{daca } x \leq 2 \\ 3x^2, & \text{daca } x > 2 \end{cases}$$

să se scrie un program Matlab care să calculeze în mod automat această funcție.

```
x=5
if x <= 2
    f=2*x+8
else
    f=3.*x^2
end
f
```

Rezultatul acestor instrucțiuni returnează valoarea lui $f = 75$.

În cazul în care funcția ce urmează a fi calculată are mai multe nivele de decizie **if-else** este necesară imbricarea comenzilor. În acest caz vom folosi sintaxa **if-elseif**.

5. Exemplu **if-elseif-end**

$$f(x) = \begin{cases} 2x + 8, & \text{daca } x \leq 2 \\ 3x^2, & \text{daca } x = 3 \\ 5x, & \text{daca } x = 4 \\ 500, & \text{daca } x \neq \text{caz}(1,2,3) \end{cases}$$

```
x=5
if x <= 2
    f=2*x+8
elseif x==3
    f=3.*x^2
elseif x==4
    f=5*x
else
    f=500
end
f
```

Rezultatul instrucțiunilor returnează $f = 500$, deoarece nici una din condițiile buclei nu a fost adevărată, iar astfel, s-a executat doar instrucțiunea **else**.

6.1.3 Instrucțiunea **switch - case**

Această instrucțiune este o instrucțiune similară cu instrucțiunea **if-else**, doar că, atunci, când avem secvențe imbricate complexe se utilizează de regulă instrucțiunea **switch-case**. Sintaxa funcției este de forma:

```
switch expresie
case valoare 1
    secventa 1 de comenzi
case valoare 2
    secventa 2 de comenzi
.....
otherwise valoare n
    secventa n de comenzi
```

6.1.4 Instrucțiunea while

Această buclă de control permite repetarea unei instrucțiuni sau a unui ciclu de instrucțiuni atâta timp cât o condiție logică este îndeplinită.

Forma generală a instrucțiuni **while** este:

```
while conditie
    instructiuni
end
```

unde: **conditie** - este o condiție logică pe care instrucțiunea while o evaluează, iar **instructiuni** reprezintă secvența de instrucțiuni ce se execută atâta timp cât condiția este îndeplinită

! Atenție. În cazul buclelor de tip **while** există posibilitatea ca pentru o condiție dată să se cicleze la infinit. Pentru a evita acest lucru este bine să se introducă un număr maxim de iterații.

Exemplu:

6. Să se determine primul număr întreg n, pentru care factorialul (n!) este un număr de 5 cifre.

Rezolvarea problemei cu ajutorul programului Matlab se poate realiza de forma:

```
n=1;
while prod(1:n)<1.e4
    n=n+1;
end
n
```

Rezultatul instructiunilor returneaza $n = 8$.

Dacă se dorește ieșirea forțată dintr-o buclă de tip **for**, **while**, **if**, se folosește instrucțiunea **break**. Această instrucțiune conferă controlul primei comenzi de după end-ul buclei.

Exemplu:

7. Fie un vector $x = [1 \ 4 \ 8 \ 13 \ 15 \ s \ 14]$. Să se calculeze restul împărțiri la 3 a elementelor vectorului. Dacă se întâlnește elementul 0 să se oprească calculul.

```
x=[1 4 8 13 15 s 14];
nt=length(x)
for i=1:nt
    if x(i)==s
        break
    end
    x(i)=rem(x(i),3)
end
x
```

Rezultatul acestor instrucțiuni este un vector cu elementele $[1 \ 1 \ 2 \ 1 \ 0 \ s \ 14]$. Putem observa că instrucțiunile au fost efectuate până la elementul s , moment în care s-a ieșit din buclă.

Instrucțiunea **return** comandă o ieșire normală din fișierul M-file către funcția care l-a apelat sau către tastatură.

- Sintaxa acestei instructiuni este: **return**

Instrucțiunea **error** permite afișarea unor mesaje la întâlnirea unei erori.

- Sintaxa acestei instructiuni este: **error('mesaj')**

Exemplu:

```
x=3
if x>5
    x=x+1
else
    error('x este mai mic decat 5')
end
```

Exemple cu ajutorul funcțiilor **sum**, **prod**, **rem**.

8. Considerând matricea $M = [1,2,3 ; 4,5,6 ; 7,8,9]$

Dacă aplicăm **sum**, **prod** și **rem** acestei matrici vom obține:

```
M=[1, 2, 3; 4, 5, 6; 7, 8, 9]
suma=sum(M)
produs=prod(M)
rest=rem(M, 2)
```

```
suma = 12 15 18
produs = 28 80 162
rest = [1,0,1 ; 0,1,0 ; 1,0,1]
```

Deci, dacă aplicăm **sum**, **prod** pe o matrice vom obține suma elementelor de pe o coloană, respectiv produsul elementelor de pe o coloană a matricii.

Funcția **rem** calculează restul împărțirii fiecărui element al matricii la scalarul ales.

- Dacă scriem $\text{suma1} = \text{sum}(M(:))$ se obține suma tuturor elementelor matricii. $\text{suma1} = 45$. În mod similar pentru produs.

- Dacă scriem $\text{suma2} = \text{sum}(M(1,:))$ se obține suma elementelor de pe prima linie. $\text{suma2} = 6$. În mod similar pentru produs.

- Dacă scriem $\text{suma3} = \text{sum}(M(:,2))$ se obține suma pe coloana 2.

- Dacă scriem matricea $M1=[M,M ; M,M]$ și apoi $\text{suma4} = \text{sum}(M1(1:5,3:6))$ vom obține $[27 \ 17 \ 22 \ 27]$, adică suma elementelor de pe liniile de la 1 până la 5 și coloanele de la 3 la 6 ale noi matrici formate.

6.2 Desfășurarea laboratorului.

1. Să se realizeze un program Matlab care să permită pe baza celor însușite în lucrarea curentă crearea unei aplicații care să însumeze exemplele prezentate.
2. Să se realizeze o interfață grafică care să permită cu ajutorul unui buton de tip slider apelarea problemei din laborator în funcție de poziția butonului și să returneze rezultatele într-un buton de tip edit.
3. Să se implementeze în funcție de problemă obiecte grafice ajutătoare astfel încât fiecare modificare să se poată realiza fără intervenție în codul programului realizat.
4. Să se creeze 7 butoane de tip text și 7 butoane de tip edit cu ajutorul instrucțiunii **for**. Butoanele de tip text vor avea string-uri de lungimi diferite selectate dintr-un vector șir de caractere, iar cele de tip edit vor avea string-urile selectate dintr-un vector.
5. În variabila lungime se află valoarea numerică a unei măsurători de lungime. Unitatea de lungime cu care s-a făcut măsurătoarea poate fi: 'ft' pentru picior, 'in' pentru inci, 'm' pentru metri și 'mm' pentru milimetru. Să se creeze o secvență **switch-case** care pune în lungime_metri valoarea în metri a măsurătorii. Convertirea se face după:
 - inci-metru $y = 0.0254 * x$
 - picior-metru $y = 0.3048 * x$
 - milimetru-metru $y = x / 1000$
6. Să se determine indicii (linie, coloană) a elementului maxim și a celui minim dintr-o matrice.
7. Temperatura mediului ambiant și în același timp și temperatura unei sticle de răcoritoare este de 30° . Dacă în interiorul unui frigider avem temperatura de 10° , în cât timp reușim să răcorim lichidul din sticla la 14° .

Notăm cu T_f temperatura din frigider și dacă considerăm că T_i este temperatura lichidului la momentul t_i , atunci temperatura T_{i+1} la momentul t_{i+1} este dată de relația:

$T_{i+1} = T_i + K(t_{i+1} - t_i)(T_f - T_i)$, unde $K = 0.05$ este coeficientul de conducție al căldurii. Considerăm că unitățile de măsură a timpului sunt minute $t_{i+1} - t_i = 1$.