

Fișiere de tip Script, Function și Callback - uicontrol

Obiectivele lucrării de laborator:

- Prezentarea și descrierea fișierelor **Script** și **Function**
- Prezentarea și implementarea parametrului **Callback** – uicontrol, uimenu
- Realizarea unei interfețe grafice pe structura **Script** – **Function** – **Callback**

5.1 Fișierele .m de tip Script și Function

Așa cum s-a enunțat și în lucrările anterioare, fișierele Matlab sunt fișiere de tip M-file, ce prezintă extensia **.m**. Aceste fișiere pot să fie: atât fișiere de tip **script** - sunt cele mai simple fișiere MATLAB, nu au argumente de intrare sau de ieșire și sunt utile pentru executarea secvențială a unor calcule care altfel ar trebui executate în mod repetat de la linia de comandă; fișierele de tip script operează cu datele din workspace sau pot crea date noi, iar aceste date sunt disponibile și după terminarea execuției fișierului, sau fișiere de tip funcție (**function**) – fișiere ce se caracterizează prin faptul că acceptă argumente de intrare și returnează valori de ieșire, folosesc atât variabile generale cât și variabile locale declarate în interiorul fișierului de tip funcție și sunt utile pentru extinderea limbajului Matlab pentru rezolvarea diverselor aplicații.

Aceste fișiere de tip funcție, obligatoriu, pe prima linie din editor au sintaxa de forma:

function [prm_ieșire1, prm_ieșire2,...] = nume_funcție (prm_intrare1, prm_intrare2,...)

unde:

- **function** – cuvânt cheie care declară tipul fișierului .m ca fiind fișier de tip funcție
- **prm_ieșire** – lista parametrilor de ieșire pe care funcția îi returnează. Parametri de ieșire sunt cuprinși între paranteze drepte și separați prin virgulă
- **nume_funcție** – numele funcției. **!Atenție:** - numele funcției este implicit și numele fișierului. Dacă fișierul se salvează cu un alt nume decât numele funcției la apelare programul Matlab returnează eroare.

- **prm_intrare** – lista parametrilor de intrare pe care funcția îi primește. Parametri de intrare sunt trecuți între paranteze rotunde și separați prin virgulă.

!Atenție: în cazul în care funcția ce se dorește a fi implementată nu prezintă parametri de ieșire pe care aceasta să îi returneze, aceștia se omit și fișierul are sintaxa de pe prima linie de forma:

function nume_funcție (prm_intrare1, prm_intrare2,...)

Dacă și parametrii de intrare lipsesc, funcția nu are nici un parametru transmis din altă parte (dintr-un fișier de tip script sau dintr-un alt fișier de tip funcție), atunci sintaxa devine de forma: **function nume_funcție.**

Să se realizeze un program Matlab care să permită afișarea funcției: $y = \sin(\omega t)$, permițând totodată modifica amplitudinii semnalului. Se va ține cont că amplitudinea implicită a semnalului este de 1V, iar frecvența de 50Hz.

1. rezolvarea cerinței se poate face simplu prin utilizarea unui fișier de tip M-file, pe liniile cărora vom scrie:

```
clear all;           %goleste variabilele utilizate
close all;          %inchide toate ferestrele Matlab deschise anterior
A=1;               %amplitudinea
f=50;              %frecventa
T=1/f;             %perioada
t=(0:T/100:2*T);  %definim vectorul timp
y=A*sin(2*pi*f*t); %functia y
plot(t,y);         %functia care reprezinta grafic
```

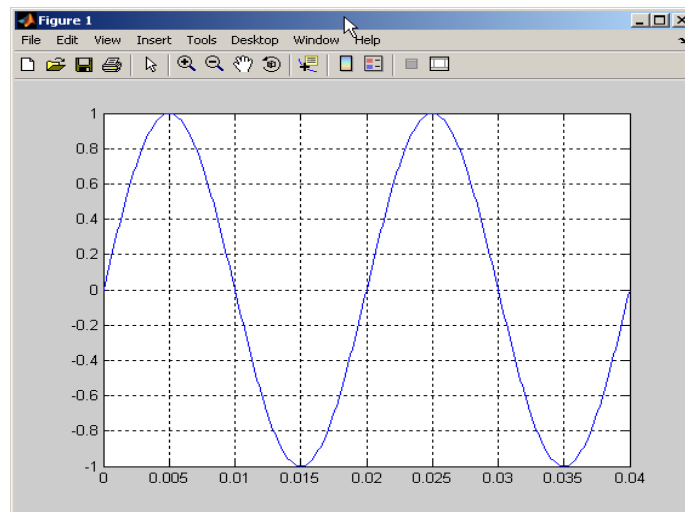


Figura 5.1. Rezultatul instrucțiunilor de la punctul 1.

- utilizarea instrucțiunilor într-un fișier simplu conduce la rezultatul prezentat în figura 5.1. Problema rezolvării constă în modificarea amplitudinii semnalului care s-ar putea realiza doar din liniile de cod prin modificarea valorii variabilei A.
- 2. Soluția corectă pentru rezolvarea cerinței se poate face dacă se folosește un fișier de tip script și un fișier de tip funcție. Astfel, în fișierul de tip script vom introduce valorile inițiale ale funcției iar în fișierul de tip funcție vom rezolva problema propusă. O schemă logică a programului este prezentată în figura 5.2.

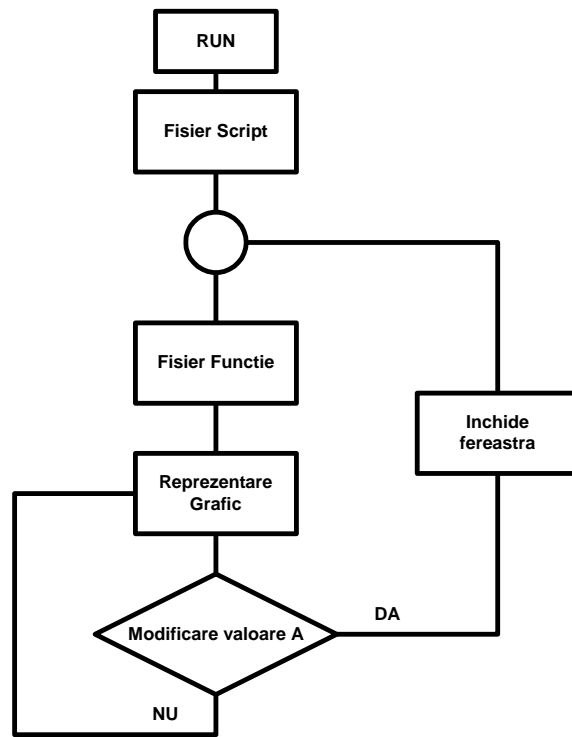


Figura 5.2. Schema logică a problemei propuse.

- Schema logică din figura 5.2 prezintă o imagine de ansamblu a programului Matlab ce trebuie scris. Se evidențiază necesitatea utilizării a două fișiere de tip M-file, un fișier de tip script, fișierul care va porni aplicația și un fișier de tip funcție în corpul căruia vom rezolva atât reprezentarea grafică a funcției y cât și modificarea amplitudinii semnalului de reprezentat grafic.

Fișierul de tip script:

```
clear all;
close all;
A=1;           %amplitudinea
f=50;         %frecventa
sinus(A,f);   %apelarea fisierului de tip functie sinus(A,f)
```

- așa cum se poate observa fișierul de tip script conține în corpul acestuia doar variabilele pe care le impunem implicit funcției y, amplitudinea A și frecvența semnalului f.
- ultima linie de cod conține numele fișierului de tip funcție și parametrii pe care dorim să îi transmitem din fișierul script spre fișierul de tip funcție. **sinus(A,f)**
- vom salva fișierul script cu numele **start_sin**, astfel încât să fie ușor de identificat în momentul apelării

Fișierul de tip funcție

```
function sinus(A,f)

Fig=figure('Name','Reprezentarea sinusului',...
           'Position',[0.3 0.3 0.5 0.5],...
           'NumberTitle','off');

uicontrol('Style','text', ...
          'Units','normalized', ...
          'Position',[0.91 0.95 0.1 0.04], ...
          'String','A');

uicontrol('Style','edit',...
          'Units','normalized',...
          'Position',[0.91 0.9 0.1 0.04],...
          'String',A,...
          'Callback','A=str2num(get(gco, 'string')), close; sinB(A, f);');

T=1/f;
t=(0:T/100:2*T);
y=A.*sin(2.*pi.*f.*t);
plot(t,y);
grid on;

%-----
```

Fișierul de tip funcție are așa cum se observă pe prima linie de cod sintaxa **function** urmată de numele funcției și parametrii de intrare pe care dorim să îi transmitem dinspre fișierul de tip script. Pentru a reuși modificarea amplitudinii s-au implementat două butoane unul de tip text cu ajutorul căruia vom eticheta butonul de tip edit ce va conține valoarea amplitudinii semnalului. Se poate observa că parametri transmiși dinspre fișierul de tip script sunt utilizați în cadrul fișierului de tip funcție, iar o altă serie de variabile sunt definite doar în cadrul acestui fișier.

- parametrul **STRING** al butonului de tip **edit** este de forma: (`'String',A,`), ceea ce implică încărcarea în string a valorii lui A transmis de la script.
- **Callback**-ul butonului permite încărcarea unei noi valori față de valoarea inițială prin câmpul editabil al acestui obiect din interfața grafică creată. Acesta este de forma: (`'Callback', 'A=str2num(get(gco, 'string'))',close;sinus(A,f);'`), unde
 - o **A=str2num** – convertește un șir de caractere introdus în câmpul butonului într-un număr pe care îl atribuie variabilei A
 - o **get** – returnează proprietățile obiectelor: ex: `get(uicontrol)`
 - o **gco** - permite manipularea proprietăților obiectelor: `get(gco,'string')` permite modificarea valorii lui A cu cel introdus din tastatură în câmpul butonului
 - o **close** – închide aplicația și `sinus(A,f)` – reapelează fișierul de tip funcție cu noua valoare a lui A transmisă din butonul de tip edit.

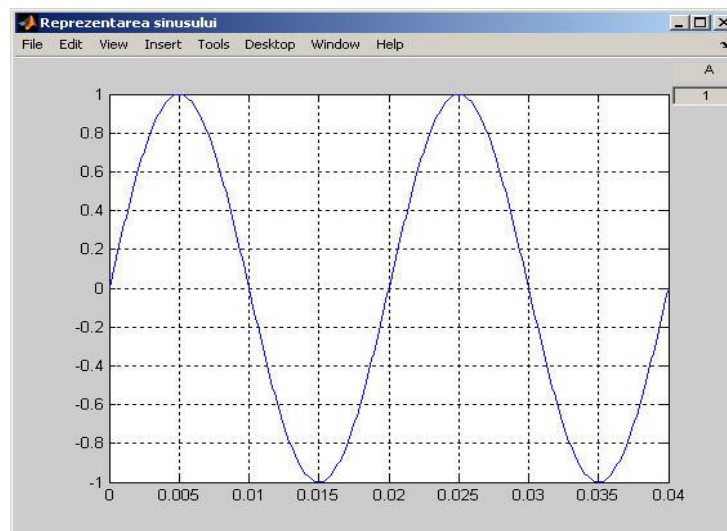


Figura 5.3. Rezultatul rulării fișierului start_sin

Problema 2:

Să se scrie un program care să reprezinte grafic funcțiile sinus, cosinus, suma acestora și să permită modificarea amplitudinii, frecvenței și a numărului de perioade de afișare a celor trei reprezentări grafice obținute, prin manipularea unui buton de tip pushbutton.

Se dau funcțiile : $X1=10\sin(\omega t)$

$X2=5\cos(5\omega t)$, frecvența semnalelor să fie implicit de 50 Hz, iar numărul de perioade de vizualizare să fie 4.

Fișierul de tip script:

```
clear all;
close all;

A=10;          % definim amplitudinea primului semnal
B=5;           % definim amplitudinea semnalului doi
f=50;          % definim frecventa
N=4;           % definim numarul de perioade de vizualizare
interfata(A,B,f,N); %transmitem variabilele fisierului interfata
```

Fișierul de tip funcție:

```
function interfata(A,B,f,N)
Fig=figure('Name','Laborator 4',... %se creaza o noua figura
           'Units','normalized',...
           'NumberTitle','off',...
           'Position',[0.1 0.1 0.8 0.8],...
           'Color',[0.5 0.5 0.9]);

T=1/f;          % definim perioada semnalului
t=0:T/100:N*T; % alegem un interval de timp de reprezentare
x1=A*sin(2*pi*f*t); % prima functie
x2=B*cos(5*2*pi*f*t); % a doua functie
x3=x1+x2;       % scriem ecuatia sumei celor doua functii

uicontrol('Style','pushbutton',... % pushbuton CLOSE
          'Units','normalized',...
          'Position',[0.9 0.9 0.08 .05],...
          'string','Close',...
          'Callback','close');

uicontrol('Style','text',... % Text Buton pentru A
          'Units','normalized',...
```

```
'Position',[0.9 0.83 0.08 .05],...
'backgroundcolor',[0.5 0.5 0.9],...
'string','A');

uicontrol('Style','edit',...           % edit pt A
'Units','normalized',...
'Position',[0.9 0.80 0.08 .05],...
'String',A,...
'Callback',['A=','str2num(get(gca,'String'))']);

uicontrol('Style','text',...          % Text Buton pentru B
'Units','normalized',...
'Position',[0.9 0.73 0.08 .05],...
'backgroundcolor',[0.5 0.5 0.9],...
'string','B');

uicontrol('Style','edit',...          % edit pt B
'Units','normalized',...
'Position',[0.9 0.70 0.08 .05],...
'String',B,...
'Callback',['B=','str2num(get(gca,'String'))']);

uicontrol('Style','text',...          % Buton pentru f
'Units','normalized',...
'Position',[0.9 0.63 0.08 .05],...
'backgroundcolor',[0.5 0.5 0.9],...
'string','f');

uicontrol('Style','edit',...          % edit pt f
'Units','normalized',...
'Position',[0.9 0.60 0.08 .05],...
'String',f,...
'Callback',['f=','str2num(get(gca,'String'))']);

uicontrol('Style','pushbutton',...    % buton de reset
'Units','normalized',...
'Position',[0.9 0.2 0.08 .05],...
'string','RESET',...
'Callback','close;interfata(A,B,f,N)'); %reapelam functia

subplot('position',[0.1 0.72 0.4 0.25]);%setam o fereastră de plotare
plot(t,x1);
grid on;
title('Sinus');
xlabel('timp [s]');
ylabel('Amplitudine [V]');
subplot('position',[0.1 0.38 0.4 0.25]);%alegem o fereastră de plotare
```

```

plot(t,x2);
grid on;
title('Cosinus');
xlabel('timp [s]');
ylabel('Amplitudine [V]');
subplot('position',[0.1 0.05 0.4 0.25]);%alegem o fereastră de plotare
plot(t,x3);
grid on;
title('Suma sin+cos');
xlabel('timp [s]');
ylabel('Amplitudine [V]');
    
```

Rezultatul este prezentat în figura 5.4. **!Atenție** pentru ca programul să fie complet trebuie să adăugați butoanele pentru modificarea numărului de perioade de vizualizare.

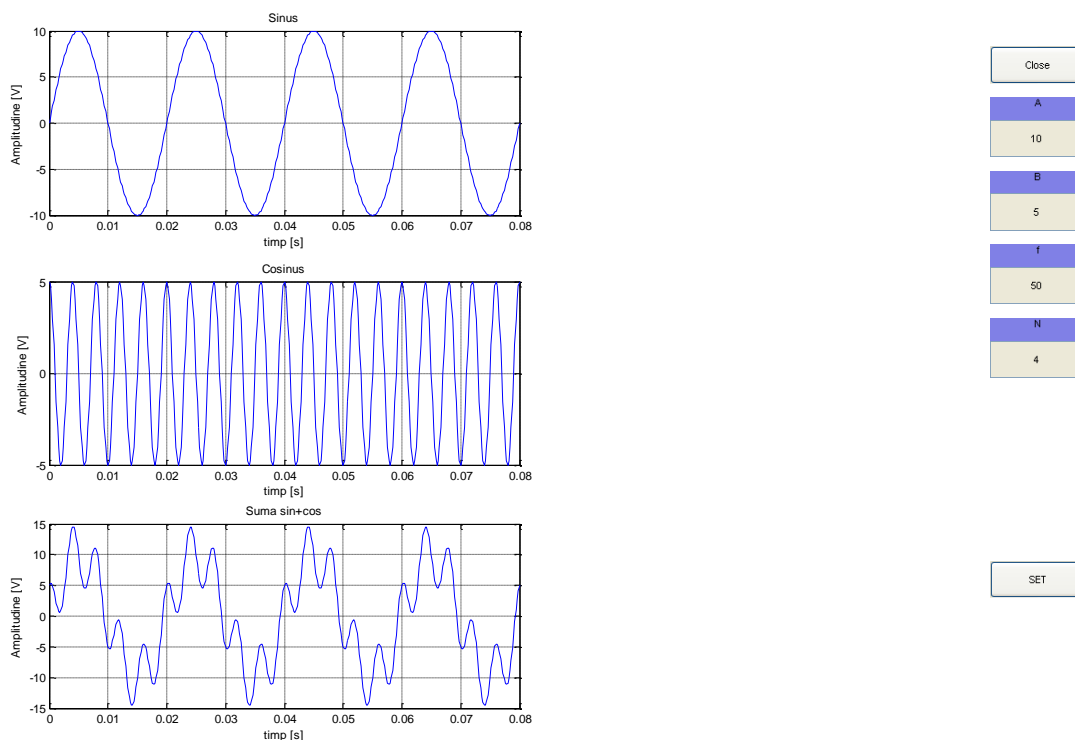


Figura 5.4. Rezultatul problemei 2

5.2 Desfășurarea laboratorului:

1. Adăugați liniile de cod necesare modificării frecvenței pentru problema 1.
2. Adăugați pentru problema 2 liniile de cod necesare astfel încât cele 2 semnale să poată fi vizualizate pe același număr de perioade.
3. Realizați o interfață grafică pentru problema 12 din lucrarea 2.
4. Realizați o interfață grafică precum cea din figura 5.5 astfel:

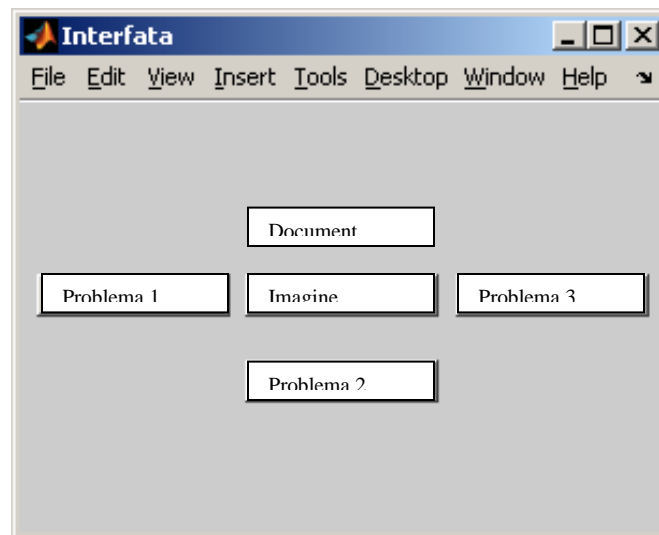


Figura 5.5. Problema 4.

- Folosind 5 butoane de tip pushbutton la manipularea acestora se va realiza callback-ul astfel: Problema 1 – apelează problema 1, Problema 2 – apelează problema 2, Problema 3 – apelează problema 3, Imagine – deschide o imagine, Document – deschide un document. Fiecare fereastră deschisă va avea un buton de închidere.
5. Se întocmește un portofoliu care să conțină: `imread`, `imshow`, `image`, `uicontrol`, `uimenu`, `uipanel`, `callback`, `open`, `cat(2)`, `num2str`, `str2num`, `plot 2D`.
 6. Să se propună 2 funcții care să fie analizate sub forma unor reprezentări grafice cu posibilitatea modificării unor proprietăți dintr-o interfață grafică.