

## TEMA 5 (optionala)

Tema se preda in saptamana a-14 a marti la ultimele sedinte de proiect.

Se realizează un proiect cu interfață grafică folosind mediul GUIDE MATLAB. Prezentarea mediului GUIDE este la sfarsitul documentului.

### Exercitiul 1. Generator de semnal



Realizati interfata grafica pentru un generator de semnal similar cu cel din laboratorul de instrumentatie electronica de masura (cel putin 5 forme de unda, reglaj frecventa, amplitudine, offset). Semnalul generat poate fi salvat in fisier si/sau reprezentat grafic cu plot intr-o fereastră similara cu un osciloscop (nu trebuie facuta interfata grafica pentru un osciloscop).

### Exercitiul 2. IIR Filter design

Se realizeaza o interfata grafica pentru proiectarea filtrelor IIR

- FTJ
- FTS
- FTB
- FOB

De tip

- Butterworth
- Cebasev 1
- Cebasev 2
- Eliptic

Prin metoda

- Invariantei rasp la impuls (cu verificarea conditiilor de proiectare)

- Transformarea biliniara
- Metoda directa

Condițiile de gabarit

- Limitele benzii de trecere si de oprire se introduc de la tastatura
- Riplurile se variaza printr-un slider

Se afiseaza caracteristica dorita, caracteristica de frecventa proiectata si semnal chirp filtrat

### Exercitiul 3. Efecte audio cu sisteme multirate. Resample. Pitch scaling



Se realizează un proiect cu interfață grafică folosind mediul GUIDE MATLAB pentru implementarea unui procesor de efecte audio.

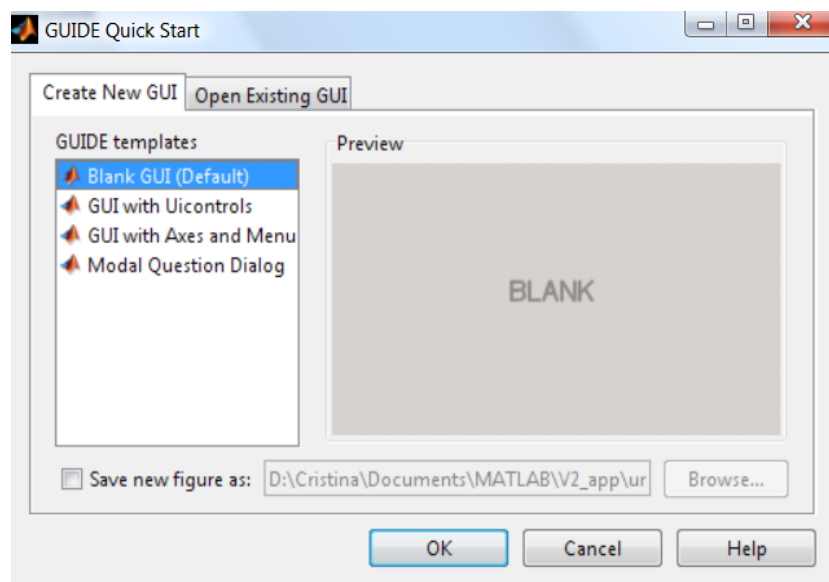
## 5.2. Interfețe grafice

### 5.2.1. Mediul de dezvoltare GUIDE

GUIDE reprezintă un mediu de dezvoltare specific Matlab, în care se pot crea cu ușurință interfețe grafice (Graphical User Interface Development Environment). Se lansează tastând la linia de comandă:

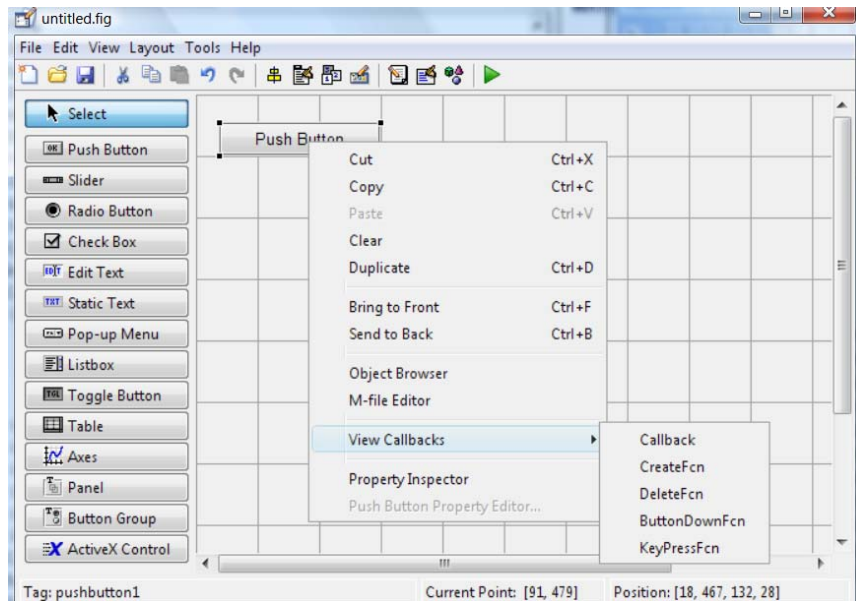
```
guide
```

În urma lansării acestei comenzi, pe ecran va apare următoarea fereastră, în care utilizatorul poate alege un anumit tip de interfață grafică în funcție de cerințele de proiectare; de obicei se alege prima opțiune ”Blank GUI (Default)”.



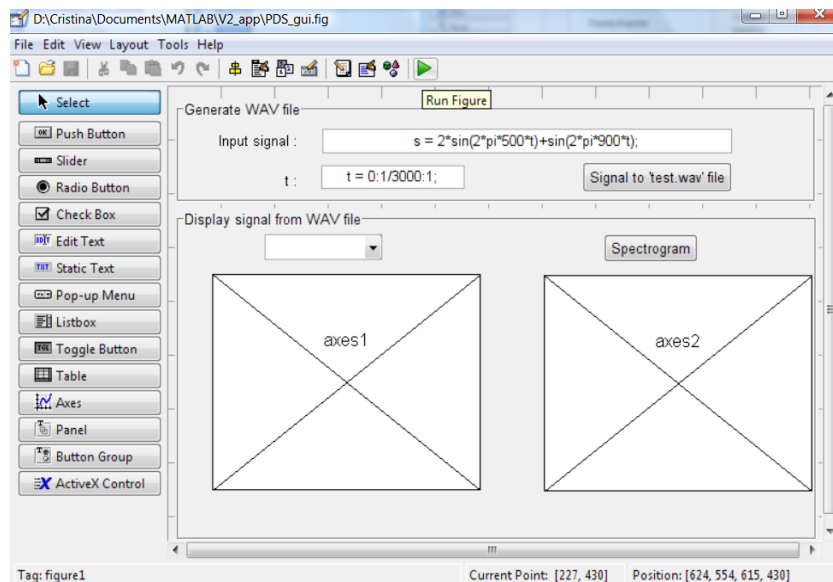
GUIDE oferă posibilitatea de a alege butoane dintr-o listă predefinită, poziționându-le pe suprafața grafică (sau canvas) cu „drag and drop”. Pe măsură ce utilizatorul alege și așează pe canvas diverse obiecte și după ce salvează interfața grafică creată sub forma unui fișier cu extensia .fig, GUIDE creează automat un fișier cu extensia .m care poate fi vizualizat selectând din meniu *View/M-File Editor*. Acest fișier conține toate funcțiile necesare rulării unei interfețe grafice în Matlab într-o formă inițială, adică needitate. Aceste funcții trebuie completate cu codul Matlab corespunzător aplicației care se dorește a fi creată.

De exemplu, pentru a putea vedea funcția chemată atunci când se dă click pe un buton de tip ”pushbutton” se dă click-dreapta pe butonul respectiv și se alege din meniul afișat opțiunea *View Callbacks/Callback*, ca în figură:




### 5.2.2. Interfața grafică "PDS\_gui"

Se deschide exemplul de interfață grafică "PDS\_gui" alegând în fereastra inițială a mediului GUIDE, GUIDE Quick Start, tab-ul *Open Existing GUI* și selectând fișierul *PDS\_gui.fig*. Va apărea următoarea fereastră:



Pentru a putea rula interfața grafică, se selectează opțiunea *Tools/Run* din meniu, sau se apasă butonul verde din meniul Toolbar.

### E2. Exerciții:

1. Pentru exemplul PDS\_gui să se studieze fișierul cu același nume și extensia .m, accesibil direct din meniul Toolbar de la butonul . Se va alcătui o listă cu numele tuturor funcțiilor existente și se va stabili corespondența obiect-funcție acolo unde se poate. De exemplu, primele patru funcții sunt:

```
function varargout = PDS_gui(varargin)
function PDS_gui_OpeningFcn(hObject, eventdata, handles, varargin)
function varargout = PDS_gui_OutputFcn(hObject, eventdata, handles)
function inSignal_Callback(hObject, eventdata, handles)
```

iar ultima funcție corespunde butonului de editare de text cu eticheta *inSignal* în care se introduce semnalul *s*.

Prima funcție nu se editează, ea reprezentând o primă etapă de inițializare. A doua funcție, *PDS\_gui\_OpeningFcn*, este apelată imediat după rularea interfeței grafice (*Run*), înainte ca utilizatorul să poată iniția vreo acțiune asupra acesteia. Aceasta reprezintă o a doua etapă de inițializare, în care proiectantul își declară și definește variabilele necesare.

#### **Observație:**

- Pentru a ști ce etichetă are fiecare buton în parte, se dă dublu-click de exemplu pe butonul de editare a semnalului *s*, deschizând astfel fereastra *Inspector* (sau *Property Inspector*). Aceasta conține toate proprietățile obiectului curent, adică ale butonului de editare de text în acest caz. Eticheta obiectului se găsește în câmpul corespunzător proprietății *Tag*. Observați ce alte proprietăți mai au obiectele din exemplul dat.

2. Să se studieze cu ajutorul Help-ului din Matlab semnificațiile următoarelor noțiuni: *GUI Data* și *handles Structure*. Pentru aceasta se va apela la meniul *Help/Product Help* al principalei ferestre Matlab. Să se noteze pe scurt aceste semnificații.

3. Reluați exercițiul 1 pentru celelalte două interfețe grafice cu care lucrează PDS\_gui: *gui\_player.fig* și *gui\_filter.fig*. Corespondențele dintre funcții și obiecte se află dând click-dreapta pe obiectul respectiv și alegând din meniul afișat opțiunea *View Callbacks/Callback*.

#### **E3. Exerciții:**

1. Pentru interfața grafică PDS\_gui, să se completeze funcția callback corespunzătoare butonului Spectrogram astfel încât, atunci când acest buton este apăsător, să fie afișată spectrograma semnalului *s* dat ca "*Input signal*". Să se verifice pentru un semnal *s* de tip sinusoidal cu frecvența de 800Hz, eșantionat cu  $F_s = 2\text{kHz}$  și generat pe durata 0,5secunde, de amplitudine unitară.

#### **Observație:**

- După completarea callback-ului, salvarea modificărilor, lansarea în execuție cu *Tools/Run*, editarea câmpurilor *Input signal* și *t* cu valorile cerute, trebuie apăsător butonul *Signal to 'test.wav' file*. Astfel, semnalul dat este scris într-un fișier *test.wav* care apoi se va deschide cu *Open* din meniul Toolbar al interfeței grafice.
2. Deschideți pe rând fișierele *baby.wav* și *speech.wav* și vizualizați formele de undă în timp, spectrele semnalelor și spectrogramele. Pentru formele de undă în timp folosiți

opțiunile *Zoom in* și *Zoom out* din Toolbar-ul interfeței grafice. Ce tip de semnal corespunde vocalei *a* ? Vocale *a* se poate identifica cu ușurință în *baby.wav*.

#### E4. Exerciții:

1. Pentru interfața grafică *gui\_player* să se identifice funcția callback asociată obiectului pop-up meniu, având eticheta *signSelect*. În această funcție, primele două linii sunt:

```
str = get(hObject, 'String');  
val = get(hObject, 'Value');
```

Identificați semnificația acestor linii căutând în *Help* sintaxa funcției *get*. Pentru aceasta tastează în *Command Window* comanda

```
help get
```

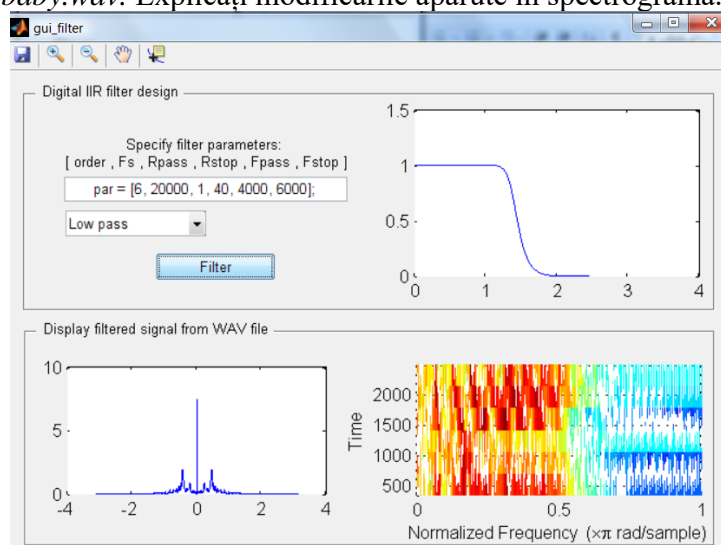
Cu ce se poate înlocui parametrul *hObject* al acestor funcții fără a schimba semnificația și comportamentul lor?

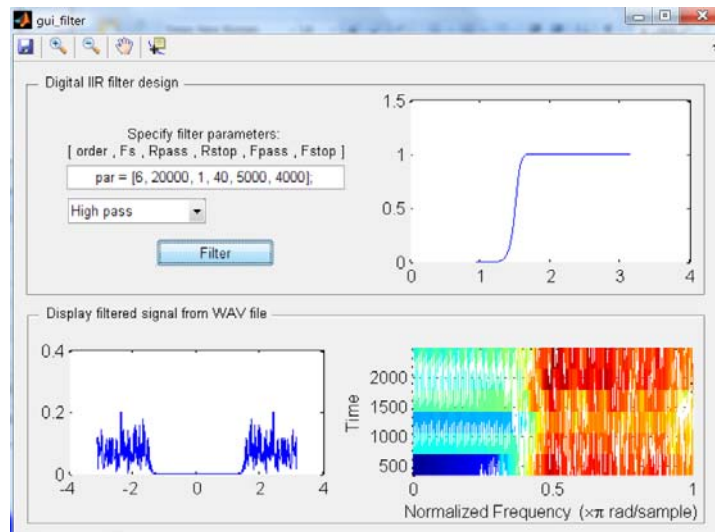
2. În aceeași funcție callback de la exercițiul anterior este creat un player asociat unui singur semnal, fie cel original fie cel filtrat. Studiați funcția *audioplayer* utilizată pentru a crea player-ul; în *Matlab/Help/Product Help* căutați documentația referitoare la *audioplayer*.

#### E5. Exerciții:

1. La interfața grafică *gui\_filter* se va studia modul de implementare a filtrului digital RII. De remarcat faptul că filtrul este reproiectat pentru fiecare modificare adusă parametrilor acestuia în căsuța de editare dedicată lor sau în meniul pop-up care permite alegerea tipului dintre FTJ, FTS, FTB și FOB! Ce metodă a fost folosită pentru proiectarea filtrului RII, directă sau indirectă?

2. În următoarele două figuri este exemplificat rezultatul filtrării trece jos și apoi trece sus a semnalului *baby.wav*. Explicați modificările apărute în spectrogramă.





3. Reluați proiectarea filtrului pentru un filtru de tip eliptic, înlocuindu-l pe cel de tip Butterworth existent. Folosiți următorii parametri:  $F_s = 25\text{kHz}$ ,  $F_{c1}=4500\text{Hz}$ ,  $F_{c2}=6000\text{Hz}$ ,  $F_{b1}=3500\text{Hz}$ ,  $F_{b2}=7000\text{Hz}$ . Filtrați semnalul *speech.wav* și observați diferențele.
4. Reluați filtrele de la exercițiile 2 și 3 și ascultați semnalele original și filtrat cu player-ul din interfața grafică *gui\_player*.
6. Creați o funcție nouă care să primească ca parametri de intrare toți parametrii setați în *gui\_filter* pentru a putea proiecta un filtru RII. Funcția trebuie să returneze coeficienții *b* și *a* ai filtrului. Această funcție va fi apelată de callback-urile asociate obiectelor cu etichetele *filtParam* și *filtType*.