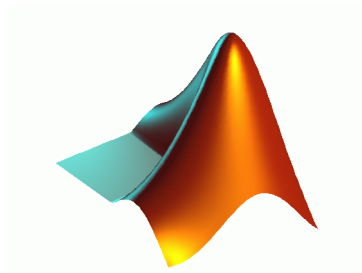




ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ-ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

ΒΟΗΘΗΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΑΞΙΟΠΟΙΗΣΗΣ ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΚΩΝ
ΜΕΤΡΗΣΕΩΝ ΣΕ ΠΕΡΙΒΑΛΛΟΝ GUI (GRAPHICAL USER INTERFACE)
MATLAB

ΣΧΕΔΙΑΣΜΟΣ ΔΙΕΠΑΦΗΣ ΕΠΙΚΟΙΝΩΝΙΑΣ



ΤΖΗΚΑΣ ΒΑΣΙΛΕΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΧΑΝΙΑ 2009

ΕΠΙΒΛΕΠΩΝ

Δρ.Α.Κωνσταντάρης

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Δρ.Κ.Πετρίδης

Α.Ζερβουδάκης

1.ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία πραγματοποιήθηκε υπο την επιβλεψη του καθηγητή Δρ.Αντώνη Κωνσταντάρα . Τον ευχαριστώ για την στήριξη,καθοδήγηση και εμπιστοσύνη που μου έδειξε καθώς και για τον χρόνο που μου αφιέρωσε.

Θα ήθελα να ευχαριστήσω τον καθηγητή και διευθυντή του τμήματος Ηλεκτρονικής Δρ.Ιωάννη Μακρή για την υποστήριξη του κατά την εκπόνηση της εργασίας αυτής.

Ευχαριστώ τον Χρήστο για την πολύτιμη βοήθειά του.

Για την συμπαράσταση όλα αυτά τα χρόνια ευχαριστώ τους φίλους και συμφοιτητές μου.

Ευχαριστώ την οικογένεια μου για την ηθική και οικονομική στήριξη όλα αυτά τα χρόνια .

ΠΕΡΙΕΧΟΜΕΝΑ

1. Ευχαριστιες.....
2. Πρόλογος.....
3. Εισαγωγή.....
4. Παρουσίαση & επεξήγηση κώδικα.....
5. Σημειώσεις-Συμπερασματα.....

2.ΠΕΡΙΛΗΨΗ

Στόχος αυτής της πτυχιακής εργασίας είναι η δημιουργία μιας διέπαφης επικοινωνίας(interface) σε περιβάλλον Matlab gui το οποίο να συντομεύει και να διευκολύνει υπολογιστικές διαδικασίες δεδομένων όπως αυτά είναι δορυφορικές μετρήσεις ηλεκτρομαγνητικών κυμάτων σε διάφορα εδάφη στην Ελλάδα με σκοπό την προβλεψη σεισμικών δονησεων.

Το Matlab gui είναι ένα πρόγραμμα στο οποίο μπορείς να δημιουργήσεις ένα γραφικό περιβάλλον χρησιμοποιώντας τις πολλές συναρτήσεις σε συνδυασμό με τα στοιχεία(components) που περιέχονται μέσα σ' αυτό. Σκοπός αυτής της πτυχιακής εργασίας είναι να δημιουργηθεί ένα πρόγραμμα στο οποίο θα χρησιμοποιήσουμε τις συναρτήσεις και τα στοιχεία αυτά μέσα απ' τα οποία θα δημιουργήσουμε τις δικές μας συναρτήσεις για την ολοκλήρωση του προγράμματος αυτού.

2.ABSTRACT

Objective of this final work is to create a matlab graphical user interface that short and facilitates the calculating procesees data as these are satelitte measurements of electromagnetic waves in various parts in Greece aiming at the forecast of seismic vibrations.

Matlab gui is a program which you can create a graphical user interaface using many of functions in compination with components that are contained in this. Aim of this final work is to create a program in that we will use functions and components giving ready by the program in from that we create our own functions for the completion of this program.

3.ΕΙΣΑΓΩΓΗ

▪ Τι είναι το MATLAB

Το **MATLAB** (MATrix LABoratory) είναι ένα διαδραστικό περιβάλλον για αριθμητικούς υπολογισμούς κυρίως. Μέσω του Symbolic Toolbox μας παρέχει τη δυνατότητα να προβούμε και σε κάποιους συμβολικούς υπολογισμούς. Το 1978 δημιουργήθηκε η πρώτη του έκδοση σε Fortran. Το 1984 η έκδοση MATLAB 1 γράφηκε σε γλώσσα C. Γρήγορα το MATLAB έγινε ένα ιδιαίτερα πολύτιμο και δημοφιλές εργαλείο για επιστήμονες και ερευνητές πολλών επιστημονικών πεδίων. Στην εποχή μας έχουμε φθάσει στην έκδοση MATLAB 7.4 Ο πυρήνας του με τις βασικές εσωτερικές του εντολές είναι αναπτυγμένος σε γλώσσα C, ενώ όλες οι υπόλοιπες εντολές του είναι αναπτυγμένες στη γλώσσα του MATLAB.

Τα κύρια χαρακτηριστικά του είναι τα ακόλουθα:

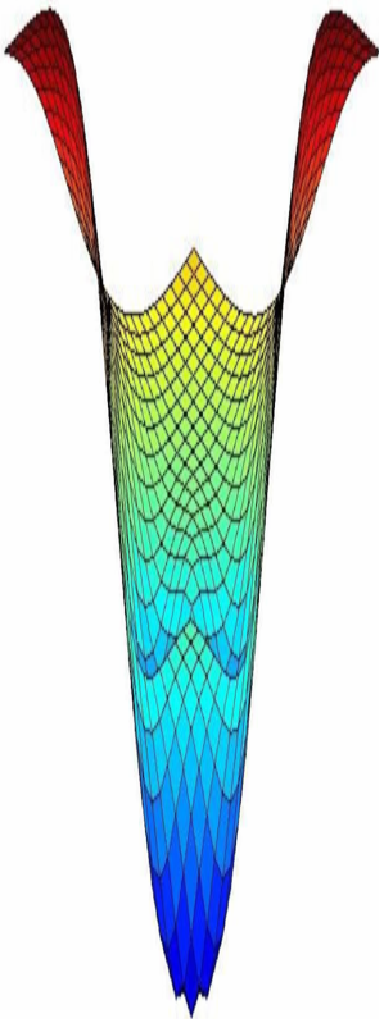
1. Επιτρέπει ταχύ και εύκολο προγραμματισμό σε μία υψηλού επιπέδου γλώσσα.
2. Παρέχει υψηλού επιπέδου εύκολα υλοποιήσιμα γραφικά.
3. Το διαδραστικό περιβάλλον επιτρέπει τον πειραματισμό πάνω στα δεδομένα και εύκολη εύρεση σφαλμάτων.
4. Είναι μία σύγχρονη γλώσσα προγραμματισμού που υποστηρίζει τον αντικειμενοστραφή προγραμματισμό, χειρίζεται τα δεδομένα με ιδιαίτερη ευκολία (π.χ. χωρίς δηλώσεις) . Τα προγράμματα MATLAB μπορούν να μεταφερθούν από ένα σύστημα σε ένα άλλο χωρίς μετατροπή.
5. Υπάρχουν πολλές βιβλιοθήκες (Toolboxes) που δίνουν τη δυνατότητα να χειριστούμε προβλήματα συγκεκριμένου πεδίου π.χ. Fuzzy Logic, Image Processing, Databases, Symbolic Maths κ.λ.π.

▪ Γνωριμία με το Graphical User Interface(GUI)

Το GUI είναι ένα interface ενός προγράμματος. Μπορεί να κάνει τα προγράμματα ευκολότερα στη χρήση παρέχοντας ωραία εμφάνιση και πρακτικές εφαρμογές όπως pushbuttons, list boxes, sliders, menus, κ.α. Το GUI πρέπει να συμπεριφέρεται με ένα κατανοητό και αναμενόμενο τρόπο, έτσι ώστε ο χρήστης να ξέρει τι να περιμένει όταν το χρησιμοποιεί. Για παράδειγμα, όταν με το ποντίκι πιάσουμε ένα pushbutton το GUI θα πρέπει να εκτελέσει την πράξη που περιγράφει η ετικέτα του pushbutton.

▪ Πως δουλεύει το Graphical User Interface

Το Gui παρέχει στο χρήστη ένα οικείο περιβάλλον στο οποίο μπορεί να εργαστεί. Αυτό το περιβάλλον περιέχει pushbuttons , toggle buttons, lists, menus, text boxes κ.α, τα οποία είναι ήδη γνωστά στον χρήστη ,έτσι μπορεί να συγκεντωθεί πάνω στην εργασία που επιθυμεί παραβλέποντας περαιτέρω καθυστερήσεις που θα οφειλόταν σε συνεχή σύνταξη εντολών. Ωστόσο, το GUI είναι δύσκολο για τον προγραμματιστή διότι ένα πρόγραμμα βασισμένο στο GUI πρέπει να προετοιμαστεί για mouse clicks(η για εισαγωγή δεδομένων απ' το πληκτρολόγιο) για κάθε στοιχείο του GUI κάθε φορά. Αυτά τα δεδομένα θα οδηγήσουν το πρόγραμμα σε νέες λειτουργίες. Παρακάτω θα δούμε τα τρία βασικά χαρακτηριστικά που απαιτούνται για να δημιουργηθεί ένα MATLAB Graphical User Interface.

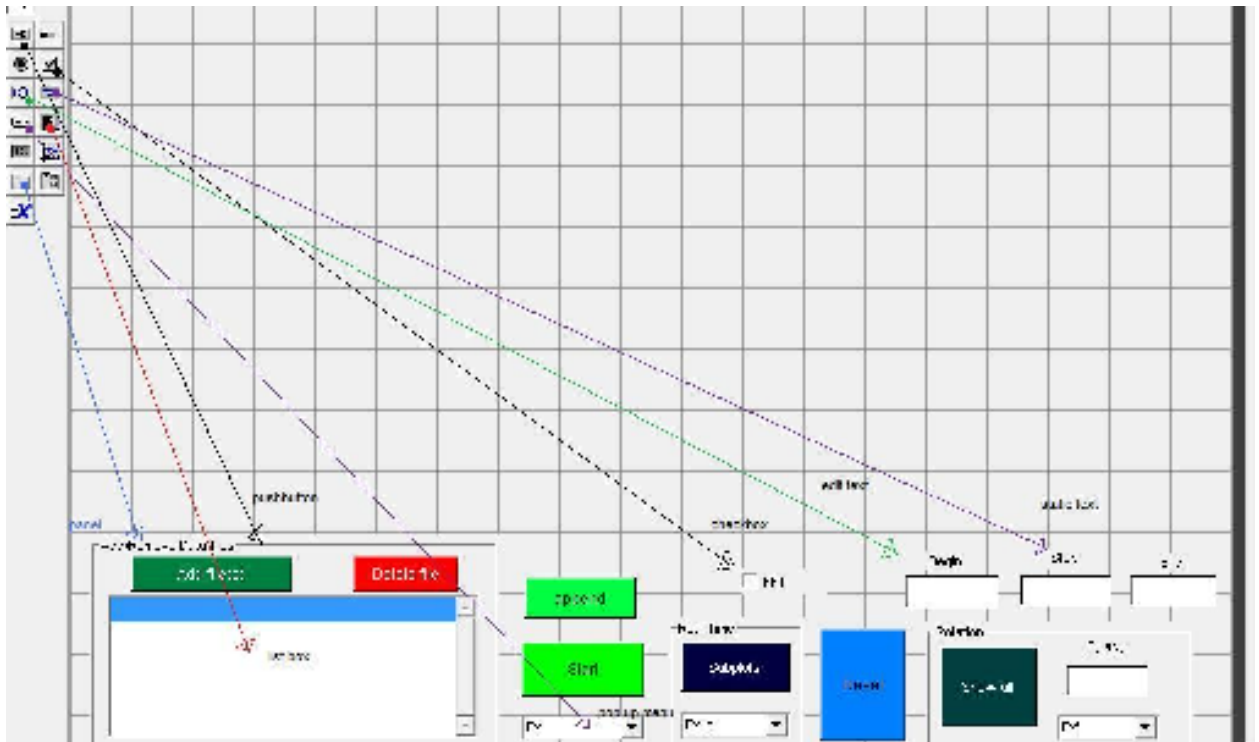


1. Στοιχεία(components). Κάθε στοιχείο του MATLAB GUI(pushbuttons, labels, edit boxes, κτλ.) είναι ένα γραφικό στοιχείο. Μερικά απ'αυτα είναι (pushbuttons, edit boxes, lists, sliders, frames, text strings, menus, axes κ.α.), οι γραφικοί άξονες χρησιμοποιούνται για να εμφανίζουν γραφικά τα δεδομένα.
2. Γραφήματα(plots).Τα στοιχεία ενός GUI πρέπει να τακτοποιούνται μέσα σ'ένα πλαίσιο το οποίο είναι ένα παράθυρο στην οθόνη του υπολογιστή. Τα γραφήματα μπορούν να δημιουργηθούν απο συναρτήσεις και μπορούν να χρησιμοποιηθούν για κάθε πιθανό συνδυασμό των στοιχείων.
3. Ρουτίνες(functions). Στο τέλος πρέπει να υπάρχει ένας τρόπος ώστε να εκτελείται μια δράση αν ο χρήστης κάνει κλικ με το ποντίκι σε ενα button η τυπώσει κάποια πληροφορία με το πληκτρολόγιο. Ενα κλικ με το ποντίκι η το πάτημα μιας πληροφορίας με το πληκτρολόγιο είναι ένα γεγονός, και το MATLAB πρέπει να ανταποκρίνεται σε κάθε γεγονός αν το πρόγραμμα είναι έτοιμο να εκτελέσει μια συνάρτηση. Για παράδειγμα , αν ο χρήστης κάνει κλικ σ'ενα κουμπι, αυτό το γεγονός πρέπει να προκαλέσει τη συνάρτηση που ανάγεται σ'αυτό το κουμπί να εκτελεστεί.Ο κώδικας που εκτελείται εξαιτίας ενός γεγονότος όπως περιγραψαμε πρίν ονομάζεται ρουτίνα. Πρέπει να υπάρχει μια ρουτίνα για να προκαλέσει τη συνάρτηση που υπάγεται σε κάθε γραφικό στοιχείο του GUI.

- **Δημιουργώντας το GUI (Graphical User Interface)**

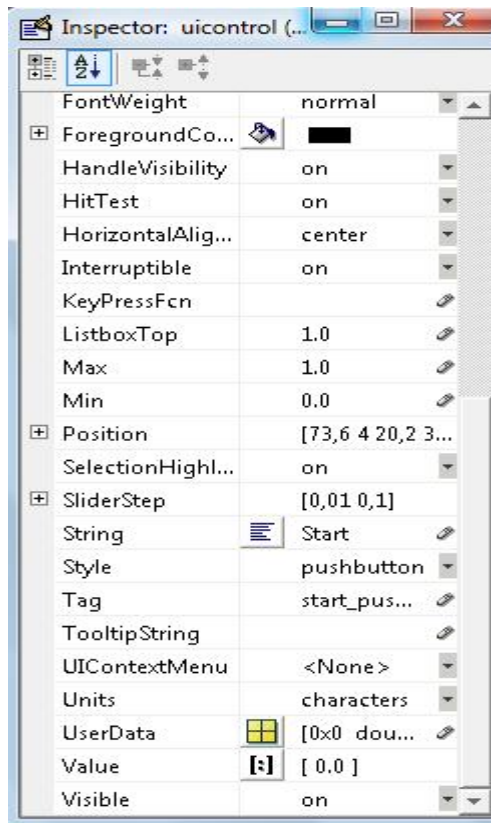
Έχοντας δημιουργήσει το figure(γράφουμε στο command window του Matlab guide επιλέγωντας creating new GUI), είμαστε έτοιμοι να προσθέσουμε σ'αυτό τα στοιχεία που επιθυμούμε.

Όπως παρατηρούμε στο παρακάτω σχήμα το figure που δημιουργήσαμε αποτελείται απο τα εξής στοιχεία: 7 pushbuttons, 4 edit texts, 3 pop up menus, 1 check box, 1 list box , 3 panels και 4 static texts.



Εφόσον έχουμε τοποθετήσει τα στοιχεία πάνω στο figure στη θέση που επιθυμούμε μπορούμε να αλλάξουμε τις ρυθμίσεις κάθε στοιχείου κάνοντας διπλό κλικ πάνω σ'αυτό. Στην συνέχεια θα δούμε τις αλλαγές που κάναμε στα properties(χαρακτηριστικά) του κάθε είδους.

Για να τοποθετήσουμε τα push buttons ή ότι άλλο επιθυμούμε πάνω στο figure πηγαίνουμε με το ποντίκι πάνω αριστερά στο toolbox(εργαλειοθήκη) του προγράμματος και πατάμε την επιλογή push button, στη συνέχεια σχηματίζουμε



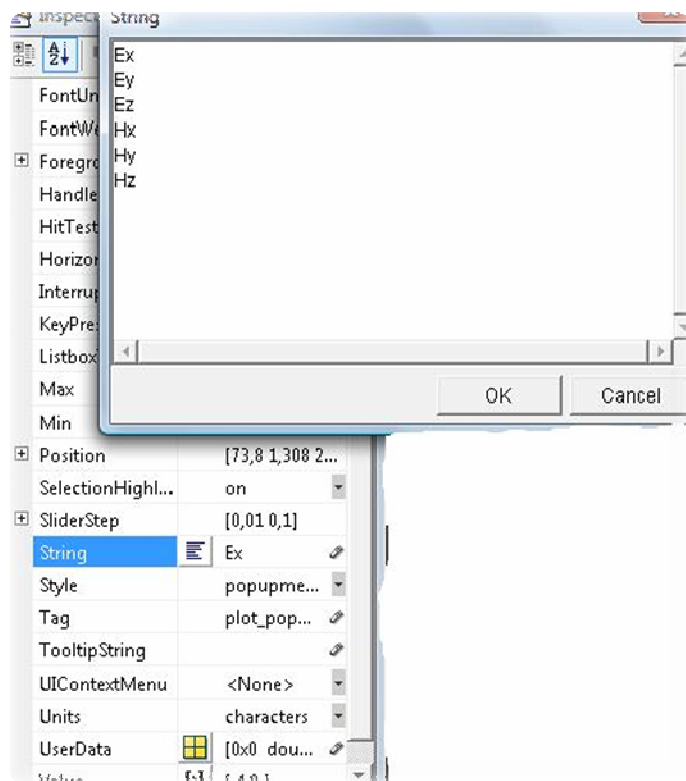
πάνω στο figure το push button όπως εμείς επιθυμούμε. Τώρα μας μένει να αλλάξουμε κάποια απ' τα χαρακτηριστικά του, για παράδειγμα θα δούμε τα χαρακτηριστικά του "start" push button.

Όταν ανοίγουμε τα χαρακτηριστικά ενός στοιχείου μας εμφανίζεται ένα νέο παράθυρο όπως της διπλανής εικόνας.

Το συγκεκριμένο είναι του "start" push button, όπως βλέπουμε τα χαρακτηριστικά που μπορούμε να μεταβάλλουμε είναι αρκετά εμείς όπως και στα περισσότερα στοιχεία του προγράμματός μας ασχοληθήκαμε με τα εξής: το string το οποίο είναι η ονομασία με την οποία θέλουμε να εμφανίζεται το push button στο figure μας, το tag είναι η ονομασία που θέλουμε να έχει το στοιχείο μας στο m-file ,

το background colour το οποίο είναι το χρώμα του push button και το foreground colour το οποίο είναι το χρώμα της γραμματοσειράς.

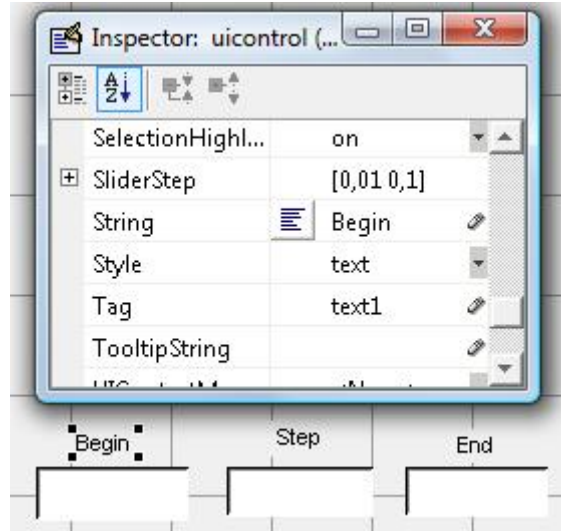
Όμοια προχωρήσαμε και στα υπόλοιπα push buttons.



Στο pop up menu υπάρχει η διαφορά ότι θέλουμε να έχουμε 6 επιλογές(ονομασίες) όπως βλέπουμε στη διπλανή εικόνα. Το παράδειγμα είναι απ' το 1^ο pop up menu που δημιουργήσαμε και έτσι βλέπουμε ότι έχουμε τοποθετήσει διαδοχικά τις επιλογές (ονομασίες) που θέλουμε αφήνωντας για 1^η επιλογή την επιλογή που θέλουμε να βλέπουμε όταν ξεκινάμε το πρόγραμμα.

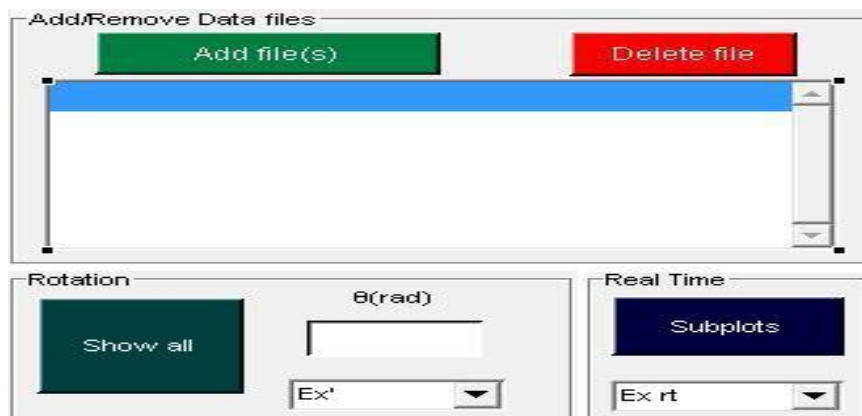
Στα υπόλοιπα pop up menus θα έχουμε πάλι 6 επιλογές μόνο που θ'αλλάζουν οι ονομασίες στα χαρακτηριστικά string και tag.

Στα edit text αφήσαμε το string(ονομασία) κενό αφού σ'αυτό το πρόγραμμα περιμένει κάποιο αριθμό να εισαχθεί, όπως και το background colour λευκό. Για να προσδιορίσουμε τι είναι τα edit texts τοποθετήσαμε απο πάνω τους static texts τα οποία είναι κατι σαν ετικέτες και δώσαμε στο string του κάθε static text την ονομασία που θα προσδιόριζε το edit text. Όπως βλέπουμε και στην εικόνα δεξιά.



Στο listbox επίσης αφήσαμε κενό string αφού σ'αυτό θέλουμε να φορτώνουμε τα data(δεδομένα) αλλά και την ακολουθία των data όταν πιέζουμε το push button "append", όπως και στα edit text αφήσαμε το background colour λευκό. Το tag του το ονομάσαμε inputfiles_listbox και έτσι δημιουργήθηκε κι' αυτό σαν αρχική συνάρτηση στο m-file του προγράμματος.

Τα panel τα χρησιμοποιήσαμε για να ομαδοποιήσουμε κάποια στοιχεία μεσ' το figure. Αυτό το βλέπουμε και στη παρακάτω εικόνα

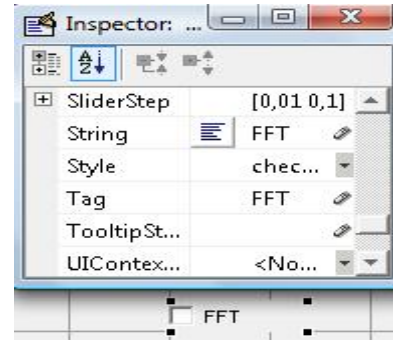


Στο panel με ονομασία(string) add/remove data files τοποθετήσαμε τα push button "add files" και "delete files" καθώς και το listbox.

Στο panel με ονομασία real time τοποθετήσαμε το push button "subplots" και το pop up menu "Ex rt".

Στο panel rotation τοποθετήσαμε το push button show all to edit text με ετικέτα το static text “ θ (rad)” και το pop up menu “Ex”.

Τέλος το check box το ονομάσαμε “FFT”(discrete fourier transform) το οποίο έχει να κάνει με το πλάτος συναρτησει της συχνότητας , το tag του το ονομάσαμε επίσης FFT και έτσι δημιουργήθηκε και η αντίστοιχη FFT_callback στο m-file.



Στη συνέχεια αφού τελειώσουμε το figure και κάνουμε save ,δημιουργούνται 2 αρχια το m-file και το figure. Στο m-file θα δημιουργηθει ο σκελετός του προγράμματος με τις συναρτήσεις(functions) του κάθε στοιχείου έτοιμες να αναμορφωθούν απ’ τον προγραμματιστή όπως αυτός επιθυμεί.

Παρακάτω θα δούμε την παρουσίαση και επεξήγηση του κώδικα όπως αυτή περιέχεται μέσα στο m-file.

4.ΠΑΡΟΥΣΙΑΣΗ-ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ

```
function varargout = tyxh(varargin)
%TYXH M-file for tyxh.fig
%   TYXH, by itself, creates a new TYXH or raises the existing
%   singleton*.
%
%   H = TYXH returns the handle to a new TYXH or the handle to
%   the existing singleton*.
%
%   TYXH('Property','Value',...) creates a new TYXH using the
%   given property value pairs. Unrecognized properties are passed
via
%   varargin to tyxh_OpeningFcn. This calling syntax produces a
%   warning when there is an existing singleton*.
%
%   TYXH('CALLBACK') and TYXH('CALLBACK',hObject,...) call the
%   local function named CALLBACK in TYXH.M with the given input
%   arguments.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help tyxh

% Last Modified by GUIDE v2.5 07-Apr-2009 23:39:59

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @tyxh_OpeningFcn, ...
                  'gui_OutputFcn',  @tyxh_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
%%
function plot_popupmenu_CreateFcn(hObject, handles)

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```

end
%%
function inputFiles_listbox_CreateFcn(hObject, handles)

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%%
function tyxh_OpeningFcn(hObject, handles, varargin)
% Choose default command line output for tyxh
handles.output = hObject;
set(hObject,'toolbar','figure'); %enables toolbar
set(handles.figure1,'CloseRequestFcn',@closeGUI);%enables function
closegui
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes tyxh wait for user response (see UIRESUME)
% uiwait(handles.figure1);

function varargout = tyxh_OutputFcn(hObject, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

function inputFiles_listbox_Callback(hObject, handles)%function for
input files
index_selected = get(hObject,'Value'); % index_selected takes the
value of the data we've input
handles.index={index_selected }; % index takes the value we've
clicked
if(index_selected<=length(handles.data))% if the length of data we've
clicked is similar too the length of the data we've create
    A=handles.data{index_selected}; % A is equal to the array we
choose from listbox when checkbox is disabled
    set(handles.edit1,'string','1');% edit1 text takes value 1
    set(handles.edit2,'string','1');% edit2 text takes value 2
    set(handles.edit3,'string',num2str(length(A))); % edit3 text
takes value the length of array A
    plotData(A,1,1,length(A)); % plotting array A combine with values
of edit texts 1,2,3
else
    errordlg('push start');%error if we click a choise of import data
before we push start button

end

guidata(hObject, handles); % store the structure so far

%%
function addFiles_pushbutton_Callback(hObject, handles)%function for
adding files which are shown up on listbox

```

```

%gets input file(s) from user. the sample data files have extension
.txt
[input_file,pathname] = uigetfile( ...
    {'*.txt', 'Data Files (*.txt)'; ...
    '*.*', 'All Files (*.*)'}, ...
    'Select files', ...
    'MultiSelect', 'on');

%if file selection is cancelled, pathname should be zero
%and nothing should happen
if pathname == 0
    return
end

%gets the current data file names inside the listbox
inputFileNames = get(handles.inputFiles_listbox, 'String');

%if they only select one file, then the data will not be a cell
if iscell(input_file) == 0

    %add the most recent data file selected to the cell containing
    %all the data file names
    inputFileNames{length(inputFileNames)+1} =
fullfile(pathname,input_file);

%else, data will be in cell format
else
    %stores full file path into inputFileNames
    for n = 1:length(input_file)
        inputFileNames{length(inputFileNames)+1} =
fullfile(pathname,input_file{n});
    end
end

%updates the gui to display all filenames in the listbox
set(handles.inputFiles_listbox, 'String',inputFileNames);

%make sure first file is always selected so it doesn't go out of
range
%the GUI will break if this value is out of range

set(handles.inputFiles_listbox, 'Value',1);

% Update handles,hobject structure
guidata(hObject, handles);
%%
function deleteFiles_pushbutton_Callback(hObject, handles)%function
for delete selected data from listbox
%get the current list of file names from the listbox
inputFileNames = get(handles.inputFiles_listbox, 'String');

%get the values for the selected file names
option = get(handles.inputFiles_listbox, 'Value');

%is there is nothing to delete, nothing happens
if (isempty(option) == 1 || option(1) == 0 )
    return
end

```

```

end

%erases the contents of highlighted item in data array
inputFileNames(option) = [];

%updates the gui, erasing the selected item from the listbox
set(handles.inputFiles_listbox, 'String', inputFileNames);

%moves the highlighted item to an appropriate value or else will get
error
if option(end) > length(inputFileNames)
    set(handles.inputFiles_listbox, 'Value', length(inputFileNames));
end

% Update handles structure
guidata(hObject, handles);
%%
function reset_pushbutton_Callback(hObject, handles)%this function is
for reset the gui program
%resets the GUI by clearing all relevant fields

handles.processDataCompleted = 0;

%clears the axes
subplot(4,2,1, 'replace');
subplot(4,2,3, 'replace');
subplot(4,2,5, 'replace');
subplot(4,2,2, 'replace');
subplot(4,2,4, 'replace');
subplot(4,2,6, 'replace');
%clears the contents of the listbox
set(handles.inputFiles_listbox, 'String', '');
set(handles.inputFiles_listbox, 'Value', 0);
set(handles.plot_popupmenu, 'Value', 1);
%restores the first values of all components
handles.data = {};
handles.legendData = {};
handles.index = {};
handles.fft_index = {-1};
handles.fftch_index={0};
handles.popvalue={1};
set(handles.FFT, 'Value', 0);
set(handles.edit1, 'string', '0');
set(handles.edit2, 'string', '0');
set(handles.edit3, 'string', '0');
set(handles.edit8, 'string', '');
set(handles.popupmenu3, 'Value', 1);
set(handles.popupmenu4, 'Value', 1);
%updates handles structure
guidata(hObject, handles);
%%
function start_pushbutton_Callback(hObject, handles)%this function
is for processing data that we've imported in listbox

inputFileNames = get(handles.inputFiles_listbox, 'String');
% %checks to see if the user selected any input files
%if not, nothing happens
if isempty(inputFileNames)
    return
end

```

```

%disables the button while data is processing
disableButtons(handles);
refresh(tyhx);

for x = 1 : length(inputFileNames)
    %gets the filename without the extension
    [ignore,fileName,ext,ignore]=fileparts(inputFileNames{x});
    %store filenames so that it will display on the legend
    handles.legendData(x) = {fileName};
    %stores the numerical data using the custom function
    handles.data{x} = importMyData(inputFileNames{x});
end

enableButtons(handles); %go to enablebuttons function
handles.processDataCompleted = 1;% process of the data completed and
get value 1
guidata(hObject, handles); %update structure
%%
function plot_popupmenu_Callback(hObject, handles)%this function is
for the pop up menu 1 which is for enlarge the selected array

%if the data hasn't been processed yet,
%nothing happens when this button is pressed
if (handles.processDataCompleted == 0)
    return
end
%get the value of the current Pop-up menu selection
%plotType = 1, if Ex option is chosen
%plotType = 2, if Ey option is chosen
%    -||- . -||-
%    -||- . -||-
%    -||- . -||-
plotType = get(hObject, 'Value');%gets the value of the option we
click
handles.popvalue={plotType};%popvalue gets the value of the option

%plots the data
    if(handles.fftch_index{1}==1)%if fftch_index value is 1
        A=handles.data{handles.fft_index{1}};% A is equal to the
array we choose from listbox when checkbox is enabled
        A=A(1:length(A)/2-200,:);%SOS SOS SOS last number trims end
of fft plot, -1 should be its actual value instead of minus whatever,
        else %h praksh panw ginetai gia thn
apofygh tou mirror effect
        A=handles.data{handles.index{1}}; % A is equal to the array
we choose from listbox when checkbox is disabled
    end
n1= str2num( get(handles.edit1, 'String') ); % n1 "edit text 1" gets
the value you type
if(n1<1)%if you type value beneath 1
    n1=1; %edit1 gets value 1
end
n2= str2num( get(handles.edit2, 'String') );% n2 "edit text 2" gets
the value you type
if(n2<1 || n2>length(A))%if you type value beneath 1 or value higher
than length of array A
    n2=1; %edit2 gets value 2
end

```



```

n3= str2num( get(handles.edit3,'String') );% n3 "edit text 3" gets
the value you type
if(n3<1 || n3>length(A))%if you type value beneath 1 or value higher
than length of array A
    n3=length(A);                %edit3 gets value of the
length of array A
end
plotData2(A,plotType,n1,n2,n3);    %go to the plotdata2
function and plot the selected option
if(handles.fftch_index{1}==1)      %if fft checkbox enabled
    xlabel('frequency(Hz)');        %name the x axis
    ylabel('magnitude');            %name the y axis
end
    guidata(hObject, handles);      %update structure
%%
function disableButtons(handles)    % disable components
while data process
set(handles.figure1,'Pointer','watch');
set(handles.start_pushbutton,'Enable','off');
set(handles.reset_pushbutton,'Enable','off');
set(handles.addFiles_pushbutton,'Enable','off');
set(handles.deleteFiles_pushbutton,'Enable','off');
set(handles.inputFiles_listbox,'Enable','off');
set(handles.append,'Enable','off');
set(handles.edit1,'Enable','off');
set(handles.edit2,'Enable','off');
set(handles.edit3,'Enable','off');
set(handles.plot_popupmenu,'Enable','off');
set(handles.FFT,'Enable','off');
set(handles.edit8,'Enable','off');
set(handles.popupmenu3,'Enable','off');
set(handles.showall,'Enable','off');
set(handles.subplots,'Enable','off');
set(handles.popupmenu4,'Enable','off');
%%
function enableButtons(handles)     %enable components
when data process ends
set(handles.figure1,'Pointer','arrow');
set(handles.start_pushbutton,'Enable','on');
set(handles.reset_pushbutton,'Enable','on');
set(handles.addFiles_pushbutton,'Enable','on');
set(handles.deleteFiles_pushbutton,'Enable','on');
set(handles.inputFiles_listbox,'Enable','on');
set(handles.append,'Enable','on');
set(handles.edit1,'Enable','on');
set(handles.edit2,'Enable','on');
set(handles.edit3,'Enable','on');
set(handles.plot_popupmenu,'Enable','on');
set(handles.FFT,'Enable','on');
set(handles.edit8,'Enable','on');
set(handles.popupmenu3,'Enable','on');
set(handles.showall,'Enable','on');
set(handles.subplots,'Enable','on');
set(handles.popupmenu4,'Enable','on');
%%
function []=plotData(data,d,s,u) %function plotdata, inputs: array
"data",edit texts 1,2,3 "d,s,u"

    subplot(4,2,1);                %creating a plot on top and left of the
figure

```

```

        plot(data(d:s:u,7),data(d:s:u,1)); % plot from array "data"
columns 7 & 1 combines edit texts 1,2,3
xlabel('Time (sec)'); %name the axis x
ylabel('Ex (mv/Km)');%name the axis y
grid on %enable grid

        subplot(4,2,3); %creating a plot on middle and left of the
figure
        plot(data(d:s:u,7),data(d:s:u,2));%plot from array "data"
columns 7 & 2 combines edit texts 1,2,3
xlabel('Time (sec)');%name the axis x
ylabel('Ey (mv/Km)');%name the axis y
grid on %enable grid

        subplot(4,2,5); %creating a plot on bottom and left of the
figure
        plot(data(d:s:u,7),data(d:s:u,3));%plot from array "data"
columns 7 & 3 combines edit texts 1,2,3
xlabel('Time (sec)');%name the axis x
ylabel('Ez (mv/Km)');%name the axis y
grid on %enable grid

        subplot(4,2,2); %creating a plot on top and right of the
figure
        plot(data(d:s:u,7),data(d:s:u,4));%plot from array "data"
columns 7 & 4 combines edit texts 1,2,3
xlabel('Time (sec)');%name the axis x
ylabel('Hx (nTesla)');%name the axis y
grid on %enable grid

        subplot(4,2,4);%creating a plot on middle and right of the
figure
        plot(data(d:s:u,7),data(d:s:u,5));%plot from array "data"
columns 7 & 5 combines edit texts 1,2,3
xlabel('Time (sec)');%name the axis x
ylabel('Hy (nTesla)');%name the axis y
grid on %enable grid

        subplot(4,2,6); %creating a plot on bottom and right of
the figure
        plot(data(d:s:u,7),data(d:s:u,6));%plot from array "data"
columns 7 & 6 combines edit texts 1,2,3
xlabel('Time (sec)');%name the axis x
ylabel('Hz (nTesla)');%name the axis y
grid on %enable grid

%%
function [parsedData]= importMyData(name)%this function is for import
data
fid = fopen(name); %open file
parsedData=importdata(name); %import data on array
temp=1:1:length(parsedData);% creating column of time
parsedData=horzcat(parsedData,temp'); % add new column on first data

%%

function edit1_Callback(hObject, handles)%this function is for edit
text1 which is the begin of selected array

```

```

% hObject      handle to edit1 (see GCBO)
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%          str2double(get(hObject,'String')) returns contents of edit1
as a double

if(handles.fftch_index{1}==1) %if fft check box is enable
    A=handles.data{handles.fft_index{1}}; % A is equal to the
array we choose from listbox when checkbox is enabled
else
    A=handles.data{handles.index{1}};% A is equal to the array we
choose from listbox when checkbox is disabled
end
n1= str2num( get(handles.edit1,'String') ); % n1 "edit text 1" gets
the value you type
if(n1<1) %if you type value beneath 1
    n1=1; % then n1=1
end
n2= str2num( get(handles.edit2,'String') );% n2 "edit text 2" gets
the value you type
if(n2<1 || n2>length(A))%if you type value beneath 1 or value higher
than length of array A
    n2=1; % then n2=1
end
n3= str2num( get(handles.edit3,'String') );% n3 "edit text 3" gets
the value you type
if(n3<1 || n3>length(A))%if you type value beneath 1 or value higher
than length of array A
    n3=length(A); % then n3=length of array A
end

plotData(A,n1,n2,n3);%go to function plotdata with inputs array A &
edit texts 1,2,3

temp=get(0,'children');% returns the children-defined value of temp
if(length(temp)>1) %% if length of temp bigger than 1
    plotData2(A,handles.popvalue{1},n1,n2,n3); %go to function
plotdata2 with inputs array A,option of pop up menu 1 & edit text
1,2,3
end

guidata(hObject, handles); %update structure

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, handles)
% hObject      handle to edit1 (see GCBO)
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit2_Callback(hObject, handles)%this function is for edit
text 2 which is the step of selected array
% hObject    handle to edit2 (see GCBO)
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2
as a double
if(handles.fftch_index{1}==1)%if fft check box is enabled
    A=handles.data{handles.fft_index{1}};% A is equal to the
array we choose from listbox when checkbox is enabled
else
    A=handles.data{handles.index{1}};% A is equal to the array we
choose from listbox when checkbox is disabled
end

n1= str2num( get(handles.edit1,'String') );% n1 "edit text 1" gets
the value you typed
if(n1<1)%if you type value beneath 1
    n1=1;% then n1=1
end
n2= str2num( get(handles.edit2,'String') );% n2 "edit text 2" gets
the value you typed
if(n2<1 || n2>length(A))%if you type value beneath 1 or value higher
than length of array A
    n2=1;% then n2=1
end
n3= str2num( get(handles.edit3,'String') );% n3 "edit text 3" gets
the value you typed
if(n3<1 || n3>length(A))%if you type value beneath 1 or value higher
than length of array A
    n3=length(A);% then n3=length of array A
end

plotData(A,n1,n2,n3);%go to function plotdata with inputs array A &
edit texts 1,2,3

temp=get(0,'children');% returns the children-defined value of temp
if(length(temp)>1)%if length of temp >1
    plotData2(A,handles.popvalue{1},n1,n2,n3);%go to function
plotdata2 with inputs array A,option of pop up menu 1 & edit text
1,2,3
end
guidata(hObject, handles); %store handles ,hobject structure

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, handles)
% hObject    handle to edit2 (see GCBO)
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function edit3_Callback(hObject, handles)%function for edit text 3
which is the length of selected array
% hObject      handle to edit3 (see GCBO)
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of edit3
as a double
if(handles.fftch_index{1}==1)%if fft check box is enabled
    A=handles.data{handles.fft_index{1}};% A is equal to the
array we choose from listbox when checkbox is enabled
else
    A=handles.data{handles.index{1}};% A is equal to the array we
choose from listbox when checkbox is disabled
end

n1= str2num( get(handles.edit1,'String') );% n1 "edit text 1" gets
the value you typed
if(n1<1)%if you type value beneath 1
    n1=1;% then n1=1
end
n2= str2num( get(handles.edit2,'String') );% n2 "edit text 2" gets
the value you typed
if(n2<1 || n2>length(A))%if you type value beneath 1 or value higher
than length of array A
    n2=1;% then n2=1
end
n3= str2num( get(handles.edit3,'String') );% n3 "edit text 3" gets
the value you typed
if(n3<1 || n3>length(A))%if you type value beneath 1 or value higher
than length of array A
    n3=length(A);% then n3=length of array A
end

plotData(A,n1,n2,n3);%go to function plotdata with inputs array A &
edit texts 1,2,3

temp=get(0,'children');% returns the children-defined value of temp
if(length(temp)>1)% if length of temp >1
    plotData2(A,handles.popvalue{1},n1,n2,n3);%go to function
plotdata2 with inputs array A,option of pop up menu 1 & edit text
1,2,3
end
guidata(hObject, handles);%store handles ,hobject structure

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject,handles)
% hObject      handle to edit3 (see GCBO)
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

%%
% --- Executes on button press in append.
function append_Callback(hObject, handles)%this function unite the
data we have imported in listbox
% hObject    handle to append (see GCBO)
% handles    structure with handles and user data (see GUIDATA)
final=[];% create an empty cell
t='appended';%t gets label appended
for i=1:length(handles.data),
    A=[handles.data{i}];% A is equal to the array we choose from
listbox when checkbox is disabled
    final=vertcat(final,A(:,1:6));%we took off the 7th column of A
and we added the 2 arrays so the rows are doubled
    b=[handles.legendData(i)];% b takes the name of data we have
added to listbox
    t=strcat(t,'-',b); % t is horizontal concatenation of t and b
end

if (length(final)~= 0)
    temp=1:1:length(final);%temp= length of array "final" which have
concationed with array A
    final=horzcat(final,temp');% final will be the new appended array
i=length(handles.data);% for i = length of array "data"
i=i+1;
handles.data{i}=final;% "data" becomes equal to array final
handles.index{1}=i;%index takes value of i

    j=length(handles.legendData); %j is equal to the number of data
we have in listbox
handles.legendData(j+1) = t;%t value is number of data in list
box + 1
set(handles.inputFiles_listbox,'String',handles.legendData);%
sets the named properties to the specified values of object
n3=length(final);% edit text 3 gets the length of array "final"
n1= str2num( get(handles.edit1,'String') );% n1 "edit text 1"
gets the value you typed
if(n1<1)%if you type value beneath 1
    n1=1;% then n1=1
end
n2= str2num( get(handles.edit2,'String') );% n2 "edit text 2"
gets the value you typed
if(n2<1 || n2>length(final))%if you type value beneath 1 or value
higher than length of array final
    n2=1;% then n2=1
end

set(handles.edit1,'string','1');%edit text 1 get value 1
set(handles.edit2,'string','1');%edit text 2 get value 2
set(handles.edit3,'string',num2str(n3));% edit text 3 get the
length of the array "final" <<num2str>>: converts numbers to their
string representations
plotData(final,n1,n2,n3);% go to plotdata with inputs array
"final" & edit texts 1,2,3
else
errordlg('load file(s) and press start'); %error if we press
start before append
end
end

```

```

guidata(hObject, handles);%update handles,hobject structure
%%
function closeGUI(src,evnt)%this function is for closing gui request
%src is the handle of the object generating the callback (the source
of the event)
%evnt is the The event data structure (can be empty for some
callbacks)
selection = questdlg('Do you want to close the GUI?',...%message in
the window
                    'Close Request Function',...
                    'Yes','No','Yes');
switch selection,%choise yes to close , no to return
    case 'Yes',
        delete(gcf)
    case 'No'
        return
end

%%

function []=plotData2(data,option,d,s,u)%function plotdata 2 is to
enlarge the selected option of pop up menu

temp=get(0,'children');% returns the children-defined value of temp
if(length(temp)>1)% if length of temp bigger than 1
    close(temp(2));%close the 2nd cell of temp
end

if (option==1)%if option 1 is selected from pop up menu 1
    figure;%create figure 1
    plot(data(d:s:u,7),data(d:s:u,1)); % plot from array "data"
columns 7 & 1 combines edit texts 1,2,3
    xlabel('Time (sec)');%name the axis x
    ylabel('Ex (mv/Km)');%name the axis y
    grid on;%enable grid

elseif(option==2)%if option 2 is selected from pop up menu 1
    figure;%create on figure 1
    plot(data(d:s:u,7),data(d:s:u,2)); % plot from array "data"
columns 7 & 2 combines edit texts 1,2,3
    xlabel('Time (sec)');%name the axis x
    ylabel('Ey (mv/Km)');%name the axis y
    grid on;%enable grid

elseif(option==3)%if option 3 is selected from pop up menu 1
    figure;%create on figure 1
    plot(data(d:s:u,7),data(d:s:u,3)); % plot from array "data"
columns 7 & 3 combines edit texts 1,2,3
    xlabel('Time (sec)');%name the axis x
    ylabel('Ez (mv/Km)');%name the axis y
    grid on;%enable grid

elseif(option==4)%if option 4 is selected from pop up menu 1
    figure;%create on figure 1
    plot(data(d:s:u,7),data(d:s:u,4)); % plot from array "data"
columns 7 & 4 combines edit texts 1,2,3
    xlabel('Time (sec)');%name the axis x
    ylabel('Hx (nTesla)');%name the axis y
    grid on;%enable grid

elseif(option==5)%if option 5 is selected from pop up menu 1
    figure;%create on figure 1

```

```

        plot(data(d:s:u,7),data(d:s:u,5)); % plot from array "data"
columns 7 & 5 combines edit texts 1,2,3
        xlabel('Time (sec)');%name the axis x
        ylabel('Hy (nTesla)');%name the axis y
        grid on;%enable grid
elseif(option==6)%if option 6 is selected from pop up menu 1
        figure;%create on figure 1
        plot(data(d:s:u,7),data(d:s:u,6)); % plot from array "data"
columns 7 & 6 combines edit texts 1,2,3
        xlabel('Time (sec)');%name the axis x
        ylabel('Hz (nTesla)');%name the axis y
        grid on;%enable grid

end

% --- Executes on button press in FFT.
function FFT_Callback(hObject, handles)%function fft is where we
transform the single arrays to fft mode
% hObject    handle to FFT (see GCBO)
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of FFT
t=get(hObject,'Value'); %t gets the value of hObject
handles.fftch_index={t};%situation of fft takes value of t
if (t==1) %if check box enabled ,fft on
    A=handles.data{handles.index{1}};%array A is equal to the array
we choose from listbox
    tmp=1:1:length(A);% tmp = length of array A ,step 1
    tmp2=fftshift(abs(fft(A(:,1:6))));% fftshift rearranges the
outputs of fft by moving the zero frequency component to the center
of the array
    temp=horzcat(tmp2,tmp'); %adds the column of time to array tmp2
    j=length(handles.data); % j takes value of the number of
imported arrays we have in listbox
    handles.fft_index = {j+1};%fft_index =j+1 for the new array
(temp1) we want to create below
    handles.data{j+1} = temp;%all arrays of listbox plus the new one
transform usinf function fftshift
    temp1=temp(1:length(temp)/2-200,:); %SOS SOS SOS last number
trims end of fft plot, -1 should be its actual value instead of minus
whatever,
    %temp1 array is created to avoid the mirror effect of fft
    n3=length(temp1);%edit text 3 takes value of length of array
temp1
    n1= str2num( get(handles.edit1,'String') );% n1 "edit text 1"
gets the value you typed
    if(n1<1)%if you type value beneath 1
        n1=1;% then n1=1
    end
    n2= str2num( get(handles.edit2,'String') );% n2 "edit text 2"
gets the value you typed
    if(n2<1 || n2>length(temp1))%if you type value beneath 1 or value
higher than length of array temp1
        n2=1;% then n2=1
    end

    set(handles.edit1,'string','1');%edit text 1 get value 1
    set(handles.edit2,'string','1');%edit text 2 get value 2

```



```

        set(handles.edit3,'string',num2str(n3));% edit text 3 get the
length of the array "templ" <<num2str>>: converts numbers to their
string representations

        plotData3(templ,n1,n2,n3);% go to plotdata with inputs array
"templ" & edit texts 1,2,3

else

        A=handles.data{handles.index{1}};% A is equal to the array we
choose from listbox when checkbox is disabled
        n1= str2num( get(handles.edit1,'String') );% n1 "edit text 1" gets
the value you type
        if(n1<1)%if you type value beneath 1
            n1=1;% then n1=1
        end
        n2= str2num( get(handles.edit2,'String') );% n2 "edit text 2" gets
the value you typed
        if(n2<1 || n2>length(A))%if you type value beneath 1 or value higher
than length of array A
            n2=1;% then n2=1
        end
        n3= str2num( get(handles.edit3,'String') );% n3 "edit text 3" gets
the value you type
        if(n3<1 || n3>length(A))%if you type value beneath 1 or value higher
than length of array A
            n3=length(A);% then n3=length of array A
        end

        plotData(A,n1,n2,n3);%go to function plotdata with inputs array A &
edit texts 1,2,3
    end

guidata(hObject, handles); %store hObject,handles structure

%%
function []=plotData3(data,d,s,u)%plotdata3 is used when we have
enable fft checkbox

        subplot(4,2,1);%creating a plot on top and left of the figure
plot(data(d:s:u,7),data(d:s:u,1)); % plot from array "data"
columns 7 & 1 combines edit texts 1,2,3
        title('Ex');%title of subplot
        xlabel('frequency(Hz)');%name the axis x
        ylabel('magnititude');%name the axis y
        grid on%enable grid

        subplot(4,2,3); %creating a plot on middle and left of the
figure
        plot(data(d:s:u,7),data(d:s:u,2));% plot from array "data"
columns 7 & 2 combines edit texts 1,2,3
        title('Ey');%title of subplot
        xlabel('frequency(Hz)');%name the axis x
        ylabel('magnititude');%name the axis y
        grid on%enable grid

        subplot(4,2,5);%creating a plot on bottom and left of the
figure

```

```

        plot(data(d:s:u,7),data(d:s:u,3));% plot from array "data"
columns 7 & 3 combines edit texts 1,2,3
        title('Ez');%title of subplot
        xlabel('frequency(Hz)');%name the axis x
        ylabel('magnitude');%name the axis y
        grid on;%enable grid

        subplot(4,2,2);%creating a plot on top and right of the
figure
        plot(data(d:s:u,7),data(d:s:u,4));% plot from array "data"
columns 7 & 4 combines edit texts 1,2,3
        title('Hx');%title of subplot
        xlabel('frequency(Hz)');%name the axis x
        ylabel('magnitude');%name the axis y
        grid on;%enable grid

        subplot(4,2,4);%creating a plot on middle and right of the
figure
        plot(data(d:s:u,7),data(d:s:u,5));% plot from array "data"
columns 7 & 5 combines edit texts 1,2,3
        title('Hy');%title of subplot
        xlabel('frequency(Hz)');%name the axis x
        ylabel('magnitude');%name the axis y
        grid on;%enable grid

        subplot(4,2,6);%creating a plot on bottom and right of the
figure
        plot(data(d:s:u,7),data(d:s:u,6));% plot from array "data"
columns 7 & 6 combines edit texts 1,2,3
        title('Hz');%title of subplot
        xlabel('frequency(Hz)');%name the axis x
        ylabel('magnitude');%name the axis y
        grid on;%enable grid

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, handles)%this function is for
enlarge the selected figure of panel rotation
% hObject      handle to popupmenu3 (see GCBO)
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu3 contents
as cell array
%      contents{get(hObject,'Value')} returns selected item from
popupmenu3
if (handles.processDataCompleted == 0)%if data not proceded
    return
end
%get the value of the current Pop-up menu selection
%plotType = 1, if Ex' option is chosen
%plotType = 2, if Ey' option is chosen
%plotType = 3, if Hx' option is chosen
%plotType = 4, if Hy' option is chosen

plotType2 = get(hObject,'Value');%plottype gets choise value we've
choose from pop up menu 3
handles.popvalue1={plotType2};%pop value is confirmed
A=handles.data{handles.index{1}};% A is equal to the array we choose
from listbox when checkbox is disabled

```

```

n4=str2num( get(handles.edit8,'String') );% n4 "edit text 8" gets the
value you type
if(n4<1)%if you type value beneath 1
    n4=1;% then n4=1
end

if plotType2==1%choise of pop up menu3 is 1
    N=(cos(n4).*(A(:,1)))-(sin(n4).*(A(:,2))); %equation of N1 using
edit text8 as n4
elseif plotType2==2%choise of pop up menu3 is 2
    N=(sin(n4).*(A(:,1)))+(cos(n4).*(A(:,2)));%equation of N2 using
edit text8 as n4
elseif plotType2==3%choise of pop up menu3 is 3
    N=(cos(n4).*(A(:,4)))-(sin(n4).*(A(:,5)));%equation of N3 using
edit text8 as n4
elseif plotType2==4%choise of pop up menu3 is 4
    N=(sin(n4).*(A(:,4)))+(cos(n4).*(A(:,5)));%equation of N4 using
edit text8 as n4
end
plotData4(A,plotType2,N);%go to function plotdata4 with inputs array
A,plottype2,and N
guidata(hObject, handles);      %hObject is the component we have
clicked
% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject,handles)
% hObject      handle to popupmenu3 (see GCBO)
% handles      empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject,handles)
% hObject      handle to edit8 (see GCBO)
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%      str2double(get(hObject,'String')) returns contents of edit8
as a double
A=handles.data{handles.index{1}};% A is equal to the array we choose
from listbox when checkbox is disabled
n4=str2num( get(handles.edit8,'String') );% n4 "edit text 8" gets the
value you type
if(n4<1)%if you type value beneath 1
    n4=1;% then n4=1
end
plotData4(A,n4);%go to f. plotdata4 with inputs array A &edit text8
guidata(hObject, handles);%update handles,hobject structure
% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, handles)
% hObject      handle to edit8 (see GCBO)
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.

```

```

%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','black');
end

function []=plotData4(data,option,N)%plotdata4 is to shown off the
data which are in rotation panel

temp=get(0,'children');% returns the children-defined value of temp
if(length(temp)>1)% if length of temp bigger than 1
    close(temp(2));%close the 2nd cell of temp
end

if (option==1) %if choise 1 is selected from pop up menu 3
    figure;%create on fig1
    plot(data(:,7),N);%plot from array data x axis time and y
axis N1 data which we've created on pop up menu 3 callback
    xlabel('Time (sec)');%name the axis x
    ylabel('Ex rotated(mv/Km)');%name the axis y
    grid on;%enable grid

elseif(option==2)%if choise 2 is selected from pop up menu 3
    figure;%create on fig1
    plot(data(:,7),N);%plot from array data x axis time and y
axis N2 data which we've created on pop up menu 3 callback
    xlabel('Time (sec)');%name the axis x
    ylabel('Ey rotated(mv/Km)');%name the axis y
    grid on;%enable grid
elseif(option==3)%if choise 3 is selected from pop up menu 3
    figure;%create on fig1
    plot(data(:,7),N);%plot from array data x axis time and y
axis N3 data which we've created on pop up menu 3 callback
    xlabel('Time (sec)');%name the axis x
    ylabel('Hx rotated(nTesla)');%name the axis y
    grid on;%enable grid
elseif(option==4)%if choise 4 is selected from pop up menu 3
    figure;%create on fig1
    plot(data(:,7),N);%plot from array data x axis time and y
axis N4 data which we've created on pop up menu 3 callback
    xlabel('Time (sec)');%name the axis x
    ylabel('Hy rotated(nTesla)');%name the axis y
    grid on;%enable grid
end

% --- Executes on button press in showall.
function showall_Callback(hObject, handles)%this function plots in a
fig the rotated data
% hObject    handle to showall (see GCBO)
% handles    structure with handles and user data (see GUIDATA)
A=handles.data{handles.index{1}};% A is equal to the array we choose
from listbox when checkbox is disabled
n4=str2num( get(handles.edit8,'String') );% n4 "edit text 8" gets the
value you type
if(n4<1)%if you type value beneath 1
    n4=1;% then n4=1
end

```

```

N1=(cos(n4).*(A(:,1)))-(sin(n4).*(A(:,2)));%equation of N1 using edit
text8 as n4
N2=(sin(n4).*(A(:,1)))+(cos(n4).*(A(:,2)));%equation of N2 using edit
text8 as n4
N3=(cos(n4).*(A(:,4)))-(sin(n4).*(A(:,5)));%equation of N3 using edit
text8 as n4
N4=(sin(n4).*(A(:,4)))+(cos(n4).*(A(:,5)));%equation of N4 using edit
text8 as n4

plotData5(A,N1,N2,N3,N4);%go to plotdata5 function with inputs array
A %N 1,2,3,4

guidata(hObject, handles);%update handles ,hobject structure

%%
function []=plotData5(data,N1,N2,N3,N4)%function plotdata5 plots the
rotation

    figure;%open figure 1

    subplot(4,2,1);%creating a plot on top and left of the
figure1
    plot(data(:,7),N1);%plot the 7th column of array data which
is time and N1 the column we created with equation N
    xlabel('Time (sec)');%name the axis x
    ylabel('Ex rotated(mv/Km)');%name the axis y
    grid on % enable grid

    subplot(4,2,5); %creating a plot on bottom and left of the
figure1
    plot(data(:,7),N2);%plot the 7th column of array data which
is time and N2 the column we created with equation N
    xlabel('Time (sec)');%name the axis x
    ylabel('Ey rotated(mv/Km)');%name the axis y
    grid on% enable grid

    subplot(4,2,2);%creating a plot on top and right of the
figure1
    plot(data(:,7),N3);%plot the 7th column of array data which
is time and N3 the column we created with equation N
    xlabel('Time (sec)');%name the axis x
    ylabel('Hx rotated(nTesla)');%name the axis y
    grid on% enable grid

    subplot(4,2,6);%creating a plot on bottom and right of the
figure1
    plot(data(:,7),N4);%plot the 7th column of array data which
is time and N4 the column we created with equation N
    xlabel('Time (sec)');%name the axis x
    ylabel('Hy rotated(nTesla)');%name the axis y
    grid on% enable grid

% -----
-
function m_about_Callback(hObject,handles)%function of about program

```

```

% hObject    handle to m_about (see GCBO)
% handles    structure with handles and user data (see GUIDATA)
about %go to function about

% --- Executes on button press in subplots.
function subplots_Callback(hObject, handles)% function subplots plots
the real time data
% hObject    handle to subplots (see GCBO)
% handles    structure with handles and user data (see GUIDATA)
A=handles.data{handles.index{1}};%A is equal to the array we choose
from listbox when checkbox is disabled

dt=24/length(A);%divide the length of array A
q=length(A)-1;%q=length of array A -1
k(1)=0;%cell 1,1 of k is 0
k(1,q+1)=dt;% from cell 2,2 to length of array A cells of k will have
value dt
k=cumsum(k);%returns the cumulative sum along different dimensions of
an array.

plotData6(A,k);%go to function plotdata6 with inputs array A & k
guidata(hObject, handles);%update handles,object structure
%%
function []=plotData6(data,k)%plotdata6 function plots the real time
subplots

    subplot(4,2,1);%creating a plot on top and left of the figure
    plot(k,data(:,1));% x axis is k & y axis column 1 of "data"
array
    xlabel('Time (hours)');%name the axis x
    ylabel('Ex RT (mv/Km)');%name the axis y
    grid on % enable grid

    subplot(4,2,3);    %creating a plot on middle and left of the
figure
    plot(k,data(:,2));% x axis is k & y axis column 2 of "data"
array
    xlabel('Time (hours)');%name the axis x
    ylabel('Ey RT (mv/Km)');%name the axis y
    grid on % enable grid

    subplot(4,2,5);%creating a plot on bottom and left of the
figure
    plot(k,data(:,3));% x axis is k & y axis column 3 of "data"
array
    xlabel('Time (hours)');%name the axis x
    ylabel('Ez RT (mv/Km)');%name the axis y
    grid on % enable grid

    subplot(4,2,2);%creating a plot on top and right of the
figure
    plot(k,data(:,4));% x axis is k & y axis column 4 of "data"
array
    xlabel('Time (hours)');%name the axis x
    ylabel('Hx rt (nTesla)');%name the axis y
    grid on % enable grid

```

```

        subplot(4,2,4);%creating a plot on middle and right of the
figure
plot(k,data(:,5));% x axis is k & y axis column 5 of "data"
array
xlabel('Time (hours)');%name the axis x
ylabel('Hy RT (nTesla)');%name the axis y
grid on% enable grid

        subplot(4,2,6);%creating a plot on bottom and right of the
figure
plot(k,data(:,6));% x axis is k & y axis column 6 of "data"
array
xlabel('Time (hours)');%name the axis x
ylabel('Hz RT (nTesla)');%name the axis y
grid on % enable grid

% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, handles)%function of pop up
menu 4
% hObject    handle to popupmenu4 (see GCBO)
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu4 contents
as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu4

if (handles.processDataCompleted == 0)
    return
end

plotType3 = get(hObject,'Value');%plottype gets choise value we've
choose from pop up menu 4
handles.popvalue2={plotType3};%pop value is confirmed
A=handles.data{handles.index{1}};%A is equal to the array we choose
from listbox when checkbox is disabled

dt=24/length(A);%devide the length of array A
q=1:length(A)-1;%q=length of array A -1
k(1)=0;%cell 1,1 of k is 0
k(1,q+1)=dt;% from cell 2,2 to length of array A cells of k will have
value dt
k=cumsum(k);%returns the cumulative sum along different dimensions of
an array.

plotData7(A,plotType3,k);%go to plotdata7 function with inputs array
A,choise of pop up menu4 and k

guidata(hObject, handles);%update handles, hobject structure

% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, handles)
% hObject    handle to popupmenu4 (see GCBO)
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.

```

```

%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function []=plotData7(data,option,k)%plotdata7 function enlarge the
subplot of the coise of pop up menu4

temp=get(0,'children');% returns the children-defined value of temp
if(length(temp)>1)% if length of temp bigger than 1
    close(temp(2));%close the 2nd cell of temp
end

if (option==1)%if choise 1 is selected from pop up menu 4
    figure;%create on fig1
    plot(k,data(:,1));% x axis is k & y axis column 1 of "data"
array
    xlabel('Time (hours)');%name the axis x
    ylabel('Ex RT (mv/Km)');%name the axis y
    grid on;% enable grid
elseif(option==2)%if choise 2 is selected from pop up menu 4
    figure;%create on fig1
    plot(k,data(:,2));% x axis is k & y axis column 2 of "data"
array
    xlabel('Time (hours)');%name the axis x
    ylabel('Ey RT (mv/Km)');%name the axis y
    grid on;% enable grid
elseif(option==3)%if choise 3 is selected from pop up menu 4
    figure;%create on fig1
    plot(k,data(:,3));% x axis is k & y axis column 3 of "data"
array
    xlabel('Time (hours)');%name the axis x
    ylabel('Ez RT (mv/Km)');%name the axis y
    grid on;% enable grid
elseif(option==4)%if choise 4 is selected from pop up menu 4
    figure;%create on fig1
    plot(k,data(:,4));% x axis is k & y axis column 4 of "data"
array
    xlabel('Time (hours)');%name the axis x
    ylabel('Hx RT (nTesla)');%name the axis y
    grid on;% enable grid
elseif(option==5)%if choise 5 is selected from pop up menu 4
    figure;%create on fig1
    plot(k,data(:,5));% x axis is k & y axis column 5 of "data"
array
    xlabel('Time (hours)');%name the axis x
    ylabel('Hy RT (nTesla)');%name the axis y
    grid on;% enable grid
elseif(option==6)%if choise 6 is selected from pop up menu 4
    figure;%create on fig1
    plot(k,data(:,6));% x axis is k & y axis column 6 of "data"
array
    xlabel('Time (hours)');%name the axis x
    ylabel('Hz RT (nTesla)');%name the axis y
    grid on;% enable grid
end

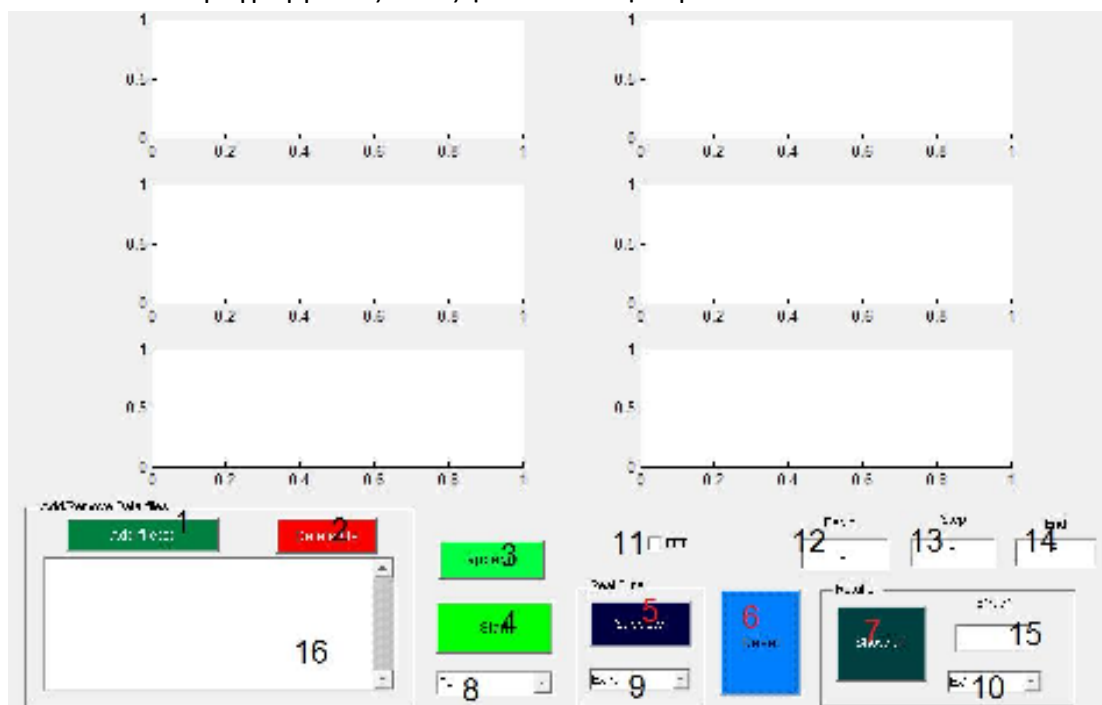
% end of program

```


5.ΣΗΜΕΙΩΣΕΙΣ-ΣΥΜΠΕΡΑΣΜΑΤΑ

- **Τρόπος λειτουργίας**

Τρέχουμε το πρόγραμμα Matlab 7.4.0 ,πηγαίνουμε στο current directory και ανοίγουμε τον φάκελο που βρίσκονται τα αρχεία του προγράμματος (m-file,figure κτλ.) Στη συνέχεια γράφουμε το όνομα του προγράμματος στο command window και πατάμε enter. Μπορούμε να τρέξουμε το πρόγραμμα επίσης απευθείας απ' το m-file. Το αποτέλεσμα θα είναι να ανοίξει το interface του προγράμματος όπως φαίνεται στη παρακάτω εικόνα.



Για να εισάγουμε δεδομένα πιέζουμε το κουμπί **1**,στη συνέχεια μπορούμε να εισάγουμε τα δεδομένα απ'το φάκελο που τα έχουμε αποθηκευμένα στον υπολογιστή μας. Μπορούμε να εισάγουμε ένα ή περισσότερα αρχεία απλά να ξέρουμε ότι αν είναι μεγάλος ο όγκος των αρχείων τόσο περισσότερο θα καθυστερήσει η επεξεργασία δεδομένων. Τα δεδομένα που εισήχθησαν θα εμφανιστούν στο στοιχείο **16** το οποίο είναι ένα listbox, τώρα αν θέλουμε να διαγράψουμε κάποιο στοιχείο το επιλέγουμε και πιέζουμε το κουμπί **2**.

Στην συνέχεια για να γίνει η επεξεργασία των δεδομένων πιέζουμε το κουμπί **4**(start). Ετσι θα μετατραπούν τα δεδομένα μας σε πίνακες και θα είναι έτοιμα για την περαιτέρω επεξεργασία. Το κουμπί **3** όταν το πιέσουμε θα μας δώσει μια ακολουθία απ' τους πίνακες που εισήχθησαν δλδ. Θα μας δώσει έναν ενιαίο πίνακα. Τώρα όταν επιλέξουμε ένα πίνακα με το ποντίκι

μας απ' το **listbox(16)** τα γραφήματα θα εμφανιστούν στα έξι sublots που βρίσκονται στο πάνω μέρος του figure μας. Αν θέλουμε να δούμε κάποιο απ' τα γραφήματα σε μεγέθυνση δε μένει παρά να επιλέξουμε απ' το στοιχείο **8**(pop up menu) αυτό που επιθυμούμε. Στα στοιχεία **12,13** και **14** έχουμε τα edit texts στα οποία μπορούμε να επέμβουμε και να μεταβάλλουμε την αρχή , το βήμα και το τέλος του γραφήματος όπως εμείς επιθυμούμε. Αν ενεργοποιήσουμε το στοιχείο **11**(checkbox) τότε θα έχουμε ένα ασυνεχή μετασχηματισμό fourier μεταλλάζοντας έτσι τους πίνακες μας και κατ'επέκταση τα γραφήματά μας. Στο στοιχείο **15**(edit text) μπορούμε να εισάγουμε μια τιμή γωνίας έτσι θα έχουμε περιστροφή των πινάκων που έχουμε εισάγει, το κουμπί **7**(show all) θα μας δώσει και τα τέσσερα γραφήματα σ'ένα figure ενώ το στοιχείο **10**(pop up menu) θα μας μεγενθίνει την επιλογή που προτιμάμε απ'τα τέσσερα γραφήματα. Το κουμπί **5** θα μας εμφανίσει τα γραφήματα με τον x άξονα να είναι σε πραγματικό χρόνο και το στοιχείο **9**(pop up menu) θα μας μεγενθίνει το γράφημα που επιθυμούμε. Τέλος το κουμπί **6**(reset) επαναφέρει όλα τα στοιχεία του προγράμματος στην αρχική τους κατάσταση. Να σημειωθεί ότι μπορεί να επιτευχθεί μερικός συνδυασμός στοιχείων του προγράμματος.

- **Συναρτήσεις(functions) του Matlab που χρησιμοποιήθηκαν**

Get : Query object properties

Set : Set object properties

ErrorDlg : Create and open error dialog box

Refresh: Redraw current figure

Str2num: Convert string to number

Subplot: Create axes in tiled positions

Horzcat: Concatenate arrays horizontally

Vertcat: Concatenate arrays vertically

Strcat: Concatenate strings horizontally

Fftshift: Shift zero-frequency component to center of spectrum

Cumsum: Cumulative sum

Grid: Grid lines for 2-D and 3-D plots

Plot: 2-D line plot

Guidata: Store or retrieve GUI data

Fopen: Open file, or obtain information about open files

Close: Remove specified figure

About: Creates an about figure that you can write a text (about m-file created separately)

Handles: A function handle is a MATLAB value and data type that provides a means of calling a function indirectly. You can pass function handles in calls to other functions (often called function functions). You can also store function handles in data structures for later use (for example, as Handle Graphics® callbacks).

Πρέπει να σημειωθεί ότι αυτές είναι οι συναρτήσεις που εμφανίζονται μες'το πρόγραμμα, δεν έχουμε σημειώσει τις υπορουτίνες οι οποίες περιέχονται μέσα σε αυτές.

- **Σημειώσεις**

Στον σχεδιασμό του figure στην αρχή δε χρειάστηκε να βάλουμε axis γιατί τα δημιουργήσαμε μέσα στο m-file ,επίσης τα panel καθώς και τα static text χρησιμοποιήθηκαν εντελώς γραφικά δε χρειάστηκε να μεταβάλουμε τίποτα απ'αυτά στο m-file του προγράμματος.

Το about m-file το ξεχωρίσαμε απ' το αρχικό tych m-file γιατί δημιουργήσαμε νέο figure γι'αυτό.

Τέλος να σημειωθεί κάθε αλλαγή στον τρόπο λειτουργίας του προγράμματος επέφερε μεγάλη αλλαγή στον κώδικα του.